MULTI-AGENT SYSTEM FOR CONSUMER-ORIENTED ELECTRONIC COMMERCE

A Thesis submitted for the degree of
DOCTOR OF PHILOSOPHY

by

MANSOOR ABDULAZIZ TURAIF

Department of Information Systems and Computing,
Brunel University

November 1999

BRUNEL UNIVERSITY

PUBLICATION OF THESES

Please complete in block capitals

NAME: MANSOOR ABDULAZIZ TURAIF

DEPARTMENT: INFORMATION SYSTEMS AND COMPUTING

TITLE OF THESIS: MULTI-AGENT SYSTEM FOR CONSUMER-ORIENTED ELECTRONIC COMMERCE.

PLEASE COMPLETE EITHER PART A OR PART B

A. I agree that the abstract of my thesis may be published by the University without further reference to me.

In accordance with the University's Handbook of Procedures, the Head of Library Services may allow my thesis to be copied in whole or in part without further reference to me. Such Authority shall apply only to single copies made for study purposes and shall be subject to normal conditions of acknowledgement.

(See note's overleaf)

Signature: ______________________

Date: 16/12/99

If you have completed Part A please return this form direct to the Library.

B. I request that my thesis be held under confidential cover in the Library for a period of ____ years for the following reasons:

____

____

____

____

Once the period of confidentiality has expired, I agree that the conditions set down in Part A will apply without further reference to me

Signature: ______________________

Date: ______________________

Please pass this form to your supervisor who should sign below indicating his/her agreement to the retention of the thesis under confidential cover.

Signature of Supervisor: ______________________

Additional comments ______________________

____

____

____

Part B, when complete, should be returned to the Assistant Registrar.

Affectionately dedicated to my family,
Manal and Muhammad

Acknowledgements

بسم الله الرحمن الرحيم

In the Name of Allah, the most Beneficent, the most Merciful

First, I thank my advisor, Professor Robert O'Keefe, for the countless support, stimulating discussions and the good care that he has taken of me academically and as a friend. His flexibility and experience have also inspired, taught and encouraged me in a number of important ways.

I gratefully thank Manal and Muhammad for always being there for me and making the time well-rounded and meaningful. Without your inspiration and support none of this would have been possible, thank you.

Last but certainly not least, I thank my parents, brothers and sisters for all the moral support that they have continuously provided me. I especially thank my mother and father for all their efforts in my upbringing, "my Lord! Bestow them the mercy as they cherished me in childhood."

And our final prayer is "all praises and thanks be to Allah, the Lord of all that exist."

ABSTRACT

With the advent of the information superhighway and the exponential growth of the Internet usage, the importance of multi-agent systems is proliferating. The central theme of this thesis is to demonstrate the benefits of adopting multi-agent system (MAS) paradigm to implement consumer oriented electronic commerce system. The discipline of computational science is exploited to provide insights into the behaviour of a model of consumer behaviour that reflect the cognitive notion that the thesis has developed. For this, a multi-agent system computational environment is used to model and investigate the consumer purchase over the Internet. The MAS is developed based on a presented taxonomy, that is most relevant to the thesis application. The thesis also presents a novel approach to negotiation. Results of empirical evaluations provide a strong support that agents using the proposed approach would achieve higher payoff than human subjects. An empirical evaluation for the usability of the prototype system is also presented. Reported results are very encouraging to implement a fieldable system. To complement the perspective for a complete consumer-oriented EC system, the thesis addresses and develops approaches for searching and extracting relevant information. Example experiments are also reported to act as indicators for the effectiveness of the developed approaches.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The importance and the increasing interest in the agents paradigm is likely to grow rapidly as a result of several developments. One is the advent of the "information superhighway" and the exponential increase and diversity of the information resources. Software agents are well suited for enhancing advances in usability for effective use of the Internet by end users. The aim of developing these agents is to assist users in a range of activities, and could be designed to simplify and/or automate functions to relieve them from complex low-level tasks.

Fuelled by the popularity of the Internet, electronic commerce (EC) is proliferating. Consumers need to find products' suppliers, products' information (including product brochure, specifications, and viable configurations) and they have pressing need to obtain experts advice (including product rating services, specialised newsgroups and online magazines) prior to the purchase. However, existing systems, such as virtual malls and online stores, are not well integrated, and the consumer has to spend time and money locating these particular sites. Software agents can assist in leveraging existing resources by electronic commerce in a number of ways. Agents could be used for Internet shopping, taking required specifications as input and returning the purchase that meets those specifications as output.

The central theme of this thesis is to demonstrate the benefits of using a Multi-Agent System (MAS) paradigm to implement a consumer-oriented electronic commerce system. Methods are developed for the entire consumer's decision making process. The approach of MAS widens the notion of software agents to include automating tasks delegated by other agents as well as end users by incorporating communication mechanisms for interactions with other agents in the system.

The thesis concentrates on the benefits, barriers and design issues of the MAS approach. Particular attention is given to the field of negotiation. Searching and information extraction methods will be also addressed.

1.1 Issues in Electronic Commerce

While electronic commerce is developing expeditiously, a number of issues remain to be resolved of which some need conceptual work. Kalakota (1996) argues that there are three key business issues to Internet-based electronic commerce that need to be considered. These are value-laden content, enhanced search capabilities and simplified buying processes. Other issues that are key to Internet-based electronic commerce include:

- Privacy and security: Online electronic commerce mandate effective and trusted mechanisms to provide confidentiality, authentication and non-repudiation for the users. Luckily, security of online systems is evolving rapidly and most of the existing systems are secure enough to be used for most commercial transactions; and with the evolving legalisation in the electronic commerce field, better systems are expected to be developed (Bloch et al. 1996).

- Electronic payment systems: The implementation of technical standards for safeguarding credit card purchases over the Internet and the evolution of revolutionary schemes for electronic cash mechanisms (such as electronic cash and value-token creation and passing) are steps towards addressing this issue.

- Implementation issues: The cohesion of new technologies and the adoption of electronic commerce in an organisation requires changes to the management structure and the redesign of the business strategy. The issues of alignment or balance required by an organisation embracing electronic commerce are alignments between: (1) strategy and technology, (2) technology and organisational processes and (3) technology and the employees. These alignment issues should be considered in the implementation plans for electronic commerce (Bloch et al. 1996).

- Process automation: Software agents present a challenge to automate the process of electronic commerce applications.

Some of the questions regarding the development of software agents for electronic commerce systems, which need further research, are:

1. Can artificial agents be relied upon to arrange for most of routine business transactions autonomously?
2. Is it possible to develop learning software agents for finding, retrieving and suggesting information of interest to the client?
3. Can intelligent software agents be developed for advertisement filtering?
4. What is required to develop these agents? How does it work?

These general questions, which lead to many more specific questions, are hot topics for researches investigating solutions for electronic commerce issues. While many avenues of investigation have been undertaken to quest solutions for these questions, the paradigm of agents using some sort of learning techniques seems to be promising to pursue.

1.2 Example Application Domain: Software Agents for EC

The recognition for the need for software agents to help consumers in their decision making process is likely to become apparent when the consumer is faced with the problem of searching the Internet for the rational product that satisfy his need. The following scenario depicts an example application domain, where the methods developed by this thesis are needed and should motivate this research from practical perspective.

> Consider the case of a consumer -call him John- who happens to need a car, and decides to purchase one via the Internet. Because he has no experience with cars, he surfs the Web for information sources that would help him in making the right choice. Besides querying popular search engines using a general keyword "car", scanning and filtering the results for useful sources of information, a myriad of search alternatives are tried. For example, John examines some of the copious online publications that deals with cars, he overviews specialised newsgroup discussions and posts questions asking for advice. John also searches yellow pages to locate cars' manufacturers Web pages, in a very lengthy pre-purchase search. Number of factors could increase the pre-purchase search in this specific situation. The fact that John has no past experience with cars that leads him to seek experts' advice is one dimension. Furthermore, a car is a long lasting product with a high price and personal factors such as how much he is willing to pay for the car affects the search time. After finishing the search stage, John starts to evaluate the evoked set of candidate cars. He uses two types of information: 1) a list of brands from which he plans to make his selection from (evoked set), and 2) the criteria he uses to evaluate each brand of the evoked set (e.g. safety, style, price, service after sale, ride and performance). John assigns weight to each criterion in terms of what is more important to him. He then ranks all available options based on the summated score of each brand option (this is called the

consumer heuristics/decision rules). The final stage before making the choice is the negotiation (bargaining) stage, which could take place via e-mail. At this stage John negotiates with the candidate supplier/s to refine and/or clarify some details, and could include price bargaining. The negotiation session could also involve discussing a range of issues that go beyond price (such as accessories, delivery and warranties). The negotiation stage helps in further distinction between available alternatives for a better choice.

The above scenario shows how much time and money the consumer needs to spend in locating relevant information to assist him in making the genuine purchase decision. Would not it be judicious to have a system that would elicit the consumer needs, and then autonomously searches the entire Internet for the required product? The system then compares available choices by consulting newsgroups and specialised online rating services and analysing manufacturers' brochures. Then, the system negotiates best deal, (maybe) purchases, and finally notifies the user of the finalised deal and expected delivery time, giving reasons for the selection "This brand is selected because it came out best when balancing the good ratings with bad ones".

1.3 A Model of Consumer Decision Making

To develop a framework for understanding how consumers make decisions, problem solving should be tied with many relevant concepts that affect decision making such as psychological and socio-cultural issues. To capture the dynamics of consumer decision making and to increase understanding of consumer behaviour, a simplified model of consumer behaviour that reflect the notion of the cognitive behaviour is provided as shown in Figure 1.1. This model borrows from comprehensive models of consumer behaviour described in the literature (Engel et al. 1978, Howard 1994).

The model consists of three stages: (1) Information input and variables influencing the decision process (2) Decision process stage and (3) Output. The information input and variables influencing the decision process stage consist of external stimulus (such as marketers' efforts in advertisements, promotions, pricing policy, etc.) together with endogenous and exogenous socio-cultural influences. Endogenous influences include cultural norms and values, personality, life style and social class. Other persons influencing the intentions and choices of the consumer constitute the exogenous influences. These include reference groups and family influences, word of mouth, product rating services (consumer reports and magazines), an editorial in a newspaper, or an article in a journal.

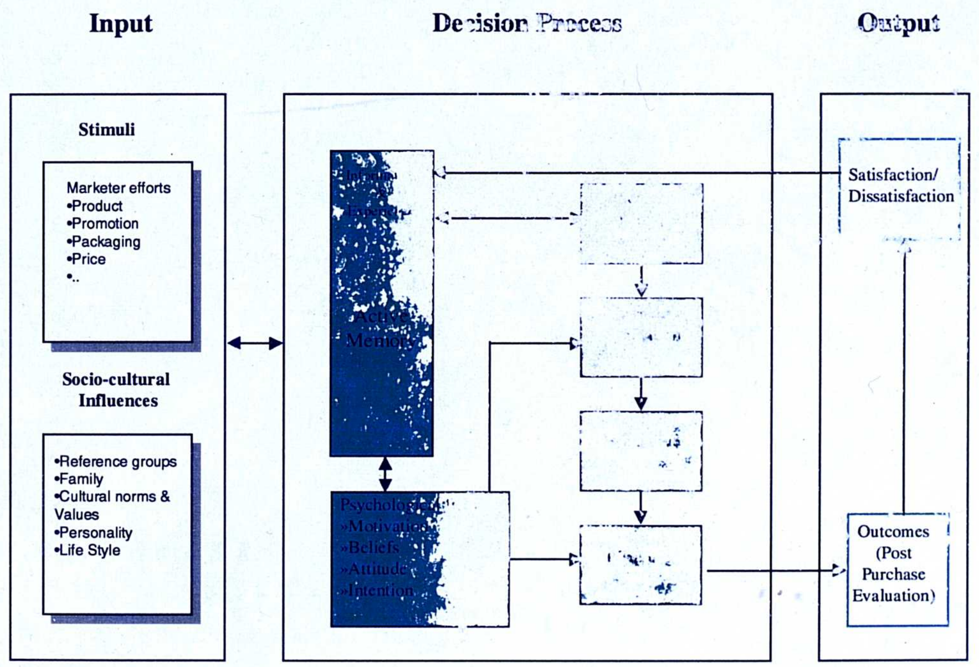


FIGURE 1.1. SIMPLIFIED MODEL OF CONSUMER DECISION-MAKING PROCESSES.

The first step in decision process stage is the search process. The extent and depth of search varies according to the experience (internal search for previous experiences stored in active memory). If the consumer lacks the desired degree of knowledge (or he thinks that additional information will be advantageous) he engages in external search in order to gather sufficient information about the number of potential suppliers and characteristics of each of their products and brands. Once the search (internal and/or external) has been finished and relevant information has been obtained, the next stage of the decision process is to sense and compare those alternatives against the consumer's perception of desirable attributes. Those desirable features result from the interaction with the psychological fields (motivation, beliefs about a product, attitude towards a product, and intention to buy a specific product) and past experience.

Evaluative criteria, which reflect the underlying motives of the consumer, are the specification criteria the consumer employs against which various alternatives are assessed. Those could be objective (such as price, quantity, delivery timing, packaging and service) and/or subjective (such as symbolic values or distinctiveness of a product). Consumers frequently link the price with the quality as a criterion, and they usually consider the brand reputation as the dominant evaluative criteria. The consumer accordingly updates his beliefs, determining the extent to which each alternative (of the evoked set) possesses the desirable attribute (payoff value). A useful attitude model for evaluating consumer preferences presented by Rosenberg (Engel et al. 1978) takes the form:

$$A_o = \sum_{i=1}^{n} ViPi$$

Where:

A_o is the overall evaluation of the attractiveness of alternative o.

V_i is the relative importance of ith value importance of ith criterion.

P_i is the perceived instrumentality of alternative o with respect to ith value (payoff).

n is the number of salient values (number of criteria).

Available alternatives may be subject to negotiation, in which case the consumer would engage in negotiation with suppliers to get least cost under most favourable conditions. The negotiation could involve discussions on a range of issues that go beyond price (such as packaging, method of shipment, delivery time, warranties, technical assistance and service, and payment terms). The negotiation would help in hammering out details and attributes of each potential supplier's product, which would be used in the evaluation process, before a choice is actually made.

Choice and outcomes of the choice are the last stage in the decision process. Outcomes of a choice could be satisfaction, dissatisfaction or post choice dissonance (doubt that a correct choice was made). These outcomes would exercise a strong effect (feedback) on future choice behaviour. An evaluation of satisfaction or dissatisfaction is stored in the active memory (learning theory).

1.4 The Problem of Automating the Consumer Decision Making Process

Based on the presented model, searching and negotiation, which is in turn a search process, are two fundamental tasks in the consumer decision making process for online purchase over the Internet. Currently, there are no tools that automate the process for the consumer. Existing tools, such as Infoseek, WebCrawler, Yahoo, and AltaVista, are unable to interpret the results of their searches. More sophisticated agent-based AI systems, including the Internet SoftBot (Etzioni & Weld 1994) and Occam (Kowk & Weld 1996), are able to reason, plan, extract information and analyse unstructured Web pages, but do not include automatic learning techniques to scale up with the immense expansion of the Internet.

BargainFinder (Krulwich 1995) and ShopBot (Doorenbos et al. 1997) are two existing examples of systems using agent technology for Internet shopping. Whereas BargainFinder is hand coded for a single product domain (Compact Disk market), ShopBot is a domain independent comparison-shopping agent. ShopBot autonomously learns how to shop at pre-given home pages of online stores and extracts information such as price and availability and summarises the results for the user. Product recommendation is not accounted for in these systems.

A satisfying implementation of software search and information extraction agents to support the consumer decision making process for this thesis purpose should be endowed with means of learning techniques to address the following questions:

1. Data mining and information discovery: How does the agent find new and unknown information resources and product suppliers? How does the agent find URLs of related newsgroup/ on-line product rating services that could advise/recommend on these products and available options?

2. Data filtering and extraction: How does the agent access an information source and parse the required data? How to generate automatic wrappers that would satisfy source query and data extraction?

3. Data interpretation: How does the agent explicate the parsed data? How to map natural language queries to the appropriate coveted data?

4. Data presentation and interaction: How does the agent present the response to the query? How does the agent inter-communicate with other agents in the Infosphere to solve larger problems?

In order to support the negotiation process, electronic commerce applications need to have the ability to negotiate. Negotiation is the process in which the consumer and the potential supplier multi-dimensionally negotiate resources for mutual intended gains. This process can be perceived as "negotiators jointly searching a multi-dimensional space and then agreeing on a single point in the space" (Oliver 1996).

Existing electronic commerce applications do not support negotiation. However, researchers are studying expedient approaches to endow intelligent artificial agents with the capacity to learn good negotiation strategies, rather than attempting to initially hand-code them with a strategy that could handle any liable situation (Oliver 1996, Dworman et al. 1996a, Zeng & Sycara 1996).

The incorporation of negotiation agents for consumer-oriented electronic commerce application appropriate for this thesis intent adds the following challenges that should be addressed:

5. Ontology issue: How does an agent understand / describe objects in meaningful and non-ambiguous way to the other party in negotiation?

6. Strategy: How does an agent determine which rules, that determine which points in the negotiation space should be propounded or accepted, should be used? What types of strategies are superlative?

7. Goals: How does an agent determine which value function or utilities over the negotiation space should be assigned for a specific negotiation session?

8. Communication: How does an agent interface to the external market/ inter-communicate with the other party's agent? (Maybe use e-mail if the supplier does not provide electronic negotiation agent).

Critical for a successful system implementation is user's acceptance. The challenge is to come to an open and viable approach for designing the interface between the end user and the multi-agent systems in an enterprise context. Some of the requirements for the user interface design include:

9. Ease of use: The interface should be designed in such a way that the requirements imposed on the user are moderate. For example, preference elicitation should not entail major requirements on the user that would outweigh the benefits of using the system.

10. User control: Users should be allowed to deputise their tasks to agents expressly. Tasks may be allocated, suspended, recommenced or cancelled at any point in time.

11. Transparency: The interface should guarantee access to knowledge on task progress (current state and approximate completion time), task succession (how a task is decomposed and which agents fulfil its part) and execution results.

1.5 Thesis Scope

This thesis focuses on the barriers, techniques, design issues and benefits of developing multi-agent system paradigm towards implementing consumer-oriented electronic commerce system. The thesis gives a special consideration to the negotiation field. The aim is to make some contributions and extend the current research in the MAS and negotiation fields. In depth research in which results are scrutinised fully to whether the designed agents can be attainably integrated into electronic commerce system will be conducted.

In the thesis, several topics are investigated including:

- Techniques for developing distributed multi-agent systems.
- Practical design issues of multi-agent systems for electronic commerce applications.
- The problem of automating learning to interact with online information sources.
- The challenge of developing artificial negotiation agents for electronic commerce systems.
- The challenge of finding unknown information resources and extracting the required information.

1.6 RESEARCH QUESTIONS

This thesis seeks to answer some interesting issues pertaining to the big deal theory of electronic commerce discussed by Kimbrough (1996). The primary research questions that this thesis will strive to answer are:

1. Can artificial agents, that are ingenious and trustworthy enough, be developed to arrange for a large percentage of routine business transactions, sometimes involving in subtle negotiations?

2. Is it feasible to develop an intelligent system that would allow consumers to obtain services or information and make effortless Internet purchase?

3. How to implement software agents for finding, retrieving, extracting and interpreting required information?

4. What is required to develop these agents? How does it work?

The consequence to a satisfactory answer to these questions and the more specific ones that ensue with it are momentous, or at least appealing adequately to motivate this research.

1.7 Statement of Thesis

This thesis presents a number of novel approaches and contributions in the different topics investigated. Those include:

- In the area of MAS, Chapter 4 presents a taxonomy of MAS architecture that elaborates on features that are most relevant to the Internet-based consumer-oriented EC application. Hybrid architecture of MAS is used to marry the best aspects of two existing standard architectures.

- In the area of negotiation, Chapter 4 presents a novel approach that incorporates decision-theoretic constructs and negotiation analysis techniques. The thesis claims that the devised negotiation strategy can do better than human subjects in similar negotiation context. Chapter 6 reports the results of experiments conducted to support this claim.

- In the areas of Internet searching and information extraction, the thesis builds on existing techniques to provide more effective approaches. The thesis claims that the pre and post processing techniques proposed in the meta-search approach would result in a higher recall and precision rates. The thesis also claims that the use of wrapper generation method for information extraction is promising to build upon. Chapter 5 reports on example experiments to act as indicator for the validity of these claims.

1.8 Thesis Outline

The rest of this thesis is structured as follows:

- Chapter 2 discusses related research. The chapter reviews relevant research that has been done in the areas of negotiation, information retrieval, information discovery and software agents. The first section is dedicated to critical analysis of the negotiation literature. The second stream of literature review is dedicated to information retrieval, information discovery and Internet searching techniques. The chapter concludes with a review of the field of software agents and design issues for multi-agent systems.

- Chapter 3 describes the pursued research methodology. The chapter starts by introducing computational science as a legitimate approach to investigate complex models. Next, the design of consumer-oriented EC system is described within which a framework for MAS is presented. The chapter concludes by presenting the measuring criteria for successful answers to the research questions.

- Chapter 4 presents the work in the MAS field. The chapter presents the developed MAS taxonomy. First, the environment occupied by the MAS is analysed. Then, the architecture of the system within which the agents interact and the architecture of the individual agents are described. The chapter concludes by giving a clear picture of the thesis contribution in the area of MAS.

- Chapter 5 presents the hypotheses and the experimental design. Section 5.1 begins with an overview of the experiments that are designed to test the efficacy of the thesis negotiation approach. The specific hypotheses to be tested are given. In Section 5.2, the experiments that are designed to test the system's usability are presented. In Section 5.3, evaluation techniques that can be used when testing the search approach are presented. The section reports an example experiment that is conducted to act as indicator for the effectiveness of the approach. The chapter concludes by discussing evaluation techniques for the wrapper generation approach and reports an example experiment to act as indicator for the effectiveness of the approach.

- Chapter 6 investigates the results of the experimental work. First, the results of the negotiation experiments are reported and analysed. Then, the results of the usability testing experiments are reported and analysed.

- Chapter 7 concludes and investigates future research. The chapter starts by presenting a statement of contributions made during the study. Second, the implications of this research are presented. Third, some of the requirements for a fieldable Internet-based consumer-oriented EC system are presented. The chapter concludes by outlining some issues that can be expanded and areas for further research.

CHAPTER 2

RELATED LITERATURE

This chapter reviews relevant research that has been done in related areas. Related projects are described at variable level of details. Some projects are described at relatively shallow level of detail because they use very different technical details compared to this work, although they address similar issues.

Since, a major piece of this research concerns negotiation, the first section is dedicated for negotiation and bargaining literature research. Game-theoretic approaches (mathematical models), decision-theoretic approaches (Decision and Negotiation Support Systems) and non-economic computational approaches (Distributed Artificial Intelligence (DAI) and Evolutionary Computation) are reviewed.

The second stream of literature review is concerned with information discovery techniques. Since the term "information discovery" is used as an amalgam of Information Retrieval (IR) and Information Extraction (IE) to describe the task of locating and extracting needed data expressed by the user as a query, both fields are addressed. In Section 2.3, various techniques that are currently used for searching the Web are examined. Following, current approaches used in systems for comparison shopping over the Internet are presented.

Finally, the field of software agents is addressed. The temptation to be involved in the arguments of classifying what is an agent, and whether, as a requirement, the software should posses some sort of learning or intelligence to be classified as an agent or not will be avoided. Instead, important properties of software agents that are useful for this system are addressed, then the design issues for multi-agent systems are studied.

2.1 NEGOTIATION

One of the main areas of multi-agent systems which this thesis contributes to is negotiation automation. First, the individual transactions (conventional commerce or Internet-based commerce) are classified into three types:

1. **Posted Price:** is a type of transaction where the seller posts a credible take-it or leave-it offer. Price bargaining is not allowed in these types of transactions, which in effect reduces issues of negotiation by one dimension (since other issues such as delivery, colour, etc. could be negotiated).

2. **Auctions:** are forms of transactions where an auctioneer offers items for sale, and potential bidders compete to acquire the item at lowest possible price. English, sealed-bid, Dutch and Vickrey are four different auction protocols.

   In the English auction, each bidder is free to raise his bid. The auction ends when no one is willing to rise anymore, and the highest bidder gets the item at the price of his bid. An agent's strategy for this type of auction is a series of bids as a function of his private value for the item, his prior estimates of other bidders' valuations and the past bids of the others. A dominant strategy for the agent is to bid $X + \varepsilon$, where X is the highest current bid and ε is a small amount, and withdraw when his private valuation is exceeded.

   In the sealed bid auction, each bidder submits a bid without knowing the others' bids. The highest bidder gets the item at the price of his bid. A possible agent's strategy for this type of auction for bidding is a function of his private valuation for the item and prior knowledge/beliefs of other bidders' valuations. There is no general dominant strategy for participation in this auction.

In the Dutch auction, the auctioneer continuously lowers the price of the item until one of the bidders accepts the offer at the current said price. A possible agent's strategy for this type of auction is his bid as a function of his private valuation for the item and prior knowledge/beliefs of other bidders' valuation with no general dominant strategy to follow.

In the Vickrey auction, each bidder submits a sealed bid without knowing the others' bid. The highest bidder gets the item, but at the price of the second highest bid (Vickrey 1961, Rasmusen 1989). Agents' strategy is his bid as a function of his private valuation and his beliefs of others' private valuation for the item, whereas agent's dominant strategy is to bid one's true valuation for the item.

Vickrey auctions have been widely studied and adopted for the use in computational multi-agent systems to allocate computation resources in operating systems (Waldspurger et al. 1992) and to allocate bandwidth in computer networks (MacKie-Mason & Varian 1993).

Cheating by the auctioneer, i.e. overstating the second highest bid for the highest bidder, has been suggested to be one of the main reasons why the Vickrey auction protocol has not been advocated in auctions among people. Another reason is that the bidder prefers not to reveal his true valuation for the auctioned item since, sometimes, this information is sensitive (Rothkopf et al. 1990).

Auctions have a number of characteristics that simplify the task of implementing autonomous negotiation agents. These include limiting the negotiation issues to a single dimension "price" negotiation, in which agents use proved strategies. The need for "special vocabulary" is somewhat resolved, since the auctioned item is displayed and should be purchased "as-is" with whatever specification (Beam et al. 1996).

Online auctions are spawning as a competitive and proliferating industry. OnSale (www.onsale.com) and AuctionWeb (www.ebay.com/aw) are examples of two popular commercial auctions' Web sites. Design features of existing Web-based electronic auctions include: (1) whether or not sellers specify reservation price for their items, (2) whether or not the auction supports automatic bidding, (3) auction closing time and the rules regarding closing, and (4) the type of the merchandised item (new or used) and the quantity offered (one or bulk). For example, in AuctionBot (Wurman et al. 1998) users choose the type of the auction they want to create to sell their products. The users also specify the parameters for the auction (e.g. method for resolving bid ties, clearing time and number of sellers permitted). Buyers then start bidding, above the seller's reservation price, according to the multi-lateral distributed negotiation protocols.

3. **Negotiation:** is a transaction in which two or more parties bargain over multidimensional issues. The following sub-sections review related streams of research that inform the design and modelling of a system of negotiation agents.

2.1.1 Game-Theoretic Models of Negotiation

The discipline of economics views negotiation from a mathematical perspective. Negotiation theories were developed, with the advent of game theory, which would predict possible outcomes in the contract curve. Nash (1950) proposed a model to predict an outcome of a bargaining based on information about each bargainer's preferences, as modelled by the expected utility function over the set of feasible (Pareto-Superior) agreements and the outcome that would result in case of disagreement. The approach of Nash's axiomatic model (also called co-operative model since it models the bargaining process as a co-operative game) of bargaining is to postulate a set of desirable properties for the solution, called axioms of the bargaining solutions as represented in terms of the utility functions of the bargainers. The solution concept that satisfies these axioms is sought (Rasmusen 1989, Osborne & Rubinstein 1994).

When more than one of the solutions are individually rational, i.e. pay more than disagreement, to both of the bargainers there may be a continuum of Nash equilibria. Due to the non-uniqueness of the equilibrium, strong axiomatic solution for Nash bargaining is proposed. Nash bargaining solution prescribe a unique solution by postulating a set of four axioms characterised by Roth (1979) as:

Axiom 1: Strong individual rationality (Pareto-Efficiency).
Axiom 2: Independence of irrelevant alternatives.
Axiom 3: Independence of utility calibration.
Axiom 4: Symmetry (Bargainers have symmetric positions).

Symmetric mechanism means that if the parties act exactly alike they would receive the same payoff, and that the mechanism does not handle any party in a priori or dissimilar manner.

Progress in the strategic approach to the theory of bargaining has been due to the developments in the general theory of non-cooperative games and the study of the roles to be ascribed to information, time and commitment within the bargaining process.

Harsanyi's (1956) theory of 'games of incomplete information' extended the strategic approach to the theory of bargaining, to allow more realistic modelling of the bargaining situation by taking into account the uncertainty about the preferences that the other player may have over the set of possible deals that can be reached, uncertainty about the extent to which commitment is possible and uncertainty over time-dependent costs (Harsanyi 1956, Binmore & Dasgupta 1987, Rubinstein 1987). Selton (1981) developed the notion of subgame-perfect equilibrium that offers a technique for reducing the multiplicity of Nash equilibria, which could be found in many non-cooperative games. He proposes criteria to select a subset of equilibria that could credibly be expected to arise from certain kinds of rational play of the game (Selton 1981, Roth 1985).

Multi-period alternating offer bargaining models, where agents alternate in making offer to each other in a pre-specified sequential order, have been an active research area (Rubinstein 1982, Osborne & Rubinstein 1994, Rasmusen 1989, Kreps 1990). Some researchers have also analysed alternating sequential bargaining with no exogenous bound on the length of the time with one-sided incomplete information (e.g. the seller's S valuation is a common knowledge and the buyer's B valuation is a private information) and two-sided incomplete information (both seller's S valuation and buyer's B valuation are private information) (Chatterjee & Samuelson 1983, Linhart et al. 1992).

The role that the time, lying, claims and cheap talk play in the description of the game tree representing the framework within which the players bargain has been studied. How players can improve their positions in the game by undertaking costly actions (such as delay) to convince their opponent that they are not keen to the deal, has been also analysed (Crawford & Sobel 1982, Fudenberg & Tirole 1983).

To Summarise, despite the theoretical insights and bargaining strategies offered by game theoretic models of bargaining, these models are not informative as to how to design computational models for negotiation. These mathematical models give information on expected outcomes of a rational negotiation, without guidelines on which strategies to use in a given negotiation context to reach these expected outcomes. Thus, in spite of the significant role game theoretic models play in the pre/post analysis of the negotiation problem, the usefulness of those models as a negotiator during the process is very limited. Also, the underlying assumptions of those models is too restrictive (e.g. assumptions of perfect rationality, single-issue negotiation, complete information) to be effectively used in real actions undertaken in decision and negotiation processes.

2.1.2 Decision Theoretic Approaches

Weaknesses in the game-theoretic models have motivated researchers in the area of Decision Analysis (DA) to study negotiation and negotiation support. Raiffa (1982) introduced the prescriptive / descriptive negotiation analysis in an attempt to provide advice and support for negotiation problems. He suggested that, in order to provide support for prescriptive method for negotiation, the opponents' decision process, needs and constrains should be taken into consideration.

Negotiation Analysis (NA) is a framework of thought in which analytical methods are used, taking the evolving nature of the negotiation process into consideration through negotiation arithmetic, adding / subtracting issues and parties and investigating changes in parties' perceptions (Lax & Sebenius 1986). To evaluate the possible responses of the opponent, NA techniques use goal-seeking and if-then functions.

Assessment of the opponent's alternatives to the negotiated deal, that is to determine "Best Alternatives To a Negotiated Agreement (BATNA)", opponent's output interests and the relative importance of each of these interests to the opponent (the utility function) determine the basic structure of the negotiation analysis approach taking into consideration the bounded rationality of the negotiator (Sebenius 1989, Fisher & Ury 1981).

The influence of cognitive heuristics on decision-making and negotiator's performance has been studied by behavioural decision theorists. The cognitive limitation and socio-emotional aspects of negotiator behavior have been shown to produce biased negotiation. Some of these interesting behaviours that could influence negotiation include (Foroughi & Jelassi 1990, Neale & Bazerman 1991):

- Framing: Negotiators employ a framing bias by evaluating alternatives as a potential loss or gain based on some reference point. They often behave in a risk-avoiding manner, i.e. accept an offered settlement, when evaluating potential gains, while they behave in a risk-seeking manner, i.e. opt for more concessions from opponent, when evaluating potential losses.

- Consideration of issues in isolation: Due to cognitive difficulty to integrate multilateral issues into a single package, negotiators tend to evaluate each issue in isolation.
- Anchoring and adjustment: Since issues under negotiation are often of uncertain value, the negotiator starts from an initial offer, and adjust from there to yield an acceptable offer.
- Information availability: Availability of information, past experiences and present knowledge play an important role in negotiator's evaluation of alternatives.
- Fixed-pie mentality: Negotiators often assume that they have conflicting interests with the other party and they are competing to split a fix-pie (win-loss).
- Negotiator overconfidence: When a negotiator is overconfident that he will achieve his favourable outcome, the likelihood of him making a compromise is reduced.
- Ineffective communication between negotiators.

A related area of research is Negotiation Support Systems (NSS), which are computer-based negotiation support tools using what-if, sensitivity analysis and simulation of potential effects of the contemplated compromises (Kersten 1997, Teich et al. 1994, Bui 1994).

NSS can be classified into different categories based on the amount of support and degrees of intervention in the negotiation process. Primary NSS aim at facilitating performing qualitative analysis during part of the negotiation process and/or facilitating communication between negotiators (Anson & Jelassi 1990). Higher category of NSS aim at supporting negotiators in their decision using multi-criteria decision-making techniques, Bayesian networks, multi-attribute preference elicitation techniques and decision trees. For example, MEDIATOR (Kolodner & Simpson 1989) uses case-based reasoning to support negotiators and mediators in conflict resolution, where previous cases can be remembered and adapted to fit new situations.

Another type of NSS employ knowledge-based systems technology to provide advice and explanations throughout the negotiation process. Negoplan (Kersten et al. 1991) system was developed for the purpose of negotiation support implemented as an expert system shell with a solution generator, a forward chaining inference engine and a restriction enforcer. Several experiments were conducted adopting Negoplan including simulation of Camp David negotiation (Kersten 1988, Neale & Bazerman 1991), negotiation with a hostage taker (Kersten & Michalowski 1989), labour negotiation (Matwin et al. 1989) and analysis of foreign investment negotiation (Cray 1994).

Matwin et al. (1991) developed GBML, which simulates the negotiation problem and generates negotiation rules and tactics based on genetic algorithm using the specified relevant issues as input.

To formalise the process of negotiations in organisations, Woo & Chang (1990) employed speech-act theory for message transmission, and the system is automated with the use of appropriate domain knowledge. INSPIRE (Kersten & Noronha 1997) is a Web-based system developed in the context of cross cultural study of negotiation and decision making based on analytical models.

Based on previous studies (Bazerman 1990, Jelassi & Foroughi 1989), the main issues important to the design of NSS can be outlined as:

1. Define the problem appropriately and separate the parties involved from the problem.
2. Provide communication for negotiators.
3. Help negotiators to identify all relevant issues and assign relative value to each of these issues.

4. Identify possible courses of action (and generate alternatives for mutual benefits of the parties involved). An optimal search to identify alternatives should continue until the cost of the search outweighs the value of the added information.
5. Use objective criteria in rating each alternative on each issue identified. Assess the potential consequences of selecting each of the alternative solutions on each of the issues and compute the optimal decision.

In this review, the contribution of NSS to negotiation analysis and support is clearly more than just considering the negotiation outcome, which is the predominant focus of game theory. The ideas used in considering the negotiation process, the use of decision criteria and interactivity designs can be adopted for more automated negotiation system.

2.1.3 Other Computational Approaches

Negotiation has been a topic of general interest for researchers in Distributed Artificial Intelligence (DAI) field. The focus of these studies has mainly tackled computational questions that further co-ordination and co-operation, while recent research has begun to study the less co-operative self-interested agents paradigms. Work in self-interested negotiation agents will be reviewed only, since albeit the co-operative approach of most DAI research is not applicable if the business negotiation orientation is viewed as competitive. Differences between competitive (or distributive) and collaborative (or integrative) orientations in negotiations result in different goals, strategies and tactics. There is considerable support for this argument and that these differences affect negotiator's outcomes (e.g. Putnam (1990)). Competitive negotiation in commerce can be viewed as the situation of a zero-sum game where as the value along a single dimension shifts in either direction, one party is better off and the other is worse off (the Win-Lose situation). Whereas, negotiations that are characterised by collaborative problem solving orientation emphasise similarities between negotiators and have as their aim the identification of common goals.

Rosenschein and Zlotkin (1994) have proposed evaluation criteria for automated negotiation agents' protocols in terms of efficiency, stability and simplicity based on game theory.

The Contract Net Protocol (CNP) (Smith & Davis 1981) for decentralised iterative task allocation negotiations, developed in DAI, was initially applied for co-operative applications. Sandholm (1993) discussed a formal method for making task announcing, bidding and awarding decisions to extend the CNP. Issues that arise in negotiation among self-interested agents whose rationality is bounded by computational complexity are addressed in a later paper (Sandholm & Lesser 1995). The paper presents a protocol that allows agents to be able to choose the level and the stage of commitment dynamically through conditional commitment breaking penalties. The implication of bounded rationality is also analysed.

Sun and Weld (1995) present a simple economic model of electronic commerce and test several bargaining strategies developed under the model. Reported results suggest that complex adaptive strategies (such as Undercut Competitor) perform significantly better than simple discounting approaches.

Kasbah (Chavez & Maes 1996, Chavez et al. 1997) is a Web-based marketplace system where autonomous user-created software agents negotiate for the selling and buying of goods on behalf of their users. The agents developed for the prototype do not use any AI or Machine Learning techniques, but rather the user specifies the negotiation strategy the agent uses. The user can choose a decay function (linear, quadratic or cubic) for the agent to use to lower the asking price over its given time frame. Feedback from experimental results was generally positive, but agents sometimes act in a clearly stupid way (such as accepting an offer when a better one was available).

To build autonomous software agents that improve their negotiation skills based on Bayesian learning mechanisms, Zeng and Sycara (1996) adopt the sequential decision making paradigm. The presented model "Bazaar" provides a computational framework that addresses multi-issue, multi-criteria negotiation and incorporates learning and operational algorithms to guide negotiation.

Several researchers have studied evolutionary computation's ability to automatically discover effective negotiation strategies. Genetic Algorithms (GA) and Genetic Programming (GP), two new techniques in evolutionary computing, were adopted to evolve effective strategies in the context of negotiation. The approach used is to represent a population of initial strategies as a set of finite-state automata (or to start with randomly generated strategies). The agent evaluates these strategies by playing against other agents' strategies. The fitness of each strategy in a population increments or decrements in accordance to the payoff of the strategy in that given round. At the end of the round, retaining "parent" strategies with highest fitness, and creating new "children" strategies through mutation and crossover between the selected parents creates the next generation of population. These children and the chosen parents, which are just new possible solutions, enter new rounds of evaluation, and the cycle continues for the specified number of generations.

Dworman et al. (1996a, 1996b) conducted a series of experiments in which they used genetic programming approach to automatically discover effective negotiation strategies. Results show that after running the system for 4000 generations, these adaptive agents co-evolve to produce coalition agreements that approximate the co-operative game-theoretic solutions, but they fail to exploit opportunities to "seize the chance" present in their opponents' strategies.

To discover automatically effective negotiation policies, Oliver (1996) presents a system of artificial adaptive agents using genetic algorithm approach. The reported simulation uses a set of initial randomly generated strategies using simple sequential threshold rules. Offers made by each player are communicated via messages. The performance of the adaptive agents is investigated across several types of games such as distributive bargaining problem, integrative bargaining problem, labour negotiation and stylised international business negotiation. Results show that typically after 400 trials (20 generations of 20 rounds), the performance of the artificial agents was similar to the human performance in the same negotiation case. The paper concludes with an outline of the requirements for a working system for electronic commerce.

To summarise, negotiation protocols that have been proposed in the DAI literature were designed based on toy scenarios (for research purposes) and do not support many of the semi-structured tasks or functions of real life business negotiation. There is a need to develop a flexible negotiation protocol that would support business work and modelled after human negotiation behaviour. The design should build upon the distinctive competencies of negotiators while compensating for their inherent weaknesses. A promising approach for negotiation protocol should benefit from combinations of different disciplines such as speech act theory in linguistics and negotiation research in psychology and sociology to model the negotiation after human negotiation behaviour. At the same time, some decision theoretic techniques to evaluate options should be applied.

2.2 INFORMATION DISCOVERY

With the fast growth of the Internet and the distribution of information, interest in research in automatic resources and information discovery has increased. This section surveys the current approaches to information discovery. Here the term "Information Discovery" is used as an amalgam of Information Retrieval (IR) and Information Extraction (IE) to describe the task of locating and extracting needed information expressed by the user as a query from a collection of documents in large volumes of information sources.

For IR, both classical IR techniques, i.e. statistical IR methods, and semantic information and machine learning techniques are addressed.

The field of IE has a rich literature, see Cowie & Lehnert (1996) for details. But since traditional IE refers to the task of identifying literal segments from a given text that instantiate some relation or concept, this thesis uses of the term differs in two ways. The first difference concerns our focus on regularity found in Internet resources to guide the extraction. Therefore this thesis assumes that the information to be extracted is delimited by appropriate HTML tags and, hence, the thesis assumes that the given text is structured. A second difference is that many knowledge intensive approaches to IE are slow and thus best suited for offline-extraction, while an information extraction agent must execute online and therefore need to be fast. Because the documents in which this thesis is interested in, i.e. HTML pages, have some kinds of structural regularities, only a review of the wrapper generation approach to information extraction that is designed to exploit these regularities to guide the extraction is presented.

2.2.1 Statistical IR Methods

Despite the fact that IR is currently commonplace with the explosive growth of the Web and the plethora of the Internet search services, the techniques used in mainstream IR applications are similar to the traditional techniques used by IR specialists 20 years ago.

Obviously, one way to locate documents containing specific string (user's query) is based on Boolean search strategy. In this method, each document is scanned searching for the specified term. The full text scanning method is not feasible for searching large document bases, since it is not efficient to scan each document in the database. Therefore, full text scanning is used in association with another IR method (such as the inversion) to limit the number of documents to be searched.

The general approach taken in statistical IR methods is to represent both the query and documents as lists of index terms. The matching criterion is to count the number of common terms between the query and documents using some measuring functions.

There are a number of IR methods that employ statistical techniques including:

- Inversion method: In this method each document is represented by a list of keywords describing the content of the document. These keywords are stored in an index file (hence indexing) together with a list of pointers to the qualifying documents of each keyword. Various methods are used to construct and maintain the index file, such as Binary-trees and hashing (Knuth 1973, Lesk 1978).

  The inversion method is relatively simple to implement, it is fast in retrieving the relevant documents and the support for synonyms can be easily embodied. For these reasons, this method has been adopted in most of the existing Internet search engines. Researchers have studied techniques to attain fast updates to the indices that could be very huge in size (Tomasic et al. 1994, Brown et al. 1994).

- Clustering method: In the methods used to represent clusters, a match to the query is based on a clustered document collection rather than on each document. IR techniques that use clustering method involve both the cluster generation and cluster search tasks.

  In the cluster generation procedure, each document is processed and keywords are extracted to represent a vector model for the document (also called indexing the document). Salton (1971) proposes the following steps for an automatic indexing procedure for a document:

  - Remove stop/ common words in the document (such as "the", "and" and "for").
  - Stem each word in the document (for example "reformation" is stemmed to "form").
  - Assign each stemmed word to a concept class.

