INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

**ACTIVE NEGOTIATION SUPPORT WITH A
SOFTWARE AGENT**

By

Gordon Lo, B.Comm

**A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of**

MASTER OF BUSINESS ADMINISTRATION

**Sprott School of Business
Carleton University
Ottawa, Ontario
August 1, 2001**

**© Copyright
2001, Gordon Lo**

**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-66864-9

Canada

The undersigned recommend to the Faculty of Graduate Studies and
Research acceptance of the thesis

ACTIVE NEGOTIATION SUPPORT WITH A SOFTWARE AGENT

Submitted by Gordon Lo, B. Comm.
In partial fulfillment of the requirements for the
degree of Master of Business Administration

Thesis Supervisor

Director, School of Business

School of Business
Carleton University

August 1, 2001

ABSTRACT

Software agent and decision support are rapidly developing information technologies due to their potential in supporting and conducting electronic transactions. Negotiation via the web is currently supported by several technologies, such as negotiation support systems, group decision support systems and negotiation software agents.

This thesis is based on research in three areas: business negotiation and negotiation support, electronic commerce and Internet technologies. A large experiment conducted in the InterNeg project led us to suggest a web-based integrated