This way a t-dimensional vector represents each document, where t is the number of concepts. Presence of a concept (term) in the document is indicated by a positive number (weight) and the absence of the concept is indicated by zero. Different weighting functions were proposed, of which *tf.idf* (Term frequency * inverse document frequency) is the most prevalent weighting function in use. The second step in document generation is to distribute these vectors represented by a point in the t-dimensional space into groups of classes. The partitioning procedure should be theoretically sound and efficient (Van-Rijsbergen 1979).

In the cluster search procedure, the input query is represented as a t-dimensional vector and compared with the cluster-centroids. Closest relevant clusters to the query, whose similarity with the query vector exceeds a given threshold, are searched. In order to measure the similarity between a query and a collection of documents, a number of metrics have been proposed and tested for this purpose. The metrics are easily adapted to provide a numerical distance between the documents and the query. The cosine similarity measure by Salton (1971) is the most widespread measure in use. The idea behind the cosine measure is that the angle between two vectors in a t-dimensional space is always measurable across 2-dimensional plane. Hence, when the vectors point in the same direction (parallel document and query) the angle between them is zero. When the vectors are perpendicular to each other the angle between them is 90^0 (and cosine 90^0 is 0). In the vector space model the cosine measure is equivalent to the cross product of the two vectors.

In addition to term weighting and document ranking, relevance feedback techniques (Rocchio 1971) can be used to improve the effectiveness of the search results. In the relevance feedback, the user judges the relevance of the presented documents to his query. These explicit remarks on the relevance of the document are fed back into the system to dynamically re-formulate the query vector and term weighting causing re-ranking of the documents. Relevance feedback is also used in probabilistic retrieval methods.

Query expansion techniques are usually used as follow-on to relevance feedback, which could be either manually or from a static structure such as a thesaurus.

Other new approaches include data fusion (Belkin et al. 1995) for IR wherein a combination of several retrieval strategies are used, has found particular interest in the Text Retrieval Conference (TREC) requirements. Data fusion involves a combination of rankings from several document-ranking approaches into one consolidated ranking. This may also involve generating more than one version of the same query to run against a document collection and then combining the retrieved document rankings. Data fusion has been consistently shown to lead to effective IR.

2.2.2 Semantic Information and Machine Learning for IR

Traditional lexical matching techniques that try to match words in the document with the query do not cater for polysemy (words having more than one meaning) and synonymy (more than one word having the same meaning). This could result in impairing the retrieval precision and the recall levels from the document collection. Precision is the ratio of the number of relevant documents to the total number of retrieved documents by the system. Recall is the proportion of all relevant document collections that are retrieved by the query.

Semantic information techniques are used to identify suitable index terms (rather than keywords) to represent documents that could be done on several levels. One level is to identify the word-level equivalence between words in the query and the indexed documents. Another level is to try to identify the conceptual-level equivalence between the query and documents (this could be phrase comparison).

Some IR methods that use semantic information and/or machine learning techniques include:

1. Latent Semantic Indexing: LSI (Berry et al. 1995, Shippy 1995) attempts to address the word-matching problem, faced in traditional lexical matching methods, by using fuzzy searching techniques to locate documents that have the same conceptual meaning as the query.

   The key idea behind LSI is to automatically analyse the semantic structure of the documents in the database, extracting keywords as terms to index the documents. A term-by-document matrix is constructed to define the relationship between terms and documents. Documents containing word usage patterns similar to the query are considered relevant. Since LSI presumes that the relevancy is partially obscure by variability in word usage (word matching problem), a Singular Value Decomposition (SVD) of the sparse term-by-document matrix is used to associate terms to documents by using only the k-largest singular values and corresponding singular vectors. The k-dimensional conceptual vector space is constructed from the singular vectors, so that each term or document can be represented as a point in the space. This reduced vector space captures the implicit high order structure in association of terms and documents and eliminates much of the word-matching problem. LSI uses similar statistical methods for term weighting and similarity metrics functions.

   LSI method, although computationally expensive, has been shown to enhance the overall IR effectiveness.

2. Natural Language Processing: Researchers have studied a variety of approaches to integrate natural language processing with IR systems (Ram 1991, Jacobs & Rau 1988). The central theme in the literature is that some , perhaps shallow, variant of the kind of syntactic and semantic analysis performed by general-purpose natural language processing system can

provide useful information to enhance the performance of the indexing and thus the retrieval of the documents.

Strzalkowski et al. (1995) presented in TREC-3 IR system that uses advanced natural language processing to improve the effectiveness of term-based document retrieval. The system consists of a pure statistical core engine and advanced NLP module. The statistical engine contains the indexer module, which builds the inverted index file from pre-processed documents, and a retrieval engine that searches and ranks the documents in response to users' queries. The NLP module is used to pre-process the documents to extract content-carrying terms and to discover the inter-term dependencies and built conceptual hierarchy specific to the database domain. The NLP module also processes user's natural language requests into apropos search queries.

Bear et al. (1998) presents an approach to apply NLP system to act as a post-filter on the output of an IR system to extract needed information. The idea is to cascade an IR system with the information extraction system in such a way as to enhance the performance of routing (filtering) tasks.

CRYSTAL (Soderland 1997a) takes as an input a set of labelled examples of documents and a set of description features for those documents, and using linguistic features such as part-of-speech tags it learns information extraction rules that are triggered by these linguistics techniques. WEBFOOT (Soderland 1997b) extends CRYSTAL to handle non-linguistic Internet–based documents. WEBFOOT implements a set of HTML-specific heuristics that are useful for Internet-based information extraction. Those heuristics are used to identify the text fragments to be extracted.

3. Neural Networks and Genetic Algorithms methods: Neural networks methods use connectionist models for the purpose of IR that include several approaches, such as spreading activation models and associative networks (Doszkocs et al. 1990). The technique is to generate an automatically created

weighted network of keywords from documents and integrate it with a thesaurus (either manually or automatically constructed).

In the genetic algorithms methods, documents are pre-processed and keywords are associated with it. Associated keywords are altered over time by using genetic mutation and crossover operations based on the score matching fitness measure. This would evolve until getting the best keywords that describe the document (Gordon 1988).

For large-scale real life applications, such as the Internet, neural networks and genetic algorithms methods are not suitable. These methods suffer from requiring extensive computation time and lack of interpretable results.

2.2.3 Wrapper Generation for IE

The notion of "wrapper" is used to refer to information extraction procedures constructed to extract relevant data from semi-structured resources. Generating wrappers for databases and Web resources and providing database-like querying for semi-structured data are research areas that have received significant attention.

There are substantial works that address building wrappers to integrate heterogeneous information sources and encapsulating legacy databases in a software layer that provides a standard interface (for example, Shklar et al. (1995) and Roth & Schwartz (1997)). Hammer et al. (1997) developed a template-based approach to generating wrappers for legacy systems and Web resources. In their approach, the user provides actions for the system to execute when a query matches a certain template. The approach provides a way to quickly constructing wrappers by example, but it could require a large number of examples to determine a proper template for a single resource.

ARIADNE (Ashish & Knoblock 1998) is a semi-automatic system for constructing wrappers for hierarchically structured Web documents. Their approach employs powerful HTML and domain-specific heuristics for postulating the structure of the document. For example, relative font size is usually a suggestion for determining when an attribute ends and secondary attributes start. Several other heuristics such as the use of bold or italic, relative indentation and sequence of alphanumeric followed by a colon are used by the system to postulate the structure of the document. Once the hierarchical structure of the document is determined, the system generates a grammar consistent with the observed structure. During the structuring step, the user has to manually correct the system's postulates of the document's hierarchical structure. Ashish & Knoblock (1998) report that only a small number of correction steps are needed in practice. After structuring the source, a parser that can extract selected sections for a document in the source is built.

Kushmerick et al. (1997) developed an approach to automatically generate wrappers for Web resources. Their approach uses inductive learning techniques to build a program that extracts relevant data from a given HTML document. Their wrapper induction approach involves generalising from a set of examples of a resource's documents, each annotated with the data fragments to be extracted. The system postulates a wrapper for labelling the document's relevant data. Their work does not address how to construct recognisers that can be used to generate examples for their inductive learning step, instead they assume that a set of recognisers are given.

2.3 Internet Search

As the Web grows, the number and variety of search services increases as well. Currently, there are more than 2,000 search services on the Web, most of them highly specialised and targeted at special subjects and media. Examples include Yahoo!, Lycos, Harvest, AltaVista, HotBot and WebCrawler. Yahoo! is a manually generated directory, which contains a hierarchy of items that have been categorised either by users' suggestions or handpicked. This type of service is precise and comprehensive for categories that are relatively small and static, but can not keep up with dynamic rapidly growing categories. Thus, the user will not be able to locate a particular page that has not yet been categorised by this service.

AltaVista (and most of the existing search engines) tries to exhaustively collect all available Web pages using automatically robot-generated (robots or spiders) Web indices. WebCrawler, for example, operates with multiple retrieval agents in a server-breadth-first search approach to index only the "good" pages using some handcrafted heuristics. The rational behind the usage of server-breadth-first search rather than document-breadth-first search (used by other search services) is that most servers have many related documents in a single subject area, rather than multiple subject fields.

AltaVista introduced the LiveTopics feature, whereby once a user's query has been initiated and a ranking of some of relevant documents has been generated, index terms from those ranked documents are then processed to determine the most customarily occurring one. Those terms are presented to the user as extra search terms candidate for query expansion. Some other services suggest manual query expansion terms, using the Trigger Pair Model (Gauch & Futrelle 1996), by presenting additional refinement keywords for the user to select from. The idea is that if a word S is significantly correlated with another word L, then the appearance of the word S in the text should make the appearance of the word L anticipated somewhere in the text with some confidence. For example, some of the trigger pairs for the word *product* (in decreasing order) are {*maker, company, corporation, industry, sale, manufacturer, sell, and* ..}. Query expansion techniques could provide considerable improvement in the retrieved results.

To relieve the user from the tedium of choosing the right search engine for a specific information need from a plethora of on-line search services, several different approaches were pursued. One approach is to build a meta-search method on top of existing search engines. Examples of meta-search services include MetaCrawler (Selberg & Etzioni 1995), ProFusion (Gauch & Wang 1996) and SavvySearch (Dreilinger 1996). These services, essentially operating in the same way, rely on existing search services to provide necessary information to fulfil users' queries. The idea is to query the underlying engines in parallel and then collate the results to the user. The difference in the operation of these services is in the processing they perform before querying the underlying engines (pre-processing to formulate the query) and before presenting results to the user (post-processing to rank and format results).

Another approach is to use more sophisticated agent-based AI systems. Softbot (Etzioni & Weld 1994) provides an integrated interface to a variety of Internet services and utilities, wherein the softbot dynamically chooses which facility to invoke and in what sequence to achieve a specific goal (satisfy a request). Amalthaea (Moukas 1996) is an ecosystem of multi-agents for personalised information discovery and filtering. Information discovery agents are based on genotype that contains information on the number of keywords they should use when querying existing Internet search engine (meta-search). Phenotypes of discovery agents include the history of transactions with different filtering agents and how well they performed in those transactions. Amalthaea uses evolution techniques to continually adapt to the user's interests.

MACRON (Decker et al. 1995) is a multi-agent system built on top of co-operative information gathering (CIG) searchbots and uses a centralised planner to generate sub-goals that are pursued by a group of co-operating agents using Knowledge Query and Manipulation Language (KQML). The system embodies low-level retrieval agents responsible for retrieving and processing URLs for higher level planning agents and interacting with existing search engines (again meta-search) and online databases.

2.4 Internet Shopping

There is a substantial commercial interest in software agents for Internet shopping. Examples include Jango (www.jango.com), BidFinder (www.vsn.net/af), Computer ESP (oracle.uvision.com/shop) and the Shopping Explorer (shoppingexplorrer.com). Jango, which is the commercial product of the ShopBot (Doorenbos et al. 1997) research prototype, is the most interesting comparison-shopping system.

ShopBot is a full-fledged domain-independent autonomous comparison-shopping agent. ShopBot learns how to shop at several on-line stores by extracting vendor's content, how to query the vendor and how to identify unusual conditions (e.g. product not found). ShopBot embodies knowledge base containing domain descriptions and URLs of possible on-line vendors. Elements of the domain description are attributes of the products in the domain, heuristics for understanding vendor's page (regular expression for recognising these attributes) and seed knowledge to support learning (description of several popular products in the domain to be used for inductive learning).

Given the domain description and the URL for an on-line vendor, the ShopBot learner function is to automatically generate vendor-description for this vendor. Basically, the learner learns how to query the vendor and how to parse the vendor's response to the query. ShopBot learner operates by looking for patterns in the HTML source of the example document. It first removes the head and tail information in the HTML and then partitions the example page into a sequence of logical records (the system assumes that these records are separated by visually salient HTML tags such as <LI>, <P> and <B>). Each record is then abstracted by removing non-HTML characters, generating a signature for each record. For example, ShopBot would generate the following signature:

```
<LI> text <a> text </a> text <BR>
<LI> text <a> text </a> text <BR>
                    .
<LI> text <a> text </a> text <BR>
```

For the following HTML source:

```
<LI> Microsoft <a href = "http://microsoft.com/encarta/"> Encarta </a> for $39.99 <BR>
<LI> Grolier <a href = "http://grolier.com /"> Grolier</a> for $29.99 <BR>
                    .
<LI>Britanica <a href = "http://britanica.com/"> Britanica </a> for $299.99 <BR>
```

ShopBot uses domain specific heuristics to rank these signatures. For example, the presence of a price in a record suggests that the generated signature is correct. ShopBot extracts the prices from these records for comparison-shopping.

The basic architecture of the ShopBot reveals a number of limitations that include the problem of scalability with the Web growth. ShopBot is limited to the comparison-shopping tasks and does not include recommendation or negotiation tasks. ShopBot is also limited to on-line stores that provide a searchable index and relies on a very strong bias towards environmental regularities in HTML pages.

2.5 SOFTWARE AGENTS

Software agents is a rapidly developing area of research. Although there is no standard definition for agent on which consensus exists, here the term "software agent" is used to refer to *any software entity that execute a specific set of operations on behalf of a user (or another program), and in doing so employing some degree of intelligence, autonomy and perception of the user's goals or desires.* Using this definition, agents then can be described in terms of several properties and classified based on the degree and variety of attributes vested in the agent. Some of the customarily (Wooldridge & Jennings 1995, FIPA 1997, Nwana 1996) identified agents' properties are:

- **Autonomy**: agents should exhibit some degree of independence or autonomy without direct intervention from their users. The degree of autonomy is enhanced if the agent is able to take pre-emptive initiatives and carry on an agenda independently of the user.

- **Social Ability**: Agents require the ability to interact with other agents as well as with their users in their course of achieving their goals. In the collaboration between the user and an agent, the user imparts his desires and the agent acquaints what can be achieved and provides progress results. Agents interact with other agents to perform tasks of common-interest, which are beyond their individual capabilities. Inter-agent co-operation requires that agents have the ability to communicate to exchange knowledge and plans to achieve the goal.

- **Reactivity**: Agents should be able to discern their environment and react in a timely manner to changes that occur in it. Reactive agent responds in a stimulus fashion to the present state of his environment using a set of triggering conditions to re-evaluate his belief space, and hence continually adapting to changes in the environment.

- **Proactiveness**: Agents should be able to act in a goal-directed mode and take initiatives where appropriate. The degree of pro-activeness of an agent is related to the scale of the reasoning and learning abilities.

- **Mobility**: Agents capable of moving to other environments are called mobile agents. Mobility is the degree to which agents themselves roam through the network. Whereas some agents are static (residing on one machine), other mobile agents may be instantiated on one machine and transported to another machine in the middle (or before) of execution while carrying accumulated state data with them. Implementing mobile agents require careful considerations to address security, privacy and transport mechanism challenges.

- **Personalisation**: Since agent's primary task is to execute a specific set of operations on behalf of the user, the agent should be personalised for that user. The degree to which an agent is personalised depends on a set of characteristics. Those include the level to which the agent captures and maintains the complete profile of the user, the degree the agent requires user-specific information and the degree to which the agent makes rational unique actions based on user's profile.

- **Trustworthiness**: The client (user) would not relinquish some of his responsibilities to an agent unless he makes sure that the agent will not run amok. So for an agent to gain the acceptance of its client it is essential that the agent can represent accurately the client. The client must be highly confident that his agent will not lie or communicate wrong information (veracity), and that the agent will always act for its client's own benefit (benevolence) (Wooldridge & Jennings 1995). The client must also weight his confidence that the agent will do the sane thing against the risk that it will do it wrong. Agents should be also reliable and exhibit graceful degradation in cases of communication or domain mismatch (the two parties are not communicating well or out of context, and they are not realising this fact) (Foner 1993).

2.5.1 Multi-agent Systems

Multi-Agent System (MAS) is a network of loosely coupled agents that work collectively to perform global tasks that are beyond their individual capabilities. The approach of MAS widens the notion of software agents to include automating tasks delegated by other agents in the system as well as end users by incorporating communication mechanisms for interactions between agents in the system. Agents in the MAS interact to co-ordinate their knowledge, skills, plans and desires to take action or perform some tasks. Accordingly, the design of MAS should address several issues as follows.

1. **Communication:** Agents in a MAS need the ability to communicate to exchange information to co-operate and co-ordinate their actions. Several techniques have been proposed for agents in a MAS to communicate with each other.

   One approach is to implement direct one-to-one communication mechanism using a protocol such as TCP/IP. Direct communication implies that the agent should be aware of other agents in the MAS in order to communicate with them. An agent wishing to communicate with another agent must obtain the address of the other agent. These addresses may be obtained either from previously broadcast messages announcing the address of each agent in the system or from a centralised look-up agent directory.

   Another approach is to organise the agents in the MAS into federated system (Genesereth & Ketchpel 1994) and communicate through facilitator agents. In the federated system, each set of agents have a facilitator (mediator) agent which takes the responsibility of the communication between agents under its control and other facilitators in the MAS. This approach is recommended for MAS containing very large number of agents.

   Third approach is to use broadcast communication mechanism to broadcast a message to all agents in the MAS. Blackboard systems (Werkman 1990, Kearney et al. 1994, Tsvetovatyy & Gini 1996) can be also used to facilitate communication between agents in the MAS. Agents use the blackboard (repository) to post messages to other agents in the MAS and also to obtain posted information.

2. **Interaction:** For agents to communicate correctly they need to understand a common language (lingua franca). The importance of having an expressive common Agent Communication Language (ACL) (Genesereth & Ketchpel 1994) where agents can communicate with their peers by exchanging messages and interact together through explicit linguistic actions is ratified. In

MAS, the ACL should be independent of the agents. Thus each agent in the MAS should implement a linguistic layer to provide a common message interface, that is independent of the agent's semantics and internal representation. This implies that each agent should cognise what knowledge to represent for communicating and how to do so. ACLs derive their inspiration from speech act theory (Austin 1962) to build the linguistic layer and formalise the linguistics actions of agents.

Many of the existing MAS applications employ an *ad hoc* set of performatives within *ad hoc* ACL. These performatives are used to allow an agent to convey its knowledge, desires and intentions. The problem of such ad hoc approaches is that it makes it difficult for agent applications designed by different entities to inter-operate. MAS involving standard ACL is an improved approach to solve agents' interaction and interoperability problems.

The Knowledge Query and Manipulation Language (KQML) (Labrou & Finin 1994) is an evolving *de facto* general-purpose standard ACL. The KQML consists of the content, message and communication layers. The content layer specifies the actual content of the message. KQML offers an extensible set of performatives, which constitute the message layer, to specify the types of communications the agent can have with others (e.g. ASK-IF, ADVERTISE, REPLY). External Interface Working Group (1993) gives a list of the reserved performatives for KQML.

KQML, however, requires further work to provide a specific semantics for the language. Without a precise semantics, agents' designers can not be certain that their interpretation to a performative is in fact the same as what the other designer intended it to have. Cohen and Levesque (1997) identify three general difficulties with KQML. First, the definitions of the performatives suffer from ambiguity and vagueness. Second, there are missing performatives, such as the commissives class of actions that commit an agent to a course of action (for example, promises and accepting proposal).

Third, there are misidentified performatives that should instead be classes as directives (for example, a request) or assertives (for example, informs).

3. **Co-ordination**: In MAS, agents co-ordinate in order to ensure that the overall community of individual agents are in harmony and acting in a coherent manner. Since no agent possesses a global view of the environment to which it belongs, local views and goals of an agent may conflict with other agents' actions and therefore may generate into a collection of chaotic incohesive individuals. The coordination of agents' behaviour is essential to prevent chaos during conflicts. Co-ordination is also vital when there is a need to meet global constraints if the system to be deemed successful (such as time constraints). Coordinating independent activities that may occur when the individual goals of the agents are related is necessary. Further reason as to why actions of multiple agents need to be co-ordinated is to blend the capabilities and expertise of the agents to solve a global problem.

Several approaches have been suggested to bring about coherence and co-ordination in MAS. One way for assuring co-ordinated behaviour is to have knowledge of possible interaction processes that could take place among agents (Corkill & Lesser 1983, Fox 1993). Agents form a plan that specifies all their possible future actions and interactions with respect to accomplishing a particular goal (Jennings 1995). These plans are formalised within a set of conversations that take into consideration the alternative courses of actions during execution based on discretion action/reaction of other agents.

Another approach is to exploit an organisational or hierarchical structure for agents in MAS by granting one agent that has a wider perspective of the system the master role. This technique renders master/slave architecture for resource and task allocation among slave agents by the master agent. The master agent could collect information from slave agents, create plans and allocate tasks to individual agents to assure global coherence.

Another approach is to view the problem of co-ordination as a multi-agent planning problem, and then employ one of the two types of planning strategies, namely, distributed or centralised planning. In distributed multi-agent planning, agents communicate in order to construct and update their individual plans and their perception of others' until all conflicts are resolved (Lesser & Corkill 1981, Georgeff 1984). In centralised multi-agent planning, individual agents form their plans and send them to a central co-ordinator who analyses the plans, finds potential conflicts and finally merges them into a global complete plan with conflicting interactions eliminated.

2.6 SUMMARY

This chapter has reviewed the research areas relevant to this research topic (i.e. Multi Agent System for Consumer-Oriented Electronic Commerce). Figure 2.1 shows the related research areas that the chapter has addressed. Since the Internet is the target environment for the system, the chapter has also addressed various techniques that are currently used for searching the Web and the current approaches used in systems for comparison shopping over the Internet.

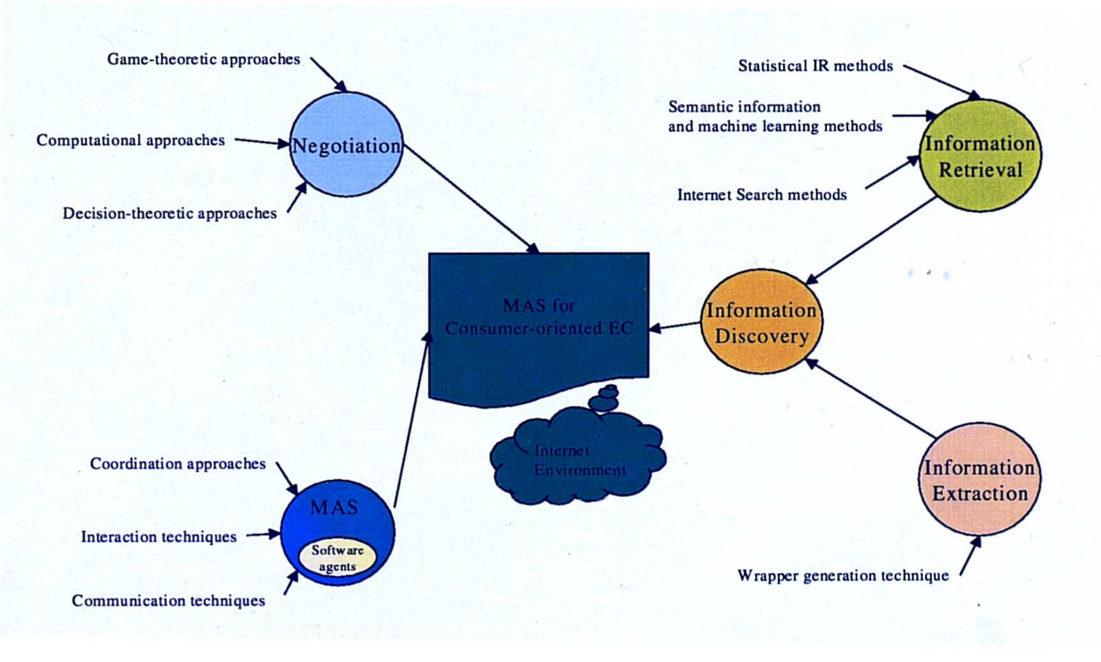


FIGURE 2.1. RELATIONSHIP BETWEEN REVIEWED RESEARCH AREAS AND THE THESIS SCOPE.

CHAPTER 3

RESEARCH METHODOLOGY

In this thesis, the discipline of computational science is exploited in which computers are used to provide insights into the behaviour of the perceived consumer-oriented electronic commerce model. The framework of computational science emphasises the development of computationally feasible models (applications) for physical systems, developing algorithms for solving issues arising during the modelling process and matching the algorithms to computer architectures (Stevenson 1994).

From Stevenson's perspective, complex econometric models (such as bargaining) are legitimate applications for computational science study. Computational models can be used to investigate complex models that can not be solved analytically. To put it differently, problems that are analytically intractable are approachable by computational means. However, computational models lack the strong proclamations that can be concluded when using mathematical proof techniques.

To illustrate, following is a more careful examination of the negotiation context. Recall that one of the behavioural game theory vigorous findings is that humans deviate from optimal behaviour, often violating tenets of rationality, and that the divergence occurs often in a predictable manner. Still, humans' performance is also typically sub-optimal and they need many trials to learn to approach optimal behaviour. Many game theorists have tried to model some of these findings by embodying especial types of non-rationality or certain learning mechanisms into formal models. However, the consolidated results of these efforts only chew at the overall problem and merely provide poke of understanding. The crucial inadequacy of the analytical approach is that tractability often requires non-economic presumptions (Oliver 1996).

Holland and Miller's (1991) view of computational science complement this perspective. Their idea is that computational science in general, and specifically software agents, occupy a vacancy in the study of complex phenomena. At one hand, purely descriptive models that apply linguistic representation can have problems of precision and consistency (Aumann 1989). On the other extreme, punctilious mathematical models, although precise, typically encompass only simplified or stylised interpretation of the actual system. Therefore, solid results that can be deduced when using analytic models have to be justified for their real life validity. Besides, many analytic models are not flexible in that slight variations in hypothesis can make the model un-resolvable. On the contrary, software agents' representation is rich and flexible and the dynamics are precise and observable. Moreover, since the agents exist in a virtual environment, the researcher has control over the configuration and parameters of the experimental situation.

In this thesis empirical approach, a multi-agent computational environment is used to model the behaviour of a consumer purchase over the Internet. The system models the decision making process of the consumer to recommend the preferred buy that would satisfy the consumer's needs based on a given utility function. The system will be also used to simulate the negotiation phase (since, currently, no supplier provides an online negotiation agent to interact with) in which the MAS engage in alternating offer bargaining with several assumed suppliers' agents. The computational approach allows the investigation of complex situations such as the satisficing (satisfactory and sufficient) behaviour observed in human decision making that runs contra to the assumption of game theory that firms maximise profit. One explanation to the satisficing behaviour would be that decision making involves bounded rationality, in which decision-makers do not have the complete information and/or decisions have to be made within a limited time frame. It is the interplay of experiment, traditional theory and computational modelling which makes the computational science approach enormously fruitful and provide strong results.

To comply with the perspective of computational science, the design of the system is described in Section 3.1 in which the perceived MAS framework to automate the consumer decision making process is presented. Then, a general description of each of these identified tasks is provided. This would provide an overview of the design requirements for the agents in charge for these tasks, to be addressed in Chapter 4. The chapter concludes by presenting the measuring criteria for successful answers to the research questions.

3.1 SYSTEM DESIGN

To facilitate the investigation of the possibility and feasibility questions raised in Chapter 1, a multi-agent system for supporting consumer-oriented electronic commerce is designed and implemented. To satisfy the perspective of computational *science, the congruous mix of applications, algorithms and* architectures is described as follows.

- **Application**

  The application is a system to help the consumer to make the rational Internet purchase that would satisfy his needs. The prototype system is implemented for experimental studies (that requires further work to be a fieldable practical system) and has at its core a set of software agents for the purpose of Internet shopping. However, the purpose of this thesis is to study the potential of software agents in retail EC rather than creating a fieldable system for EC. The principal efforts are towards creating a computational platform that is a research tool that can be used to evaluate a concept. The task of creating a fieldable system for retail EC is a different assignment that needs in addition to addressing the usability issue, which will be tackled, to further address logistical and maintenance issues. Few experiments that are designed to answer the research questions are presented and the results of these investigations are reported.

- **Algorithms**

  A set of algorithms are developed for the identified tasks in the consumer decision making process. First, a set of text analysis algorithms and IR methods are employed in the search for the required items. A meta-searching method utilising existing Internet search services was adopted, while different pre and post processing methods were implemented to increase the accuracy of retrieving relevant information. Second, a number of information extraction techniques were used to identify relevant data and attributes in a given HTML document. Some recognition heuristics and wrapper generation method for this task are developed. Third, a protocol and strategy for negotiation are designed. Algorithms for sustaining the designed strategy that would conform to the protocol are developed.

- **Architectures**

  The software is written entirely in Java using object-oriented approach. Since Java is claimed to be platform-independent, the application could be virtually used with any hardware. Several PCs are used to perform all the testing of the model. Appendix C gives the high level object-oriented design of the system.

The following subsection discusses the framework in more details.

3.1.1 Framework for Multi Agent System to Automate Consumer Decision Making Process

The proposed architecture for automating the consumer decision-making process is based on co-operative multi-agent paradigm as shown in Figure 3.1. In the framework, the consumer (user) input provoke the creation of initial plans which results in the deployment of multiple co-operative search agents for the purpose of achieving primary goals set by the planner agent. Subsequent plans are created based on the search outcome, resulting in the deployment of multiple co-operative suggestion agents for achieving sub-goals set by the planner. These agents could use existing search services and information resources on the Internet (such as the Web, newsgroups, on-line journals and magazine databases) to accomplish their assigned objectives. Agents may co-operate and co-ordinate with one another to achieve the goal set by the planner. Agents share information with one another via messaging mechanism.

Based on the search outcome, the planner agent employs multiple information extraction agents to extract relevant data in each of the results' documents. Information extraction agents use wrapper generation technique to provide a wrapper for the document of interest to automatically extract relevant data. Generated wrapper for online stores' Web pages is used to extract available information of the required product that is used for comparison-shopping. The agent utilises information and seed knowledge stored in the knowledge-base that include descriptions of several popular products and their specific attributes.

The planner agent also deploys multiple co-operative negotiation agents to negotiate with the evoked set of potential suppliers' agents over the Internet. Alas, the negotiation phase is simulated, because currently there are no suppliers who provide online negotiation agents for this purpose. In the stylised negotiation, agents use different negotiation strategies and apply some objective measures to aid decision-making. Multi-Attribute Utility Theory (MAUT) is applied to select the best choice from a given set of alternatives (proposed by potential suppliers) and, accordingly, may use this information to counter-propose.

The learning module is included so as to store user's and agents' experiences and strategies to evolve high quality results. To maintain autonomous design for the architecture, the feedback of the purchase experience (satisfaction, dissatisfaction or post purchase dissonance) is used as an input to the system. The user is asked whether the last purchase experience made by the system should be used as a basis in the current round. Though autonomous system is investigated, the possibility of a practical system, at least in the medium term, might not be fully autonomous.

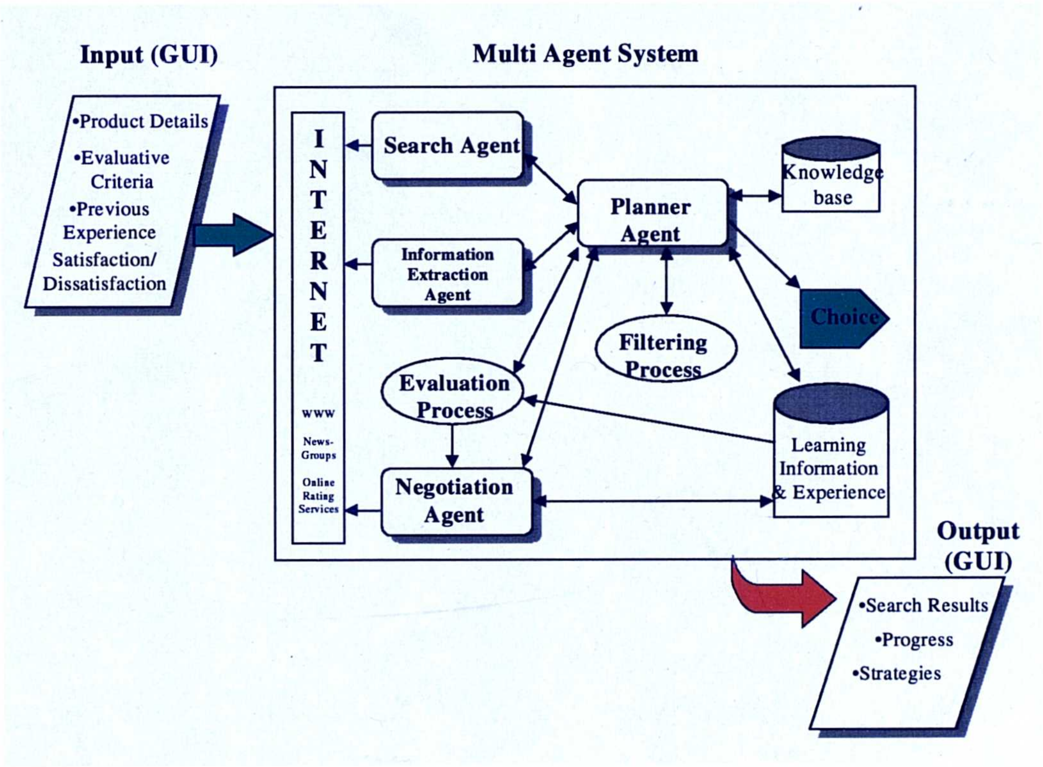


FIGURE 3.1. FRAMEWORK OF MULTI-AGENT SYSTEM TO AUTOMATE CONSUMER DECISION-MAKING PROCESS.

The benefits of adopting a multi-agent approach to automate the tasks of the consumer decision-making processes include:

- The use of concurrent parallel multiple agents for searching, suggestion and information extraction would substantially help to avoid the time constraint that could be set by the planner for each agent to achieve their assigned goal.

- An agent exchanges messages within structured conversations with other agents to co-ordinate their knowledge, skills, goals and plans jointly to take action. Agent interactions wherein one agent takes an action based on the presence or knowledge of another agent could use speech act language.

Example of agent interaction is that the knowledge of the search agent about the presence of a product-rating magazine could influence the actions of a suggestion agent.

- Co-ordination and interaction between negotiation agents would influence the decision-making strategies of those agents for the benefit of the consumer. The following scenario explains this benefit:

  Assume that one agent (B1) from the MAS community engaged in negotiation with one of the potential suppliers' agent (S1), and that the proposed asking price by the supplier's agent of the current round is X. The current asking price of another potential supplier (S2) negotiating with another agent (B2) is Y, where X > Y. Then, B1 would interact with B2 to co-ordinate their counter proposal for their counterpart agents before committing to any agreement. Agent B1 may inform supplier S1 that supplier S2 is offering the product at price of Y and this is subject to negotiation too. This is a unique feature that could greatly increase the quality of negotiation agreements for the benefits of the consumer (but could have the reverse effect on the supplier).

3.2 Meta Searching for IR

The proposed approach is to deploy meta-service search agents to leverage existing services and collate their results. The search phase employs multi-threaded parallel Web search agents utilising existing search engines and other Internet services. These agents are responsible for finding, retrieving and processing URLs for the planner agent. Those agents would use domain specific knowledge about existing on-line services to determine how to interact with search engines and query forms and how to properly collate the returned results.

To improve the relevancy of the retrieved results, heuristics to select meaningful and non-ambiguous keywords should be applied. The following example explains how a simple keyword selection heuristics would help to improve search results:

> Suppose that the consumer is interested in purchasing a "car". Querying WebCrawler with a single word "car" would result in more than 36,000 hits, most of which are irrelevant to the intended query. Knowing that some of the evaluative criteria the user is interested in are the performance, style and safety of the car, then querying Webcrawler using "car AND performance AND safety AND style" would reduce the returned results to 347 only.

Another simple elimination process is to prune out URLs that are clearly not for suppliers, such as documents located at educational hosts (e.g. that ends with .edu or .ac.uk).

Similar approach can be used for the support of product recommendation. However, recommendation support requires natural language understanding and extensive domain knowledge to extract the required details about the product attributes useful for discriminating between different products and between variants of the same product. The agent needs to use seed knowledge to support learning and find-out product specifications and reviews extracted from specialised news groups, such as rec.autos for cars, and specialised Web sites such as AutoWeb (www.autoweb.com).

3.3 Wrapper Generation for IE

The design of the model would not be complete without the consideration of the information extraction module. However, because a thorough study of wrapper generation techniques is a huge research area in itself, a decision was made to limit the target information sources to only those presenting the data in a table format. To put it differently, this thesis focuses on online resources that present information in a structured fashion and exhibiting some sort of regularity for extracting information. Limiting the focus on such resources, though, does not abate the usefulness of the approach since relatively large fraction of actual Internet information resources exhibit such semi-structured behaviour (Perkowitz et al. 1997). The approach is to provide a procedure (wrapper), for each resource of interest, to automatically extract relevant information. The generated wrapper, when invoked on such online resource's document, extract the document's content and identify relevant fragments in the document while ignoring extraneous text. Domain-specific heuristics (such as regular expressions for recognising price availability) and a seed knowledge for automatically labelling the given documents are used. This knowledge base includes descriptions of several popular products in the domain and their corresponding attributes.

3.4 NEGOTIATION CONTEXT

For this thesis purpose, the term "negotiation" is defined as a decision making process in which the consumer and the potential supplier jointly search a multi-dimensional space with an objective of finding a single point in the space that satisfies mutual intended gains. This interaction (negotiation session) between the consumer and the potential supplier is described in terms of protocols and strategies.

Negotiation protocols comprise the valid rules and legitimate mechanisms that can be used in the dynamic process of transacting a deal (reaching a consensus). The richness of a negotiation protocol depends on a set of characteristics that include the degree to which negotiators (artificial agents) can exchange offer and ask alternatives' options over time. This implies that the protocol should enable agents to have the ability to recognise time and could consider a sufficient number of constraints. The extent to which agents have the versatility to make counter offer is also an important characteristic of a negotiation protocoL

In terms of negotiation strategies, a negotiator may elect to use a rational plan of action to maximise his utility. In this thesis stylised negotiation, a multi-period alternating offer bargaining model with two sided incomplete information (both supplier's valuation and consumer's valuation and preferences over the possible set of alternatives are private information) is employed.

In the negotiation simulation, multiple agents from the MAS community negotiate concurrently with potential suppliers' agents; one agent from the MAS community negotiates with one of the potential suppliers' agent, as depicted in Figure 3.2. This idea is inspired from consumer behaviour models, wherein the consumer compares offers of different potential suppliers (sometimes informing one of the suppliers that he found similar product offered cheaper by another one). The difference is that the comparison happens in real-life sequentially rather than concurrently in this system.

Offers made by each party are communicated via Knowledge Query and Manipulation Language (KQML) like messages. The negotiation session begins when multiple agents from the MAS community, concurrently, send a Request For Proposal (RFP) message to their peer potential suppliers' agents.

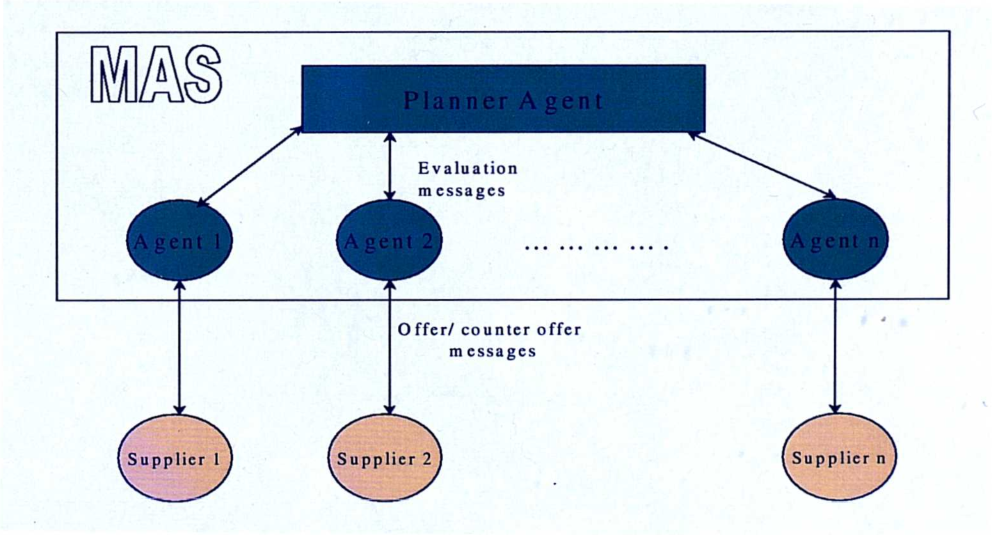


FIGURE 3.2. MAS COORDINATION AS A NEGOTIATION STRATEGY.

Each of the potential suppliers' agents interprets the received message and accordingly replies with an offer that would maximise his private utility payoff. Since multiple agents in the MAS community are negotiating concurrently with multiple suppliers' agents, each agent evaluates the offer received from the peer supplier agent and then, with the aid of the planner agent, collectively compares the offers. The MAS negotiation agents co-ordinate their counter offer based on the best received offer and consequently each agent conveys appropriate message to the respective other party. This offer and counter offer continues until agreement is reached or one of the parties exhausts his strategy and sends 'quit' message. Figure 3.3 describes the flow of the negotiation session of two agents S (supplier's agent) and agent B (one of the MAS negotiation agents).

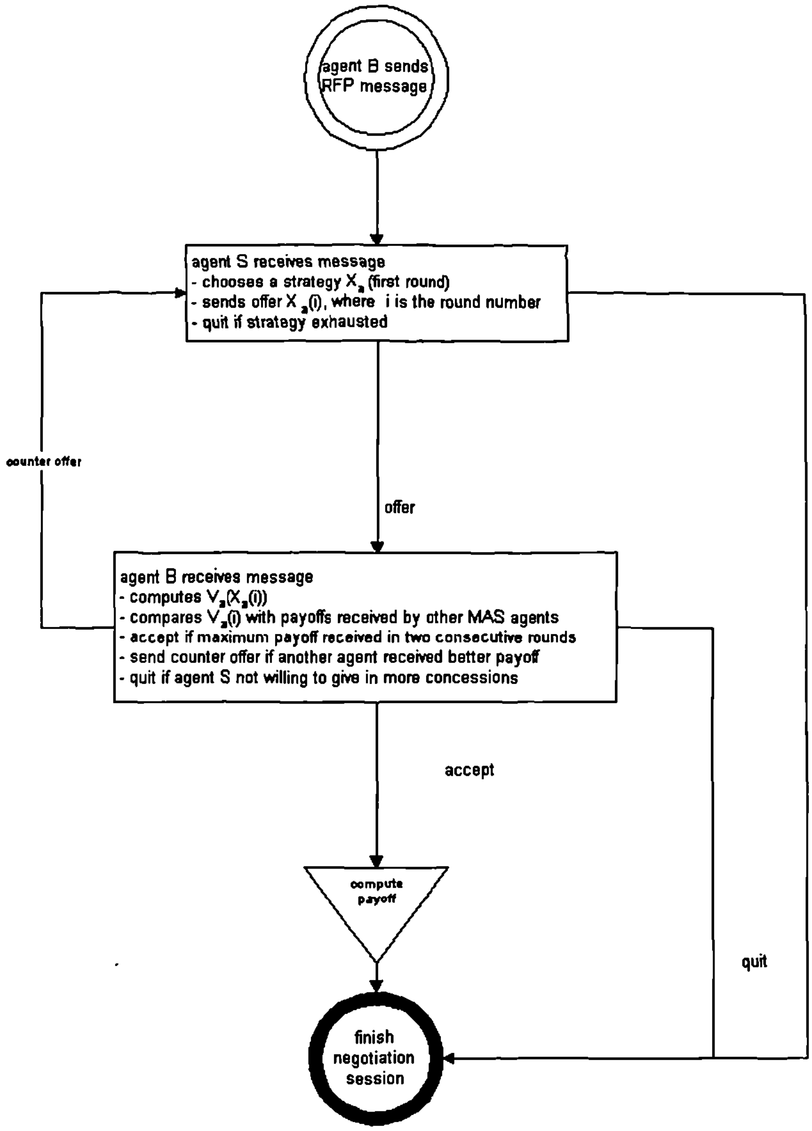

agent B sends RFP message
agent S receives message
- chooses a strategy X_a (first round)
- sends offer $X_a(i)$, where i is the round number
- quit if strategy exhausted
counter offer
offer
agent B receives message
- computes $V_a(X_a(i))$
- compares $V_a(i)$ with payoffs received by other MAS agents
- accept if maximum payoff received in two consecutive rounds
- send counter offer if another agent received better payoff
- quit if agent S not willing to give in more concessions
accept
compute payoff
quit
finish negotiation session

FIGURE 3.3. FLOW OF THE NEGOTIATION SESSION FOR TWO AGENTS A (SUPPLIER'S AGENT) AND B (ONE OF THE MAS AGENTS').

A basic heuristic that is often used to evaluate the overall attractiveness of a given alternative with multiple objectives is the utility theory approach. Appendix B gives an overview of some of the most widespread decision analysis that are used to systematically make a choice among alternatives. In Appendix B, the decision to choose the utility theory for alternatives evaluation is justified.

Both parties use utility theory to apply objective measurement to decision making (which includes offer and counter offer). The payoff of an alternative is assumed to consist of measures over the criteria that add to worth, all converted to a common scale. This valuation is different for both parties and is a private information. In the prototype, the consumer can choose from either a discrete, linear or a curvilinear function to specify his attributes preferences structure. A standard functional form of type U (x) = a + b e^{-cx} has proven to fit commonly encountered risk averse human decision-makers (Olson 1995).

The following section describes the application of the MAUT for sustaining agents' negotiation strategies.

3.5 Multi-Attribute Utility Theory

The basic conjecture of MAUT (Olson 1995) is that in any decision problem, there exists a real valued function U defined on the set of feasible alternatives that the decision-maker tries to maximise. This function aggregates the criteria C_1, C_2, .., C_x. Utility theory provides a numerical scale to order preferences.

For illustration, consider a stylised negotiation wherein a consumer agent needs to purchase a personal computer from a supplier. The consumer might be interested in the issues of price, hard disk, processor, memory, CD-ROM and screen size. Regarding the price, the value-worth would be a linear decay function with a cut-off maximum price (i.e. as the price increases the utility of the price-measure decreases). The price measure, in hundreds of pounds, is given with a worst measure of 350 and best measure of 150. Cut-off value is applied in which any price less than 150 get a measure-value of 1. For the processor dimension, he might have increasing utilities for P II-200, P II-233, P II-266, P II-300 and P II-333. The value-worth for the main memory might be decreasing utilities for 64MB, 32MB and 16MB. For the hard disk measure, he might have a linear function, with a maximum of 6GB and a minimum of 2GB. For the CD-ROM, he has discrete preferences wherein 8x get a value of 0, 12x get a value of 0.5 and 24x get a value of 1. A discrete measure is also applied for screen size wherein 15" screen gets a value of 0.5 and 17" screen gets a value of 1. Here we use a utility value of 1 for the best measure and a value of 0 for worst measure. In contrast to the consumer, the supplier would have different, nearly opposing, valuations with respect to these measures.

The consumer assigns the relative importance (weight) that would be used for objectives (attributes) to evaluate trade-offs in each alternative. The consumer might assign a relative weight of 0.3 to the price measure, 0.2 for the processor, 0.2 for the memory, 0.1 for hard disk, 0.1 for CD ROM and 0.1 for the screen size.

The overall utility score is calculated using the utility function $U_j = \Sigma_k w_k u_{jk}$. Where U_j is the utility value of alternative j, w_k is the normalised weight for attribute k and u_{jk} is the scaled value for alternative j on attribute k. Therefore, for the alternative (PC1) as shown in Table 3.1 the overall utility score is:

u(£2800)* 0.3 + u(P II-300) * 0.2 + u(64MB) * 0.2 + u(6GB) * 0.1+ u(12x CD ROM)*0.1+ u(17" screen)*0.1 = 0.7.

Where u(£2800)=(350-280)/(350-150), u(P II-300)=3/4, u(64MB)=1.

Alt.	Price	CPU	RAM	Screen	CD	HD	Utility
PC1	2800	P-300	64	17"	12x	6GB	0.700
PC2	2500	P-266	32	15"	8x	2GB	0.260
PC3	3000	P-333	64	17"	24x	6GB	0.875

TABLE 3.1. EXAMPLE OF SCORES FOR ALTERNATIVES' OVERALL UTILITIES.

3.6 USER INTERFACE

The design of consumer-oriented electronic commerce system requires mechanisms for eliciting user's preferences. However, from the user's perspective the interface should be kept as easy as possible to use for the system to be acceptable. To satisfy both requirements the preference's elicitation should not entail major requirements on the user. The design needs to carefully consider the user interface to perform functions for:

1. Eliciting preferences from the user: the system requires knowing about specific attributes of the products of interest. When the user asks for a specific product for the first time, the system requests from him to identify relevant dimensions and attributes of the product, which will be used for evaluation of alternatives. The user is also asked to rank these dimensions in order of importance, which will be used for developing weights. Subsequent user's requests for the same product need not to go through the same procedures, since these information are stored in a knowledge base. However, the user should be able to modify his preferences whenever he wishes.

2. Provide reports and support for browsing and monitoring: the user will be interested in knowing how the system is performing and what is the current status of his request. He should feel that the agents he is delegating are reliable and predictable and he remains in ultimate controL The user should also have a reasonable expectation of the results of the system in which the benefits he gains (in terms of time, information, etc.) should be greater than the cost (in terms of price of delegation, time, re-work, etc.). The design of the user interface should take into consideration these concerns.

3.7 PERFORMANCE MEASURES

The developed MAS should be tested for a set of products and the outcomes are evaluated on several dimensions. A thorough empirical approach to evaluation is taken, aiming to answer the research questions posed in Section 1.6. Here those questions are re-iterated wherein the research strategy for a plausible answer to each of these questions is described.

1. Can artificial agents, that are ingenious and trustworthy enough, be developed to involve in routine business negotiations?

A satisfactory answer to this question should investigate the ability of these agents to use effective negotiation strategies and compare their performance with human subjects. The system will be used to simulate several types of buyer-supplier negotiations and the outcomes will be evaluated on multiple dimensions. Simulating different types of business negotiations gives more conviction in predicting the performance of the system. Strategic effectiveness will be evaluated by quantitative analysis of agents' payoffs. The relationship between the bargaining behaviour of the agents and humans will be also investigated by comparing the results of these experiments with human experiments of similar bargaining cases.

2. How to implement software agents for finding, retrieving, extracting and interpreting required information?

The two important measures that are defined in classic information retrieval to describe system's effectiveness, i.e. document recall and precision, cannot be exactly calculated. The reason is that the relevance of a document is subjective even for humans and the number of actual relevant documents (even if considered objective) is unknown. However, in this thesis meta-searching approach the number of retrieved results from underlying search engines is limited, and that it is the techniques the thesis employs for post processing that would influence the relevance of retrieval towards higher precision. Similarly, pre-processing techniques (before the query to the search engine) are employed towards higher recalL

However, due to the thesis and time limits, decision is made to concentrate on a thorough evaluation for the negotiation approach and the usability of the system. For this, the study will be content with conducting an example experiment to act as an indicator to the effectiveness of the searching approach and leaving the thorough testing as an area for further research.

The approach for information extraction is to automatically *construct wrappers* that would be used to extract relevant information from structured online sources. To measure the performance of the wrapper construction approach, the behaviour of the system should be measured against actual online information sources on the Internet. The extent to which the results are relevant and important needs an investigation by examining the applicability of the approach on a number of Internet information resources. Whereas wrapper induction addresses the information extraction problem, a domain-specific knowledge and some recognition heuristics are used for interpreting the extracted data.

The approach to evaluation in this thesis, similar to the searching, will be limited to conducting an experiment to act as an indicator to the effectiveness of the wrapper generation method, while leaving thorough testing for further research.

3. Is it feasible to develop a software system that would allow consumers to make effortless Internet purchases?

Ease of use of any system's features is an important indicator of its potential success. From the user's perspective, the evaluation of the system would involve evaluation of the system's usability. An important objective of this study is the analysis of the effects and usefulness of the decision analytic techniques in practical negotiations. Therefor, the extent of requirements on the user, which the chosen MAUT for preference elicitation impose, should be measured.

4. What is required to develop these agents? How does it work?

A MAS framework is developed that supports agents' interaction and co-ordination. The following chapter studies practical design for Internet-based EC MAS, in which MAS taxonomy is presented. Application structure and MAS issues such as planning, interaction, coordination and mechanisms for exchanging messages will be addressed.

CHAPTER 4

PRACTICAL MAS DESIGN FOR INTERNET-BASED EC

The choice of MAS as a solution strategy for consumer oriented Internet-mediated EC was motivated by several desirable features of such systems and the dominant characteristics of the environment of the application.

Significant advantages inherent in MAS approach include:

- MAS architectures, akin to other distributed systems, offer many advantages like scalability and modularity (Parunak 1996). Agents are powerful entities because of the separation of concern and factorisation of problem they impart. Agents can be constructed and maintained separately and therefore help in the incremental growth and flexible expansion of the system.

- The distributed nature of MAS facilitates the tolerance for uncertain data/knowledge and fault, and thus assists building more robust applications. The concept of a single point of failure vanishes, instead the system degrades gracefully even when some of the agents are out of service temporarily. As the overall system behaviour emerges from local decisions, the system readjusts itself automatically to environment noise or entrance and exit of agents. Accordingly, MAS is self-reconfigurable as it runs.

- MAS are well suited to solve problems that may be overly large for a centralised single agent. Having a number of agents with different capabilities dynamically teaming up to solve a large problem constructs more flexible system. Likewise, MAS are well suited to domains that require the integration of multiple resources of information dispersed over a wide geographical area like the Internet.

- In accordance with the insight gained over the past decades in disciplines like psychology and AI that intelligence and interaction are highly and inevitably coupled, interaction allows the agents to increase their level of intelligence (Parunak 1996). On the other hand, interactions in such systems are fairly sophisticated including negotiation, information sharing and coordination that requires the complex social skills with which agents are endowed.

- MAS provide enhanced speed and reliability by exploiting parallelism and concurrency.

Several taxonomies of MAS have been published (Bond & Gasser 1988, Decker et al. 1989, Grant 1992). In this chapter, a taxonomy developed from these is presented, giving special attention to features that are most relevant from the application perspective. First, since any system always has to be considered in its environment context in order to really understand its functionality, the environment occupied by the MAS is analysed in Section 4.1. Then, the architecture of the system within which the agents interact and the individual agents' architecture are described. The chapter concludes, in Section 4.4, by giving a clear picture of the contributions of the chapter.

4.1 SYSTEM ENVIRONMENT

The design of Internet-based MAS should take into consideration the dominant characteristics of the Infosphere. Key properties of the Internet environment that makes it a major target for systems based on agents and MAS technologies include:

- The availability of ample amounts of data, rich diversity and heterogeneity of information sources "information-rich", open, large-scale environment and the very dynamic and unpredictable nature of the setting are some of these characteristics.

- Lack of global control of the accuracy of the content of information sources.

- Contain information sources that are added or removed in an unpredictable manner.

Given these dominant characteristics, the design of open MAS for the Internet faces many problems. One of the basic problems is the connection problem, i.e. finding the potential agents (suppliers) who might have the product/information the agent (consumer) needs. The fact that the system is open and distributed over the entire Internet precludes broadcast communication solutions. To address this issue, mediator agents (Kuokka & Harada 1996) have been proposed for open environments such as the Internet. If the exchanges between agents are to be efficient and protected on matters of privacy, more sophisticated mediators are required. In such setting, each agent advertises its capability to some mediator agent. A number of different approaches for mediator agents that have been tested (Decker et al. *1997)* include:

- **Matchmakers** allow one agent with some goal to learn the name of another agent that could satisfy the goal (yellow pages).

- **Brokers** provide services such as locating the other agents by name (white pages) or by capabilities (yellow pages). Brokers manage a name space service and could have the ability to store and route messages and locate messages recipient. Broker agents also function as a communication facilitator by managing communications among various agents, databases and applications in the environment.

In some settings, agents do not directly communicate with one another. Instead, they communicate through the mediator (facilitator) agent. In such a setting, a group of agents register with a mediator who is kept informed about their capabilities and needs. Agents send information and requests to these mediators and, in return, accept information and requests. Mediators route information provided by those agents to appropriate places. FIPA (1997a) specification defines a specialised **Domain Facilitator** agent for each domain whose job is to maintain an agent directory for that domain and facilitate communication between agents in the domain.

In preliminary experiments (Decker et al. *1997*), it was seen that the behaviour of each type of mediator agent has certain performance characteristics. For example, while brokered systems are more vulnerable to certain failures; they are also able to cope more quickly with a rapidly fluctuating agent work force and could offer centralised ontological translation or even simple integration facilities. On the other hand, matchmakers offer the most flexible response to random agents entering and exiting the system, however they lack the centralised ontological facilities.

Mediators are advantageous since they allow a system to operate robustly in the face of agents' appearance and disappearance and intermittent communication. However in this work, the task of developing mediator agents will not be undertaken, which is a requirement for a fieldable system. Instead, for simplicity the Java RMI registry will be used to do the job of 'white pages' directory to know the name of other available agents.

Another critical issue that needs special care when designing open systems is the ontological mismatch (Gruber 1993) problem. As the requirement for a higher degree of autonomy and complexity of cognitive attitudes and capabilities for the MAS agents' increases, the need becomes apparent for formally specifying the meaning of the terms and definitions of the entities involved in the information transactions among the agents. Within the knowledge engineering field, the notion of **ontology** is identified with the set of formal terms with which one represents knowledge (Gruber 1993). To clarify, ontology is specified by a set of definitions of both the objects relating to some area of interest (objects' names) and the relationships hypothesised to hold up among them (relations and/or functions) defined in both machine and human readable form. Such a specification when performed through a logic-based language provided with declarative semantics and, in general, allowing to correlate natural language descriptions with formal statements, form a special axiomatization of an application area. The logic-base and declarative semantics approach is proposed to make it easier both to understand the meaning and the implications of the definitions and to maintain overall consistency. The use of an **open standard ontology** specification language allows to employ an ontology relative to a distributed application, in this case heterogeneous MAS, as a coupling mechanism (for sharability) among the interacting parties, all perpetrate the chosen definitions while exchanging information/ messages. This would provide the foundation for many applications programs (or heterogeneous agents) to inter-operate on the basis of shared definitions.

Ontolingua (Gruber 1993) is a public domain tool used as a mechanism for defining and maintaining common ontologies portably, in a form that is compatible with multiple representation languages. It reads the definitions written in a simple syntax and translates them into forms that can be used by a variety of knowledge representation languages. The syntax of Ontolingua definitions is based on the standard notation and semantics of predicate calculus called Knowledge Interchange Format (KIF) (Genesereth & Fikes 1992).

In this thesis, the task of designing ontology language for EC applications will not be undertaken, instead useful shared domain ontologies are assumed to exist and are being used by the agents.

4.2 APPLICATION ARCHITECTURE

To describe the MAS architecture, the number of agents, the communication channels and protocols that they use, how agents are configured relative to each other and how they coordinate their activities as the system runs must be identified. The identified requirements dictate the design principles and the pursued methodology to ensure communication, coordination and co-operation between agents in the MAS. Then based on these guidelines, the basic infrastructure required to facilitate the interaction and coordination mechanisms between the agents in the MAS is developed. In the following subsections, those requirements are described followed by the overall MAS architecture.

4.2.1 Agents Interaction

In all communicating MAS and particularly in domains that include agents built by different designers (such as consumers and suppliers), there must be some set of language and protocol for the agents to use when interacting. Therefore, the system must support and provide a means for individual agents to communicate with one another and define protocols for the resulting conversations. Moreover, systems developed for information rich environment have requirement for a wide-scale interoperability that requires **open standard protocols** and interfaces. Independent aspects of such protocols are information content, message format and coordination conventions. Among many others, ARPA Knowledge Sharing Effort (Patil et al. 1992) has produced the KIF (Genesereth & Fikes 1992) predicate logic language for describing the information content and KQML (Finin et al. 1994) to delineate the message format and message handling protocol to support knowledge-sharing and interaction among agents.

KQML has been designed as a general-purpose standard language for inter-agent communication through the support of explicit linguistic actions called "performatives" based on the speech act theory. The notion of speech acts goes back to the philosophical work of Austin (1962), who remarked that most things that people say are not simply propositions that are either true or false, but performatives that could succeed or fail. That is to say, utterances are endeavours by the speaker to do something. This notion has been used in the agent community as the basis for understanding how agents communicate.

KQML offers an extensible set of performatives that specify the kinds of communications an agent can have with another KQML-speaking entity. The performative indicates that the content of the message is a query, an assertion, a command or any mutually agreed upon speech act. It also determines how the sender would like the reply (if any) to the message to be delivered. Abstractly, a KQML message consists of a performative specifying the agent's intent, the identity of the sender, recipient, the message language and the ontology it assumes. Some of the identified problems with the current implementations of KQML are addressed earlier in Section 2.5.1.

Performative	Meaning
RFP	Sender requesting a proposal from the recipient.
OFFER	Sender sending offer in-reply-to RFP or RE-OFFER message.
RE-OFFER	Sender commenting to previously received offer.
ACCEPT	Sender signalling acceptance to an offer.
REJECT	Sender signalling rejection to an offer.
QUIT	Sender informing recipient that it is terminating the conversation.

TABLE 4.1. MAS COMMUNICATION PERFORMATIVES.

The fact that KQML is widely accepted as the de facto standard for agent communication language and in order to easily allow agents developed by outside parties (for example, suppliers' agents) to communicate with the MAS, a choice is made to adhere to KQML message format. Therefore, the design of the message format for the MAS inter-agent communication protocol is based on KQML, though a set of higher performatives specific to the application area such as RFP (Request For Proposal), OFFER, RE-OFFER, ACCEPT, REJECT and QUIT have been added. Table 4.1 gives a description of each performative.

A message consists of a performative conveying the agent's intention, fields specifying the identity of the sender, recipient, the thread of the conversation that the message is part of (hence enabling multiple conversation to be commenced concurrently which is an extension to KQML), the language used for communication, the name of the ontology assumed in the message content and, finally, the content of the message. The general structure of the message format is shown in Figure 4.1.

```
(       Type:          <performative>
        Sender:        <agent-id>
        Receiver:      <agent-id>
        Conversation:  <conversation-id>
        Language:      <language-name>
        Ontology:      <ontology-name>
        Content:       <expression>  )
```

FIGURE 4.1. THE GENERAL STRUCTURE OF AN INTER-AGENT COMMUNICATION MESSAGE

With regard to coordination, the approach is to make conversations as the basic unit for the interaction rather than messages in isolation and, hence, restricting messages to take place only within the context of conversations. A conversation is a sequence of messages between two agents, taking place over a period of time that maybe of arbitrary length, yet is bounded by specified termination conditions for any given occurrence. Conversations may initiate other conversations as deemed appropriate.

The policies governing conversational and other social behaviour among agents are implemented using an off-line design approach (Walker and Wooldridge 1995) in which social laws are hard-wired. The most important construct of this approach is the use of conversation plans to explicitly define what sequence of which messages are permissible between a given set of participating agents. Conversation plans specify the states and associated rules that are specific to a conversation type, together with rules for error recovery. The rules specify how an agent in a given state receives messages of specific type, check several conditions and, accordingly, transmit messages and update the local status. These rules are indexed on a finite set of values of a special variable called current-state of the conversation. Therefore, finite state machines are used to represent these conversations. The states of the finite state machine represent the states a conversation can be in. The initial state specifies the condition for initiating the conversation. Any *state with no transition rule leaving it is a final state; reaching a* final state terminates the conversation.

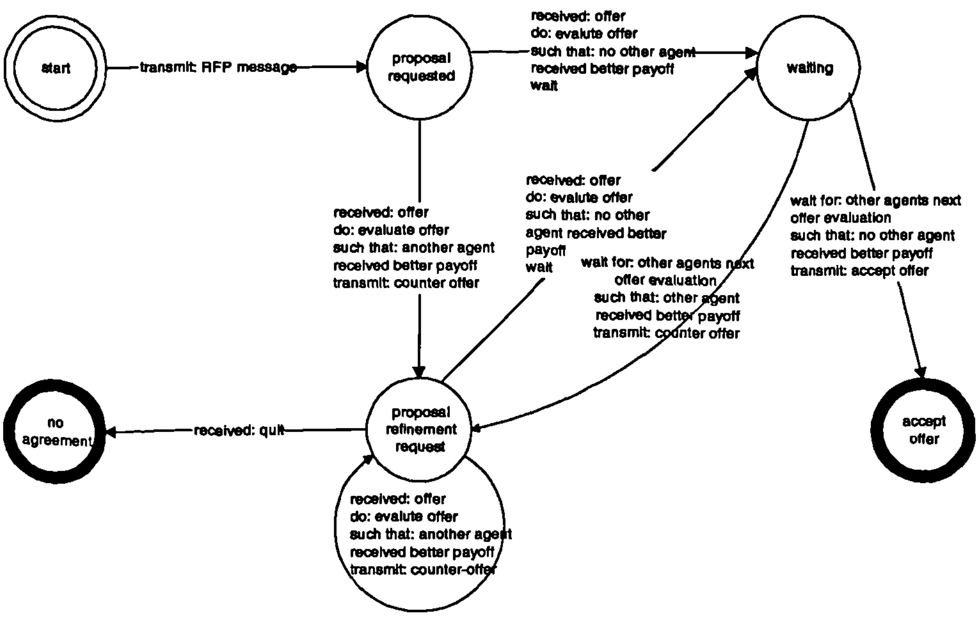


FIGURE 4.2. FINITE STATE MACHINE REPRESENTATION FOR CONSUMER'S CONVERSATION POLICY WITH SUPPLIER.

Figure 4.2 depicts an example of a transition diagram for a conversation policy governing the consumer's conversation with a supplier's agent in the application. Nodes represent the states of the conversation, arrows indicate the existence of a transition rule.

4.2.2 MAS Architecture

The design principles and the proposed methodology outlined in previous subsection ensure communication, coordination and co-operation between agents in the MAS. Based on these guidelines, a generic MAS infrastructure in Java have been developed as shown in Figure 4.3.

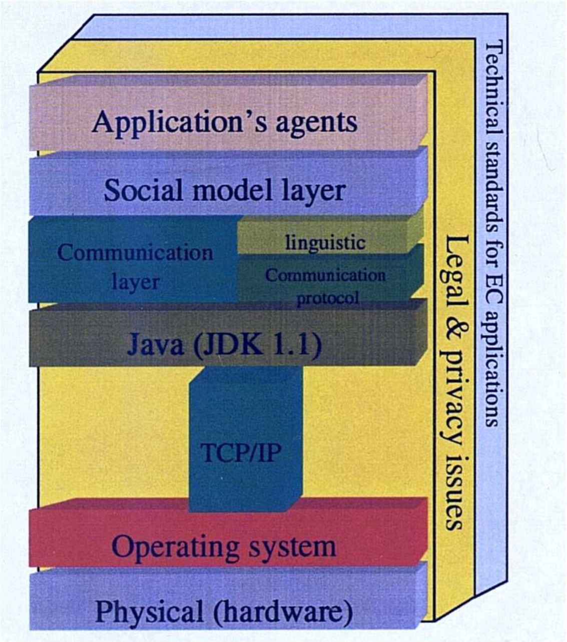


FIGURE 4.3. MAS ARCHITECTURE.

The architecture provides the basic infrastructure required to facilitate the interaction and coordination mechanisms between the agents in the MAS. To support these mechanisms the structure implements several layers. The communication layer defines the common language for agents' communication (linguistic sub-layer) that follows the KQML format, and the communication protocol sub-layer that defines the protocols used in communication (direct messaging, broadcasting and RMI). The social model layer implements the conversation classes and plans that are used to coordinate interactions between the agents.

4.3 Agents' Architecture

In describing individual agents, it is important to understand how agents are mapped onto the application domain as described in Section 3.1.1. Following subsections describe how those identified agents model their environment and the internal structure of individual agents. First, the architecture of the search agent is described. Algorithms used for the searching and analysing results are addressed and advantages of our approach are explained. Second, the architecture of the information extraction agent is illustrated. However, it should be clear that this study of information extraction techniques is only to complement the global perspective of a complete consumer-oriented Internet EC system, and that no attempt is made to design or conduct a thorough research into the field. Research dedicated on information extraction is described earlier in Section 2.2. Finally, the architecture of the negotiation agents is described. The negotiation strategies, payoffs evaluation and communication design are detailed.

4.3.1 Search Agent Architecture

The search subsystem was designed based on a meta-searching approach. The idea is to develop more effective searching techniques on top of existing search services. The agent-based approach employs intelligent search agents capable of searching multiple sites, translating a single query into its optimal form for each existing engine, receive the results back, aggregate them in some way and then merge the results. Figure 4.4 shows the architecture of the searching subsystem.

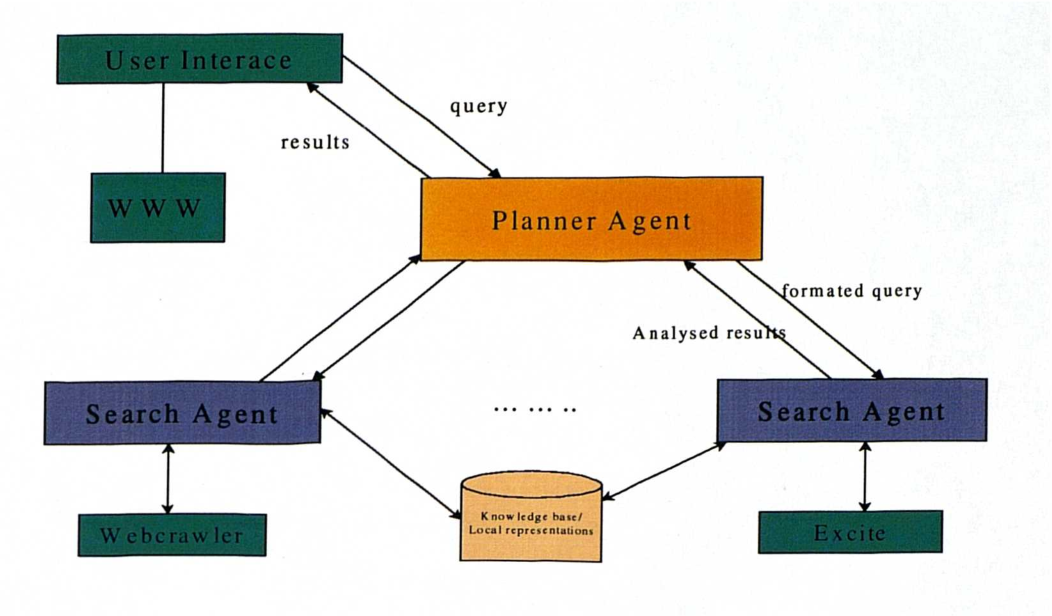


FIGURE 4.4. SEARCHING MAS ARCHITECTURE VIEW.

The design of the search subsystem considers the following issues:

- Query capabilities of each of the existing Web search engines vary to a great extent from each other and each service provides incomplete snapshot of the Web. Also, those engines will give the best performance for queries structured and tailored to the internal design of the engine. Therefore, queries should be formulated for each search engine to optimise the accuracy of retrieved results.

- Search engines return many results that are irrelevant, out dated or unavailable. The system should sift through those results searching for useful information.

- Network delays and time spent waiting for busy servers is sometime intolerable. The system should use effective concurrent programming methods to reduce search time and put limitations on waiting time for busy servers and network delays.

- With the exponential growth of the Internet, many new search services are added daily. Therefore, the design of the system should be flexible to add additional search services with minimal effort.

The search subsystem utilises multithreading techniques to instantiate multiple instances of search agents which receive formulated queries from the planner agent. The planner agent formulates the query based upon the requested product and information obtained from the knowledge base. Then, the planner agent instantiates a number of search agents equivalent to the number of enabled search engines.

Each search agent consists of two main modules, i.e. search engine and results analyser modules, accountable for returning results of a query to the planner agent (see Figure 4.5).

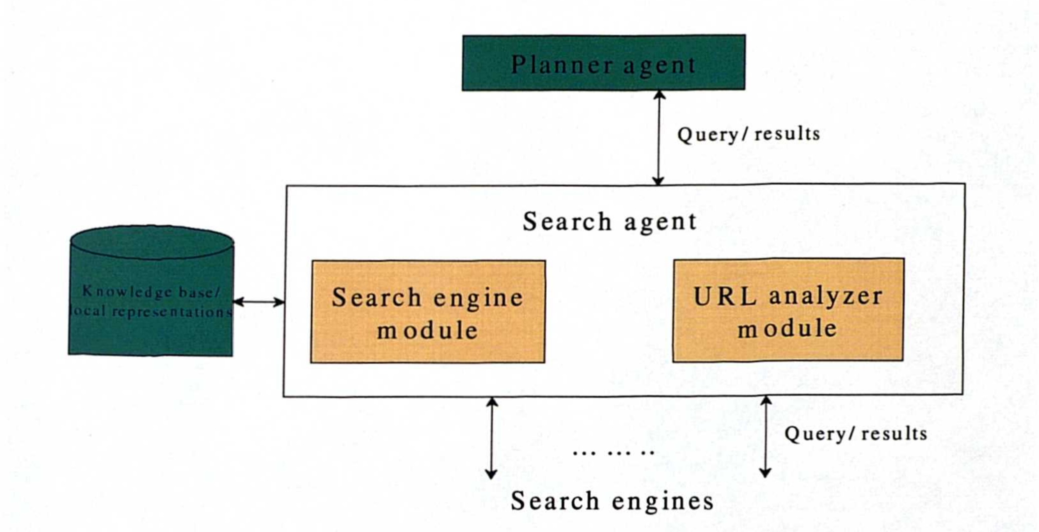


FIGURE 4.5. SEARCH AGENT INTERCOMMUNICATION AND DATA FLOW.

The search engine module applies the following algorithm to query the underlying Web search engines and extract the returned URL results.

1. Optimise query – formulate query specific to the search engine to be queried with the aid of the knowledge base.

```
2. Start timer.
3. Query underlying search engine.
4. IF timer > maximum time to query THEN terminate.
5. Retrieve results page.
6. Extract URLs from results page – add URLs to URL-table.
7. WHILE (timer < maximum time to query AND search engine has more results) THEN
   7.1 Formulate query to retrieve next results page.
   7.2 GO TO 3.
8. Terminate.
```

Concurrently, while the search module is extracting the results' URLs, the results' analyser module proceeds analysing each URL added to the URL-table to filter out irrelevant URLs and dead links (URLs that are no longer exist or URLs that have very heavy network delay). The results' analyser module applies the following algorithm.

```
1. Get URL from URL-table.
2. Start timer.
3. Try connecting to URL.
4. IF (not connected OR timer > time to connect to URL) THEN GO TO 7.
5. Analyse retrieved document – whole HTML document is analysed to provide more reliable weighting for the document, as follows.
   5.1 Parse HTML page; deleting non-stop words (non-informative words such as "the", "after", etc.) and stemming the other words to their original form using Porter stemming algorithm.
   5.2 Remove meta-tags that are used by some publishers to force appearance of their document or to obtain a higher rating if any.
   5.3 Weight the relevance of the document – using normalised term frequency (tf). Term frequency is the frequency of a concept's occurrence within a document. The more often a concept is used, the more to be expected that
```

the concept is important within the document. Normalisation is used to prevent very long documents from dominating much smaller ones.

6. Cache document locally, send URL and the weight to the planner agent.
7. IF URL-table has more entries GO TO 1.
8. Finish.

Combination of different schemes for document weighting can also be used. One scheme is to give words inside HTML tags higher ratings, for example in HTML <B> agent </B> would appear in bold face, and thus would be assumed to be more indicative of the document content. Another scheme is to use structural features such as the document length or the number of pictures or hyperlinks it contains.

Since the analyser module caches each HTML result document locally, the parsed document (i.e. HTML tags removed) can be summarised using a summarising algorithm similar to the one provided by MS Word to summaries a document. This way each possible related URL is presented with the document summary (approximately 150 words), instead of the first 100 words provided by most of the search engines.

To summarise the searching approach, meta-searching is used to query existing search engines and then collate the results. The intelligence of the searching agents' lies in the processing they perform before querying the underlying engines to formulate optimal query, and after retrieving results from the engines. Post processing combines different methods to increase the accuracy of relevant results. These include: (1) analysing the whole document to provide more reliable weighting, (2) techniques for pornography avoidance, and (3) document summarising.

4.3.2 Information Extraction Agent Architecture

Information extraction refers to the process of identifying and classifying relevant fragments in a document while discarding extraneous text. The complete design of information extraction agent for this thesis purpose should involve constructing recognisers for the products and their relevant attributes to be extracted. The agent employs some recognition heuristics to extract the required attributes while ignoring extraneous elements such as advertisements and HTML formatting constructs. The required recognition heuristics might be very primitive, for example using regular expression £[0-9]{1,} to identify a price. At the other extreme, recognition might require natural language processing.

For this application, many attributes that are to be extracted can be recognised perfectly. For example, regular expressions can be used to perfectly match attributes such as URLs, HTML tables, credit card numbers, price and Zip codes. Although, regular expression mechanisms are sometimes trivial, these can be amazingly powerful for implementing recognisers even for structured and sophisticated attributes.

The advanced research in the natural language processing community concerned with the identifying classes of entities in free text has matured to the point that high quality commercial recognisers are now available. For example, the 'Named Entity' task at the sixth Message Understanding Conference (MUC-6) (ARPA 1995) involves locating people's names, companies, locations, dates, times and so forth.

The strategy for information extraction is to build a wrapper for information sources that present their data in a table format. Central to this technique, is a library of recognisers and domain specific procedures to identify instances of particular attributes. However, the limitation for the type of information sources to only those presenting their data in a tabular form is due to the thesis and time limits and that it should be clear that a practical system should include wrappers for other information sources. Building upon previous works on wrapper construction described earlier can be very helpful for this task.

The algorithm that is used to wrap information sources, presenting data in a table format, is simply a text-searching algorithm to locate the HTML imbedded tables and extract each row and column of these tables as possible candidate attributes. Then by consulting pre-compiled specialised products' databases and a table of regular expressions, attributes' recognisers are built. Once the information source has been successfully wrapped, the resource can be further mined for additional data that can subsequently be used to build recognisers for other sources and attributes. For example, in Figure 4.6 if source A is successfully wrapped for extracting the price and speed attributes, a wrapper for source B can be successfully generated recognising that the speed is the same attribute as the processor. The resource can be further mined to recognise the memory attribute and hence the RAM capacity in source B.

Speed	Pentium 233
Memory	64MB
Price	£ 1200

Source A

Processor	RAM	Price
Pentium 233	64MB	£1000

Source B

FIGURE 4.6. EXAMPLE OF HTML TABLE INFORMATION EXTRACTION.

4.3.3 Negotiation Agents' Architecture

Here business negotiation is considered from the point of view of the consumer. In the proposed model of business negotiation, a front-end GUI interacts with the user who provides the specifications for the planner agent which, in turn, deploy the negotiation agents and aggregate those specifications. Negotiation agents interact with other agents or suppliers. This model involves both individual decision making and negotiations. Individual decisions include specifications of the items attributes significant to the consumer, and of his preferences. Negotiation entails the analysis of the offers, offer rejection or acceptance and the construction of counter offers. Figure 4.7 shows the overall design of the negotiation agent.

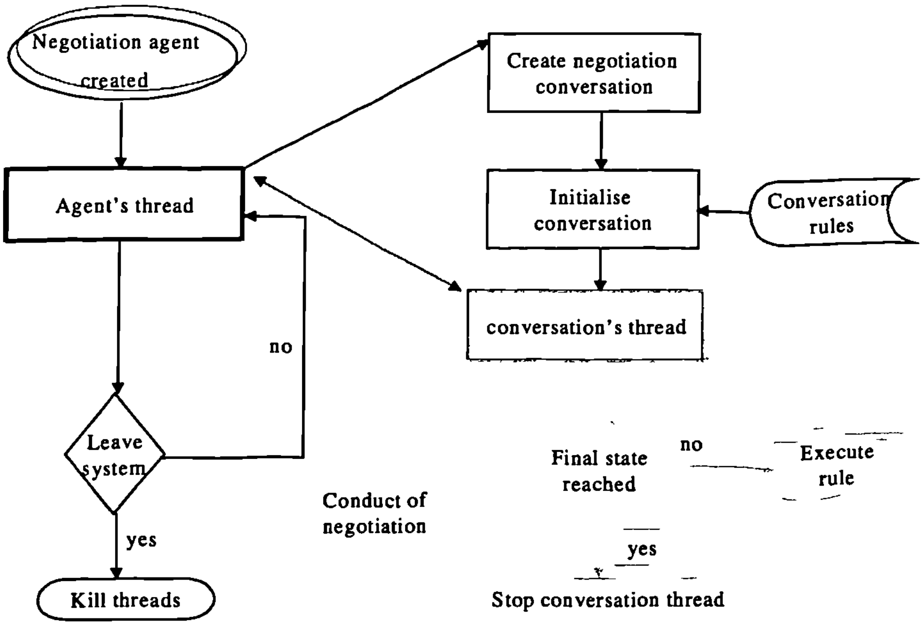


FIGURE 4.7. THE DESIGN OF NEGOTIATION AGENT.

An important feature of the proposed model is the incorporation of decision and negotiation theoretic constructs and a methodology based on the negotiation analysis approach. The negotiation agents follow the prescriptions of the decision theory and negotiation theory, employing mechanisms designed within these theories.

The negotiation process comprises the pre-negotiation and the conduct of negotiation phases described in negotiation analysis theory. In the pre-negotiation phase, the situation and the decision problem are analysed. This requires the specification of item's attributes, together with the consumer's preferences and criteria for evaluating the options. This procedure is called the preference elicitation, which leads to the construction of a utility function. As described earlier, MAUT is the tool used for preference elicitation and utility function construction. For this the user is presented with attributes and options and is asked to assign a weight to each of them separately.

In the negotiation conduct phase, the session starts with the agent sending an RFP message to the potential supplier and then waits for an offer message. When the agent receives the offer message, it applies heuristics to analyse the offer and accordingly decide to accept, counter *offer or reject and quit the negotiation* session. The offer analysis procedure is described earlier in Section 3.3, whereas the conversation plan is shown in Figure 4.2.

A deep analysis of the negotiation protocol reveals a sort-of reverse auction mechanism. The basic idea is to force all potential suppliers to compete to win a contract with the customer. Each supplier starts with an initial offer, then the consumer evaluates each offer (a function of a number of quantitative (e.g. price, quantity) and qualitative (e.g. brand, service) attributes). Supplier with best offer (decreasing-price English auction) stays silent, while others are asked to further lower their price. When no one is willing to lower his price further, the supplier with lowest offer wins the contract. Furthermore, the protocol can be easily expanded to include a trading-off process capitalising on the followed speech act approach. Trading-off (also known as logrolling) is the process of offering concessions on issues that are less-valued in exchange for receiving concession on issues that are more-valued (e.g. faster delivery for higher price).

4.4 Summary

The design of the Consumer-Oriented EC system as MAS presents a useful software-engineering paradigm where the system is described as individual problem-solving agents pursuing higher-level goals. However, MAS development is a technically difficult task challenged by distributed programming issues and complexities associated with supporting agents' coordination and interaction.

In this chapter, a MAS architecture is proposed that models open environments composed of logically distributed functions where agents can be used. The architecture provides a framework for constructing those various agents by extending the basic functionality provided by the abstract common infrastructure. The basic agents in the proposed MAS architecture are:

- Abstract agent: This is an abstract entity for all agents in the architecture. This conforms to the recommendations of Agent-Oriented Programming (AOP) paradigm and offers many advantages like scalability and modularity. The abstract agent implements a multi-layer approach to facilitate the basic communicational functionality required for agent-to-agent interactions. The choice of adopting the KQML format in the communication layer is due to the wide adoption of the language as a de-facto standard for inter-agents communication. A set of higher level of performatives was added to the core KQML specification specialised to the application. In the social layer model, a choice was made to make conversations as the basic unit for the interaction. Social interaction among the agents is more appropriately modelled when conversations rather than isolated messages are taken as the primary unit of agents' interaction.

  All of the identified agents in the system, i.e. search, information extraction and negotiation agents, extend the abstract agent class to benefit from the provided infrastructure to facilitate the interaction and coordination mechanisms and provide regulations that agents follow to communicate and understand each other, thereby enabling knowledge-sharing.

- Planner agent: This plays the role of the central coordinator that helps agents' interactions and coordination and provides a 'white page' directory service of agents in the MAS. This implies a centralised control mechanism in which a central coordinator to whom everyone communicates his solution is used.

- Matchmaker: Although this functionality is not implemented, its advantages and its importance for any robust open system was emphasised.

Comparing this thesis MAS architecture approach with current efforts by groups working on the standardisation of the dynamic MAS architectures, an approach that corporates aspects of both the Foundation for Intelligent Physical Agents (FIPA) MAS model and the Knowledgeable Agent-oriented System (KAoS) MAS architecture model was proposed. Appendix B gives an overview of those MAS architectures.

Mainly, the FIPA's approach to MAS development was followed which is based on a "minimal framework for the management of agents in an open environment." This framework is described using a reference model that specifies the normative environment within which agents exist and operate, and an agent platform that specifies an infrastructure for deployment, interaction and coordination of agents. However, since their recommendation lacks the ideas and specifications of how to go about developing the interaction and coordination techniques. A hybrid architecture was investigated to marry the best aspects of both recommendations (i.e. FIPA and KAoS). The conversations and conversations' policies approach was adopted from the KAoS model that they use to elaborate on the interactive dynamics of agent-to-agent messaging communication by using these policies.

This chapter has also addressed the internal architecture of the application's specific agents. The design of search, information extraction and negotiation agents were presented, and the rational behind the employed techniques were addressed. Elaborating on the small details of the implementation was avoided, because what makes the difference is the ideas behind the design, not the implementation itself. For example, a major piece that this thesis contributes to in the negotiation field is the idea of incorporating decision and negotiation theoretic constructs and a methodology based on the negotiation analysis approach. The use of multiple concurrent negotiation agents and utilising the utility theory as a strategy to evaluate offers and coordinate counter-offers, is what, is a novel idea, would result in appreciable outcomes.

With regard to searching, although the idea of meta-searching is far from novel, however what is different in this approach lies in the pre and post processing techniques employed to enhance the relevancy of retrieved results. An example to evaluate the effectiveness of those techniques will be provided.

In the information extraction approach, though limited to Web resources presenting information in a table format, using a table of regular expression and pre-compiled specialised products database to built attributes recognisers that would aid the wrapping process was proposed. Expanding on those methods to construct more general Web resources wrappers should be relatively easy. An example to evaluate the effectiveness of the wrapper generation method will be also provided.

CHAPTER 5

EXPERIMENTAL DESIGN

The mere building of a system or a tool is development and not research. Research involves gaining comprehension about how and why a particular type of a tool might be useful and validating that a tool has certain properties or certain effects by carefully designing an experiment/s to measure those properties or to compare it with alternatives. Scientific experimental paradigms serve as a basis for differentiating research activities from development activities. A scientific method can be used to gain the understanding of the effects of a particular tool usage in some environment and to validate hypothesis about how the tool is performing. The experimental paradigms require an experimental design, observation, data collection and validation of hypothesis or theory of the model being studied.

There are a number of dimensions that could be used as the determining measures for the system's goodness. On one level evaluations could focus on functionality and system performance, while user's evaluations are of little importance. Another view is that evaluation of a system involves evaluation of the total system usability. Also, it is important to understand that there are a variety of techniques available that assist different evaluation purposes. However, techniques that are essential for general theory building and hypothesis testing can be far too expensive and time consuming.

A thorough empirical evaluation for this system requires several experiments to evaluate all the dimensions that contribute to and are determining measure for how good or bad the system is. This involves evaluations for the system's functionality and performance, ease of use and usefulness. Therefore, to test the efficacy of the system, many experiments should be conducted to test each of the functional components of the system. This would involve conducting experiments to test the efficacy of the approaches used in the "search", "retrieval", "extraction" and "negotiation" subsystems, both each in isolation and as integral systems. However, as mentioned earlier this would be far too expensive and daunting task and very time consuming. So, instead only two sets of experiments will be conducted to empirically test the efficacy of the negotiation approach and the usability of the system. Then, a set of criteria will be provided for testing the other subsystems, i.e. the search, retrieval and extraction subsystems, that need to be considered when testing their efficacy. However, we these will not be tested, instead an example experiment will be provided to act as an indicator to the effectiveness of the approach then references will be given to previous empirical evaluations of existing information retrieval and extraction systems that use similar approaches.

Two sets of experimental tests are designed. The first set of experiments are designed to test the efficacy of the negotiation approach. First, a role playing experiment with students in the roles of suppliers and customer was used to conduct stylised sales bargaining cases. Second, simulated replications of the first set of experiments were conducted in which human consumer subjects are replaced with the system. Those experiments will enable the investigation of the relationship between the bargaining behavior of the system and humans. Section 5.1 begins with an overview of the experiments then the specific hypotheses to be tested are given and the experimental design is presented.

The second set of experiments are designed to test the ease of use and usefulness of the system. To test the system's usability, a questionnaire method is used to gauge user's subjective satisfaction with the system. Subjects received hands-on demonstration on the system, then they completed the questionnaire. This is described in Section 5.2.

In Section 5.3, evaluation techniques that can be used when testing the search approach are discussed. References to previous evaluation works for similar task are also provided. Then, an example experiment is reported that is conducted to act as indicator to the effectiveness of the approach.

Finally in Section 5.4, evaluation techniques that can be used when investigating the wrapper generation approach of similar previous works in the area are discussed. Then, an example experiment is reported that is conducted to act as indicator to the effectiveness of the approach.

5.1 Negotiation Evaluation

To test the efficacy of the negotiation system, a laboratory experiment was designed using simulated multi-party, multi-issue sales bargaining. First, a role playing experiment with students in the roles of suppliers and customer was used to provide values of conventional face-to-face human subjects bargaining. Then, a simulated replication to the experiment with the software agents playing the roles of the customer and interface to the suppliers was used to provide values of the proposed approach to negotiation. The design of the two experiments gives a common ground for the comparison and analysis of the bargaining behaviour of both approaches. In the following subsections the specific hypotheses to be tested are presented, then the experimental design is addressed.

The design of the experiments is set in the context of previous "laboratory" experiments in Management Science. For example, Rangaswamy and Shell (1997) report on four dimensional stylised business negotiation similar to this thesis approach. They present a negotiation support system that enables negotiators to analyse their own preferences and provides a structured negotiation process to help parties move towards optimal trade. The underlying model is based on a multi-attribute representation of preferences and communications over a computer network. Their results suggest that parties using the system in structured negotiation settings would achieve better outcomes than parties negotiating face-to-face or over an e-mail messaging facility.

5.1.1 Hypotheses

In designing the experiment, answers to two principal questions are sought: (1) would software agents acting on behalf of customers achieve a higher proportion of efficient agreements as compared to customers using conventional face-to-face negotiations? and (2) can software agents learn effective general bargaining strategies?

Humans have to make decisions and are aware of the process attributes and their cognitive limitations, time and effort constraints (Pruitt 1981). Therefore, they often make decisions accepting inconsistencies and errors. The amount of mental effort and the psychological costs of decision making engendered by a decision process can change individual's preferences. Further, any measurement of values, criteria and preferences is bound by simplifications and errors which are difficult to assess.

Humans, sometimes, are not clear about their own priorities and not always too good at finding optimal trade because of the complex pattern that characterises multi-issue negotiation. Further, human emotions often interfere with rational judgement (Pruitt 1981). On the other hand, negotiation systems can be designed to foster more effective results by using the computational strength to compensate for the inherent weaknesses of humans. Algorithms can be written to help humans to identify their preferences and evaluate the attractiveness of alternatives. Further, these systems are well suited for removing the bias in the decision making process with respect to anchoring and adjustment (Hoch 1994).

The above argumentation leads to the following hypotheses:

HYPOTHESIS 1. Software agents enhance achievement of agreements with better payoff outcomes.

HYPOTHESIS 2. Software agents can learn general bargaining strategies that can be used in general trade scenarios.

5.1.2 The Negotiation Scenario

To facilitate the presentation, consider a case where a customer's agent is negotiating a deal to buy a number of PCs for his client with three candidate suppliers namely IBM, NCR and Compaq. The five important issues to be negotiated at the same time are price, CPU speed, RAM size, quantity and delivery time. A range of options was stipulated for each issue and the preference for each option and the relative importance for each issue for all parties are as shown in Figures 5.1-5.5. Due to reasons such as brand loyalty, newness of the product in the marketplace, marketing expertise, etc. the customer adds the brand as a further issue that affects the overall utility payoff of an offer, as depicted in Figure 5.4. Here normalised values are used for issues' preferences so that the total will be 100, and hence the maximum achievable payoff by any party is 1000. It should be clear that these data are not real about the named suppliers, and that the use of existing names was chosen to stimulate the experiment. However, the selection of those data is not random or chosen to suit or affect the results of the thesis.

The data were established from experience and the observed reality of the suppliers' behaviour. For example, because of the high demand on a specific brand or the limitation of the production line of some suppliers (e.g. IBM), they would always prefer to set delayed delivery time and this would have a high priority in their negotiations. It should be also clear that although the descriptive validity of non-linear preference functions was acknowledged and that non-linear contingencies are the rule rather than exception, a decision was made not to have any because of the complexity and the problems posed for negotiators by non-linear functions (Northcraft et al. 1995). It will be very difficult for a human subject to evaluate alternatives' attractiveness (payoff) when non-linear preferences are introduced.

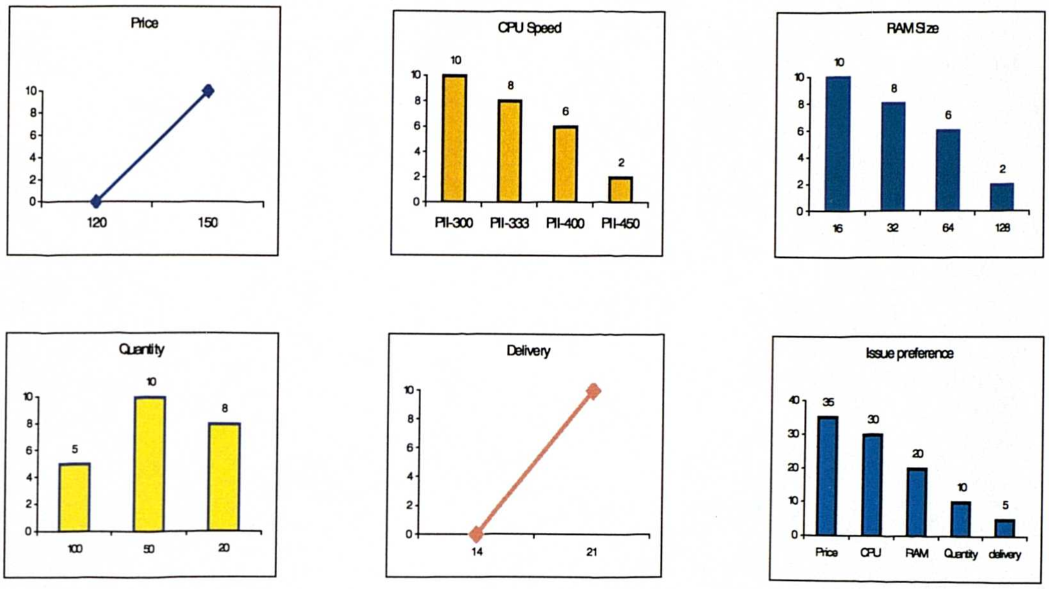


FIGURE 5.1. IBM SUPPLIER UTILITIES.

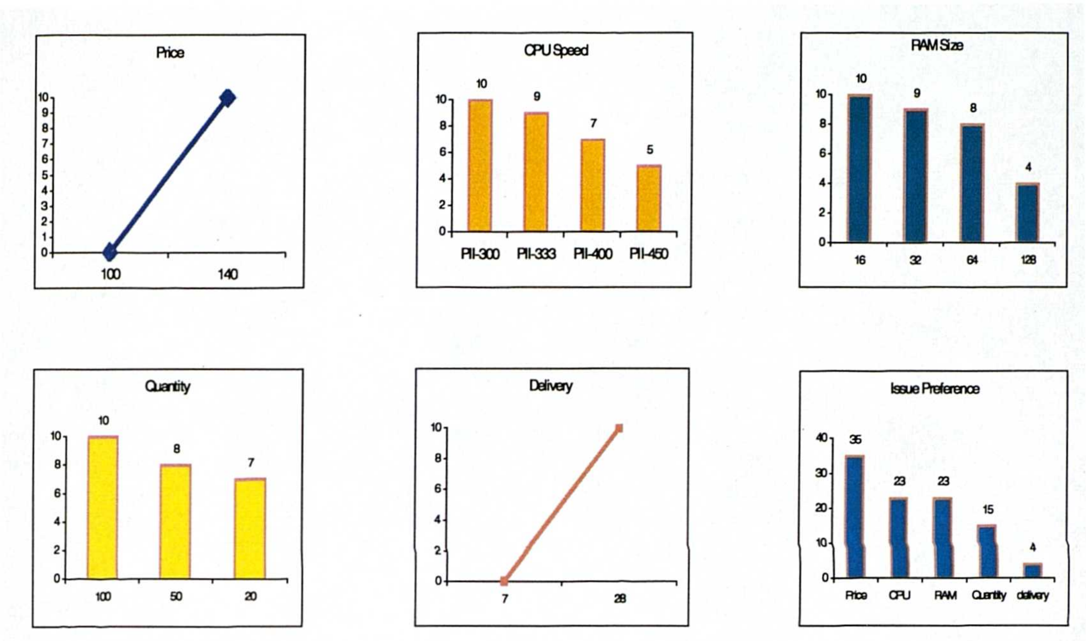


FIGURE 5.2. NCR SUPPLIER UTILITIES.

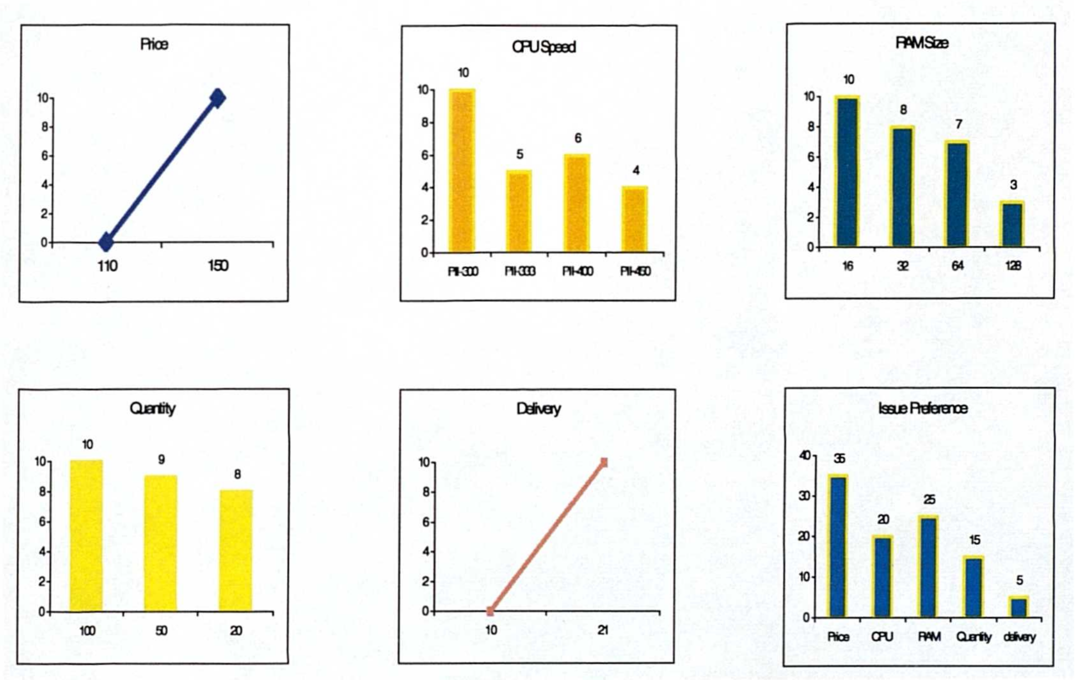


FIGURE 5.3. COMPAQ SUPPLIER UTILITIES.

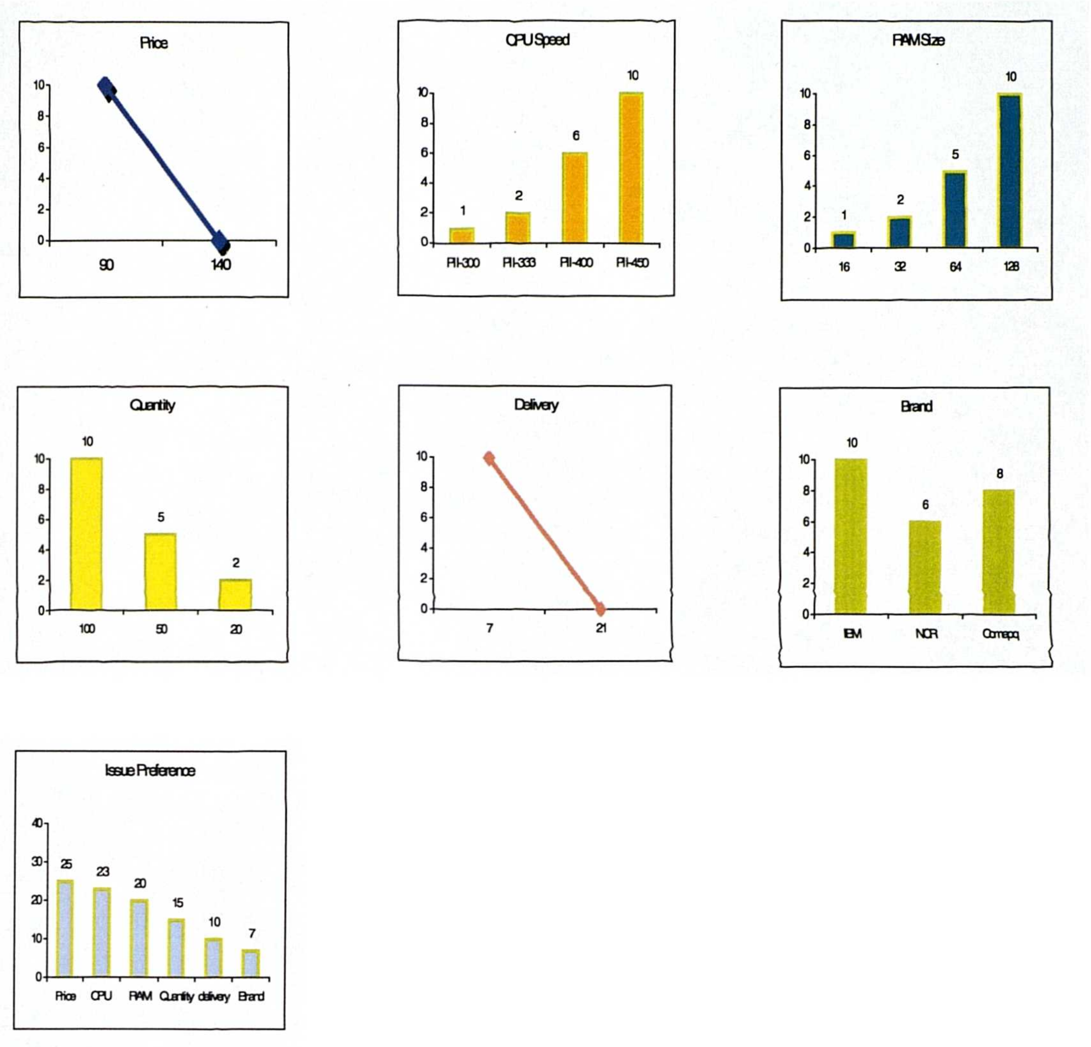
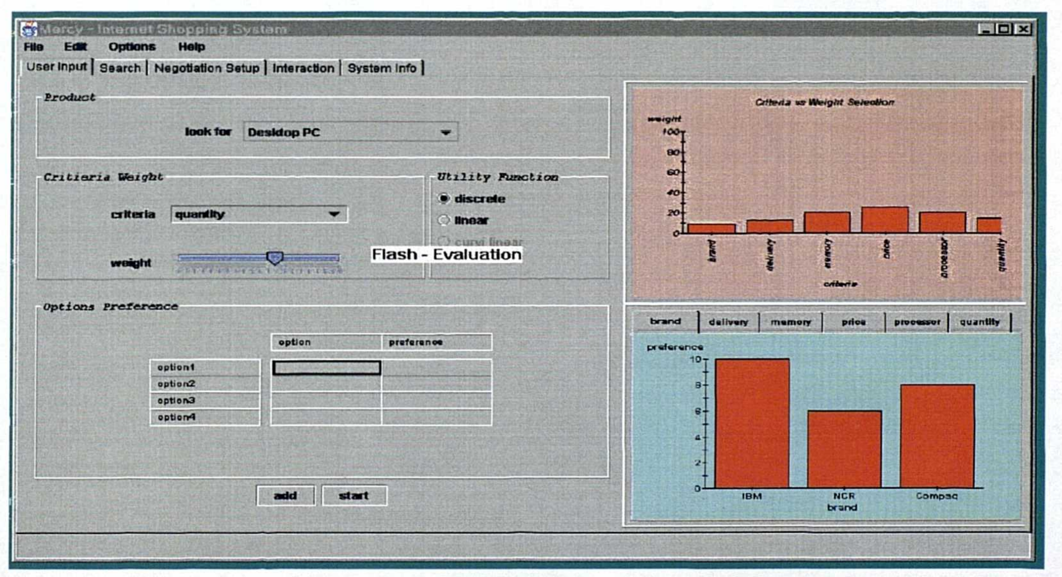


Figure 5.4. Customer's utilities.



look for: Desktop PC

criteria: quantity

weight

● discrete
○ linear
○ curvi linear

	option	preference
option1		
option2		
option3		
option4		

FIGURE 5.4. SNAPSHOT OF CUSTOMER'S PREFERENCE SETUP.

5.1.3 Experimental Setup

Twenty-four student from different levels (12 first year, 4 third year and 8 Masters students) at the department of Information Systems and Computing of Brunel University participated in this experiment. Students were randomly assigned to represent either a customer's agent or a supplier in negotiations regarding the trade for each of the three different brands. A classroom session was held in which the goal of the experiment and the utility function is introduced to all participating subjects. To give the subjects a tangible incentive to negotiate towards the goals stated in their respective role instructions, subjects were further informed that a prize would be awarded to the buyer and seller who gets the best payoff. Then, every subject was provided with the experimental sheet and only the utility function for the role to play. In Appendix I, the experimental sheets for both suppliers and consumer subjects are shown.

The experiment was conducted over four sessions. Six students participated in each session of the experiment, of which three performed the role of the suppliers and the other three played the role of the customer's agent. During the conduct of the negotiation sessions, subjects who played the role of the suppliers were negotiating with either a subject who played the role of a customer or the negotiation agent software. In each negotiation session, the three subjects who played the role of suppliers were distributed in three separate supervised rooms and each was provided with a PC running the supplier's subsystem to communicate with the negotiation agent software. Subjects who played the role of a customer's agent were permitted to freely negotiate with any of the suppliers in whatever order and as many times as they wished for as long as it took them to reach an agreement.

Similarly, the utility function and the preferences of the customer were fed into the developed application, then the system started to simultaneously negotiate with all the three suppliers. The result of the negotiated agreement was recorded and the system was started for another two times. Therefore in each experiment session, the system made three negotiated agreements with the same subjects. For the experiment to be repeatable, the Java source code of the developed system that includes the negotiation agent code is given in Appendix H.

The following rules governed the negotiation sessions:

1. Transmission of any information amongst the negotiators was permitted. Thus, for example, it was legal for the customer's agent to say to one of the suppliers " I received better offer from supplier X, therefore I will buy from him if you do not improve your offer". It was also legal to say " How about increasing the RAM to 128MB and keep everything else the same". However, communication between suppliers themselves was illegal.

2. The goal for each of the participating parties was to maximise his outcome payoff. Therefore, each supplier was trying to get the best deal under the most favourable settings. Similarly, each customer's agent was trying to get the least cost under most favourable conditions.

3. The profit margin for all of the suppliers was set to 30%. For simplicity, it was assume that the payoff of an offer is the same as the selling value, which means that an offer with a payoff of less than 700 (where 1000 is the maximum achievable payoff) was not feasible.

After each session, negotiated agreements from all subjects are gathered and the payoff for each deal is accordingly evaluated. Similarly, the payoff of the negotiated agreements received by the negotiation agent system is evaluated. The experiment scenario is depicted in Figure 5.6.

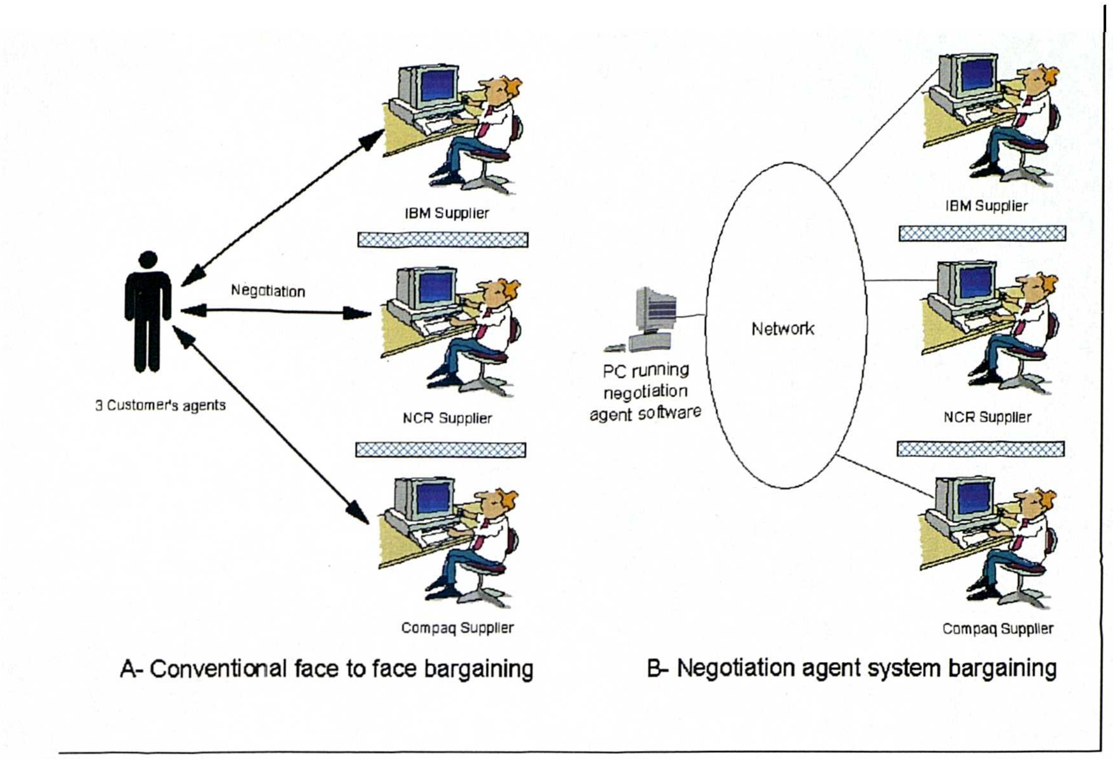


FIGURE 5.6. EXPERIMENTAL SETUP – NEGOTIATION SCENARIO.

5.2 USABILITY EVALUATION

Software usability is a multidimensional and diverse concept. The various perspectives on usability are due to different ideas concerning the referent of the concept, its scope, specificity and objectivity. The referent of usability refers to different approaches to usability; usability as a design approach, usability as a product attribute, usability as a measurement and subjective usability measurement. Depending on the context, usability may be a property of a development process, interaction, user's experience or expectations.

Some approaches focus on the core human-computer interaction in which the usefulness of the system is taken for granted. On the other extreme, others strongly urge the importance of paying attention to all possible contextual variables.

This thesis approach to usability focuses on subjects' personal experience with the system (subjective usability measurement). Instead of theoretical considerations, one of the available scales for measuring users' subjective experience as suggested by one of the well-known tools is used. Here, Davis (1989) Technology Acceptance Model (TAM) is used to explain and predict user acceptance of the system from measure taken after a brief period of interaction with the system. TAM predicts that the usage is determined by the behavioural intention that is jointly determined by the perceived usefulness and ease of use. Perceived usefulness is defined as the prospective users subjective probability that using a specific application system will increase his job performance. Ease of use refers to the degree to which the prospective user expects that the target system to be free of effort. The relationships between the factors are presented in Figure 5.7. Attitude is determined by cognitive beliefs, i.e. TAM follows the model of attitude formation suggested by Fishbein & Ajzen (1975).

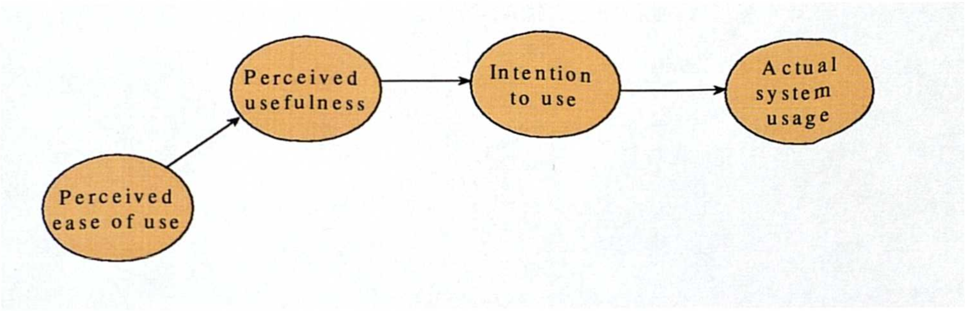


FIGURE 5.7. TECHNOLOGY ACCEPTANCE MODEL

Beliefs are measured with a 7-point Likert scale with high reliability (perceived usefulness r= 0.97 and perceived ease of use r= 0.91). The items of perceived usefulness address the quality and quantity of work accomplished. Special emphasis is given to measuring temporal efficiency (3 items) and the feeling of being in control of the work (3 items). Perceived ease of use measures the users' ideas on learning to use the system, the control experienced over the system and the mental effort involved in use. Appendix D presents the scale items of TAM; perceived usefulness and perceived ease of use.

Many researchers have conducted several studies to examine the relationship between perceived ease of use, perceived usefulness and the system usage (Adams et al. 1992, Chau 1996, Davis et al. 1989, Mathieson 1991). The original scales for measuring the TAM constructs have been confirmed to be reliable and valid in those studies, using several replications and applications including e-mail, spreadsheets and word *processing (Adams et al. 1992). Thus TAM's* attitude-intention-behaviour relationship is used as a valid model to predict users' acceptance of the system.

To test the usability of the proposed consumer-oriented EC system, an experiment was ran in which the questionnaire method was used to predict the user's behavioural intention to use the system. The usability is tested as surrogates for technology acceptance.

5.2.1 Hypothesis

The main reason for conducting this experiment is to test how usable the proposed solution to Internet shopping is. To put it in another way: Is it feasible to develop an intelligent system that would allow consumers to obtain services and make effortless Internet purchase?

Since the usability emphasis here is on subjects' personal experience with a product (subjective measurement), then the core focus of the test, as stated earlier, is on; accomplishment, efficiency, learn-ability, control and mental effort required form the user. To address the control requirement, the design of the system considered many factors to assure that the user of the system remains in ultimate control. The system was designed to provide reports and support for browsing and monitoring so the user is always informed how the system is performing. The design of the preferences' elicitation approach was done in such a way not to entail major requirements and mental effort from user. The requirement for efficient procedures is a core focus of this thesis, therefore it is expected that the thesis meets this requirement.

This leads to the following formal hypothesis:

HYPOTHESIS 3. If the design requirements for Internet-based consumer-oriented EC system are carefully considered for subjective usability measurement, a usable system can be developed.

5.2.2 Experiment Design

The subjects were fifteen students who received a hands-on demonstration followed by thirty minutes interaction with the system. The requirement for a random sample of subjects that would include a range of skills and experiences is acknowledged. This means that the sample should include subjects with none to only some experience with the Internet shopping. Therefore, students from different levels (first, second, third, Masters and PhD students) who come from different backgrounds (different departments at Brunel University) were selected. Following the interaction, the students completed the perceived usefulness/ ease of use instrument based on Davis's (1989). Appendix D shows the usefulness/EOU questionnaire as given to the subjects.

5.3 SEARCH EVALUATION

An information retrieval method for the Web can be generally divided into two parts, indexing and query. The first part concerns the creation of an index to the documents in the Web (described in Section 2.2). This reduces, in effect, the Web known contents by the search engine being used to the indexed information. The second part consists of matching a search query against the document representatives in the index, to find relevant ones. The meta-searching approach does not discuss the problem of indexing the documents on the Web, instead the indexing provided by the underlying search engine is just used as it is. The focus is on what can be done with the search query to the underlying search engine, that can be divided into the collection step (pre processing) and analysis step (post processing). In the collection step, finding all conceivably relevant documents by means of properly formated query is attempted, while trying to keep the irrelevant noise to the minimum to reduce computation time for the next step. In the analysis step, the documents' contents returned as a result from the first step are analysed to filter out irrelevant documents. The traditional performance measures notions used in IR, i.e. recall and precision, are not directly applicable and are not suitable for dynamic environments. Recall and precision assume that a document is either relevant or not, whereas the relevance of a document is subjective measure and that the number of actual relevant documents on the Web is unknown even if considered as objective.

This thesis claim that the developed techniques result in a performance improvement can use modified performance measures as follows:

- Towards high recall: locating as many documents as possible that are conceivably relevant to the query. This is done in the pre processing phase before querying the underlying search engine.

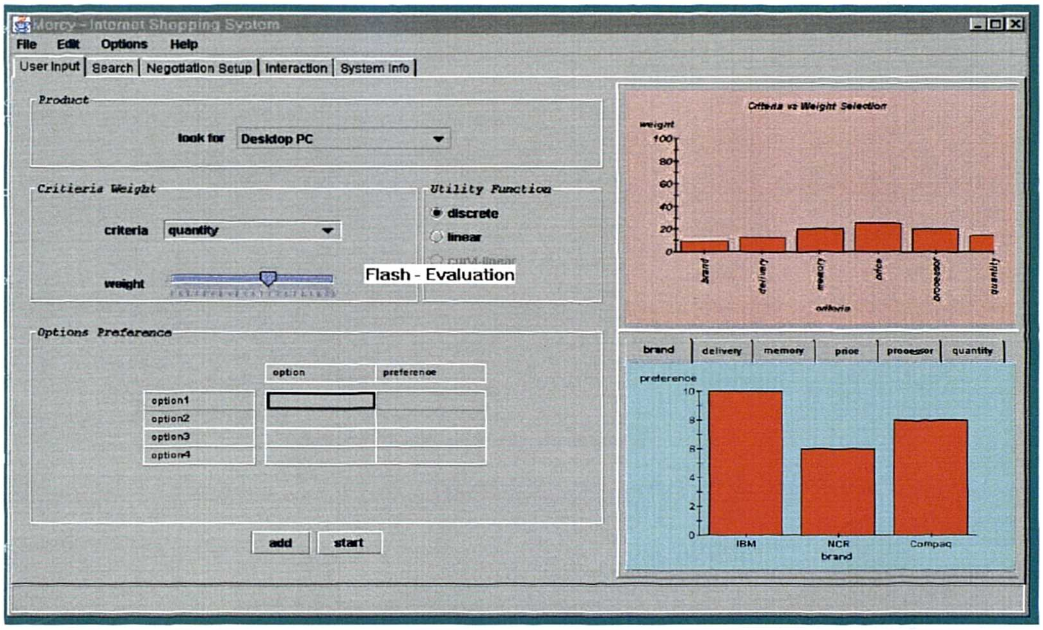
- Towards high precision: Filtering out irrelevant documents by analysing documents' content. This is done in the post processing phase after retrieving the results.

With the emergence of many Internet search services, there have been many different efforts to create single interface to these services. Therefore, unifying these services under one interface is far from novel, instead it is the pre-processing and post processing steps that makes the difference in implementation success. Several of these works has been published describing the performance measures that support their claims that their approach is viable. Similar approaches to those works, such as MetaCrawler (Selberg & Etzioni 1995), ProFusion (Gauch 1996) and SavvySearch (Dreilinger 1996), can be used to test the performance of our search procedure.

In the following subsection, an experiment is presented to act as an indicator to the effectiveness of the developed techniques to improve search results.

5.3.1 Example of search effectiveness

In this experiment, the consumer "user" is interested in buying a personal computer. The system is started and the user selected the product he is interested in and specified his selection criteria and preference weights. Then the user clicked on start to begin the search process. A snapshot of the user interface showing the selection criteria for the user is given in Figure 5.8.



Product

look for: Desktop PC

Critieria Weight

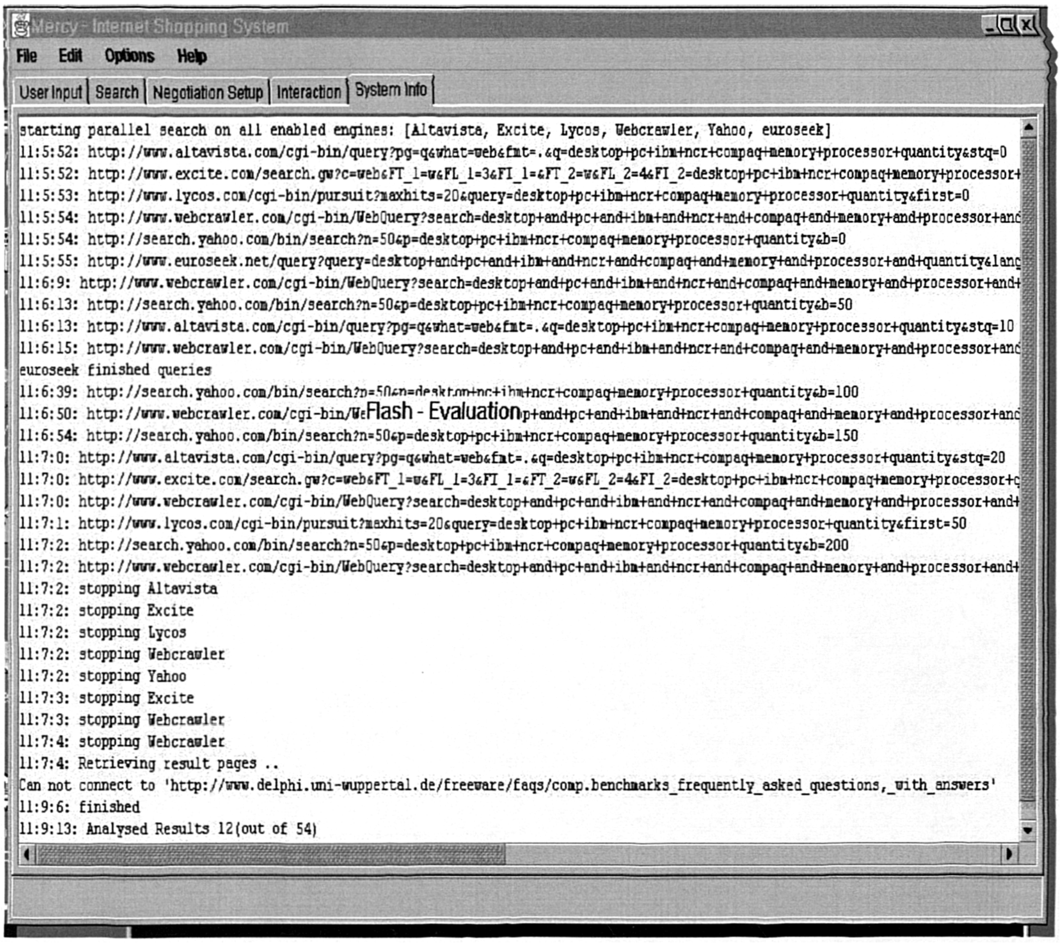
criteria: quantity

Utility Function

● discrete
○ linear
○ curvi-linear

FIGURE 5.8. SNAPSHOT OF THE USER INTERFACE SHOWING USER'S PREFERENCES.

A snapshot of the system's search progress is given in Figure 5.9. Here the user chose to enable a number of search engines on which to execute the query. The user has the capability to choose what search engines to enable, disable or add a new search engine to the given list of search engines (see Figure 5.10). The progress shows how the query is composed of the product name and the specified attributes (to minimise the noise in the retrieved results). The figure also shows the time taken to query the underlying search engines (one minute) and the time taken to analyse those retrieved results (two minutes). The user can modify those parameters. Finally, the figure shows the number of retrieved results (54 URLs) and the number of analysed results (12 URLs) before time-out.



```
starting parallel search on all enabled engines: [Altavista, Excite, Lycos, Webcrawler, Yahoo, euroseek]
11:5:52: http://www.altavista.com/cgi-bin/query?pg=q&what=web&fmt=.&q=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&stq=0
11:5:52: http://www.excite.com/search.gw?c=web&FT_1=w&FL_1=3&FI_1=&FT_2=w&FL_2=4&FI_2=desktop+pc+ibm+ncr+compaq+memory+processor+
11:5:53: http://www.lycos.com/cgi-bin/pursuit?maxhits=20&query=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&first=0
11:5:54: http://www.webcrawler.com/cgi-bin/WebQuery?search=desktop+and+pc+and+ibm+and+ncr+and+compaq+and+memory+and+processor+and
11:5:54: http://search.yahoo.com/bin/search?n=50&p=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&b=0
11:5:55: http://www.euroseek.net/query?query=desktop+and+pc+and+ibm+and+ncr+and+compaq+and+memory+and+processor+and+quantity&lang
11:6:9: http://www.webcrawler.com/cgi-bin/WebQuery?search=desktop+and+pc+and+ibm+and+ncr+and+compaq+and+memory+and+processor+and+
11:6:13: http://search.yahoo.com/bin/search?n=50&p=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&b=50
11:6:13: http://www.altavista.com/cgi-bin/query?pg=q&what=web&fmt=.&q=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&stq=10
11:6:15: http://www.webcrawler.com/cgi-bin/WebQuery?search=desktop+and+pc+and+ibm+and+ncr+and+compaq+and+memory+and+processor+and
euroseek finished queries
11:6:39: http://search.yahoo.com/bin/search?n=50&p=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&b=100
11:6:50: http://www.webcrawler.com/cgi-bin/WeFlash - Evaluationp+and+pc+and+ibm+and+ncr+and+compaq+and+memory+and+processor+and
11:6:54: http://search.yahoo.com/bin/search?n=50&p=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&b=150
11:7:0: http://www.altavista.com/cgi-bin/query?pg=q&what=web&fmt=.&q=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&stq=20
11:7:0: http://www.excite.com/search.gw?c=web&FT_1=w&FL_1=3&FI_1=&FT_2=w&FL_2=4&FI_2=desktop+pc+ibm+ncr+compaq+memory+processor+c
11:7:0: http://www.webcrawler.com/cgi-bin/WebQuery?search=desktop+and+pc+and+ibm+and+ncr+and+compaq+and+memory+and+processor+and+
11:7:1: http://www.lycos.com/cgi-bin/pursuit?maxhits=20&query=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&first=50
11:7:2: http://search.yahoo.com/bin/search?n=50&p=desktop+pc+ibm+ncr+compaq+memory+processor+quantity&b=200
11:7:2: http://www.webcrawler.com/cgi-bin/WebQuery?search=desktop+and+pc+and+ibm+and+ncr+and+compaq+and+memory+and+processor+and+
11:7:2: stopping Altavista
11:7:2: stopping Excite
11:7:2: stopping Lycos
11:7:2: stopping Webcrawler
11:7:2: stopping Yahoo
11:7:3: stopping Excite
11:7:3: stopping Webcrawler
11:7:4: stopping Webcrawler
11:7:4: Retrieving result pages ..
Can not connect to 'http://www.delphi.uni-wuppertal.de/freeware/faqs/comp.benchmarks_frequently_asked_questions,_with_answers'
11:9:6: finished
11:9:13: Analysed Results 12(out of 54)
```

FIGURE 5.9. SNAPSHOT OF THE SYSTEM'S SEARCH PROGRESS FOR THE EXAMPLE.

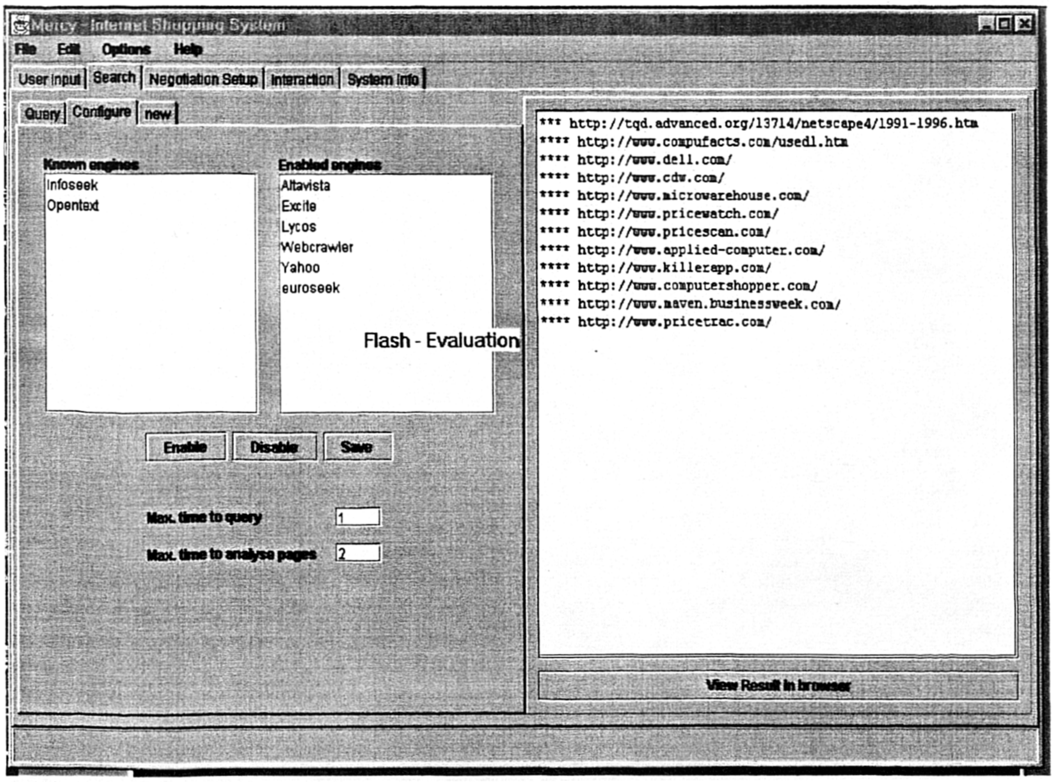
A snapshot of the system's analysed results is shown in Figure 5.10. Out of the provided twelve results, eleven hits are relevant. Those relevant results are ranked

"****" while the one that is not of interest is ranked "***". The relevancy percentage is 92%.

A further analysis of the remaining results not reported in the system as related links either because of time-out of the analysis time or because those are irrelevant, (Appendix E lists all results retrieved from Altavista by the system before time-out), the following is found.

Out of the twenty results returned by Altavista (Altavista is taken randomly as an example) the following is found.

- Number of dead links is two (i.e. percentage of 10%)
- Number of duplicate results is one (i.e. percentage of 5%)
- Number of irrelevant results is ten (i.e. percentage of 50%)
- Number of relevant results is five (i.e. percentage of 25%).



Max. time to query 1

Max. time to analyse pages 2

```
*** http://tqd.advanced.org/13714/netscape4/1991-1996.htm
**** http://www.compufacts.com/usedl.htm
**** http://www.dell.com/
**** http://www.cdw.com/
**** http://www.microwarehouse.com/
**** http://www.pricewatch.com/
**** http://www.pricescan.com/
**** http://www.applied-computer.com/
**** http://www.killerapp.com/
**** http://www.computershopper.com/
**** http://www.maven.businessweek.com/
**** http://www.pricetrac.com/
```

FIGURE 5.10. SNAPSHOT OF THE SYSTEM'S SEARCH RESULTS FOR OUR EXAMPLE.

Therefore, Altavista's relevancy percentage for this example experiment is 25%. Comparing the two relevancy percentages, a noticeable enhancement is realised when using the system. In addition to this difference, only two of the results reported as relevant by the system came from Altavista. A possible reason for other results not reported by altavista is that the region of the resources from which the results came from are not covered by Altavista and hence not indexed by the engine. Another possibility, is that those results have a lower ranking by the engine (i.e. the results would have been reported later by the engine if the timer did not time-out).

To summarise, although more thorough investigation is required before a statement can be made, this example experiment acts as indicator to the effectiveness of the pre and post processing techniques the system employs to enhance the search results.

5.4 Information Extraction Evaluation

With regard to information extraction, the wrapper generation approach to information extraction is used for the reasons explained in Section 2.2. Wrappers are built for structured Internet sources by exploiting the formatting information in the Web pages to guess the relevant content of the page. Different wrapper classes are designed for different structures of HTML pages. For example, Kushmerick et al. (1997) LR wrapper class is designed for resources that format their pages in a simple, left-right, manner, whereas HLRT wrapper class is designed to avoid distracting material in a page's head and tail.

Evaluating a given wrapper generation technique can be addressed from different dimensions. One dimension is to evaluate the usefulness of the wrapper class (in the case of this thesis, wrapper for resources that present information in a table format) for handling actual Internet sources. This would include the percentage coverage of actual Internet sites the wrapper can handle. Another dimension, is the efficiency of the wrapper generation techniques. This would include evaluating the complexity of the recognition heuristics that aid wrapper generation and required processing time for generating a wrapper for a given resource.

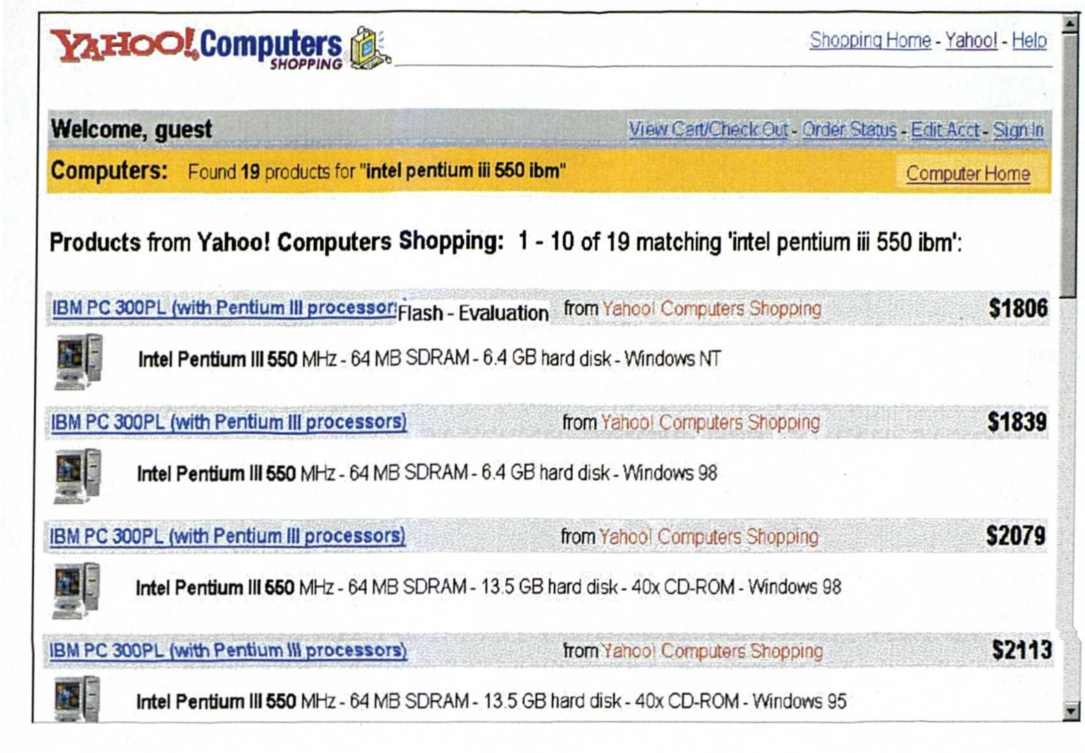
Many published researches on wrapper generation include the performance measure criteria and empirical evaluations to support their claim that their approach is viable. These include Kushmerick et al. (1997), Soderland (1997a) and Ashish & Knoblock (1998), from which evaluation approach can be inspired when testing this thesis information extraction techniques.

In the following subsection, an example experiment is presented to act as indicator of the effectiveness of the wrapper generation approach for extracting relevant information.

5.4.1 Example of the wrapper generation effectiveness

In this experiment, the system is used to construct a wrapper for a given Web source that presents the information in HTML tables. "Yahoo! Computer Shopper" Web site was chosen to give an appreciation to the real applicability of the approach.

The input to the system was an example HTML page from the Web site that is shown in Figure 5.11. The HTML document source is given in Appendix F. The system is also provided with a number of regular expressions for price and product's specific attributes recognition (for processor, memory, hard disk, CD-ROM and operating system). OROMatcher (www.oroinc.com) Java-library was also used for regular expression compiler to generate and compile these patterns.



YAHOO! Computers SHOPPING Shopping Home - Yahoo! - Help

Welcome, guest View Cart/Check Out - Order Status - Edit Acct - Sign In

Computers: Found 19 products for "intel pentium iii 550 ibm" Computer Home

Products from Yahoo! Computers Shopping: 1 - 10 of 19 matching 'intel pentium iii 550 ibm':

IBM PC 300PL (with Pentium III processor Flash - Evaluation from Yahoo! Computers Shopping $1806

Intel Pentium III 550 MHz - 64 MB SDRAM - 6.4 GB hard disk - Windows NT

IBM PC 300PL (with Pentium III processors) from Yahoo! Computers Shopping $1839

Intel Pentium III 550 MHz - 64 MB SDRAM - 6.4 GB hard disk - Windows 98

IBM PC 300PL (with Pentium III processors) from Yahoo! Computers Shopping $2079

Intel Pentium III 550 MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM - Windows 98

IBM PC 300PL (with Pentium III processors) from Yahoo! Computers Shopping $2113

Intel Pentium III 550 MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM - Windows 95

FIGURE 5.11. SNAPSHOT OF THE EXAMPLE HTML PAGE FOR OUR EXAMPLE.

A snapshot of the system's progress is given in Figure 5.12. The figure shows the results of the extraction process and the subsequent identification of the required attributes. The full listing of the system's progress is shown in Appendix G. The results show that the wrapper perfectly recognised the prices and the attributes of the personal computer given by the source.

From wrapper's results the following is postulated by the system:

- Prices recognised in the following tuples, where a tuple is given by (table number, row number, column number).
  (7, n, 4) for (n =1, n<= 27, n+3).

- Attribute "processor" is recognised in:
  (7, n, 3) for (n=2, n<= 27, n+3).

- Attribute “memory” is recognised in:
  (7, n, 3) for (n=2, n<= 27, n+3).

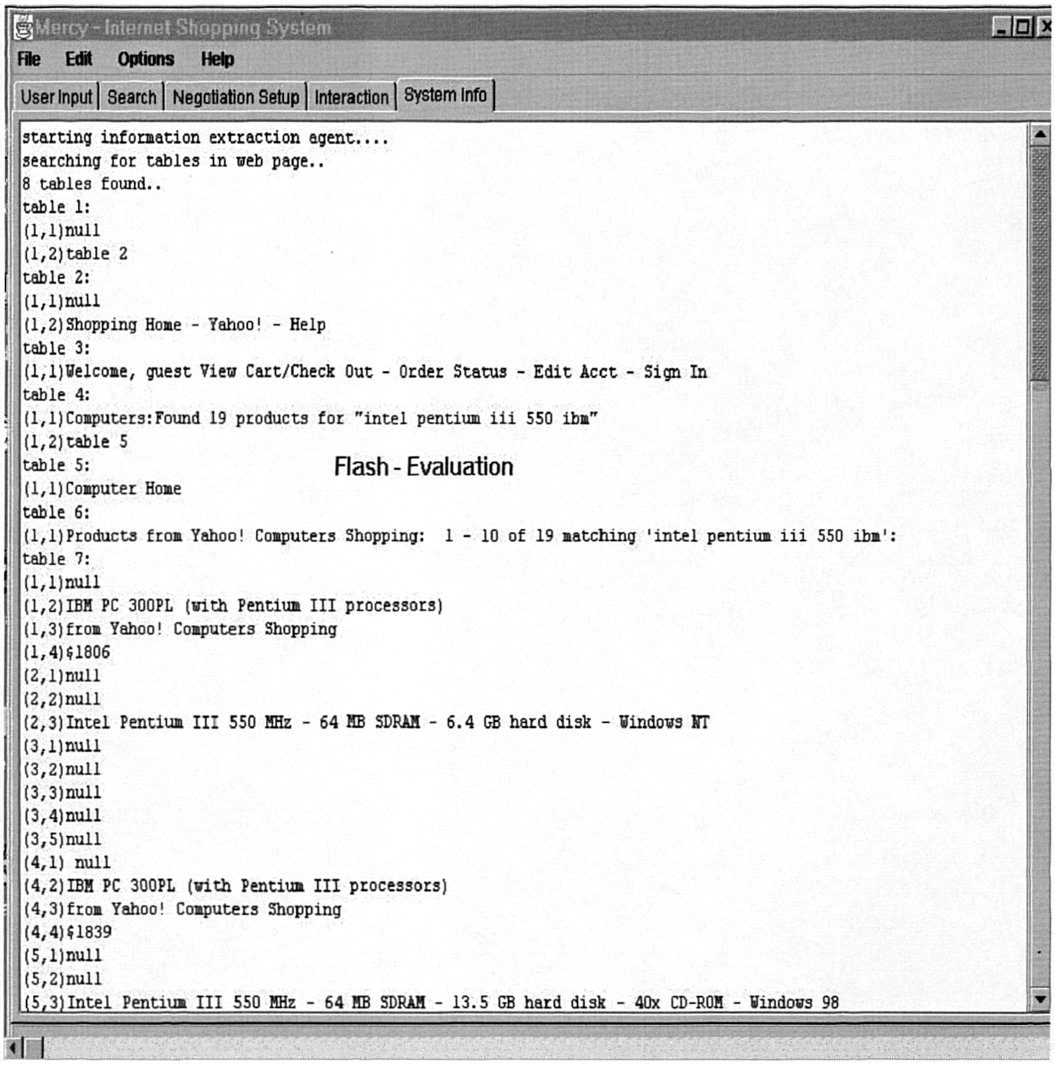
- Attribute “hard disk” is recognised in:
  (7, n, 3) for (n=2, n<= 27, n+3).

- Attribute “CD-ROM” is recognised in:
  (7, n, 3) for (n=2, n<= 27, n+3).

- Attribute “operating system” is recognised in:
  (7, n, 3) for (n=2, n<= 27, n+3).

In addition to those postulates the system records the ontology used in this resource (i.e. the way each attribute is represented such as Intel Pentium III for the processor instead of P-III). This would help in avoiding (or at least partially avoiding) the process of pattern matching in subsequent information extraction from the same source. Therefore, subsequent information extractions would be restricted to the seventh table and the patterns observed earlier.

Because the OROMatcher’s Java class library for pattern matching was borrowed, the computational complexities it involves is not known. However, the wrapper algorithm for HTML-tables the thesis employ uses a simple string search operations (for tags such as <TABLE>, </TABLE>, <TR>, </TR>, <TD> and </TD>) that can use efficient techniques (for example, Knuth (1973)). Subsequent data extraction from other HTML documents from the same source need not to go through the pattern matching step and hence would be faster.



```
starting information extraction agent....
searching for tables in web page..
8 tables found..
table 1:
(1,1)null
(1,2)table 2
table 2:
(1,1)null
(1,2)Shopping Home - Yahoo! - Help
table 3:
(1,1)Welcome, guest View Cart/Check Out - Order Status - Edit Acct - Sign In
table 4:
(1,1)Computers:Found 19 products for "intel pentium iii 550 ibm"
(1,2)table 5
table 5:
(1,1)Computer Home
table 6:
(1,1)Products from Yahoo! Computers Shopping:  1 - 10 of 19 matching 'intel pentium iii 550 ibm':
table 7:
(1,1)null
(1,2)IBM PC 300PL (with Pentium III processors)
(1,3)from Yahoo! Computers Shopping
(1,4)$1806
(2,1)null
(2,2)null
(2,3)Intel Pentium III 550 MHz - 64 MB SDRAM - 6.4 GB hard disk - Windows NT
(3,1)null
(3,2)null
(3,3)null
(3,4)null
(3,5)null
(4,1) null
(4,2)IBM PC 300PL (with Pentium III processors)
(4,3)from Yahoo! Computers Shopping
(4,4)$1839
(5,1)null
(5,2)null
(5,3)Intel Pentium III 550 MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM - Windows 98
```

FIGURE 5.12. SNAPSHOT OF SYSTEM'S EXTRACTION RESULT FOR THE EXAMPLE.

CHAPTER 6

RESULTS AND ANALYSIS

In this chapter, the results of the experiments presented in Chapter 5 are reported. First, the results of negotiation experiments and the analysis of these results are presented. Observed humans' bargaining behaviour that, possibly, affected the negotiations' outcomes is reported. Both, the quantitative and qualitative analysis, are presented in Section 6.1. Second, the results of the usability testing and the analysis of these results are presented. Some of the comments made by the evaluators are also reported.

6.1 NEGOTIATION RESULTS

Investigating the relationship between the negotiation behaviour of humans and the software agents could follow many avenues. The thesis focuses on the individual payoffs and joint payoffs of each scenario for the evaluation. Table 6.1 shows the performance payoff of humans' negotiation (i.e. both parties are humans) for each of the conducted twelve negotiations sessions. Table 6.2 shows the performance payoff of the negotiation agents and the humans' playing the role of suppliers for each of the twelve sessions conducted.

To test **HYPOTHESIS 1**, a t-test is performed on the difference between the two means of both the human subjects' and the negotiation agents' payoff values. The t-test value provides a statistical evaluation of the null hypothesis (i.e. the difference between the two means is 0). The mean of the difference variable is

0.79, which is tested against the null value of 0.0. The standard error of the difference is about 0.154 leading to a t-statistic of 5.125 at a p-value of 0.0003. The null hypothesis of no change can be rejected, supporting the hypothesis that the negotiation protocol design for software agents can do better than a human subject in the given trade negotiation scenario.

	Customer's payoff	Supplier's payoff (1)	Joint payoff
Session 1	4.7	7.0	11.7
Session 2	5.7	7.5	13.2
Session 3	5.5	8.1	13.6
Session 4	5.5	8.2	13.7
Session 5	4.7	7.5	12.2
Session 6	4.9	7.7	12.6
Session 7	4.7	8.3	13
Session 8	5.5	8.1	13.6
Session 9	4.7	8.3	13
Session 10	5.2	7.8	13
Session 11	5.1	7.5	12.6
Session 12	5.5	8.2	13.7
Average	5.14	7.85	13.0
S.D.	0.39	0.42	0.63

TABLE 6.1. PAYOFFS' VALUES OF HUMAN SUBJECTS NEGOTIATIONS.

	Agent's payoff	Supplier's payoff (2)	Joint payoff
Session 1	6.1	7.3	13.4
Session 2	5.7	7.5	13.2
Session 3	6.1	7.3	13.4
Session 4	6.3	7.1	13.4
Session 5	6.2	7.0	13.2
Session 6	5.6	7.2	12.8
Session 7	6.7	7.5	14.2
Session 8	6.1	7.3	13.4
Session 9	6.3	7.0	13.3
Session 10	5.3	7.4	12.7
Session 11	6.1	7.3	13.4
Session 12	5.7	7.5	13.2
Average	5.93	7.28	13.3
S.D.	0.32	0.18	0.37

TABLE 6.2. PAYOFFS' VALUES OF NEGOTIATION AGENTS AND HUMAN SUBJECT SUPPLIER.

The above test, i.e. t-test, assumes that the data are measurements from a normal distribution. However, t-test is a robust procedure in that if the measurements come from a continuous distribution which is non-normal, say for a distribution that is slightly skewed, the significance level will not be seriously affected. Table 6.3 presents a descriptive statistics for the data.

Name	median	One quartile	Three quartile	Skewness	Kurtosis
Agent	6.1	5.7	6.1	-0.643	-0.632
Human	5.1	4.7	5.5	-0.001	-1.852

TABLE 6.3. DESCRIPTIVE STATISTICS OF THE EXPERIMENTAL RESULTS.

In addition to the parametric test, non-parametric test is used that does not depend on specific distributional assumptions about the probability distribution from which the measurements are taken. Wilcoxon signed rank test is used with $\alpha = 0.05$, n = 11 (session 2 values are removed because the payoff is the same i.e. the difference is 0). Table 6.3 gives the values for both cases and the sign and rank of the difference between the two payoffs.

Agent	6.1	6.1	6.3	6.2	5.6	6.7	6.1	6.3	5.3	6.1	5.7
Human	4.7	5.5	5.5	4.7	4.9	4.7	5.5	4.7	5.2	5.1	5.5
Difference	1.4	0.6	0.8	1.5	0.7	2.0	0.6	1.6	0.1	1.0	0.2
Sign	+	+	+	+	+	+	+	+	+	+	+
Rank	4	8.5	6	3	7	1	8.5	2	11	5	10

TABLE 6.4. DATA USED IN WILCOXON SIGNED RANK TEST ANALYSIS.

The critical values are determined from Wilcoxon tables to be 10, 56. That is reject the null hypothesis if $T^+ \leq 10$ or $T^+ \geq 56$.

Since all signs are positive then $T^+ = n(n+1)/2 = 66$. Because this value is in the critical region then the null hypothesis is rejected. This concludes that there is a difference in the means of distribution of the payoffs for both human subjects' and agents' negotiation.

To test HYPOTHESIS 2, the different payoffs' values of negotiation agents are compared. The basic idea is that a software agent will have the opportunity to learn general negotiation strategies when faced with opponents with different strategies and preferences. Also, the agent could be exposed to completely different trade scenarios with different number of issues and that the agent should react according to the features of the scenario. Therefore, the standard deviation of 0.32 for the payoffs' values of the agents faced with different suppliers (and hence different strategies) presents an appreciation of the general performance of the agent. Although statistical significance of the result is not scrutinised, this would provide strong evidence supporting the hypothesis. Furthermore, another experiment was conducted in which the screen size was added as an issue in the negotiation. The outcome of the negotiation was similar to others (payoff value 5.9).

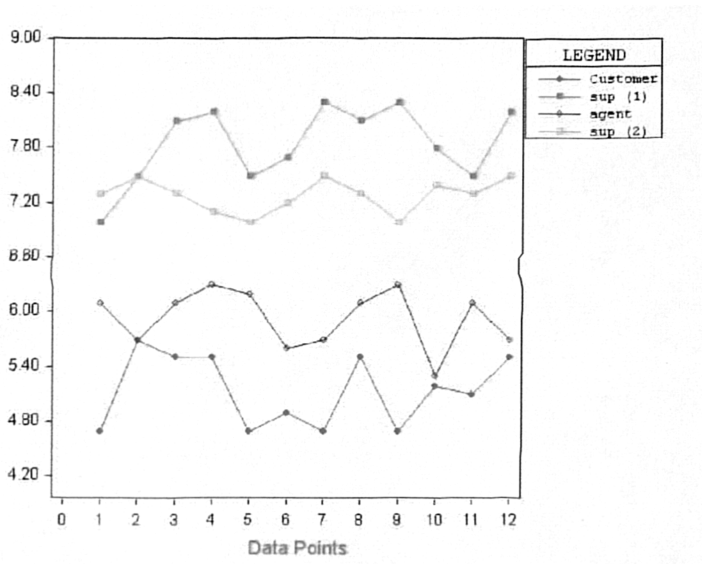


FIGURE 6.1. EXPERIMENTAL PAYOFFS' VALUES FOR ALL NEGOTIATIONS

In addition to the quantitative analysis that provided a support for the hypotheses, further qualitative analysis reveals the following:

- The results suggest that the negotiation is not always of distributive type. The joint payoff and the individual payoff of the winning suppliers shows that the subjects used different strategies to win the contract at the most favourable attainable value. An integrative type of negotiation (higher joint payoff) is used more frequently in the case of agents than humans.

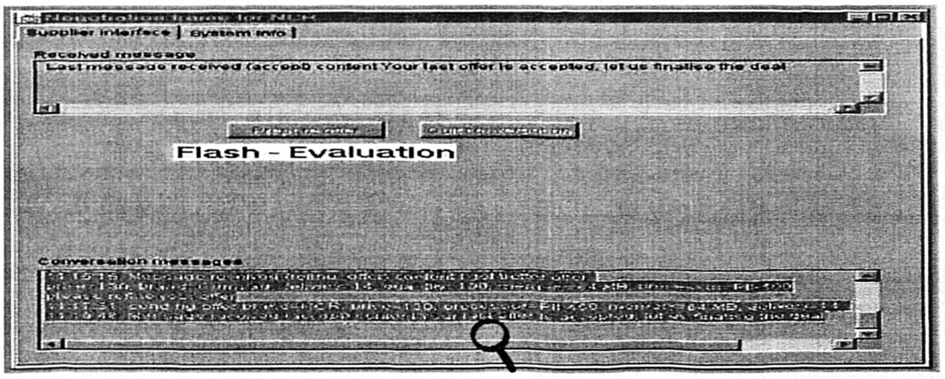
- The mean and standard deviation of payoffs' received by the suppliers negotiating with the system's agents suggest that the introduction of the system in real markets will produce fierce competition and eventually force the suppliers to reduce their profit margins. Further analysis needs to be conducted to study the effect of the system on the supply and demand curve and the potential change of the market from supplier driven to customer driven.

- Negotiation agents negotiated with four sets of suppliers (i.e. four different sessions with three different suppliers), and each three consecutive values in the table present repeated negotiation with same subjects on same issues at same preferences. Hence, if the software agents were exploitable then repeating the same scenario with the same subjects should result in inferior payoffs. The achieved results suggest that even when the opponents get a feeling of the agent's preferences, the performance of the agent is not affected. Further analysis is required for the effect of coalition formation between suppliers, however the fixing of minimum payoff value that the agent should not settle for less than should provide a guard.

- At the end of the negotiation conduct, subjects playing the role of suppliers against system's agents were asked what negotiation strategy, they think, the agents used? Amazingly, none of them did realise that he was competing with others "in what we called virtual reverse auction setup" and that the highest bid was out-cried to every other bidder.

In addition to the above-mentioned experiments, another experiment was conducted in which six human subjects played the role of suppliers (two for IBM, two for NCR and two for Compaq). The idea is to find out whether the system scales up with the increase in number of suppliers. Although the system took a little bit longer (3 seconds) to instantiate all of the six negotiation agents in this experiment, once the agents were created; the offer and counter-offer analysis and creation was instantaneous (no realisable difference between the case of three and six agents). The initial delay is because of the use of RMI registry for agents' communication binding and look up. As stated in Section 3.1, a more flexible and scalable design is to use a specialised matchmaker for this task.

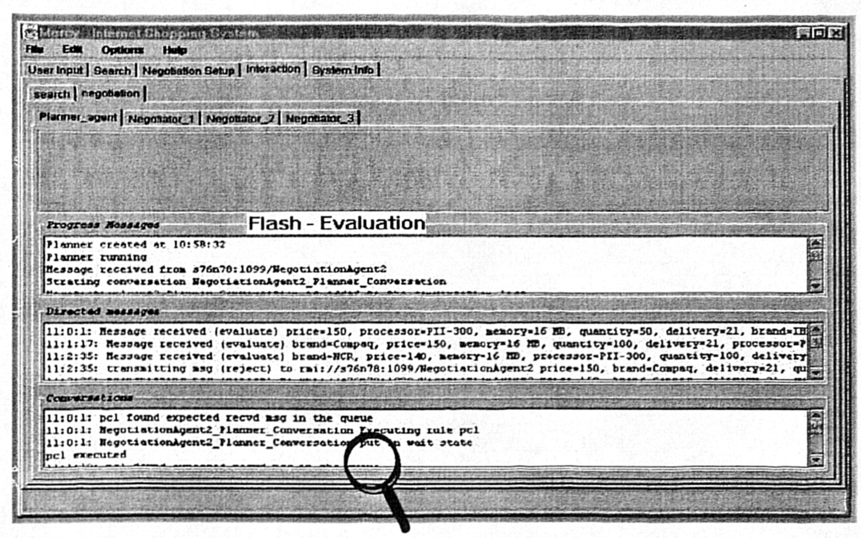
Another interesting observation, is that when two suppliers gave similar offer, the two negotiation agents negotiating with those suppliers kept silent in the next round of the negotiation. Although the outcome of the negotiation is expected to be of higher payoff (due to the increase in the competition), the result was similar to other reported sessions with a payoff value of 6.1. One reason for this is that the profit margin within which the subjects are bounded by is fixed and the same for all of them. This imposes restriction on the space area in which the subjects can manipulate their strategies to construct competing bids.



Last message received (accept) content Your last offer is accepted, let us finalise the deal

```
10:58:37: Message received (RFP) content pc
11:2:35: sending offer brand=NCR, price=140, memory=16 MB, processor=PII-300, quantity=100, delivery=28,
11:2:35: Message received (refine_offer) content I got better offer:
        (price=150, brand=Compaq, delivery=21, quantity=100, memory=16 MB, processor=PII-300) please refine your offer
11:5:33: sending offer brand=NCR, price=140, memory=16 MB, processor=PII-333, quantity=100, delivery=28,
11:8:42: Message received (refine_offer) content I got better offer:
        (price=140, brand=Compaq, delivery=21, quantity=100, memory=16 MB, processor=PII-400) please refine your offer
11:11:56: sending offer brand=NCR, processor=PII-400, price=130, memory=32 MB, quantity=100, delivery=14,
11:15:15: Message received (refine_offer) content I got better offer:
        (price=130, brand=Compaq, delivery=14, quantity=100, memory=64 MB, processor=PII-400) please refine your offer
11:17:31: sending offer brand=NCR, price=130, processor=PII-450, memory=64 MB, delivery=14, quantity=100,
11:19:48: Message received (accept) content Your last offer is accepted, let us finalise the deal
```

FIGURE 6.2. PROGRESS OF ONE NEGOTIATION SESSION FOR THE DEAL WINNER SUPPLIER.



Progress Messages

```
Planner created at 10:58:32
Planner running
Message received from s76n78:1099/NegotiationAgent2
Strating conversation NegotiationAgent2_Planner_Conversation
```

Directed messages

```
11:0:1: Message received (evaluate) price=150, processor=PII-300, memory=16 MB, quantity=50, delivery=21, brand=IB
11:1:17: Message received (evaluate) brand=Compaq, price=150, memory=16 MB, quantity=100, delivery=21, processor=P
11:2:35: Message received (evaluate) brand=NCR, price=140, memory=16 MB, processor=PII-300, quantity=100, delivery
11:2:35: transmitting msg (reject) to rmi://s76n78:1099/NegotiationAgent2 price=150, brand=Compaq, delivery=21, qu
```

Conversations

```
11:0:1: pcl found expected recvd msg in the queue
11:0:1: NegotiationAgent2_Planner_Conversation Executing rule pcl
11:0:1: NegotiationAgent2_Planner_Conversation put in wait state
pcl executed
```

```
11:0:01: offer received price=150, processor=PII-300, memory=16 MB, quantity=50, delivery=21, brand=IBM
                    (payoff) 193
11:2:35: offer received price=150, brand=Compaq, delivery=21, quantity=100, memory=16 MB, processor=PII-300
                    (payoff) 252
11:2:35: offer received price=140, brand=NCR, delivery=28, quantity=100, memory=16 MB, processor=PII-300
                    (payoff) 234
11:2:35: transmitting msg (reject) to rmi://s76n78:1099/NegotiationAgent2
         price=150, brand=Compaq, delivery=21, quantity=100, memory=16 MB, processor=PII-300
11:2:35: transmitting msg (reject) to rmi://s76n78:1099/NegotiationAgent3
         price=150, brand=Compaq, delivery=21, quantity=100, memory=16 MB, processor=PII-300
11:5:33: offer received price=150, brand=IBM, delivery=21, quantity=50, memory=32 MB, processor=PII-300
                    (payoff) 220
11:5:33: offer received price=140, brand=NCR, delivery=28, quantity=100, memory=16 MB, processor=PII-333
                    (payoff) 294
--
11:8:41: offer received price=140, brand=IBM, delivery=21, quantity=50, memory=32 MB, processor=PII-333
                    (payoff) 280
11:8:41: offer received price=140, brand=Compaq, delivery=21, quantity=100, memory=16 MB, processor=PII-400
                    (payoff) 352
--
11:11:57: offer received price=135, brand=IBM, delivery=21, quantity=50, memory=32 MB, processor=PII-400
                    (payoff) 345
11:11:57: offer received price=130, brand=NCR, delivery=14, quantity=100, memory=32 MB, processor=PII-400
                    (payoff) 469
--
11:15:15: offer received price=135, brand=IBM, qunatity=50, delivery=21, memory=64 MB, processor=PII-400
                    (payoff) 335
11:15:15: offer received price=130, brand=Compaq, delivery=14, quantity=100, memory=64 MB, processor=PII-400
                    (payoff) 547
--
11:17:32: offer received price=135, brand=IBM, qunatity=50, delivery=21, memory=64 MB, processor=PII-400
                    (payoff) 335
11:17:32: offer received price=130, brand=NCR, delivery=14, quantity=100, memory=64 MB, processor=PII-450
                    (payoff) 609
--
11:19:46: offer received price=135, brand=IBM, qunatity=50, delivery=21, memory=64 MB, processor=PII-400
                    (payoff) 335
11:19:46: offer received price=125, brand=Compaq, delivery=14, quantity=100, memory=64 MB, processor=PII-400
                    (payoff) 572
11:19:46: transmitting msg (quit) to rmi://s76n78:1099/NegotiationAgent2 ,
         msg (accept) to rmi://s76n78:1099/NegotiationAgent3, msg (quit) to rmi://s76n78:1099/NegotiationAgent1
```

FIGURE 6.3. PROGRESS OF ONE NEGOTIATION SESSION FOR THE NEGOTIATION AGENTS.

6.1.1 Humans' Bargaining Behaviour

Overall the experiment was rated as interesting and the participants followed the rules and the utilities they were provided with. During the negotiation experiments' conduct, many persistent phenomena were realised which were confirmed by the players in the experiment. Of these observations:

- Although the players (the customers and the suppliers) were provided with the utility function for the role they were playing, sometimes it was realised that they were not clear about their own priorities. It was also obvious that due to the complex communication pattern that characterise the negotiation with many issues, opportunities were sometimes lost. Some players expressed their behaviour by stating that the utility function is "difficult to follow" or "sometimes seemed illogical when used". The thesis admits that the methods of multi-attribute analysis used do not easily model the various interactions among issues that sometimes exist in complex bargaining situations. For example, some interactions alter the preference of an issue under special specified assumptions.

- The player's emotions often interfere with the rational judgement based on the given utility values. For example, one of the players accepted an offer from Compaq while rejected similar offer (same payoff value) from NCR, and when he was asked for the reason behind his decision he replied "I think Compaq is more reliable brand".

- Logrolling (trade-off process) was the norm in almost all of the negotiations. For example, IBM supplier offered a lower price for a delayed delivery in return. Also, the customer offered higher price for a higher processor speed in return. However, negotiating parties sometimes misled others regarding their preferences and priorities, and because the utility function was a private information for each player, integrative bargaining opportunities were missed out. A bias towards "fair" solutions sometimes led negotiators to exhibit a compromise-bias. Parties tried to adjust the initial offer on each issue to suit both parties rather than explore trade-offs between issues that would generate better payoff for both parties.

- All of the participants playing the role of consumer were negotiating with one supplier at a time, then they proceeded to other suppliers. Sometimes, they closed a deal with one supplier without discovering what the others could have offered. The consumer terminated the negotiation with one supplier then proceeded to the other one. No one backtracked to re-open a terminated negotiation, instead they accepted the final offer of previously terminated negotiation (when they found that this was the best offer they received). Sometimes participants (playing the role of consumers) accepted a satisfactory offer without exhaustively trying to get a better one.

- Parties employed several tactics in their negotiation. Suppliers, sometimes, used "one-time take-it-or-leave-it offer" tactic to put a pressure on the customer. Whereas customers threatened to accept another offer if the supplier would not make further concessions.

6.2 Usability Results

There were two constructs' ratings to which the users responded (i.e. usefulness and ease of use). Table 6.1 presents the mean and standard deviation of both ratings. Histograms of the ratings for both constructs are shown in Figures 6.4 and 6.5. The overall ratings of the respondents were very encouraging and showed a general satisfaction of the prototype and that the possibility of a wide scale adoption for the system is excellent.

Construct	Mean	Standard deviation
Ease of use	5.90	0.989
Usefulness	5.89	0.566

TABLE 6.4. USABILITY TESTING FINDINGS.

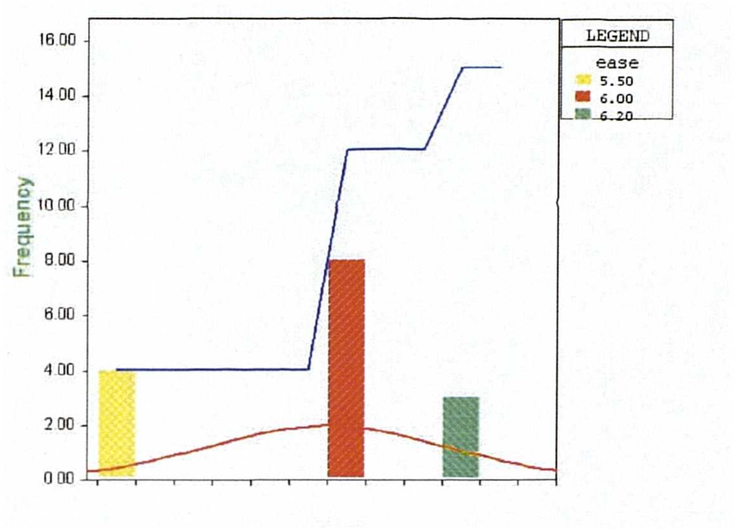


FIGURE 6.4. HISTOGRAM OF USERS' RATINGS FOR THE SYSTEM'S EASE OF USE.

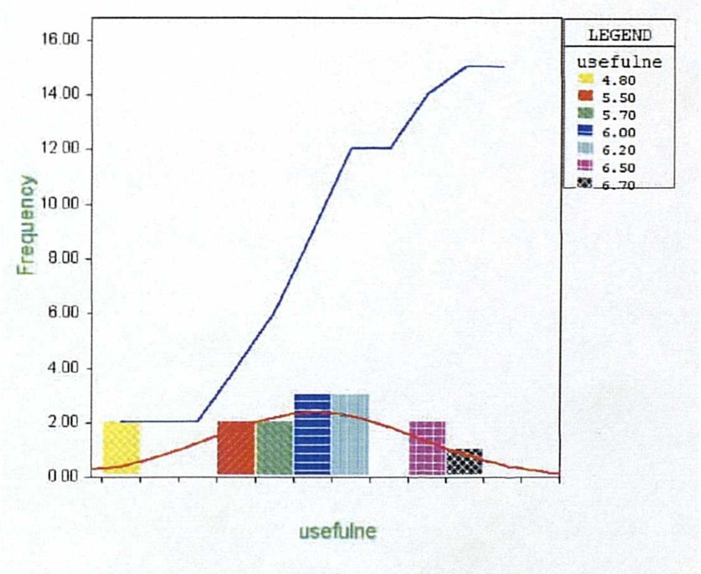


FIGURE 6.5. HISTOGRAM OF USERS' RATINGS FOR THE SYSTEM'S USEFULNESS

To provide a statistical significance for **HYPOTHESIS 3**, a t-test is performed on a single mean value for both of the ease of use and the usefulness. The null hypothesis to be tested is that the mean of each the ease of use and usefulness is 4 (the midpoint in our questionnaire scale 1-7).

For ease of use, the mean of the difference variable is 1.91. The standard error of the difference is about 0.069 leading to a t-statistic of 27.78 at a p-value of 0.0. For the usefulness, the mean of the difference variable is 1.89. The standard error of the difference is about 0.146 leading to a t-statistic of 12.9 at a p-value of 0.0. This shows that the overall respondents' ratings for the system's usability is significantly higher than average, supporting the hypothesis that the development of usable consumer-oriented EC system is achievable.

In addition to the TAM findings, recorded comments that the respondents have indicated include:

- One evaluator said that although he finds the prototype very promising and could revolutionise humans' buying behaviour, he would find it difficult to buy something without feeling or seeing it. He added that he would use the system for purchasing things such as books and computers but not furniture and perfumes. Suggestions such as showing pictures or videos or even using sophisticated virtual reality technologies for the products should be used in marketable EC system could provide some comfort for him to use the system. The socio-cultural and personal buying behaviour effects on Internet shopping behaviour is an out of scope research area.

- All respondents were questioning the usefulness of the system if the suppliers do not have a similar negotiation system at their end to conduct the negotiation. How to convince the suppliers to adopt this approach and implement compatible negotiation system is another field of research.

- Some respondents found that determining the user's utility function and preferences setting is a difficult task. One respondent raised a concern that the simple additive combination of preferences for individual attributes to construct the utility function (i.e. MAUT) is not precise in every case. This ignores certain fundamental characteristics of the effects of interactions between the attributes. Suggestions include presenting several packages for the user and asking him to rank those according to his preferences. The system should elicit the value function from these rankings. In Appendix B, the decision analysis techniques are reviewed in which the conjoint analysis method is discussed together with the advantages and disadvantages of adopting this technique.

6.2.1 System's Delegation Requirements

The experiments and their results raise a number of issues about the delegation of this type of system. Financial, trust and legal issues could impede the use of the system. Although certain benefits can be derived by consumers using EC system such as lower prices and search costs, it also adds new forms of consumer risk. The risk dimensions typically considered are economic risk, performance risk and personal risk (Simpson & Lakner 1993). Economic risk stems from the possibility of monetary loss associated with buying a product. Performance risk represents the consumers' perception that a product may fail to meet expectations. Personal risk relates to the possibility of harm to the consumer resulting from the shopping process.

As the consumer's trust in the system increases, the emphasis on the risk dimensions reduces. Trust is enhanced if one is using a system that have been demonstrated to be trustworthy in the past. Now that the proactive system has the ability to transact on behalf of the user, does the user trust the system to do the right thing? A positive answer to this question is an important requirement for delegating tasks to such system.

Second, legal implications of delegating tasks to software systems need to be addressed. Significant academic and commercial research in the AI field are currently dedicated towards addressing this issue (Oliver 1996). Those include researching questions such as whose responsibility it is if the system made wrong decision that could incur big loss such as monetary loss? Both the developer and the user of the system should be clear about their legal obligations and rights. This important area is left for others to research.

Another issue that deserves consideration is whether the attributes of the goods to be purchased by the system can be entirely defined. High technology products such as hardware can be easily defined through technical specifications. Whereas if the goods involve the sense of feel, smell or finely detailed inspection that the Internet can not provide are unlikely to be bought using the system.

CHAPTER 7

CONCLUSION

The thesis reinforces the potential benefits of adopting multi-agent system design to undertake some of the routinely tasks humans presently doing themselves. To conclude this work, the chapter starts by presenting a statement of contributions made during this study. Then, the implications of the research are presented. Then some of the requirements for a fieldable consumer-oriented EC system are discussed. The chapter concludes by outlining some issues in the thesis that can be expanded and areas for further research.

7.1 STATEMENT OF CONTRIBUTIONS

The individual elements of the contributions made by this work arise from different elements in the thesis; from the contextual information provided in Chapters 1 and 2, to the approach of the research methodology reported in Chapter 3, through the design and implementation work discussed in Chapters 3 and 4, and finally the empirical evaluation work reported in Chapters 5 and 6.

Following is a review of what is believed to be the four main contributions of this thesis.

Contribution 1: MAS paradigm as a solution strategy to automate the consumer decision making process.

In Chapter 1, a description of the consumer decision- making behaviour was given in which a simplified model that depicts this observed behaviour was presented. Then the challenges of automating the presented model were described.

In Chapter 3, the approach to automate the model and the solution strategy to overcome those challenges were described. A framework for MAS architecture as a viable solution was presented. The framework naturally impersonates the cognitive behaviour of humans by planning and sequencing the tasks of the decision making process. The MAS employs a central coordinator concept (planner agent) to plan and sequence those tasks. For each task in the process, the planner agent creates "sufficient" multiple cooperative specialised agents to fulfil the task's requirements. First, multiple search agents are created for finding and collecting relevant information. Second, multiple information extraction agents are created for filtering and extracting relevant data for information made available in the first task. Finally, multiple negotiation agents are created for negotiating best deal with potential suppliers. Then the benefits of the MAS approach were outlined.

In Chapter 4, a taxonomy of MAS architecture was presented that elaborates on features that are most relevant from the application perspective. First, the design requirements for open Internet-based MAS were studied and approaches to address those requirements were addressed. Second, the presented taxonomy elaborated on interaction and coordination requirements and presented the design strategy to ensure communication and coordination among the agents. Finally, the architectural design of all agents in the system was described (i.e. search, information extraction and negotiation agents) and the design rationale was elaborated.

Contribution 2: A novel negotiation approach.

In Chapter 2, a review of the current approaches to negotiation was given. A substantial amount of time and effort was dedicated to this review in which the thesis critically analysed those approaches and their practical applicability to provide insights for the design. During the course of the review, some approaches were eliminated (i.e. game-theoretic approaches), some inspiring ideas that can be used in the design that are used in some approaches were pointed out (i.e. decision-theoretic approaches) and the toy scenarios used when designing negotiation protocols that are used in other approaches (i.e. computational approaches) were criticised.

In Chapter 3, the negotiation context and the design of the negotiation strategy were described. The rationale of using multiple concurrent negotiation agents in the negotiation context was given.

In chapter 4, the internal design of the negotiation agents was addressed. An important feature of the proposed system is the incorporation of decision-theoretic constructs and negotiation analysis techniques. The heuristics used for preference elicitation and payoff evaluation were described. An appraisal for the negotiation approach was given.

In Chapter 6, the result of the conducted experiments (addressed in Chapter 5) to measure the effectiveness of the negotiation approach were reported. Reported results provided a strong support for the claim that the devised negotiation strategy can do better than human subjects in a given negotiation scenario. Results have also shown that the design of the system would scale up with the increase in the number of competing suppliers. The setup of the experiments to simulate real-life scenario gave an appreciation of the practical applicability of the thesis approach.

Contribution 3: Exploiting computational science to provide insights into the behaviour of the consumer decision-making model.

In Chapter 3, the legitimacy of the application for computational science study was justified. The rationale of choosing the computational science approach within which interplay of experiment, traditional theory and computational modelling is used to provide strong results was discussed. To comply with the perspective of computational science, the computational model, the algorithms used in modelling the process and the computer architecture were described.

In Chapter 5, the experimental design for testing and gaining insights into the computational model (here the thesis is concentrating on the proposed model for negotiation) were presented. The real-life applicability of the experimental scenario was discussed and argumentation for the experimental data selection criteria was provided.

In Chapter 6, the results of the experiment were reported. Then both qualitative and quantitative analysis of the results were provided. The observed behaviour of the computational model and the human subjects were elaborated.

Contribution 4: Usable Consumer-Oriented EC system.

In Chapter 3, the system's acceptability requirements were discussed from users' perspective. The thesis argued that the ease of use of the user interface and the system's reporting mechanisms together with the support for browsing and monitoring constitute the critical user interface design criteria for the system's acceptability and hence usability.

In Chapter 5, TAM was used to predict users' acceptance of the system. An experiment was conducted to test the usability of the system as surrogates for technology acceptance. In the experiment, the ease of use/ usefulness instrument was used for subjects to complete.

In Chapter 6, the results of the experiment were reported. Results suggested a strong evidence for the usability of the developed prototype system. Some remarks made by the evaluators to further enhance the system were also reported.

In addition to those main contributions, the work in the fields of Internet searching and wrapper generation together with the developmental work are considered as other minor contributions, as follows.

In the area of **Internet searching,** in Chapter 2 the approaches used in general IR methods were discussed, followed by a specific examination of the various techniques that are currently used in searching the Web.

In Chapter 3, the approach of Meta searching was advocated. Then the design for the search agent based on this approach was presented.

In Chapter 4, the specifics of the design of the search approach were presented. The rational behind the design decision was discussed. The thesis also claimed that the pre and post processing techniques employed in the search approach would result in a higher recall and precision rates.

In Chapter 5, an example experiment was conducted to act as indicator for the effectiveness of the search approach in improving the search results. Reported results of the experiment, though more thorough investigation is required before a statement can be made (hence classified as a minor contribution), justified the claim about the effectiveness of the techniques.

In the area of **wrapper generation,** Chapter 2 reviewed related works in the field and discussed the advantages of the structural regularities found in HTML pages.

In Chapter 3, the wrapper generation method was limited to only Web sources presenting their information in a tabular form due to the thesis and time limits. Then the requirements for the wrapper generation method were discussed.

In Chapter 4, we presented our wrapper generation method for information extraction. We also emphasised the requirement for recognisers to aid the process.

In Chapter 5, an example experiment was presented to act as indicator of the effectiveness of the wrapper generation method in extracting relevant data. Reported results are very encouraging and suggest that this approach is promising to build upon.

A final remark; out of the three years of this study, one year was dedicated to reviewing others' works, another year was dedicated for designing and developing the system and the last year was dedicated to the experimentation and completing the writing. This is mentioned to give an appreciation for the efforts dedicated for the developmental work.

7.2 Research Implications

The implications of this research are categorised into three areas: theoretical, empirical and technological.

- Theoretical: The thesis presents a framework for automating the consumer decision making process. The framework combines several novel approaches and modified existing techniques for the purpose. Theoretical aspects of MAS, information retrieval, information extraction and negotiation are studied. Although some of the approaches used are already in use, the novel idea is marrying the best aspects of those existing approaches for a viable system. The negotiation field received much attention in which a new strategy for multi-attribute negotiations is devised. Human negotiation behaviour was also studied to gain insights and provide groundwork for further research.

- Empirical: The wide scope of the thesis necessitated the limit of the focus on only two aspect of the system; the negotiation area and the usability of the system. Empirical testing shows the viability of the thesis approach to negotiation. Reported results strongly supported the thesis claim that well-thought design for negotiation agents could out-perform human subjects. The performance of the negotiation agents gives reason for hope to develop automated negotiation systems in the very near future. As a work in information system, the perceived usability of the system's prototype was tested. The result shows strong evidence for a possible wide scale adoption when such a system becomes available in the market.

- Technological: the results presented here are encouraging to create a working fieldable system for consumer-oriented EC. Following are two scenarios for possible implications of such system on (1) people and companies' purchasing behaviour and (2) the electronic markets.

Imagine the potential advantages of using such system in governmental sectors and big private companies for all of the purchasing. Currently, the purchasing departments in those bodies are managing the acquisition process as follows: (1) prepare the "Request For Bid (RFB)", (2) distribute the RFB to all interested suppliers, (3) collect bids, (4) compare offers and (5) grant contract to the supplier with best offer. A properly implemented purchasing system can be used to manage the whole process more efficiently and, therefore, replace the whole of the purchasing department. In addition to this, any potential of cheating in the traditional process would be eliminated (for example, one employee in the purchasing department reveals other bidders' offers for the benefit of one bidder).

The long-term effect of the increase in the number of such system for conducting transactions on the electronic markets could be numerous. It is known that today's electronic markets are closed. Examples of those e-markets are virtual superstores (for example, Amazon), online auctions, electronic storefronts to existing services (for example, Internet banking) and virtual trading marketplaces (for example, PartsNet). They are closed because they are run by one company and include specified business partners or loyal consumers. They portray a micro-representation of the true global potential, because their inventory is only a subset of what is available globally. However, the increase in use of intelligent purchasing systems for global comparison of available offerings would eventually force those closed electronic markets to enter into alliances to create global open markets for competencies. Secondly, those systems disconnect users from direct contact with brand and hence put the traditional brand value and loyalty in jeopardy. This would result in the dominance of what is called "digital brand" rather than traditional brand value. Third, businesses would think again about their strategy of putting firewalls to stop intelligent agents to do comparative shopping. The worst thing that could happen to a Web business is not to be able to respond to an agent's query that could result in a potential sale. Instead, those businesses would structure their information source in such a way that can be easily pulled by agents.

7.3 Requirements for Fieldable EC System

Significant growth is expected in terms of the use of the Internet for commerce. Electronic commerce promises to change the fundamental relationship between buyers and sellers. The task of developing effective commercial Web application should investigate its social acceptability and address the users' concerns. Next, the economical, technical, operational and organisational viability issues should be analysed.

One issue that is yet a concern for customers is that the Internet is not secure enough for electronic commerce, and they require a formalisation of network security policies and procedures to reflect their concerns. In electronic commerce, security and reliability refers to positive trust that is shown in the consistency and assurance between what a trading partner promises and actually does. Electronic commerce security, customers refer to, is not limited to only the confidentiality, authenticity, non-repudiation, availability of access control of the electronic transactions but more importantly the reliability of the trading parties involved in the electronic commerce. Issues such as: What is the legal status of a contract over the Internet? What body has legal jurisdiction over the contract? How payments are made and confirmed? What taxes and customs charges apply to a product? Some of these concerns are addressed in Section 1.1 while the others remain pending for further research.

Following are additional requirements for a fieldable EC system specific to this thesis implementation.

- The technical details of the system require carefully designed user interface that would meet the social acceptability requirements. Presenting a list of possible packages for the user and asking him to rank them in order of preference can be used for preference elicitation (see conjoint analysis for preference elicitation in appendix B). The system then calculates the preference function for the user, eliminating the requirement from the user to list and rank each attribute and their related options, which some users find difficult. This would also require a database of products information, which would contain information about the products' attributes and options and that could be accessed globally. The database would be consulted for any ontological mismatch and other unique name violations that could arise during negotiation.

- Searching would be better replaced by the use of mediator agents that can be queried for the address of potential suppliers and the availability of negotiation capabilities in their Web sites. Traditional search techniques can be still used for finding experts' advises and product rating services.

- Information extraction should be expanded to cover a wider variety of Web pages' structures and formats. A more effective information extraction technique needs to use natural language processing technologies to extract the attributes of the required information from HTML pages resulting from the searching phase.

- Because, currently, no supplier is offering automated negotiation, a practical implementation should consider negotiating with potential suppliers via e-mails. This would entail the requirements of robust speech act protocol and including natural language processing technologies. Offers' analysis should consider logrolling gestures.

- At the end of negotiation session, contract finalisations and payment mechanisms that would assure the user should be implemented.

- Learning mechanisms need to be implemented in which the whole process need to be recorded in a knowledge base for future references and for the application to be personalised, such as personal preferences and satisfaction records.

7.3 FURTHER RESEARCH

In this section areas for further research are outlined, based on what was learned from the investigations. There are many issues not examined in this thesis that need to be researched and resolved, where each one can be a huge research topic in itself. Some of these issues, mentioned earlier, include designing ontological databases, designing mediator agents and requirements for fieldable EC system.

Specific research ideas that are motivated by the limitations of the investigations reported in this work include:

- In the negotiation study, one important area of study is to extend the limited types of strategies the agent use. The availability of more than one strategy the agent can select from and learn to effectively choose the strategic structure and the complexity in any general negotiation setup, is an important area of investigation. Auction mechanisms, although effective in most cases, would do very little in cases of monopoly where no more than one supplier for a specific product can be negotiated with. Integrative bargaining will be a better strategic choice to follow.

- Negotiation is a rich area with many basics still not fully comprehended. The computational approach pursued in the negotiation testing facilitates examining the dynamics of the negotiation process in a great detail. Further research performing simulations of negotiations using particular strategies and various elements such as threats and bluffs would allow comparisons to be made with humans to understand how the software agents are similar and how they are different. The effect of noisy communications, time pressure and communication cost on the outcome and the path of the negotiation process is another area for further research.

- Using different preference elicitation techniques (for example the conjoint analysis) is another area for further research. The effect of using such techniques on the requirements imposed on end users and the computational complexities involved in creating preference structures should be addressed.

- In the MAS design, the effect of using different coordination mechanisms and interaction approaches on the overall performance of the system is an issue to be further examined.

- Testing criteria and performance measures for both the searching and information extraction techniques presented in Chapter 5 are areas for further research. The presented example experiments provided encouraging insights of the effectiveness of the developed techniques, however a thorough investigation is required.

- In the search approach, the result of adding additional analysis methods, such as automatic keyword expansion and the use of trigger pair model, could be an area for further research.

- Expanding on the wrapper generation approach to cover a wider variety of Web pages structures deserves further work. The presented example experiment suggests that the wrapper generation method is promising to build upon.

References

Adams, D., Nelson, R. & Todd, P. (1992) Perceived Usefulness, Ease of Use and Usage of Information Technology: A Replication. MIS Quarterly, 16(2). pp. 227-247.

Anson, R. & Jelassi, T. (1990) A Development Framework for Computer-Supported Conflict Resolution. European Journal of Operational Research, (46). pp. 181-199.

ARPA (1995) Proceedings of the 6th Message Understanding Conference. Morgan Kaufmann.

Ashish, N. & Knoblock, C. (1998) Semi Automatic Wrapper Generation for Internet Information Sources. Technical Report. University of Southern California, Marina Del Ray, CA.

Aumann, R. (1989) Perspectives on Bounded Rationality. IMSSS – Economics, Stanford University.

Austin, J. (ed.) (1962) How to do Things With Words. Oxford University Press.

Bazerman, M. (1990) Judgement in Managerial Decision Making. John Wiley, New York.

Beam, C., Segev, A. & Shanthikumar, G. (1996) Electronic Negotiation Through Internet-based Auctions. CITM Working Paper 96-WP-1019. Fisher

Center for Information Technology & Management, Walter A. Haas School of Business, University of California, Berkeley, CA.

Bear, J., Israel, D., Petit, J. & Martin, D. (1998) Using Information Extraction to Improve Document Retrieval. Proceedings of the 6th Text Retrieval Conference TREC-6.

Belkin, N., Kantor, P., Fox, E. & Shaw, J. (1995) Combining the Evidence of Multiple query Representations for Information Retrieval. Information Processing & Management, 31(3). pp. 431-448.

Berry, M., Dumais, S. & Shippy, A. (1995) A Case Study of Latent Semantic Indexing. Technical Report CS-95-271. Computer Science Department, University of Tennessee, Knoxville.

Bhargava, H., Kimbrough, S. & Krishnan, R. (1991) Unique Name Violations, A problem for Model Integration -or- You Say Tomato, I Say Tomahto. ORSA Journal on Computing, 3(2). pp. 107-120.

Binmore, K. & Dasgupta, P. (1987) The Economics of Bargaining. Basil Blackwell Inc., New York. ISBN 0-631-14254-1.

Bloch, M., Pigneur, Y. & Segev, A. (1996) On the Road of Electronic Commerce – a Business Value Framework, Gaining Competitive Advantage and Some Research Issues. Fisher Center for Information Technology and Management, University of California, Berkeley.

Bond, A. & Gasser, L. (1988) An Analysis of Problems and Research in DAI. In A. Bond & L. Gasser (eds.), Readings in Distributed Artificial Intelligence. Morgan Kaufmann. pp. 3-36.

Bradshaw, J., Dutfield, S., Benoit, P. & Woolley, J. (1997) KAoS: Towards An Industrial Strength Open Agent Architecture. In J. Bradshaw (ed.) Software Agents. Menlo Park, Calif, AAAI Press.

Brown, E., Callan, J. & Croft, B. (1994) Fast Incremental Indexing for Full-Text Information Retrieval. Proceedings of VLDB Conference. pp. 192-202.

Bui, T. (1994) Software Architectures for Negotiator Support: Co-op and Negotiator. Computer-Assisted Negotiation and Mediation Symposium. Harvard Law School, Cambridge.

Chatterjee, K. & Samuelson, W. (1983) Bargaining Under Incomplete Information. Operations Research, 31(5).

Chau, P. (1996) An Empirical Assessment of a Modified Technology Acceptance Model. Journal of Management Information Systems, 13(2). pp.185-204.

Chavez, A. & Maes, P. (1996) 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. pp. 75-90.

Chavez, A., Dreilinger, D., Guttman, R., & Maes, P. (1997) A Real Life Experiment in Creating an Agent Marketplace. Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97). London, UK. pp. 75-90.

Cohen, P. & Levesque, H. (1990) Intention is Choice with Commitment. Artificial Intelligence, 42. pp. 65-72.

Cohen, P. & Levesque, H. (1997) Communicative Actions for Artificial Agents. In J. Bradshaw (ed.), Software Agents. AAAI Press, Menlo Park, CA.

Corkill, D. & Lesser, V. (1983) The Use of Meta-Level Control for Coordination in a Distributed Problem Solving Network. Proceedings of the 8th IJCAI. Karlsruhe, Germany. pp. 748-756.

Cowie, J. & Lehnert, W. (1996) Information Extraction. Communications of the ACM, 39(1). pp. 90-91.

Crawford, V. & Sobel, J. (1982) Strategic Information Transmission. Econometrica, 50. pp. 1431-1452.

Cray, D. (1994) Simultaneity and Sequencing in a Strategic Decision. Proceedings of the XI MCDM Conference.

Davis, F. (1989) Perceived Usefulness, Ease of Use, and User Acceptance of Information Technology. MIS Quarterly, 13(3). pp. 319-339.

Davis, F., Bagozzi, R. & Warshaw, P. (1989) User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. Management Science, 35(8). pp. 982-1003.

Decker, K., Dufreee, E. & Lesser, V. (1989) Evaluating Research in Cooperative Distributed Problem Solving. In L. Gasser & M. Huhns (eds.), Distributed Artificial Intelligence, vol. II. Pitman. pp. 487-519.

Decker, K., Lesser, M., Prasad, N. & Wagner, T. (1995) MACRON: An Architecture for Multi-agent Cooperative Information Gathering. Technical Report. Department of Computer Science, University of Massachusetts, Amherst MA.

Decker, K., Sycara, K. & Williamson, M. (1997) Middle-agents for the Internet. Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97). Nygoya, Japan.

Doorenbos, R., Etzioni, O. & Weld, D. (1997) A Scalable Comparison-Shopping Agent for the World-Wide Web. Proceedings of the First International Conference on Autonomous Agents. Marina del Ray, CA.

Doszkocs, T., Reggia, J. & Lin, X. (1990) Connectionist Models and Information Retrieval. In M. Williams (ed.) Annual Review of Information Science and Technology (ARIST), 25. pp.209-260.

Dreilinger, D. (1996) Savvy Home Page. http://www.cs.colostate.edu/~dreiling/smartform.html.

Dreilinger, D. (1997) Integrating Hetrogeneous WWW Search Engines. ftp://132.239.54.5/savvy/report.ps.gz.

Dworman, G., Kimbrough, S. & Laing, J. (1996a) On Automated Discovery of Models Using Genetic Programming in Game-Theoretic Contexts. Journal of Management Information Systems, 12(3).

Dworman, G., Kimbrough, S. & Laing, J. (1996b) Bargaining by Artificial Agents in Two Coalition Games: A Study in Genetic Programming For Electronic Commerce. Proceedings of the AAAI Genetic Programming Conference. Stanford, CA.

Engel, J., Blackwell, R. & Kollat, D. (1978) Consumer Behaviour. Third Edition, The Dryden Press, Illinois. ISBN 0-03-089673-8.

Etzioni, O. & Weld, D. (1994) A Softbot-Based Interface to the Internet. CACM, 37(7). pp. 72-76.

External Interfaces Working Group, A.K.S.I. (1993) Specification of the KQML Agent Communication Language. Working paper.

Finin, T., Fritzson, R., McKay, D. & McEntire, R. (1994) KQML as an Agent Communication Language. Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94). ACM Press.

FIPA (1997a). FIPA 97 Specification, Reston draft (v1.0). http://www.cselt.stet.it/fipa/spec/fipa97reston.htm.

FIPA (1997b) Part 2: Agent Communication Language. Foundation for Intelligent Physical Agents, version 1.2.

FIPA (1997c) Part 1: Agent Management. Foundation for Intelligent Physical Agents, version 1.2.

Fisher, R. & Ury, W. (1981) Getting to Yes: Negotiating Agreements Without Giving In. Houghton-Mifflin, Boston, MA.

Foner, L. (1993) What is an Agent, Anyway? A Sociological Case Study. Agents Memo 93-01. MIT Media Lab, Cambridge, MA.

Foroughi, A. & Jelassi, T. (1990) NSS Solutions to Major Stumbling Blocks. INSEAD, France.

Fox, M. (1993) A Common-Sense Model of the Enterprise. Proceedings of Industrial Engineering Research Conference. pp. 178-194.

Fudenberg, D. & Tirole, J. (1983) Sequential Bargaining with Incomplete Information. Review Economic Studies, 50. pp. 221-247.

Gauch, S. & Futrelle, R. (1996) Experiments in the Automatic Word Class and Word Sense Identification for Information Retrieval. Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval.

Gauch, S. & Wang, G. *(1996) ProFusion: Intelligent Fusion from Multiple* Distributed Search Engines. WebNet '96: The First World Conference of the Web Society. San Francisco, CA.

General Magic (1998) Odyssey Web site. http://www.genmagic.com/technology/odyssey.html.

Genersereth, M. & Fikes, R. (1992) Knowledge Interchange Format, Version 3.0, Reference Manual. Technical Report Logic 92-1. Computer Science Department, Stanford University.

Genesereth, M. & Ketchpel, S. (1994) Software Agents. Communications of the ACM, 37(7). pp. 48-53.

Georgeff, M.(1984) A Theory of Action for Multi Agent Planning. Proceedings of the 4th National Conference on Artificial Intelligence. Austin, Texas. pp. 121-125.

Gordon, M. (1988) Probabilistic and Genetic Algorithms for Document Retrieval. Communications of the ACM, 31(10). pp.1208-1218.

Grant, T. (1992) Agents that Learn to Plan. DECUS Newsletter, AI, 3(29).

Green, P. & Wind Y. (1973) Multiattribute Decisions in Marketing: A Measurement Approach. The Dryden Press, Hinsdale, IL.

Gruber, T. (1993) A Translation Approach to Portable Ontologies. Knowledge Acquisition, 5(2). pp. 199-220.

Hammer, J., Brennig, M., garcia-Molina, H., Nesterov, S., Vassalos, V. & Yerneni, R. (1997) Template-based Wrappers in the Tsimmis System. Proceedings of the ACM SIGMOD International Conference on Management of Data. Tucson, AZ.

Harsanyi, J. (1956) Aproaches to the bargaining Problems Before and After the Theory of Games. Econometrica, 24. pp. 144-157.

Holland, J. & Miller, J. (1991) Artificial Adaptive Agents in Economic Theory. AEA Papers and Proceedings. pp. 365-370.

Hoch, S. (1994) Experts and Models in Combinations. In Blattberg, R., Glazer, R. & Little, J. (eds.) The Marketing Information Revolution. Harvard Business School Press. Boston, MA.

Howard, J. (1994) Buyer Behaviour in Marketing Strategy. Second Edition, Prentice Hall, New Jersey. ISBN 0-13-669011-4

IBM (1997) Aglets Workbench. http://www.trl.ibm.co.jp/aglets/.

Jacobs, P. & Rau, L. (1988) Natural Language Techniques for Intelligent Information Retrieval. In Yves Chiaramella (ed.), 11th International Conference on Research and Development in Information Retrieval. Presses Universitaires De Grenoble, France. pp. 85-99.

JATLite (1997) JATLite Overview. Stanford University. http://java.stanfod.edu/java_agent/htmlJATLiteOverview.html.

Jelassi, T. & Foroughi, A. (1989) Negotiation Support Systems: An Overview of Design Issues and Existing Software. The International Journal of Decision Support Systems, 5. pp. 167-181.

Jelassi, T., Kersten, G. & Zionts, S. (1990) An Introduction to Group Decision and Negotiation Support. INSEAD, France.

Jennings, N. (1995) Coordination Techniques for Distributed AI. In G. O'Hare and N. Jennings (eds.) Foundations of Distributed Artificial Intelligence. John Wiley. pp. 187-210.

Kalakota, R. (1996) Debunking Myths About Internet Commerce. Journal of Internet Banking and Commerce JIBC, 1(2).

Kearney, P., Sehmi, A. & Smith, R. (1994) Emergent Behaviour in a Multi-agent Economics Simulation. In A. Cohen (ed.) Proceedings of the 11th European Conference on Artificial Intelligence. John Wiley, London.

Keeney, R. & Raiffa, H. (1976) Decisions with Multiple Objectives: Preferences and Value Tradeoffs. John Wiley & sons.

Kersten, G. & Michalowski, W. (1989) A Cooperative Expert System for Negotiation With a Hostage-Taker. International Journal of Expert Systems, 2(3/4). pp. 357-376.

Kersten, G. & Noronha, S. (1997) Supporting International Negotiation with a WWW-Based System. Centre for Computer Assisted Management, Carleton University. http://interneg.carleton.ca/interneg/research/interneg/inspire_intro/.

Kersten, G. (1988) A Procedure for Negotiating Efficient and Non-Efficient Compromises. Decision Support Systems, 4. pp. 167-177.

Kersten, G. (1997) Support for Group Decisions and Negotiations. Carleton University. http://interneg.carleton.ca/interneg/research/misc/intro_gdn.html.

Kersten, G., Michalowski, W., Szpakowicz, S. & Koperczak, Z. (1991) Restructurable Representations of Negotiation. Management Science, 37(10). pp. 1269-1290.

Kimbrough, O. (1996) Remarks on Research Issues in Electronic Commerce: Krugman's Challenge and the Pooh- Pooh Theory. Technical Paper.

Knuth, D. (1973) The Art of Computer Programming. Sorting and Searching, vol. 3. Addison-Wesley, Reading, MA.

Kolodner, J. & Simpson, R. (1989) The MEDIATOR: Analysis of an Early Case-Based Problem Solver. Cognitive Science, I3. pp. 507-549.

Kowk, C. & Weld, D. (1996) Planning to Gather Information. Technical Report UW-CSE-96-01-04, Department of Computer Science & Engineering, University of Washington, WA.

Kreps, D. (1990) A Course in Microeconomic Theory. Princeton University Press.

Krulwich, B. (1995) Bargain Finder Agent Prototype. Technical Report. Anderson Consulting. http://bf.cstar.ac.com/bf/.

Kuokka, D. & Harada, L. (1996) Issues and Extensions for Information Matchmaking Protocols. International Journal of Cooperative Information Systems, 5(2/3). pp. 251-274.

Kushmerick, N., Weld, D. & Doorenbos, R. (1997) Wrapper Induction for Information Extraction. IJCAI-97.

Labrou, Y. & Finin, T. (1994) A Semantic Approach for KQML – a General Purpose Communication Language for Software Agents. Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94).

Lax, D. & Sebenius, J. (1986) The Manager as Negotiator. The Free Press, New York. ISBN 0-02-918770-2.

Lesk, M (1978) Some Applications of Inverted Indexes on the UNIX System. Bell Laboratories, Murray Hill, New Jersey.

Lesser, V. & Corkill, D. (1981) Functionally Accurate, Cooperative Distributive Systems. IEEE Transactions on Systems, Man and Cybernetics, 11(1). pp. 81-96.

Linhart, P., Rander, R. & Satterthwaite, M. (1992) Bargaining with Incomplete Information. Academic Press Inc., California. ISBN 0-12-451050-7.

MacKie-Mason, J. & Varian, H. (1993) Pricing the Internet. Proceedings of the Public Access to the Internet Conference. JFK School of Government.

Mathieson, K. (1991) Predicting User Intentions: Comparing the Technology Acceptance Model With the Theory of Planned Behavior. Information Systems Research, 2(3). pp. 173-91.

Matwin, S., Szapiro, T. & Haigh, K. (1991) Genetic Algorithms Approach to a Negotiation Support System. IEEE Transactions on Systems, Man, and Cybernetics, 21(1). pp.102-114.

Matwin, S., Szpakowicz, S., Koperczak, Z., Kersten, G. & Michalowski, W. (1989) Negoplan: An Expert System Shell for Negotiation Support. IEEE Expert, 4(4). pp. 50-62.

Moukas, A. (1996) Amalthaea: Information Discovery and Filtering Using a Multiagent Evolving Ecosystem. Technical report. MIT Media Laboratory, MA.

Nash, J. (1950) The Bargaining Problem. Econometrica, 18. pp. 155-162.

Neale, M. & Bazerman, M. (1991) Cognitive and Rationality in Negotiation. The Free Press, New York. ISBN 0-02-922515-9.

Northcraft, G., Brodt, S. & Neale, M. (1995) The Prospect of Negotiating with Non-linear Utilities: Why Some Concessions are More Equal than Others. Organizational Behaviour and Human Decision Processes, 61. pp. 67-76.

Nwana, H. (1996) Agents: An Overview. Knowledge Engineering Review, 13(2). pp. 205-244.

Oliver, J. (1996) A Machine Learning Approach to Automated Negotiation and Prospects of Electronic Commerce. URL: http://opim.wharton.upenn.edu/~oliver27/papers/jmis.ps.

Olson, D. (1995) Decision Aids for Selection Problems. Springer-Verlag, New York.

Osborne, M. & Rubinstein, A. (1994) A Course in Game Theory. The MIT Press, Cambridge, MA.

Parunak, H. (1996) Chapter 4- Applications of Distributed Artificial Intelligence in Industry. In G. O'Hare & N. Jennings (eds.) Foundations of Distributed Artificial Intelligence. John Willey & Sons. pp.139-163.

Patil, R., Fikes, R., Patel-Schneider, P., McKay, D., Finin, T., Gruber, T. & Neches, R. (1992) The ARPA *Knowledge Sharing Effort: Progress* Report. In B. Nebel, C. Rich & W. Swartout, (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR'92). Morgan Kaufmann, San Mateo, CA.

Perkowitz, M., Doorenbos, R., Etzioni, O. & Weld, D. (1997) Learning to Understand Information on the Internet: An Example-Based Approach. Journal of Intelligent Information Systems, 8(2). pp.133-153.

Pruitt, D. & Carnevale, P. (1993) Negotiation in Social Conflict. Open University Press, Buckingham. ISBN 0-335-09866-5.

Putnam, L. (1990) Refining Integrative and Distributive Bargaining: A Process Perspective. Research on Negotiation in Organizations, 2. pp.3-30.

Raiffa, H. (1982) The Art & Science of Negotiation. Harvard University Press, Cambridge, MA. ISBN 0-674-04812-1.

Ram, A. (1991) Interest-Based Information Filtering and Extraction in Natural Language Understanding Systems. Proceedings of the Bellcore Workshop on High Performance Information Filtering.

Rangaswamy, A. & Shell, G. (1997) Using Computers to Realize Joint gains in Negotiations: Toward an Electronic Bargaining Table. Management Science, 43(8). pp. 1147-1163.

Rasmusen, E. (1989) Games and Information. Basil Blackwell, New York.

Rocchio, J. (1971) Performance Indices for Document Retrieval. In G. Salton (ed.) The SMART Retrieval System – Experiments in Automatic Document Processing, Prentice Hall Inc, Englewood Cliffs, New Jersey.

Rosenschein, J. & Zlotkin, G. (1994) Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. The MIT Press, Cambridge, MA.

Roth, A. (1974) Axiomatic Models of Bargaining. Springer, Berlin.

Roth, A. (1985) Game-theoretic Models of Bargaining. Cambridge University Press, Cambridge. ISBN 0-521-26757-9.

Roth, M. & Schwartz, P.(1997) Don't Scrap it, Wrap it! A Wrapper Architecture for Legacy Data Sources. Proceedings of the 22nd VLDB Conference.

Rothkopf, M., Teisberg, T. & Kahn, E. (1990) Why are Vickrey Auctions Rare?. Journal of Political Economy, 98(1). pp. 94-109.

Rubin, J. & Brown, B. (1975) The Social Psychology of Bargaining and Negotiation. Academic Press, New York. ISBN 0-12-60125-4.

Rubinstein, A. (1982) A Perfect Equilibrium in a Bargaining Model. Econometrica, 50. pp. 97-109.

Saaty, T. (1980) The Analytic Hierarchy Process. McGraw Hill, New York.

Salton, G. (ed.) (1971) The SMART Retrieval System – Experiments in Automatic Document Processing, Prentice Hall Inc, Englewood Cliffs, New Jersey.

Sandholm, T. & Lesser, V. (1995) Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework. Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95). San Francisco, CA. pp. 328-335.

Sandholm, T. (1993) An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations. Proceedings of the 11th National Conference on Political Intelligence (AAAI-93). pp. 256-262.

Sebenius, J. (1989) International Negotiation: Problems and New Approaches. Working Paper. Kennedy School of Government, Cambridge, MA.

Selberg, E. & Etzioni, O. (1995) Meta-Service Search and Comparison Using the MetaCrawler. Proceedings of the 1995 World-Wide Web Conference.

Selton, R. (1981) A Noncooperative Model of Characteristic-Function Bargaining. In V. Boehm & H. Nachthamp (eds.), Essays in Game Theory and Mathematical Economics in Honour of Oscar Morgenstern. Morgan Kaufmann.

Shippy, A. (1995) An Analysis of the Latent Semantic Indexing Retrieval System. Master's thesis. The University of Knoxville, Tennessee.

Shklar, L., Shah, K.& Basu, C. (1995) Putting Legacy Data on the Web: A Repository Definition Language. Proceedings of the 3rd International WWW Conference.

Smith, R. & Davis, R. (1981) Frameworks for Cooperation in Distributed Problem Solving. IEEE Transactions on Systems, Man, and Cybernetics (SMC-11), 1. pp. 61-70.

Soderland, S. (1997a) Learning Text Analysis Rules for Domain-Specific Natural Language Processing. PhD dissertation. University of Massachusetts, MA.

Soderland, S. (1997b) Learning to Extract Text-based Information from the World Wide Web. Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97).

Stevenson, D. (1994) Science, Computational Science and Computer Science: At a Crossroads. Communications of the ACM, 37(12). pp. 85-96.

Strzalkowski, T., Carballo, J. & Marinescu, M. (1995) Natural Language Information Retrieval: TREC-3 Report. Proceedings of the Third Text Retrieval Conference TREC-3.

Sun, Y. & Weld, D. (1995) Automated Bargaining Agents. Working Paper. University of Washington.

Teich, J., Wallenius, H. & Wallenius, J. (1994) Advances in Negotiation Science. Transactions in Operational Research, 6. pp. 55-94.

Tomasic, A., Garcia-Molina, H. & Shoens, K. (1994) Incremental Updates of Inverted Lists for Text Document Retrieval. ACM SIGMOD. pp. 289-300.

Tsvetovatyy, B., & Gini, M. (1996) Towards a Virtual Marketplace. Proceedings of the First International Conference on the Practical Applications of Intelligent Agents and Multi-agent Technology.

Van-Rijsbergen, C. (1979) Information Retrieval. Butterworths, London, England.

Vickrey, W. (1961) Counter Speculation, Auctions, and Competitive Sealed Tenders. Journal of Finance, 16. pp. 8-37.

Virdhagriswaran, S., Osisek, D. & O'Connor, P. (1995) Standardising Agent Technology. ACM StandardView, 3(3). pp.96-101.

Waldspurger, C., Hogg, T., Huberman, B., Kephart, J. & Stornetta, W. (1992) Spawn: A Distributed Computational Economy. IEEE Transactions on Software Engineering, 18(2). pp. 103-117.

Walker, A. & Wooldridge, M. (1995) Understanding the Emergence of Conventions in Multi-Agent Systems. In V. Lesser (ed.) Proceedings of the First International Conference on Multi-Agent Systems. AAAI, Menlo Park, CA. pp.384-389.

Werkman, K. (1990) Knowledge-based Model of Negotiation Using Sharable Perspectives. Proceedings of the 10th International Workshop on DAI. Texas.

Woo, C. & Chang, M. (1990) An Approach to Facilitate the Automation of Semistructure and Recurring Negotiations in Organizations. Journal of Organizational Computing, 2(1). pp. 47-76.

Wooldridge, M. & Jennings, N. (1995) Intelligent Agents: Theory and Practice. Knowledge Engineering Review, 10(2). pp. 115-152.

Wurman, P., Wellman, M. & Walsh, W. (1998) The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents. Proceedings of the 2nd International Conference on Autonomous Agents.

Young, H. Peyton (ed.) (1991) Negotiation Analysis. The University of Michigan Press, Ann Arbor. ISBN 0-472-10251-6.

Zeng, D. & Sycara, K. (1996) Bayesian Learning in Negotiation. Working Notes of the AAAI-96 Stanford Symposium Series on Adaptation Co-evolution, and Learning in Multi-agent Systems.

APPENDIX A

MAS ARCHITECTURES

The main objective of this appendix is to give a basic overview of the terminology and current efforts by groups working on the standardisation of dynamic MAS architectures.

First, some of the terminology used for multi-agent systems are clarified and MAS architectures are elaborated, as follows:

- **Agent architectures** analyse agents as independent entities. Agents' architectures conceptualise agents as being made of perception, action and reasoning components.

- **Agent infrastructures** provide the regulations that agents follow to interact and coordinate with each other. Agent infrastructures deal with issues such as the interaction protocols, communication means and protocols and coordination mechanisms.

- **Agent frameworks** are programming tools to aid building agents. A number of agents' building tools have been developed recently. These tools vary from being platforms and frameworks for agent development, to being agent languages (in which agents are written) and agent communication languages (in which communication between agents are described). Examples of those tools include:

- **Odyssey** (General Magic 1998) is an implementation of mobile agents in Java. The agent system is implemented as a set of Java classes that provide support for building distributed mobile computing based on their conceptual model for MAS architecture for electronic commerce. Odyssey technology models a MAS architecture as an electronic marketplace that lets suppliers and customers find each other and conduct business transactions. This marketplace is modelled as a collection of network computers, modelled as a collection of places, which offer services to mobile agents. Those mobile agents occupy particular places and have the abilities to migrate, meet other agents and in doing so call each others procedures, and they can create connections to allow communication with another agent in different place.

- JATLite (JATLite 1997) provides a set of Java libraries that facilitates agent framework development. JATLite provides basic communication tools and templates for developing agents using KQML messaging format over TCP/IP connection. IT defines an Agent Name Server (ANS) for storing all the names and addresses of the existing agents. It also defines Agent Router (AR) to allow Java applets to exchange messages with any registered agent on the Internet. Though JATLite provides the essential communicational functionality required to build MAS application, it does not define a methodology to specify the social behaviour of agents.

- Aglets workbench (IBM 1997) is a visual environment for building Java-based network applications that use mobile agents. Aglets are mobile agents that may migrate and execute in specialised nodes in the network. The application's programming interface of the framework defines methods for aglets creation, dispatching, retraction, cloning, activation/ deactivation and disposing together with message handling procedures. Though aglets workbench is a very versatile tool for creating secure mobile agent-based systems, it does not deal with the issue of coordination in MAS applications.

- **MAS architectures** describes the system as individual problem solving agents pursuing high-level goals. Few efforts have been directed towards a definition of an acceptable MAS architecture. Recently, several independent industrial and research groups started to pursue the standardisation of the MAS technology. The General Magic's group, which their approach is described earlier, is one of the key players. Other prominent efforts include:

    - Knowledgeable Agent-oriented System (KAoS) (Bradshaw et al. 1997) model is described as "an open distributed architecture for software agents." This architecture describes agent implementations starting form the notion of generic agents to role-oriented agents such as mediators and matchmakers. The architecture pays special attention to the interaction and elaborates on the interactive dynamics of agent-to-agent communication by using conversation policies.

    - OMG (Virdhagriswaran 1995) model outlines the characteristics of an agent environment composed of agents (called components) and agencies (called places) as entities that collaborate using general patterns and policies of interaction. Under this model, agents are characterised by their capabilities (for example, planning, reasoning, inferencing and so on), type of interaction (for example, synchronous or asynchronous) and mobility (for example, static or movable with or without states). Agencies, support concurrent agent execution, security and agent mobility among others.

    - Foundation for Intelligent Physical Agents (FIPA) model is based on a "minimal framework for the management of agents in an open environment." This framework is described using a reference model that specifies the normative environment within which agents reside, and an agent platform that specifies an infrastructure for the deployment and interaction of agents. How to go about developing these policies is not detailed in their model. FIPA has made available a series of specifications to direct the development of MAS systems. Of particular importance are their agent communication language specification (FIPA 1997b) and agent management (FIPA 1997c).

APPENDIX B

CHOICE MAKING

Many selection problems involve multiple objectives (attributes). It is often true that no leading alternative will exist that is better than all alternatives in terms of all of these attributes. Basically, the choice maker is faced with the problem of trading off the achievement of one attribute against another. If there is no uncertainty in the problem (choice under certainty), i.e. if the multi-attribute consequence of each alternative is known, then the core of the issue is "How much achievement on attribute A is the choice maker willing to give up in order to improve the achievement on attribute B by a specific amount?" If there is uncertainty in the problem (choice under uncertainty), the trade-off issue remains but difficulties are intricate because the consequence of each of the alternatives is not clear. The trade-off issue often becomes a personal value question and it requires the subjective judgement of the choice maker. Different individuals may have very different value structures. A systematic trade-off analysis requires, from the choice maker, to formalise explicitly his value structure and use this to evaluate the available alternatives.

The choice maker is assumed to be aware of his alternatives and all information pertaining to the various levels of the objectives and is capable of evaluating them. Therefore the choice under uncertainty will not be considered, instead some of the techniques that aid choice under certainty are surveyed.

Following is a survey of three of the most widespread decision analysis techniques: utility theory, analytic hierarchy process and conjoint analysis. the decision to choose the utility theory is also justified.

B.1 UTILITY THEORY

Utility theory (Keeney & Raiffa 1976) assumes that an individual can choose among the alternatives available to him in such a way that the satisfaction derived from his choice is as large as possible. It also assumes that various levels of the objectives can be captured by an individual's utility function. In essence, an individual's utility function is a formal mathematical representation of his preference structure.

However, it is generally that an individual's preferences must satisfy certain conditions in order to be representable by a utility function. A utility function for an individual can be derived that expresses his preferences if he is rational and consistent in choosing among alternatives.

Rendering decision analysis for a multi-attribute problem entails assessing a choice maker's utility function in a vector indicating levels of the various attributes. This is called the Multi-Attribute Utility Theory (MAUT). This function has the property of "u(A) > u(B) if and only if A is preferred to B" which makes it useful in addressing the issue of trade-offs between attributes functions. To clarify, the alternative with the largest overall utility value is the most desirable one.

MAUT uses either a multiplicative or additive functional forms. The additive form is given by:

$$A_0 = \sum_{i=1}^{n} ViPi$$

Where:

A_o is the overall evaluation of the attractiveness of alternative o.

V_i is the relative importance of ith value importance of ith attribute.

P_i is the perceived instrumentality of alternative o with respect to ith value (payoff of the ith attribute of alternative o). This is a value between 0 for the worst case and 1 for best case.

n is number of attributes.

The overall value of this additive function ranges between 0 for worst alternative and 1 for best conceivable alternative.

However, this simple additive form assumes that there are no interaction effects between attributes. The assumption of attributes' independence, is not always true. For example, it might be the case that high attainment in all attributes is more valued than the addition of all individuals attributes utilities. In order to express the interdependencies among two attributes the utility function is extended to:

$$u(a,b) = c_1 + c_2\, u_a(a) + C_3\, u_b(b) + c_4 u_a(a) u_b(b).$$

As stated earlier, if the choice maker is rational and consistent in choosing the attributes that are important in ranking the alternatives, then specifying those utility functions for each attribute would be sufficient to rank any possible alternative without the need for any prior knowledge of them. For the thesis purpose, the available alternatives are not a prior knowledge, all what could be known is the evaluative criteria for those alternatives and the utility function for each of those criteria. Alternatives' generation is a dynamic process based on the negotiation's dynamics of the offers and counter-offers, therefore the attractiveness of those alternatives should be evaluated if and when they appear.

B.2 Analytic Hierarchy Process

The Analytical Hierarchy Process (AHP) (Saaty 1980) allows decision makers to model complex problem in a hierarchical structure showing the relationship of the goal, attributes, sub-attributes and alternatives.

The principle of comparative judgements is applied to construct pairwise comparisons of all combinations of elements in a cluster with respect to the parent of the cluster. Whereas AHP utilises ratio scales for the hierarchy, MAUT utilises an interval scale for the alternatives. AHP is based on the mathematical structure of consistent matrices and their associated right eigenvector's ability to generate true or approximate weights. The principle of hierachrical composition or synthesis is applied to multiply the local priorities of the elements in the cluster by the global priority of the parent element, producing global propitious throughout the hierarchy and then adding the global priorities of the lowest level elements (usually the alternatives).

To use AHP, available alternatives should be known in advanced. However, this is not practical in the thesis application since alternatives are not know prior to the negotiation process and that they change dynamically with the dynamic of the negotiation process.

B.3 Conjoint Analysis

Conjoint analysis (Green & Wind 1973) helps to examine the trade-offs for the choice maker and can be used to construct additive utility functions. In this technique, alternatives are described by "profiles". Each profile is a combination of one arbitrarily selected value for each of the attributes. First, coefficients, called utilities, among different values of attributes are estimated. Second, the relative importance among attributes and profile utilities are developed to quantitatively measure preferences. A utility is a numerical expression of the value that the choice maker gives for each option of each attribute. Interpreting utilities involves analysis of the gaps between the utility values. The absolute values of the utilities have no inherent meaning. The computation of the relative importance for each attribute depends on the relative range between the minimum and maximum value utilities within an attribute. The relative importance is given as a percentage value to reflect its weighted importance across all involved attributes.

The calculation of an attribute's relative importance is given by:

$$W_j = (\mathrm{Max}(u_{ij})-\mathrm{Min}(u_{ij})) / \Sigma (\mathrm{Max}(u_{ij})-\mathrm{Min}(u_{ij})) * 100$$

Where:

W_j is the relative importance of attribute j.

$\mathrm{Max}(u_{ij})$ is the maximum value utility of attribute j.

$\mathrm{Min}(u_{ij})$ is the minimum value utility of attribute j.

The profile utility is the overall utility of a profile calculated by summing all utilities of attribute levels defined in that profile.

To use this technique for the thesis purpose the following are needed:

1) Gather the important attributes involved in the choice making. This requires from the user to list the attributes that contribute to the overall attractiveness of an alternative.
2) Based on those attributes, the system need to arbitrary construct a set of possible alternatives and then ask the user to rank those alternative based on the attractiveness.

3) Based on those ranking, the system first estimates the part worth among different values of attributes and then calculates the relative importance of each attribute.

The process entails complicated calculations and estimation for the part-worth and relative importance of the attributes. It is also not clear how many examples or arbitrary set of alternative need to be constructed for the system to evaluate those values. However, this approach seems promising and could impose less requirements on the user. The attributes utility construction is done internally instead of eliciting it directly from the user as in the case of MAUT.

APPENDIX C

OBJECT ORIENTED DESIGN

Here, the high level object-oriented design for the prototype consumer-oriented Electronic Commerce application is given. The design is composed of four interconnected modules as follows.

- MAS infrastructure module to implement the basic abstract agent which all other agents in the system extend. The module implements basic communicational and interaction functionality required by MAS agents. High-level design is shown in Figure C.1.

- Search module to implement the functionality requirements for the search subsystem. The search agent extends the abstract agent in the MAS module. High-level design is shown in Figure C.2

- Information extraction module to implement the functionality requirements for wrapper generation. The information extraction agent extends the abstract agent in the MAS module. High-level design is shown in Figure C.3.

- Negotiation module to implement the functionality requirements for negotiation subsystem. The negotiation agent extends the abstract agent in the MAS module. High-level design is shown in Figure C.4.

In addition to those modules, a user interface module is designed to interface those modules and present a graphical interface for users interaction. Screens design is shown in the provided snapshots of the system in operation.

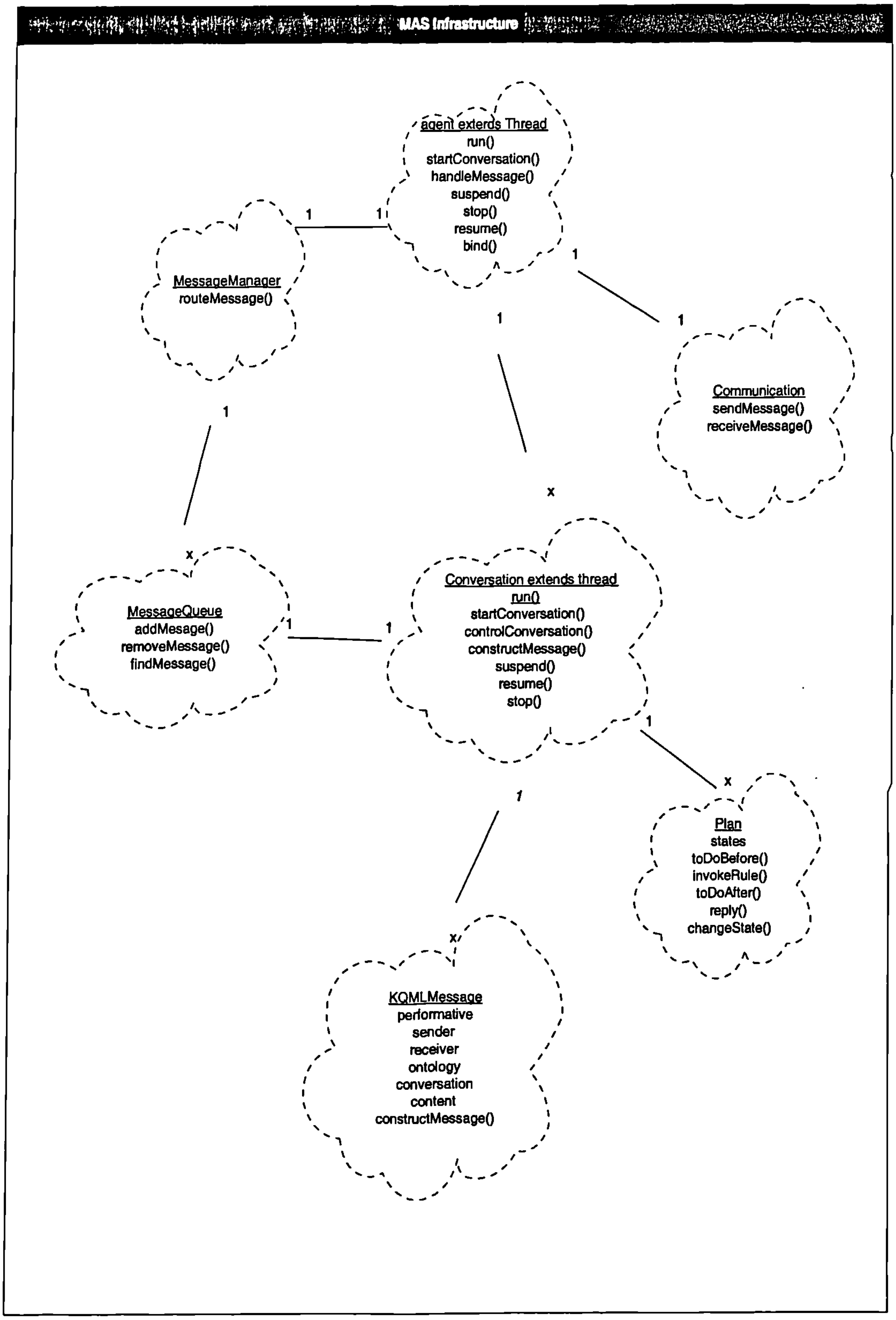


FIGURE C.1. OBJECT ORIENTED DESIGN OF MAS INFRASTRUCTURE.

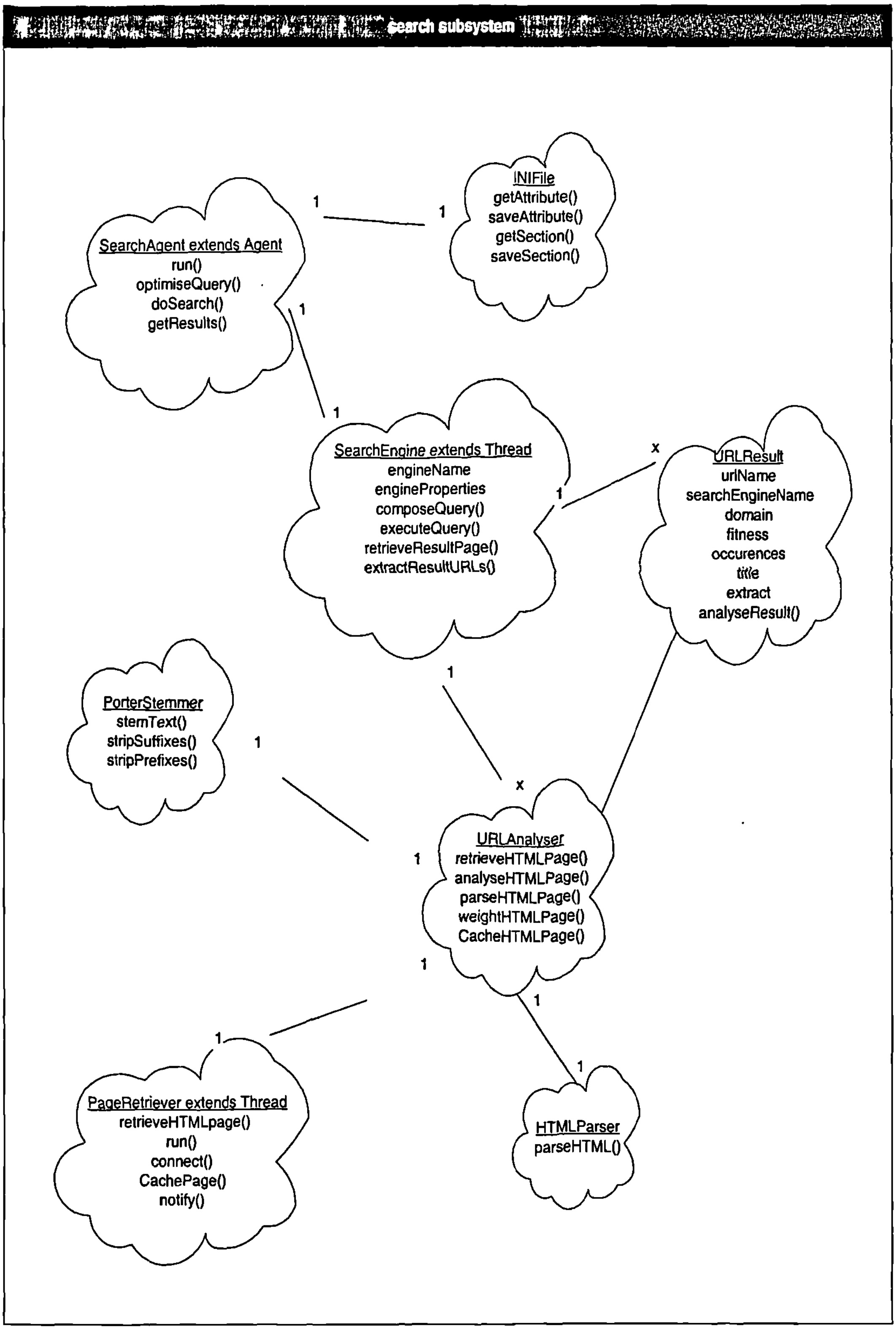


FIGURE C.2. OBJECT ORIENTED DESIGN OF THE SEARCH SUBSYSTEM.

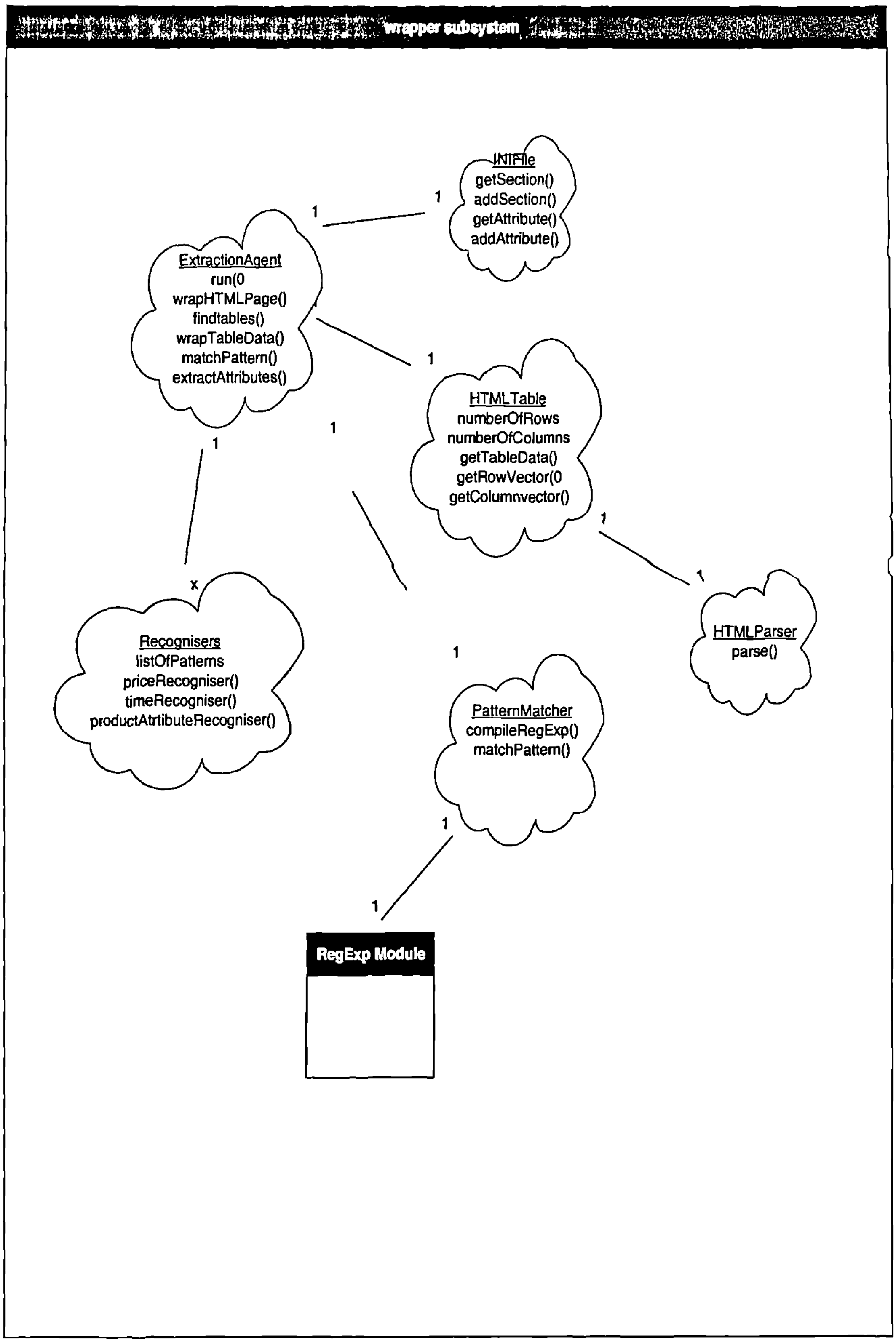


FIGURE C.3. OBJECT ORIENTED DESIGN OF THE INFORMATION EXTRACTION SUBSYSTEM.

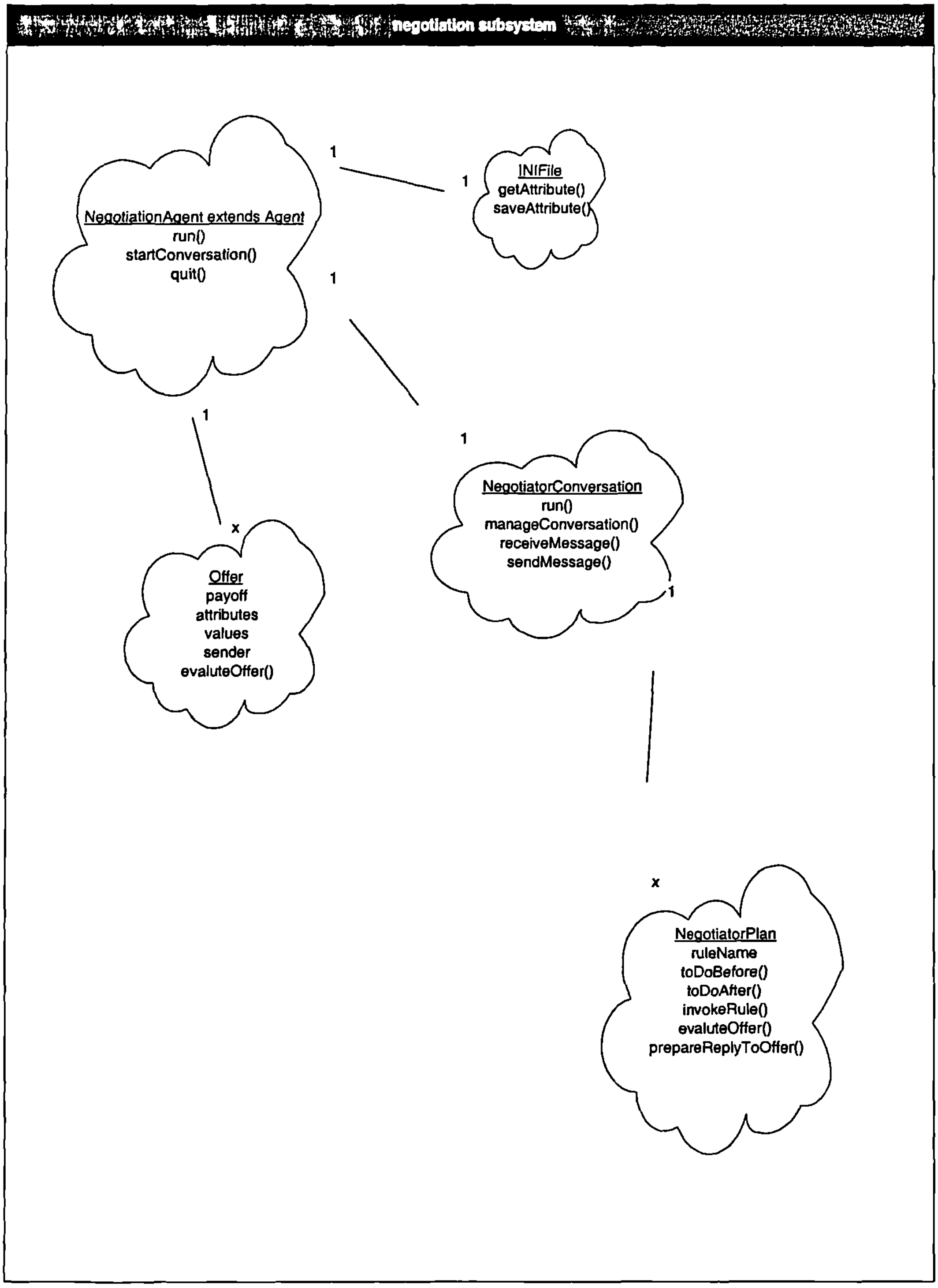


FIGURE C.4. OBJECT ORIENTED DESIGN OF THE NEGOTIATION SUBSYSTEM.

APPENDIX D

USEFULNESS / EASE OF USE QUESTIONNAIRE

In this appendix, the usefulness ease of use questionnaire that was used to measure the evaluators' acceptability of the prototype system is given. The questionnaire is adopted as provided by Davis (1989) in his TAM. TAM's attitude-intention behaviour relationship is used as a valid model to predict users' acceptance of the model. Usability is tested as surrogates for technology acceptance.

Please rate the usefulness and ease of use of the system

Try to respond to all the items.
For items that are not applicable, use: NA

Evaluator's name:
Title:
Email:

PERCEIVED USEFULNESS

	Unlikely 1	2 3 4 5 6 7 likely
1. Using the system in my job would enable me to accomplish tasks more quickly	1	2 3 4 5 6 7
2. Using the system would improve my job performance	1	2 3 4 5 6 7
3. Using the system in my job would increase my productivity	1	2 3 4 5 6 7
4. Using the system would enhance my effectiveness on the job	1	2 3 4 5 6 7
5. Using the system would make it easier to do my job	1	2 3 4 5 6 7
6. I would find the system useful in my job	1	2 3 4 5 6 7

PERCEIVED EASE OF USE

7. Learning to operate the system would be easy for me	1	2 3 4 5 6 7
8. I would find it easy to get the system to do what I want it to do	1	2 3 4 5 6 7
9. My interaction with the system would be clear and understandable.	1	2 3 4 5 6 7
10. I would find the system to be flexible to interact with.	1	2 3 4 5 6 7
11. It would be easy for me to become skillful at using the system	1	2 3 4 5 6 7
12. I would find the system easy to use	1	2 3 4 5 6 7

List the most negative aspect(s):

1.
2.
3.

List the most positive aspect(s):

1.
2.
3.

APPENDIX E

AN EXAMPLE SEARCH RESULTS FORM ALTAVISTA

In this appendix, the results retrieved by the system when querying Altavista for "desktop pc and ibm and ncr and compaq and processor and memory" are listed. The following twenty results are retrieved before the query time finishes. Here, the retrieved results are listed together with the comments on the relevancy of those results for the thesis purpose.

1) Compaq.com - VINES 6.0: Installing NCR SCSI Device Drivers
http://www.compaq.com/support/techpubs/customer_advisories/8651.html

comment : not related.

2) New Page 2
www.technology2ventures.com/newpage2.htm

comment: not related.

3) Empower PC Cards
www.empowergroup.com/pc/memory/memory.htm

comment: not related.

4) Ong Kok Leong - Hardware Links
www.tbnc.com/kokleong/hardware.htm

comment: good link.

5) DPE Products Page
www.dpe-northeast.com/products.htm

comment: not related.

6) NETIX International s.r.o.
www.netixraid.sk/symbios_cz.htm

comment: not related.

7) Dell computer
www.dell.com

comment: good link. Analysed by the system as a good link.

8) PriceTrac
www.pricetrac.com

comment: good link. Analysed by the system as a good link.

9) UCF Members
www.sco.com/ucf/members.html

comment: not related.

10) NAVEG/IHARD.HTM
www.rscin.com/NAVEG/IHARD.htm

comment: dead link.

11) The AMD K6 MMX
www.powerleap.com/amd_k6_mmx.htm

comment: could be useful.

12) MemProducts.htm
www.t2000llc.com/Memory/MemProducts.htm

comment: not related.

13) HARDWARE PLATFORM
www.karrox.com/hardware.htm

comment: not related.

14) Memory and CPU Pricing
www.needmemory.com/simmcpus.htm

comment: dead link.

15) DFMA & Office Products
www.design-iv.com/csoffice.htm

comment: not related.

16) Company links
paradise.caltech.edu/~fan/companies.html

comment: not related.

17) Contents
www.crm-forum.com/tosy/conten.htm

comment: not related.

18) Hardware Systems

www.loop.com/~superscooter/System.htm

comment: good link.

19) Claude LaBadie – Resume
www.openface.ca/~clabadie/Resume/Resume.html

comment: not related.

20) The AMD K6 MMX
www.upgradesint.com/amd_k6_mmx.htm

comment: duplicate.

APPENDIX F

AN EXAMPLE HTML SOURCE FOR WRAPPER GENERATION TESTING

In this appendix, the HTML source is listed for the example Web resource "Yahoo! Computer Shopping" that is used in the example to evaluate the effectiveness of the wrapper generation method. The HTML is shown exactly as supplied from the source, except added white spaces are added for clarity.

Following is the HTML source.

```
<html>
<head><title>Yahoo! Computer Shopping</title>
</head>
<body BGCOLOR="#FFFFFF">
<center>

<table border=0 width="100%" cellspacing=0 cellpadding=0>
        <tr><td valign=middle width="1%">
                <ahref="http://shopping.yahoo.com/computers/"><IMG S
                SRC="http://us.yimg.com/i/sh/shc41.gif
                " ALT="Yahoo Shopping" WIDTH="319" HEIGHT="40"
                BORDER="0"></A>
                </td><td valign=middle>
                <table border=0 cellspacing=0 cellpadding=0 width="100%">
                        <tr>
                                <td align=left valign=bottom>
                                <font face="arial" size="-1">&nbsp;</font></td>
                                <td align=right valign=bottom><font face=arial size="-1">
                                <A href="http://shopping.yahoo.com/">Shopping Home</A>
                                - <a href="http://www.yahoo.com/">Yahoo!</a> -
                                <ahref="http://help.yahoo.com/help/shop/">Help</A></font>
                                </td>
                        </tr>
                </table>
                <hr size=1>
                </td>
        </tr>
</table>
```

```
<p>

<table border=0 cellpadding=4 cellspacing=0 width="100%">
        <tr bgcolor="#dcdcdc">
                <td align=left><font face=arial><b>Welcome,&nbsp;guest</b></font></td>
                <td align=right valign=middle>
                <font face=arial size=-1>
                <A HREF="https://st0.yahoo.com/cgi-bin/wg-order?shopping">
                View Cart/Check Out</a> -
                 <A HREF="https://st0.yahoo.com/cgi-bin/wg-order-status?shopping">
                Order Status</A> -
                <ahref="http://edit.my.yahoo.com/config/shop_pref?.done=http://shopping.yah
                oo.com">Edit Acct</a> -
                <AHREF="http://edit.my.yahoo.com/config/login?.src=shp&.done=http://shop
                ping.yahoo.com">Sign In</A></font>
                </td>
        </tr>
</table>

<center>
<table border="0" cellpadding="4" cellspacing="0" width="100%">
        <tr>
                <td colspan=2 align=left bgcolor="#ffcc33">
                <font face=arial><b>Computers:</b></font>&nbsp;&nbsp;
                <font size=-1 face=arial>Found <b>19</b> products for "<b>
                intel pentium iii 550 ibm</b>"</font></td>
                <td valign=top bgcolor="#ffcc33" align=right  width="1%">
                <table border="0" cellpadding="2" cellspacing="0" width="100%"
                bgcolor="#ffe566">
                        <tr>
                                <td bgcolor="#ffe566" align=center nowrap>
                                <font size="-1" face=arial>&nbsp;
                                <AHREF="http://shopping.yahoo.com/computers/">
                                Computer Home</A> &nbsp;</font>
                                </td>
                        </tr>
                </table>
                </td>
        </tr>
</table>
</center>
<p>

<center>
</center>

<table border=0 width="100%" cellpadding=4 cellspacing=0>
        <tr>
                <td><font face=arial><b>Products</b> from <b>
                Yahoo! Computers Shopping:</b>&nbsp;&nbsp;1 - 10 of 19 matching 'intel
                pentium iii 550 ibm':</font></td>
        </tr>
</table><p>
<table border=0 width="100%" cellpadding=2 cellspacing=0>
        <tr bgcolor=eeeeee>
                <td align=right valign=top>
```

```
          <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
          height=5><br></td><td colspan=2>
          <font size=2 face=arial>
          <a href="http://shopping.yahoo.com/shop?d=m&id=844"><b><b>IBM</b>
           PC 300PL (with <b>Pentium</b> <b>III</b>
          processors)</b></a></font></td><td><font size=3><small>from
           <font color=brown>
          Yahoo! Computers Shopping</font></small></nobr></font>
          </td>
          <td align=right>
          <b>$1806</b>
          </td>
</tr>
<tr>
          <td></td>
          <td>
          <a href="http://shopping.yahoo.com/shop?d=m&id=844">
          <IMG SRC="http://shopping.yahoo.com/images/computer/t/844.gif"
          HEIGHT=50 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a>
          </td>
          <td colspan=3>
          <font size="-1"><b>Intel</b> <b>Pentium</b> <b>III</b> <b>550</b>
           MHz - 64 MB SDRAM - 6.4 GB hard disk - Windows NT</font>
          </td>
</tr>
<tr><td></td><td></td><td>
          <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
          height=5></td><td></td><td></td>
</tr>
<tr bgcolor=eeeeee>
          <td align=right valign=top>
          <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
          height=5><br>
          </td>
          <td colspan=2><font size=2 face=arial>
          <ahref="http://shopping.yahoo.com/shop?d=m&id=842"><b><b>IBM</b>
          PC 300PL (with <b>Pentium</b> <b>III</b> processors)</b></a></font>
          </td>
          <td><font size=3><small>from
          <font color=brown>
          Yahoo! Computers Shopping</font></small></nobr></font>
          </td>
          <td align=right>
          <b>$1839</b>
          </td>
</tr>
<tr>
          <td></td><td>
          <a href="http://shopping.yahoo.com/shop?d=m&id=842">
          <IMG SRC="http://shopping.yahoo.com/images/computer/t/842.gif"
          HEIGHT=50 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a></td>
          <td colspan=3><font size="-1"><b>Intel</b> <b>Pentium</b> <b>III</b>
          <b>550</b> MHz - 64 MB SDRAM - 6.4 GB hard disk - Windows 98
          </font></td>
</tr>
<tr>
```

```
        <td></td><td></td><td>
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
        height=5></td><td></td><td></td>
</tr>
<tr bgcolor=eeeeee>
        <td align=right valign=top>
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
        height=5><br></td><td colspan=2><font size=2 face=arial>
        <ahref="http://shopping.yahoo.com/shop?d=m&id=845"><b>
        <b>IBM</b> PC 300PL (with <b>Pentium</b> <b>III</b>
        processors)</b></a></font></td><td><font size=3><small>from <font
        color=brown>
        Yahoo! Computers Shopping</font></small></nobr></font></td>
        <td align=right><b>$2079</b></td>
</tr>
<tr>
        <td></td><td>
        <a href="http://shopping.yahoo.com/shop?d=m&id=845">
        <IMG SRC="http://shopping.yahoo.com/images/computer/t/845.gif"
        HEIGHT=50 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a></td>
        <td colspan=3><font size="-1"><b>Intel</b> <b>Pentium</b> <b>III</b>
        <b>550</b> MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM -
        Windows 98</font></td>
</tr>
<tr>
        <td></td><td></td><td>
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
        height=5></td><td></td><td></td>
</tr>
<tr bgcolor=eeeeee>
        <td align=right valign=top>
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
        height=5><br>
        </td>
        <td colspan=2><font size=2 face=arial>
        <ahref="http://shopping.yahoo.com/shop?d=m&id=848">
        <b><b>IBM</b> PC 300PL (with <b>Pentium</b> <b>III</b>
        processors)</b></a></font>
        </td>
        <td><font size=3>
        <small>from <font color=brown>
        Yahoo! Computers Shopping</font>
        </small></nobr>
        </font>
        </td>
        <td align=right><b>$2113</b></td>
</tr>
<tr>
        <td>
        </td>
        <td>
        <ahref="http://shopping.yahoo.com/shop?d=m&id=848">
        <IMG SRC="http://shopping.yahoo.com/images/computer/t/848.gif"
        HEIGHT=50 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a>
        </td>
        <td colspan=3><font size="-1">
```

```
          <b>Intel</b> <b>Pentium</b> <b>III</b> <b>550</b> MHz - 64 MB
          SDRAM - 13.5 GB hard disk - 40x CD-ROM - Windows 95</font>
          </td>
</tr>
<tr>
          <td></td><td></td><td>
          <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
          height=5></td><td></td><td></td>
</tr>
<tr bgcolor=eeeeee>
          <td align=right valign=top>
          <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
          height=5><br></td><td colspan=2><font size=2 face=arial>
          <ahref="http://shopping.yahoo.com/shop?d=m&id=846"><b><b>IBM</b>
           PC 300PL (with <b>Pentium</b> <b>III</b>
          processors)</b></a></font></td><td><font size=3><small>from <font
          color=brown>
           Yahoo! Computers Shopping</font></small></nobr></font></td>
           <td align=right><b>$2149</b></td>
</tr>
<tr>
          <td></td><td>
          <ahref="http://shopping.yahoo.com/shop?d=m&id=846">
          <IMG SRC="http://shopping.yahoo.com/images/computer/t/846.gif"
          HEIGHT=50 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a></td>
          <td colspan=3><font size="-1"><b>Intel</b> <b>Pentium</b> <b>III</b>
          <b>550</b> MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM -
          Windows NT</font></td>
</tr>
<tr>
          <td></td><td></td><td>
          <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
          height=5></td><td></td><td></td>
</tr>
<tr bgcolor=eeeeee>
          <td align=right valign=top>
          <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
          height=5><br></td><td colspan=2><font size=2 face=arial>
          <a href="http://shopping.yahoo.com/shop?d=m&id=847"><b><b>IBM</b>
          PC 300PL (with <b>Pentium</b> <b>III</b>
          processors)</b></a></font></td><td><font size=3><small>from <font
          color=brown>
          Yahoo! Computers Shopping</font></small></nobr></font></td><td
          align=right><b>$2164</b></td>
</tr>
<tr>
          <td></td><td>
          <a href="http://shopping.yahoo.com/shop?d=m&id=847">
          <IMG SRC="http://shopping.yahoo.com/images/computer/t/847.gif"
          HEIGHT=50 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a></td>
          <td colspan=3><font size="-1"><b>Intel</b> <b>Pentium</b> <b>III</b>
          <b>550</b> MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM -
          Windows NT</font></td>
</tr>
<tr>
          <td></td><td></td><td>
```

```
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
        height=5></td><td></td><td></td>
</tr>
<tr bgcolor=eeeeee>
        <td align=right valign=top>
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
        height=5><br></td><td colspan=2><font size=2 face=arial>
        <a href="http://shopping.yahoo.com/shop?d=m&id=879"><b><b>IBM</b>
         PC 300PL (with <b>Pentium</b> <b>III</b>
        processors)</b></a></font></td><td><font size=3><small>from <font
        color=brown>
         Yahoo! Computers Shopping</font></small></nobr></font></td><td
        align=right><b>$2224</b></td>
</tr>
<tr>
        <td></td><td><a href="http://shopping.yahoo.com/shop?d=m&id=879">
        <IMG SRC="http://shopping.yahoo.com/images/computer/t/879.gif"
        HEIGHT=50 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a></td>
        <td colspan=3><font size="-1"><b>Intel</b> <b>Pentium</b> <b>III</b>
        <b>550</b> MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM -
        Windows NT</font></td>
</tr>
<tr>
        <td></td><td></td><td>
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
        height=5></td><td></td><td></td>
</tr>
<tr bgcolor=eeeeee>
        <td align=right valign=top>
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
        height=5><br></td><td colspan=2><font size=2 face=arial>
        <a href="http://shopping.yahoo.com/shop?d=m&id=868"><b><b>IBM</b>
         E Pro 2D</b></a></font></td><td><font size=3><small>from <font
        color=brown>
         Yahoo! Computers Shopping</font></small></nobr></font></td>
         <td align=right><b>$2791</b></td>
</tr>
<tr>
        <td></td><td><a href="http://shopping.yahoo.com/shop?d=m&id=868">
        <IMG SRC="http://shopping.yahoo.com/images/computer/t/868.gif"
        HEIGHT=50 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a></td>
        <td colspan=3><font size="-1"><b>Intel</b> <b>Pentium</b> <b>III</b>
        <b>550</b> MHz - 128 MB SDRAM ECC - 13.5 GB hard disk - 40x CD-
        ROM - Windows NT</font></td>
</tr>
<tr>
        <td></td><td></td><td>
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
        height=5></td><td></td><td></td>
</tr>
<tr bgcolor=eeeeee>
        <td align=right valign=top>
        <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
        height=5><br></td><td colspan=2><font size=2 face=arial>
        <a href="http://shopping.yahoo.com/shop?d=m&id=905"><b><b>IBM</b>
```

```
                 M Pro 2D</b></a></font></td><td><font size=3><small>from <font
                color=brown>
                 Yahoo! Computers Shopping</font></small></nobr></font></td>
                 <td align=right><b>$3184</b></td>
        </tr>
        <tr>
                <td></td><td>
                <a href="http://shopping.yahoo.com/shop?d=m&id=905">
                <IMG SRC="http://shopping.yahoo.com/images/computer/t/905.gif"
                HEIGHT=40 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a></td>
                <td colspan=3><font size="-1"><b>Intel</b> <b>Pentium</b> <b>III</b>
                <b>550</b> MHz - 128 MB SDRAM ECC - 13.5 GB hard disk - 40x CD-
                ROM - Windows NT</font></td>
        </tr>
        <tr>
                <td></td><td></td><td>
                <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
                height=5></td><td></td><td></td>
        </tr>
        <tr bgcolor=eeeeee>
                <td align=right valign=top>
                <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=1
                height=5><br></td><td colspan=2><font size=2 face=arial>
                <a href="http://shopping.yahoo.com/shop?d=m&id=869"><b><b>IBM</b>
                 E Pro 3D</b></a></font></td><td><font size=3><small>from <font
                color=brown>
                 Yahoo! Computers Shopping</font></small></nobr></font></td>
                 <td align=right><b>$3317</b></td>
        </tr>
        <tr>
                <td></td><td><a href="http://shopping.yahoo.com/shop?d=m&id=869">
                <IMG SRC="http://shopping.yahoo.com/images/computer/t/869.gif"
                HEIGHT=50 WIDTH=50 BORDER=0 HSPACE=0 VSPACE=0></a></td>
                <td colspan=3><font size="-1"><b>Intel</b> <b>Pentium</b> <b>III</b>
                <b>550</b> MHz - 128 MB SDRAM ECC - 13.5 GB hard disk - 40x CD-
                ROM - Windows NT</font></td>
        </tr>
        <tr>
                <td></td><td></td><td>
                <img src=http://query.store.yahoo.com/Img/trans_1x1.gif border=0 width=200
                height=5></td><td></td><td></td>
        </tr>
</table>

<center>
</center>


<p>
<table border=0 cellpadding=2 cellspacing=0 width="100%">
        <tr>
                <td bgcolor="#dcdcdc">
                <b><font face=arial>Search Yahoo! Shopping Again</font></b>
                </td>
        </tr>
        <tr>
```

```
                <td valign=top>
                <center>
                <form method=get action="/search">
                <input size="24" name="p" value="intel pentium iii 550 ibm">&nbsp;
                <Input Type="Submit" Value="Search">&nbsp;&nbsp;
                <select name=P>
                <option selected value=all>All of Y! Shopping
                <option value=apparel>Apparel
                <option value=books>Books
                <option value=computer>Computers
                <option value=electronics>Electronics
                <option value=gift>Flowers & Gifts
                <option value=food>Food & Drink
                <option value=health>Health & Beauty
                <option value=garden>Home & Garden
                <option value=video>Movies & Video
                <option value=music>Music
                <option value=office>Office
                <option value=sports>Sports & Fitness
                <option value=toy>Toys & Games
                <option value=travel>Travel
                </select>
                <br><br><Br>
                </form>
                </center>
                </td>
        </tr>
</table>


<br>
<center><font face=arial size="-2"><hr size=1 width=620>
Copyright &copy; 1994-1999 <a href="http://www.yahoo.com">Yahoo! Inc.</a>
All rights reserved.
<a href="http://www.yahoo.com/info/privacy/">Privacy Policy</a> -
<a href="http://shopping.yahoo.com/info/security.html">
Security in Cyberspace</a>.<br>
Questions, comments, suggestions? Send us
<ahref="http://add.yahoo.com/fast/help/shop/cgi_feedback">feedback</a>
<p></font></center>
</body></html>
```

APPENDIX G

SYSTEM'S RESULTS OF THE WRAPPER EXAMPLE EXPERIMENT

In this appendix the system's progress is listed together with the results for the example experiment to test the effectiveness of the wrapper generation approach. The HTML source for which the wrapper generated is given in Appendix F. Following is the progress report as given by the system.

```
starting information extraction agent....
searching for tables in web page..
8 tables found..
table 1:
(1,1)null
(1,2)table 2
table 2:
(1,1)null
(1,2)Shopping Home - Yahoo! - Help
table 3:
(1,1)Welcome, guest View Cart/Check Out - Order Status - Edit Acct - Sign In
table 4:
(1,1)Computers:Found 19 products for "intel pentium iii 550 ibm"
(1,2)table 5
table 5:
(1,1)Computer Home
table 6:
(1,1)Products from Yahoo! Computers Shopping:  1 - 10 of 19 matching 'intel pentium iii 550
ibm':
table 7:
(1,1)null
(1,2)IBM PC 300PL (with Pentium III processors)
(1,3)from Yahoo! Computers Shopping
(1,4)$1806
(2,1)null
(2,2)null
(2,3)Intel Pentium III 550 MHz - 64 MB SDRAM - 6.4 GB hard disk - Windows NT
(3,1)null
(3,2)null
(3,3)null
(3,4)null
(3,5)null
(4,1) null
(4,2)IBM PC 300PL (with Pentium III processors)
```

```
(4,3)from Yahoo! Computers Shopping
(4,4)$1839
(5,1)null
(5,2)null
(5,3)Intel Pentium III 550 MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM -
Windows 98
(6,1)null
(6,2)null
(6,3)null
(6,4)null
(6,5)null
(7,1)null
(7,2)IBM PC 300PL (with Pentium III processors)
(7,3)from Yahoo! Computers Shopping
(7,4)$2079
(8,1)null
(8,2)null
(8,3)Intel Pentium III 550 MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM -
Windows 95
(9,1)null
(9,2)null
(9,3)null
(9,4)null
(9,5)null
(10,1)null
(10,2)IBM PC 300PL (with Pentium III processors)
(10,3)from Yahoo! Computers Shopping
(10,4)$2149
(11,1)null
(11,2)null
(11,3)Intel Pentium III 550 MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM -
Windows NT
(12,1)null
(12,2)null
(12,3)null
(12,4)null
(12,5)null
(13,1)null
(13,2)IBM PC 300PL (with Pentium III processors)
(13,3)from Yahoo! Computers Shopping
(13,4)$2164
(14,1)null
(14,2)null
(14,3)Intel Pentium III 550 MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM -
Windows NT
(15,1)null
(15,2)null
(15,3)null
(15,4)null
(15,5)null
(16,1)null
(16,2)BM PC 300PL (with Pentium III processors)
(16,3)from Yahoo! Computers Shopping
(16,4)$2224
(17,1)null
(17,2)null
```

```
(17,3)Intel Pentium III 550 MHz - 64 MB SDRAM - 13.5 GB hard disk - 40x CD-ROM - Windows NT
(18,1)null
(18,2)null
(18,3)null
(18,4)null
(18,5)null
(19,1)null
(19,2)IBM E Pro 2D
(19,3)from Yahoo! Computers Shopping
(19,4)$2791
(20,1)null
(20,2)null
(20,3)Intel Pentium III 550 MHz - 128 MB SDRAM ECC - 13.5 GB hard disk - 40x CD-ROM - Windows NT
(21,1)null
(21,2)null
(21,3)null
(21,4)null
(21,5)null
(22,1)null
(22,2)IBM M Pro 2D
(22,3)from Yahoo! Computers Shopping
(22,4)$3184
(23,1)null
(23,2)null
(23,3)Intel Pentium III 550 MHz - 128 MB SDRAM ECC - 13.5 GB hard disk - 40x CD-ROM - Windows NT
(24,1)null
(24,2)null
(24,3)null
(24,4)null
(24,5)null
(25,1)null
(25,2)IBM E Pro 3D
(25,3)from Yahoo! Computers Shopping
(25,4)$3317
(26,1)null
(26,2)null
(26,3)Intel Pentium III 550 MHz - 128 MB SDRAM ECC - 13.5 GB hard disk - 40x CD-ROM - Windows NT
(27,1)null
(27,2)null
(27,3)null
(27,4)null
(27,5)null
table 8:
(1,1)Search Yahoo! Shopping Again
(2,1)<Form method>
```

APPENDIX H

JAVA SOURCE CODE FOR APPLICATION PROGRAM

In this appendix, the core Java source code for the developed Internet-based Consumer oriented EC prototype application is listed. Because the program is very lengthy (more than 3000 lines of code), only the core of the algorithmic parts of the code is given whereas the user interface classes are omitted. These are provided so that the reported experiments can be reproduced and it should be clear that the program should not be copied or reproduced without a prior consent from the author.

```
package mt.mas;

import java.lang.*;
import java.io.*;
import java.util.*;
import java.rmi.*;
import java.rmi.server.*;
import java.net.*;
import java.rmi.RMISecurityManager;
import com.sun.java.swing.*;
import mt.mercy.*;


// package MAS

/* abstract agent class in MAS
* @by MT */
public abstract class Agent extends Thread
{
   //name of the agent
   public String name;
   //Host Name
   public static String localHost = null;
   //agent's group
   public String group = null;
   //the port on which the agent listens for commuication
   public final static int PORT = 1099;
   //communication object of agent
   public Communication communication = null;
   //list of conversations the agent is engaged in
   public Vector convList = new Vector();
  public static int counter;
   public static String registryHost = null;

   //text area to display agent's messages
   public  JTextArea infoTA = null;
   public JTextArea convTA = null;
    public JTextArea commTA = null;
   //messages manager/ router for the agent
   public MsgManager msgManager = null;
   // state of the agent (running/stopped)
   public boolean status = true;


   //constructor
   public Agent (String name,String group, String registry)
   {
```

```
   counter++;
   this.group = group;
   this.name = name;
   if (registry != null)
   registryHost = registry;
   initialize();
 }

void initialize()
{
  setLocalHost();
  if ( registryHost == null)
  registryHost = getLocalHost();
  msgManager = new MsgManager(this);
  bindComm();
}

//method to set the local host name of the machine on which the agent resides
private void setLocalHost()
{
   //Utility.print(counter+" Agent.setLocalHost");
   InetAddress localAdd = null;
   try{
      localAdd = InetAddress.getLocalHost();
      localHost = localAdd.getHostName();
      //Utility.print("//"+getLocalHost()+":"+PORT+"/"+name);

   }catch (Exception e){
      Utility.print ("unknown host: "+e.getMessage());
      localHost = "localhost";
   }
}

//procedure to bind remote communication objects to bootstrap registry
private void bindComm()
{

   try{
      communication = new Communication(this);
      //Naming.rebind("//"+getRegistryHost()+":"+PORT+"/"+name, communication);
      System.out.println("1");
      showBindings();
      Naming.bind("//"+getRegistryHost()+":"+PORT+"/"+name, communication);
      Utility.print("//"+getRegistryHost()+":"+PORT+"/"+name+
         " bound in Registry ");

     System.out.println("2");
```

```
    showBindings();

  }catch(RemoteException e){
     Utility.print("RemoteException occured: "+e.getMessage());
     StringWriter sw= new StringWriter();
     e.printStackTrace(new PrintWriter(sw));
     Utility.print(sw.toString());
  }catch(MalformedURLException ex){
     Utility.print("MalformedURLException occured: "+ex.getMessage());
  }catch(Exception e){
      Utility.print("Exception: "+e.toString());
      //e.printStackTrace(System.out);
  }
}


//method to add conversation to the agent's conversation list
public void addToConvList (Conversation conv)
{
   convList.addElement(conv);
   infoTA.append(conv.getConvName()+
        " is added to the conversation list \n");
}

//method to remove conversation from the agent's conversation list
public void removeFromConvList(Conversation conv)
{
   convList.removeElement(conv);
   infoTA.append(conv.getConvName()+
        " is removed from the conversation list \n");
}

//run method of the agent thread
public void run()
{
   infoTA.append(name+" running \n");
   while (status){
       yield(); }
}

//method to manage received messages by the agent
public void handleMsg (Message msg)
{
   infoTA.append("Message received from "+msg.getSender()+"\n");
   msg.setRxTime();
   msgManager.handleMsg(msg);
}
```

```
//method to list all remote objects bindings registered with this machine's registery
public String[] getBindings(){
   try{
      return Naming.list("//"+registryHost+":"+PORT+"/");
   }catch(Exception e)
   { Utility.print("error: "+ e); }
   return null;
}

 /* This method shows the list of all the remote objects registered
 * with this machines bootstrap registry service
 */
public static void showBindings()
{
 String url = "rmi://"+registryHost+":"+PORT+"/";
 try
 {
               String[] bindings = Naming.list("//"+registryHost+":"+PORT+"/");
               for (int i=0;i<bindings.length; i++)
                 System.out.println(bindings[i]);
 }
 catch(Exception e){
               System.out.println("Error: " + e);
 }
}

 //this method must be called when extending this class to set agentInfoArea
 public void setInfoTA(JTextArea textArea){ infoTA = textArea;}
 public void setConvTA(JTextArea convT){ convTA = convT;}
 public void setCommTA(JTextArea commT){ commTA = commT;}
 //accessor methods
 public String getAgentName() { return name; }
 public String getAgentFullName(){ return getLocalHost()+":"+PORT+"/"+name;}
 public String getLocalHost() { return localHost; }
 public String getRegistryHost() { return registryHost; }
 public Vector getConvList() { return convList; }
 public JTextArea getInfoTA() { return infoTA; }
 public JTextArea getConvTA() {return convTA;}
 public JTextArea getCommTA() {return commTA;}
 public int getServerPort() { return PORT; }
 //public String getGroupAddr() { return groupAddr;}

 //stub method converse to be implemented as needed
 //start conversations and register them in listOfconvs
 public abstract void startConversation(Object obj);
}
```

```
/* Communication class uses RMI and implements CommInterface remote interface.
// Each agent creates it's own remote Communication object, which resides on the server
// Other agents call the remote method rxMsg() to send message to owner agent of this
// communication object. */


public class Communication extends UnicastRemoteObject implements CommInterface
{
  private Agent owner = null;
  private JTextArea infoTA = null;

  public Communication (Agent parent)throws RemoteException
  {
    super();
    owner = parent;
    infoTA= parent.getInfoTA();
    Utility.print("communication created for "+owner.getAgentName());
  }

  public synchronized void rxMsg(Message msg) throws RemoteException
  {

   //correct this commented line first
   //infoTA.append("Received message type "+
   // msg.getType()+" from "+msg.getSender()+"\n");
   owner.handleMsg(msg);

  }

  public void txMsg(Message msg)
  {
    String dest = msg.getReceiver();

      (owner.getConvTA()).append("Sending message: "+
         msg.getType()+" to "+dest+"\n");
      if ((msg.getConvName()) != null)
      (owner.getConvTA()).append("message conversation name: "+
         msg.getConvName()+"\n");
      //get CommInterface of receiving agent
      CommInterface commInterface= null;
      try{
       commInterface = (CommInterface)Naming.lookup(dest);
       commInterface.rxMsg(msg);
      }catch (Exception e){
      (owner.getConvTA()).append("Unable to send message: "+
         msg.getType()+" to "+dest+"\n");
```

```
            (owner.getConvTA()).append("Exception occured: "+
                e.getMessage()+"\n");
            }


    }//sendMsg

}//Communication :-)

public abstract class Conversation extends Thread
{
    protected Agent owner = null;
    //conversation name
    protected String name = null;
    //initial state of conversation
    protected String iState = null;
    //present state of conversation
    protected String pState = null;
    //list of final states of conversation
    protected Vector fStates = null;
    //message queue of this conversation object
    protected MessageQueue convMsgQ = null;
    //conversation status
    public boolean wait = false;
    //list of conversations waiting for this conversation to terminate
    protected Vector waitingConvs = null;
    //list of executed rules
    protected Vector listOfRulesExec = null;
    protected JTextArea convTA= null;
    //list of conversation rules in finite state automata
    protected Vector listOfRules= null;

    public Conversation(Agent parent,String state)
    {
        owner = parent;
        iState = state;
        pState = iState;
        fStates = new Vector();
        convTA= owner.getConvTA();
        listOfRules = new Vector();
        listOfRulesExec = new Vector();
        initializeRules();
    }

    public Conversation(Agent agent,String conv, String state)
    {
        this(agent,state);
    setConvName(conv);
  }

  protected void setConvName(String name)
  {
    this.name = name;
    convMsgQ = new MessageQueue(owner);
  }

  //set present state
  public void setPState(String state)
  { pState = state;}

  public void run()
  {
    applyFsaRule();
    while (!isFState()){
      yield();
      applyFsaRule();
    }
    if (isFState()){
      owner.removeFromConvList(this);
      convTA.append("Final state reached for "+name+" ");
      if (listOfRulesExec != null)
      for (int i=0; i<listOfRulesExec.size(); i++)
      convTA.append(((String)listOfRulesExec.elementAt(i))+", ");
      convTA.append("\n");
    }
  }

  //manage conversation as fsa and apply current rule
  protected void applyFsaRule()
  {
    // if waiting for a message then keep polling the msgQueue
    if (wait){
      if (!convMsgQ.isEmpty()){
        convTA.append("Message recieved by conv "+name+"\n");
        Date date = new Date();
        getOwner().getCommTA().append(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":  Message
received (" +((Message)convMsgQ.msgList.firstElement()).getType()+")");
        String                    content=((((Message)convMsgQ.msgList.firstElement()).getContent()
!=null)?((Message)convMsgQ.msgList.firstElement()).getContent():"");
        getOwner().getCommTA().append(content+"\n");
       wait = false;
      }
    }
    boolean currentRuleExec = false;
    Enumeration e = listOfRules.elements();
    while ((e.hasMoreElements()) && (!currentRuleExec) && (!wait)){
      ConvRule activeRule = (ConvRule)e.nextElement();
      if (pState.equals(activeRule.getCurrentState())){
        currentRuleExec = activeRule.execute();
        if (currentRuleExec){
          convTA.append(activeRule.getRuleName()+
           " executed \n");
         listOfRulesExec.addElement(activeRule.getRuleName());
        }
      }
    }

  }//end applyFsaRule

  /* Input Parameter: condition denoting whether to suspend or resume
  * conversation
  * This method either suspends or resumes conversation rule execution
  */
  public void setWait(boolean t){
   Date date= new Date();
  convTA.append(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+": "+this.getConvName()+" put in wait
state \n");
  wait = t;
  }

  public boolean isFState()
  {
    if (fStates.contains(pState)){
      return true;
    }
    return false;
  }

  public abstract void initializeRules();
  public String getConvName(){ return name;}
  public String getIState(){ return iState;}
  public String getPState(){ return pState;}
  public Vector getFStates(){ return fStates;}
  public JTextArea getConvTA(){  return convTA;}
  public Agent getOwner(){ return owner;}
  public Vector getListOfRulesExec(){ return listOfRulesExec;}
  public MessageQueue getConvMsgQ(){ return convMsgQ;}
}
```

```
public abstract class ConvRule
{
  public String ruleName = null;
  public Conversation ownerConv = null;
  public String currentState = null;
  public String nextState = null;
  public Message rxMsg = null;
  public Message txMsg = null;

  // List of waited for conversations this rule is waiting for to finish
  public Vector waitFor = null;

  public ConvRule(Conversation owner, String rule, String current, String next)
  {
    ruleName = rule;
    ownerConv = owner;
    currentState = current;
    nextState = next;
  }

  //method to execute rule
  public boolean execute()
  {
    boolean ruleExec = false;
    if (matchRxMsgFound())
    {
             if ((suchThat()) && (waitsForTest()))
             {
       ownerConv.setPState(nextState);
       Date date= new Date();
                 (ownerConv.getConvTA()).append(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":
"+ownerConv.getConvName()+" Executing rule "+ruleName+"\n");
       ruleExec = true;
       doBefore();
               waitForConvs();
       reply();
       doAfter();
     }
     else {
              ;
      //(ownerConv.getConvTA()).append("Input condition not satisfied for rule "+ruleName+"\n");
     }
    }
    else {
     ;
     //(ownerConv.getConvTA()).append("Input condition not satisfied for rule "+ruleName+"\n");
    }
    return ruleExec;
  }

  public void reply()
  {
    setTxMsg();
    if (txMsg != null){
       txMsg.setConvName(ownerConv.getConvName());
       txMsg.setSender((ownerConv.getOwner()).getAgentFullName());
       Date date= new Date();
       (ownerConv.getConvTA()).append(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":  "+ruleName
+" transmitting msg type "+txMsg.getType()+ " to "+txMsg.getReceiver()+"\n");
       (ownerConv.getOwner().getCommTA()).append(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":
transmitting msg ("+txMsg.getType()+ ") to "+txMsg.getReceiver()+" "+(txMsg.getContent() != null?
txMsg.getContent(): "")+"\n");
       (ownerConv.getOwner()).communication.txMsg(txMsg);
    }
   }

   /* This method waits for the list of converstion in it's waitForList to
   * terminate before allowing the parent conversation to proceed
   */
   public void waitForConvs()
   {
     if (waitFor !=null){
       for (int i=0;i< waitFor.size();i++){
                 Conversation c = (Conversation)waitFor.elementAt(i);
                 Utility.print("Waiting For " + c.getConvName()+ " to end");
                 (ownerConv.getConvTA()).append("Waiting For " + c.getConvName()+ " to end"+"\n");
                 while (!c.isFState()){
                    try {ownerConv.sleep(5);}
                    catch (InterruptedException e){
                    Utility.print("InterruptedException occurred");}
                 }
       }
     }
    }

    //method to chech if any message in owner conv's msgQueue matches expected received message
    public boolean matchRxMsgFound()
    {
      boolean found = false;
      setRxMsg();
      if (rxMsg == null)
      found = true;
      else {
       Message matchedMsg = null;
```

```
        String sender = rxMsg.getSender();
        MessageQueue q = ownerConv.getConvMsgQ();
        Vector v = null;
        if (sender == null){
                    v = q.findByType(rxMsg.getType());

        }else {
                    v = q.findByTypeSender(rxMsg.getType(),sender);

        }
        if (!v.isEmpty())
                    {
                    if (rxMsg.getContent() != null){
                      matchedMsg = matchContent(v,rxMsg.getContent());
                    }else {
                      matchedMsg = (Message)v.firstElement();
                    }
                    }
        if (matchedMsg != null) {
          Date date= new Date();
                              (ownerConv.getConvTA()).append(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":
"+ruleName+" found expected recvd msg in the queue"+ "\n");
          try{
          (ownerConv.getOwner().getCommTA()).append(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":
message received ("+txMsg.getType()+ ") "+(txMsg.getContent() != null? txMsg.getContent(): "")+"\n");
          }catch(Exception ex){}
                    found = true;
                    rxMsg = matchedMsg;
                    q.removeMsg(matchedMsg);
        }
        else {
          ;
                    //(ownerConv.getConvTA()).append(ruleName+" could not find expected recvd msg in the queue"+
"\n");
        }
      } //end else
      return found;
    }

  public Message matchContent(Vector v, Object o)
    {
      boolean found = false;
      Message msg = null;

      if (o instanceof String){
        String t = (String)o;
        int index = 0;
        while ((index < v.size()) && (!found)) {
          msg = (Message)v.elementAt(index);
          if (msg.getContent() instanceof String){
            String str = (String) msg.getContent();
            if ( t.equals(str))
            found = true;
          }
        }
      }
      return msg;
  }

  //STUB methods to be defined by inheriting classes as approriate
  public void setRxMsg(){ };
  public void setTxMsg(){ };
  public void doBefore(){ };
  public boolean suchThat(){return true;};
  public boolean waitsForTest(){return true;};
  public void doAfter(){ };

  public String getCurrentState(){ return currentState;}
  public String getRuleName(){ return ruleName;}

}//ConvRule

public class MessageQueue
{
  // list of messages in queue
  public Vector msgList = new Vector();
  //owner agent for msg queue
  Agent owner = null;

  MessageQueue(Agent name){
    owner = name;
  }


  public void addMsg(Message msg){
    msgList.addElement(msg);
  }
  public void removeMsg(Message msg){
    msgList.removeElement(msg);
  }
  public boolean isEmpty(){
    return msgList.isEmpty();
  }
```

```
//method to find all messages of given performative (type)
public Vector findByType(String type){
   Vector result= new Vector();
   Enumeration e= msgList.elements();
   while(e.hasMoreElements()){
      Message msg=(Message)e.nextElement();
      if(type.equalsIgnoreCase(msg.getType()))
          result.addElement(msg);
   }
   return result;
}

//method to find all messages for given conversation name
public Vector findByConvName(String name){
   Vector result= new Vector();
   Enumeration e= msgList.elements();
   while(e.hasMoreElements()){
      Message msg=(Message)e.nextElement();
      if(name.equals(msg.getConvName()))
      result.addElement(msg);
   }
   return result;
}

public Vector findBySender(String sender){
   Vector result= new Vector();
   Enumeration e= msgList.elements();
   while(e.hasMoreElements()){
      Message msg=(Message)e.nextElement();
      if(sender.equals(msg.getSender()))
      result.addElement(msg);
   }
   return result;
}

//method to find all messages of given performative (type) and sender

public Vector findByTypeSender(String type, String sender){
   Vector result= new Vector();
   Enumeration e= msgList.elements();
   while(e.hasMoreElements()){
      Message msg=(Message)e.nextElement();
      if(type.equals(msg.getType()) && sender.equals(msg.getSender()))
      result.addElement(msg);
   }
   return result;
}
```

```
  }

  public class MsgManager
  {
    private Agent owner = null;

    public MsgManager (Agent agent){
      owner = agent;
    }

    public void handleMsg(Message msg){
      Vector listOfConvs = owner.getConvList();
      String convName = msg.getConvName();
      if (listOfConvs.isEmpty()){
        (owner.getInfoTA()).append("Strating conversation "+
         msg.getConvName()+"\n");
        owner.startConversation(msg);
      }

      else{
        boolean convFound = false;
        int index = 0;
        while ((index < listOfConvs.size()) && (!convFound)){
          Conversation conv = (Conversation)listOfConvs.elementAt(index);
          if (convName.equals(conv.getConvName())){
            (conv.getConvMsgQ()).addMsg(msg);
            convFound = true;
          }
          index++;
        }
        if (!convFound){
         (owner.getInfoTA()).append("Strating conversation "+
          msg.getConvName()+"\n");
         owner.startConversation(msg);
        }
      }
    }
  }

package mt.negotiation;
import mt.mas.*;

public class NegoPlannerConversation extends Conversation {
  // negotiator agent associated with this conversation
  private String planner = null;
  private String request = null;
```

```
public String offerMsg=null;


NegoPlannerConversation(Negotiator agent, String offerToEvaluate)
{
 super(agent,"start");
 fStates addElement("accept");
 fStates.addElement("quit");
 fStates.addElement("reject");
 offerMsg = offerToEvaluate;
 String name = owner.getAgentName()+"_Planner_Conversation";
 setPlanner(((Negotiator)getOwner()).getPlanner());
 setConvName(name);
}


public void setPlanner(String planner){ this.planner = planner;}
public String getPlanner(){return planner;}



// This method initializes and instantiates the rules within this conversation
public void initializeRules()
{
  listOfRules.addElement(new NegPlanConvRule1(this));
  listOfRules.addElement(new NegPlanConvRule2(this));
  listOfRules.addElement(new NegPlanConvRule3(this));
  listOfRules.addElement(new NegPlanConvRule4(this));
}

}

// Class for first conversation rule of NegotiatorConversation
class NegPlanConvRule1 extends ConvRule
{
 // Associated conversation
 private NegoPlannerConversation conv = null;

 NegPlanConvRule1(NegoPlannerConversation c) {
   super(c, "np1","start","working");
   conv = (NegoPlannerConversation)ownerConv;
 }

 // This method sets the message to be transmitted if the rule  executes
  public void setTxMsg()
  {
   txMsg = new Message("evaluate");
 txMsg.setReceiver("rmi://"+conv.getPlanner());
 txMsg.setContent(conv.offerMsg);
}

// This method defines the action to be done after transmitting the message
public void doAfter(){
  conv.setWait(true);        //wait for a reply from the planner (keep polling the msgQ)
 }

}


class NegPlanConvRule2 extends ConvRule
{
  private NegoPlannerConversation conv = null;

  NegPlanConvRule2(NegoPlannerConversation c) {
    super(c,"np2","working","accept");
    conv = (NegoPlannerConversation)ownerConv;
    }

   public void setRxMsg()
   {
    rxMsg = new Message("accept");
   }

  }


  class NegPlanConvRule3 extends ConvRule
  {
   private NegoPlannerConversation conv = null;

   NegPlanConvRule3(NegoPlannerConversation c) {
     super(c,"np3","working","reject");
     conv = (NegoPlannerConversation)ownerConv;
    }
   public void setRxMsg()
    {
     rxMsg = new Message("reject");
    }
    public void doAfter(){
     ((Negotiator)conv.getOwner()).setBestOffer(rxMsg.getContent());
    }
   }
```

```
class NegPlanConvRule4 extends ConvRule
{
 private NegoPlannerConversation conv = null;

 NegPlanConvRule4(NegoPlannerConversation c) {
  super(c,"np4","working","quit");
  conv = (NegoPlannerConversation)ownerConv;
 }

 public void setRxMsg()
 {
  rxMsg = new Message("quit");
 }
}

public class Negotiator extends mt.mas.Agent
{
  // Negotaitor Conversation
  private  NegotiatorConversation conv = null;
  public static final String group ="Negotiation";
  public int agentCount;
  private String supplier;
  public String planner=null;
  public String currentBestOffer = null;


  Negotiator(int number,JTextArea infoA, JTextArea convA,JTextArea commA, String plannerAgent)
  {
   super(group+"Agent"+ number,group, null);   //last parameter for registry
   Utility.print("Negotiation Agent created: "+ name);
   planner = plannerAgent;
   infoTA = infoA;
   convTA = convA;
   commTA = commA;
   infoTA.append(name+ " created: "+ new Date().toLocaleString()+"\n");
  }

  public void setBestOffer(String offer){ currentBestOffer = offer;}
  public String getBestOffer(){ return currentBestOffer;}

  public String getPlanner(){ return planner;}

 public void setRequest(String order, String supplier) {
   request = order;
   this.supplier = supplier;
   startConversation(request); // request should be Message object
   super.start();
  }


 public void startConversation(Object o)
  {
     conv = new NegotiatorConversation(this);
     conv.setRequest(request);
     conv.setSupplier(supplier);
     addToConvList(conv);
     conv.start();

  }

 } // end class Customer



public class NegotiatorConversation extends mt.mas.Conversation
{

/**NegotiatorConversation extends Conversation class and it
 runs within the Negotiator agents. This class defines the conversation
 which is started when a Negotiator Agent (on behalf of the consumer) sends RFP.
 This conversation defines the interactions between Negotiator and Supplier
 agent during negotiation session. It implements the abstract
 initializeRules() method of Conversation class to initialize and
 instantiate  it's conversation rules. */

 // supplier agent associated with this conversation
 private String supplier = null;
 private String request = null;

 //conversation started with the agent
 public NegoPlannerConversation negoPlannerConversation= null;
 public boolean offerEvaluated = false;
 public boolean isBestOffer = false;
 public boolean quit = false;
 //current received offer to be evaluated
 public String offerToEvaluate= null;


/* Constructor
 * Input Parameter: Parent agent
 */
NegotiatorConversation(Negotiator agent)
{
 super(agent,"start");
```

```
  fStates.addElement("accept");
  fStates.addElement("quit");
  String name = owner.getAgentName()+"_Conversation";
  setConvName(name);
 }

 public void setRequest(String request){ this.request = request;}
 public void setSupplier(String supplier){ this.supplier = supplier;}

 // ACCESSOR METHODS
 public String getSupplier(){return supplier;}
 public String getRequest(){ return request; }


 public void getEvaluation(){
  if("accept".equals(negoPlannerConversation.getPState()))
    isBestOffer = true;
  else if ("reject".equals(negoPlannerConversation.getPState()))
  {
    isBestOffer = false;
  }
  else if ("quit".equals(negoPlannerConversation.getPState()))
  {
    isBestOffer = false;
    quit = true;
  }
 }
 // This method initializes and instantiates the rules within this conversation
 public void initializeRules()
 {
  //if (listOfRules == null)
  //listOfRules = new Vector();
   listOfRules.addElement(new NegoConvRule1(this));
   listOfRules.addElement(new NegoConvRule2(this));
   listOfRules.addElement(new NegoConvRule3(this));
   listOfRules.addElement(new NegoConvRule4(this));
   listOfRules.addElement(new NegoConvRule5(this));
   listOfRules.addElement(new NegoConvRule6(this));
   listOfRules.addElement(new NegoConvRule7(this));
  //listOfRules.addElement(new NegoConvRule8(this));
 }

}

// Class for first conversation rule of NegotiatorConversation
class NegoConvRule1 extends ConvRule
{
```

```
// Associated conversation
private NegotiatorConversation conv = null;

NegoConvRule1(NegotiatorConversation c) {
  super(c, "cs1","start","proposeal_requested");
  conv = (NegotiatorConversation)ownerConv;
}

// This method sets the message to be transmitted if the rule  executes
public void setTxMsg()
{
  txMsg = new Message("RFP");
  txMsg.setContent(conv.getRequest());
  txMsg.setReceiver(conv.getSupplier());
}

// This method defines the action to be done after transmitting the message
public void doAfter(){
  conv.setWait(true);        //wait for a reply from the supplier (keep polling the msgQ)
 }
}


class NegoConvRule2 extends ConvRule
{
 private NegotiatorConversation conv = null;

 NegoConvRule2(NegotiatorConversation c) {
   super(c,"cs2","proposeal_requested","evaluating");
   conv = (NegotiatorConversation)ownerConv;
  }

 /* This method sets the expected received message pattern
  */
 public void setRxMsg()
 {
   rxMsg = new Message("offer");
 }

 public void doBefore(){
   conv.offerToEvaluate = rxMsg.getContent();

 }

}
```

```
class NegoConvRule3 extends ConvRule
{
 private NegotiatorConversation conv = null;

 NegoConvRule3(NegotiatorConversation c) {
  super(c,"cs3","evaluating","evaluating");
  conv = (NegotiatorConversation)ownerConv;
  }

 public boolean suchThat(){
  return (!conv.offerEvaluated); } //not yet evaluted offer

 //start negoPlanConversation to compare the offer with others
 public void doBefore(){
  conv.negoPlannerConversation = new NegoPlannerConversation((Negotiator)conv.getOwner(), conv.offerToEvaluate);
  (conv.getOwner()).addToConvList(conv.negoPlannerConversation);
  conv.negoPlannerConversation.start();
  waitFor = new Vector();
  waitFor.addElement(conv.negoPlannerConversation);
 }

 public void doAfter(){
  conv.offerEvaluated = true;
  conv.getEvaluation();
 }

} //end class NegoConvRule3


class NegoConvRule4 extends ConvRule
{
 private NegotiatorConversation conv = null;

  NegoConvRule4(NegotiatorConversation c)  {
   super(c,"cs4","evaluating","refine_req");
   conv = (NegotiatorConversation)ownerConv;
  }

 //wait for the negoPlannerConversation to finish then check if refinement is requested
 public boolean waitsForTest(){
   return (!conv.isBestOffer && ! conv.quit && conv.offerEvaluated);
 }

  public void setTxMsg()
  {
   txMsg = new Message("refine_offer");
   txMsg.setReceiver(conv.getSupplier());
  txMsg.setContent(" I got better offer: \n"+((Negotiator)conv.getOwner()).getBestOffer()+"\nplease refine your offer");
 }

 public void doAfter(){
  conv.setWait(true);  // now wait for a refined offer
 }

} //end class CustConvRule4

class NegoConvRule5 extends ConvRule
{
 private NegotiatorConversation conv = null;

  NegoConvRule5(NegotiatorConversation c){
   super(c,"cs5", "refine_req","evaluating");
   conv = (NegotiatorConversation)ownerConv;
  }

 public void setRxMsg()
  {
   rxMsg = new Message("offer");
  }

 //set the offer as not evaluated yet
 public void doBefore(){
    conv.offerEvaluated = false;
    conv.offerToEvaluate = rxMsg.getContent();
  }

 }

 class NegoConvRule6 extends ConvRule
 {
   private NegotiatorConversation conv = null;

   NegoConvRule6(NegotiatorConversation c) {
     super(c,"cs6","evaluating","accept");
     conv = (NegotiatorConversation)ownerConv;
    }

   //wait for the negoPlannerConversation to finish then check if this is bestOffer
   public boolean waitsForTest(){
     return (conv.isBestOffer && ! conv.quit && conv.offerEvaluated);
   }

    public void setTxMsg()
    {
```

```
   txMsg = new Message("accept");
   txMsg.setReceiver(conv.getSupplier());
   txMsg.setContent("Your last offer is accepted, let us finalise the deal");
  }

}

class NegoConvRule7 extends ConvRule
{
 private NegotiatorConversation conv = null;

 NegoConvRule7(NegotiatorConversation c) {
   super(c,"cs7","evaluating","quit");
   conv = (NegotiatorConversation)ownerConv;
 }
//wait for the negoPlannerConversation to finish then check if asked to quit
public boolean waitsForTest(){
   return (conv.quit);
}

 public void setTxMsg()
 {
   txMsg = new Message("quit");
   txMsg.setReceiver(conv.getSupplier());
   txMsg.setContent("Sorry - I accepted better offer");
 }
}


public class Offer {
 Hashtable offer;
 int receivingAgent = -1;


 public Offer() {
    offer= new Hashtable();
 }
 public Offer(String offr, int agentNum) {
    offer = new Hashtable();
    StringTokenizer st = new StringTokenizer(offr.trim(),",");
    while(st.hasMoreTokens())
    {
     StringTokenizer st2= new StringTokenizer(st.nextToken().trim(),"=");
     this.add(st2.nextToken().trim(), st2.nextToken().trim());
    }
    receivingAgent = agentNum;
 }
```

```
  public void add(String criteria, String option) {
    offer.put(criteria, option);
  }

 public int getReceiver(){ return receivingAgent;}

 //implement
 public int getPayoff(){
  UserInputPanel userPanel= (UserInputPanel)Mercy.sharedInstance().userInputPanel;
  //String product= userPanel.((String)productCB.getSelectedItem()).trim();
  Vector cwVector= userPanel.cwVector;
  Enumeration e=offer.keys();
  String criteria=null;
  int optionValue=0;
  int payoff=0;
  while (e.hasMoreElements())
  {
   criteria= (String)e.nextElement();
   String option = (String)offer.get(criteria);
   boolean hit = false;
   for( int i=0; i< cwVector.size(); i++)
   {
   try{
   if (((CWValue)cwVector.elementAt(i)).getC().equalsIgnoreCase(criteria)&& !hit)
   {
    hit = true;
    if(((CWValue)cwVector.elementAt(i)).getF().equals("linear"))
    {
     //since this is linear function I would expect the option to be Integer
     int value=0;
     try{
       value= Integer.parseInt(option);
     }catch(Exception ex){
      Utility.print("integer value was expected for option");
    }
   JCVector v= ((CWValue)cwVector.elementAt(i)).getOptions();
   int min= Integer.parseInt((v.getFirst().toString()).replace('[',' ').replace(']',' ').trim());
   int max= Integer.parseInt((v.getLast().toString()).replace('[',' ').replace(']',' ').trim());
  //the equation to use is 10/|max-min| *|value-min|
  //to give the highest option a value of 10 and the lowest a value of 0
  //where min is the minimum payoff value
  //first check which is higher in value min or max
  if (value > min && min > max) //decrease is better than increase
   optionValue = 0;
  else if ( value > max && max > min) //increase is better than decrease
   optionValue = 10;
  else
    optionValue = 10*Math.abs(min-value)/Math.abs(min-max);
   payoff += optionValue* ((CWValue)cwVector.elementAt(i)).getNW();
   //Utility.print(value+" payoff "+optionValue);
  }
  else if (((CWValue)cwVector.elementAt(i)).getF().equals("discrete")) {
    Vector v= ((CWValue)cwVector.elementAt(i)).getOptions();
    boolean found=false;
    Enumeration e1= v.elements();
    while (e1.hasMoreElements() && ! found)
    {
      JCVector row =(JCVector) e1.nextElement();
      if ( row.elementAt(0).toString().equalsIgnoreCase(option))
      {
        found = true;
        optionValue = Integer.parseInt(row.elementAt(1).toString());
        payoff += optionValue* ((CWValue)cwVector.elementAt(i)).getNW();
        //Utility.print(option+" payoff "+optionValue);
      }
     }
    }//else if
   }//if
  }catch(Exception ex){Utility.print(ex.getMessage());}
  }//for
 }//while
 return payoff;
}

public String toString(){
   String reply= null;
   Enumeration e= offer.keys();
   String token = null;
   if( e.hasMoreElements())
   {
     token = (String)e.nextElement();
     reply = token+"="+(String)offer.get(token);
    }
    while( e.hasMoreElements())
    {
      token = (String)e.nextElement();
      reply += ", "+token+"="+(String)offer.get(token);
     }
     return reply.trim();
   }
  }

  public class PlannerConversation extends Conversation {
```

```
// negotiator agent associated with this conversation
private String negotiator = null;
private String request = null;
//public boolean bestOffer = false;

/* Constructor
 * Input Parameter: Parent agent
 */
PlannerConversation(PlannerAgent agent, String name)
{
  super(agent,"start");
  fStates.addElement("accept");
  fStates.addElement("quit");
  fStates.addElement("reject");
  //String name = owner.getAgentName()+"_Conversation";
  setConvName(name);
}

public void setNegotiator(String negotiator){ this.negotiator = negotiator;}

// ACCESSOR METHODS
public String getNegotiator(){return negotiator;}
public int getNegotiatorNumber(){
  return (Integer.parseInt(negotiator.substring(negotiator.indexOf("Agent")+5)));  //recheck the beginIndex
}
public boolean isBestOffer(){ return
  ((PlannerAgent)getOwner()).isBestOffer[getNegotiatorNumber()];
} //implement to evalute offer -need to move it to Negotiator class


// This method initializes and instantiates the rules within this conversation
public void initializeRules()
{
  listOfRules.addElement(new PlanConvRule1(this));
  listOfRules.addElement(new PlanConvRule2(this));
  listOfRules.addElement(new PlanConvRule3(this));
  listOfRules.addElement(new PlanConvRule4(this));
  listOfRules.addElement(new PlanConvRule5(this));
  listOfRules.addElement(new PlanConvRule6(this));
}

}

// Class for first conversation rule of NegotiatorConversation
class PlanConvRule1 extends ConvRule
{
  // Associated conversation
  private PlannerConversation conv = null;

  PlanConvRule1(PlannerConversation c) {
   super(c, "pc1","start","working");
   conv = (PlannerConversation)ownerConv;
  }

  // This method sets the message to be transmitted if the rule  executes
  public void setRxMsg()
  {
   rxMsg = new Message("evaluate");
  }

  //collect and evalute all offers
  public void doBefore(){
    conv.setNegotiator(rxMsg.getSender());
    synchronized (conv.getOwner()){
     if (((PlannerAgent)conv.getOwner()).getListOfOffers().size() <
((PlannerAgent)conv.getOwner()).numberOfNegoAgents )
      ((PlannerAgent)conv.getOwner()).getListOfOffers().addElement(new Offer(rxMsg.getContent(),
conv.getNegotiatorNumber()));

     if(((PlannerAgent)conv.getOwner()).getListOfOffers().size() ==
((PlannerAgent)conv.getOwner()).numberOfNegoAgents)
     {
       ((PlannerAgent)conv.getOwner()).evaluateOffers();
     }
     else
      conv.setWait(true);
    }
  }
}


class PlanConvRule2 extends ConvRule
{
  private PlannerConversation conv = null;

  PlanConvRule2(PlannerConversation c) {
   super(c,"pc2","working","wait");
   conv = (PlannerConversation)ownerConv;
   }

  //bestOffer but wait for second round
  public boolean suchThat(){
   return (conv.isBestOffer()); /*&& !(((PlannerAgent)conv.getOwner()).offerAccepted)); */
```

```
 }

 /*public boolean waitsForTest(){
   return ((PlannerAgent)conv.getOwner()).canReplyToAgent[conv.getNegotiatorNumber()];
 } */

 public void doBefore(){
   Date date= new Date();
   conv.getOwner().getInfoTA().append(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":
"+conv.getNegotiator()+" has best offer (1st round)");
   synchronized(conv.getOwner())
   {
    if                                        (((PlannerAgent)conv.getOwner()).getListOfOffers().size()<
((PlannerAgent)conv.getOwner()).numberOfNegoAgents)
     conv.setWait(true);
     else
     ((PlannerAgent)conv.getOwner()).evaluateOffers();
   }
  }


}


class PlanConvRule3 extends ConvRule
{
 private PlannerConversation conv = null;

 PlanConvRule3(PlannerConversation c) {
   super(c,"pc3","working","reject");
   conv = (PlannerConversation)ownerConv;
  }

 public boolean suchThat(){
   return (!conv.isBestOffer() && !(((PlannerAgent)conv.getOwner()).offerAccepted));
  }

 /* public boolean waitsForTest(){
   return ((PlannerAgent)conv.getOwner()).canReplyToAgent[conv.getNegotiatorNumber()];
  }*/

 public void setTxMsg()  //ask for refinement to the offer
  {
   txMsg = new Message("reject");
   txMsg.setReceiver("rmi://"+conv.getNegotiator());
   txMsg.setContent(((PlannerAgent)conv.getOwner()).currentBestOffer.toString());
  }
```

```
}


class PlanConvRule4 extends ConvRule
{
 private PlannerConversation conv = null;

 PlanConvRule4(PlannerConversation c) {
  super(c,"pc4","working","quit");
  conv = (PlannerConversation)ownerConv;
 }

 public boolean suchThat(){
  return (!conv.isBestOffer() && (((PlannerAgent)conv.getOwner()).offerAccepted));
 }

 /*public boolean waitsForTest(){
  return ((PlannerAgent)conv.getOwner()).canReplyToAgent[conv.getNegotiatorNumber()];
 } */

 public void setTxMsg()
 {
  txMsg = new Message("quit");
  txMsg.setReceiver("rmi://"+conv.getNegotiator());
 }

 public void doAfter(){
 ((PlannerAgent)conv.getOwner()).killNegotiator();
 }

}

class PlanConvRule5 extends ConvRule
{
 private PlannerConversation conv = null;

 PlanConvRule5(PlannerConversation c){
  super(c,"pc5", "wait","reject");
  conv = (PlannerConversation)ownerConv;
 }

 public boolean suchThat(){
  return (!conv.isBestOffer());
 }

/* public boolean waitsForTest(){
   return ((PlannerAgent)conv.getOwner()).canReplyToAgent[conv.getNegotiatorNumber()];
  }*/

  public void setTxMsg()
   {
    txMsg = new Message("reject");
    txMsg.setReceiver("rmi://"+conv.getNegotiator());
    txMsg.setContent(((PlannerAgent)conv.getOwner()).currentBestOffer.toString());
   }
 }

class PlanConvRule6 extends ConvRule
{
 private PlannerConversation conv = null;

 PlanConvRule6(PlannerConversation c){
   super(c,"pc6", "wait","accept");
   conv = (PlannerConversation)ownerConv;
  }

 public boolean suchThat(){
   return (conv.isBestOffer()&& (((PlannerAgent)conv.getOwner()).offerAccepted));
  }

 public void setTxMsg()
  {
   txMsg = new Message("accept");
   txMsg.setReceiver("rmi://"+conv.getNegotiator());
  }

 public void doAfter(){
  ((PlannerAgent)conv.getOwner()).killNegotiator();
 }
}//plannerConversation


public class PlannerAgent extends Agent {

   // Planner_Negotaitor Conversations
  private PlannerConversation conv = null;
  private String request; //could be changed to request object
  private String supplier;
  private Negotiator negotiator;
  public static final String group="Planner";

  //to check if an offer is already selected
  public boolean offerAccepted = false;
 //to update the list of current offers
 private static Vector listOfOffers = null;
 public static int numberOfNegoAgents = 0;
 public static int repliedToCount = 0;
 public Offer currentBestOffer = null;
 //public boolean[] canReplyToAgent;
 public boolean[] isBestOffer;
 Mercy main;


 public PlannerAgent(String name,JTextArea infoA, JTextArea convA, JTextArea commA)
 {
  super("Planner",group, null); //third parameter for registryHost
  Utility.print("Planner_agent created: "+ name);
  infoTA = infoA;
  convTA = convA;
  commTA = commA;
  Date date= new Date();
  infoTA.append(name+ " created at "+ date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+"\n");
  listOfOffers = new Vector();
  createNegotiators();

 }

 /*
 * This method creates a number of negotiators equivalent to number of suppliers
 */
public void createNegotiators(){
 try
 {
              String[] bindings = Naming.list("//"+getLocalHost()+":"+PORT+"/");
   negotiationPanel panel;
              for (int i=0;i<bindings.length; i++)
   {
    //testing
    try{
              //System.out.println(bindings[i]);
    if (bindings[i].indexOf("Supplier") != -1){ //if this is a supplier
     //create a negotiator to negotiate with this supplier
     numberOfNegoAgents++;
     panel= new negotiationPanel();
     ((AgentsInteractionPanel)
Mercy.sharedInstance().agentsInteractionPanel).getNegotiationTabs().add("Negotiator_"+numberOfNegoAgents, panel);
     ((AgentsInteractionPanel)
Mercy.sharedInstance().agentsInteractionPanel).getNegotiationTabs().setSelectedIndex(0);
     negotiator = new Negotiator(numberOfNegoAgents, panel.nGeneralTA,panel.nConvTA,panel.nCommTA,
this.getAgentFullName());
```

```
                negotiator.setRequest("pc",bindings[i]);
        }
        }catch(Exception ex){ }
      }
      //canReplyToAgent = new boolean[numberOfNegoAgents+1];
      isBestOffer = new boolean[numberOfNegoAgents+1];
      for (int i=0; i< isBestOffer.length; i++)
       // canReplyToAgent[i]=false;
       isBestOffer[i]=false;

     }
    catch(Exception e){
                System.out.println("Error: " + e);
     }
  }

public void startConversation(Object m)
  {
   if (m instanceof Message){
      Message msg = (Message)m;

      if ("evaluate".equals(msg.getType())){
        conv = new PlannerConversation(this,msg.getConvName());
        addToConvList(conv);
        (conv.getConvMsgQ()).addMsg(msg);
        Date date= new Date();
        getCommTA().append(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":    Message    received
(evaluate) ");
        getCommTA().append(msg.getContent()+"\n");
        conv.start();
      }
   }
  }

 public synchronized Vector getListOfOffers(){ return listOfOffers;}


 public synchronized void evaluateOffers(){
   currentBestOffer = (Offer)listOfOffers.firstElement();
   int bestPayoff=currentBestOffer.getPayoff();

   for (int i=1; i<listOfOffers.size(); i++)
   {
      Offer offer = (Offer)listOfOffers.elementAt(i);
      try{
       int payoff = offer.getPayoff();
       Date date= new Date();
      Utility.print(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":  offer  received  "+offer.toString()+"
 (payoff) "+payoff);
      if ( payoff > bestPayoff)
      {
       currentBestOffer = offer;
       bestPayoff = payoff;
      }
     }catch(Exception e){Utility.print("payoff can not be calculated for "+offer.toString());}
   }

   //for (int i=0; i<listOfOffers.size(); i++)
      // listOfOffers.removeElementAt(i);
   listOfOffers.removeAllElements();
   listOfOffers.addElement(currentBestOffer);
   if( isBestOffer[currentBestOffer.getReceiver()]){ //this is bestoffer for 2 rounds
        offerAccepted = true;
   }
   else {
    for (int j=1; j<isBestOffer.length; j++)
          isBestOffer[j] = false;
    isBestOffer[currentBestOffer.getReceiver()]=true;
   //Utility.print("best offer is: "+currentBestOffer.toString()+", receiver "+ currentBestOffer.getReceiver());
   }

   for (int i=0; i<convList.size(); i++)
   {
    Conversation conv= (Conversation)convList.elementAt(i);
    Date date= new Date();
    Utility.print(date.getHours()+":"+date.getMinutes()+":"+date.getSeconds()+":    "+conv.getConvName()+"    resume
execution");
    conv.setWait(false);
   }
 }

 public synchronized void killNegotiator(){
  --numberOfNegoAgents;
 }
```

Appendix I

Sheet for Negotiation Experiment

In this appendix, the experimental sheets given to subjects participating in the negotiation experiments are shown. A classroom session was held in which the goal of the experiment and the utility function is introduced to all participating subjects. Then, every subject was provided with the experimental sheet and only the utility function for the role to play.

GUIDELINES FOR CONSUMER'S AGENT SUBJECTS

Name:

Year:

Please follow the following rules in the negotiation session:

1. Transmission of any information to suppliers' subjects is permitted. Thus, for example, it is legal for a customer's agent to say to one of the suppliers " I received better offer from supplier X, therefore I will buy from him if you do not improve your offer". It is also legal to say " How about increasing the RAM to 128MB and keep everything else the same".

2. Your goal of the negotiation is maximise the achieved payoff of the negotiated agreement (i.e. try to get the least cost under most favourable conditions).

Please follow the given customer's preferences and the utility function described in the session.

Remember that the overall utility score is calculated using the utility function $U_j = \Sigma_k w_k u_{jk}$. Where U_j is the utility value of alternative j, w_k is the weight for attribute k and u_{jk} is the scaled value for alternative j on attribute k.

Therefore, for the alternative (£900, PII-300, 32MB, qty 50, delivery 7, brand IBM)

u(£900)* 25 + u(P II-300) * 20 + u(32MB) * 20 + u(qty 50) *15+ u(del 7)*10+ u(IBM)*7 =
10 * 25 + 2 *20 + 2* 20 + 5 *15 + 10 * 10 + 10 * 7

Please list down all the offers received and the final agreed negotiation.

Remember that a prize would be awarded to the buyer who gets the best-negotiated payoff.

GUIDELINES FOR SUPPLIERS' SUBJECTS

Name:

Year:

Please follow the following rules in the negotiation session:

1. Transmission of any information to the negotiators is permitted. Thus, for example, it is legal for the customer's agent to say to one of the suppliers " I received better offer from supplier X, therefore I will buy from him if you do not improve your offer". It is also legal to say " How about increasing the RAM to 128MB and keep everything else the same". However, communication between suppliers themselves is illegal.

2. Your goal of the negotiation is maximise the achieved payoff of the negotiated agreement (i.e. to get the best deal under the most favourable settings).

3. Your profit margin is set to 30%. For simplicity, it is assumed that the payoff of an offer is the same as the selling value, which means that an offer with a payoff of less than 700 (where 1000 is the maximum achievable payoff) is not feasible.

Please follow the given supplier preferences and the utility function described in the session.

Remember that the overall utility score is calculated using the utility function $U_j = \Sigma_k w_k u_{jk}$. Where U_j is the utility value of alternative j, w_k is the weight for attribute k and u_{jk} is the scaled value for alternative j on attribute k.

Please list down all the offers received and the final agreed negotiation.

Remember that a prize would be awarded to the seller who gets the best-negotiated payoff.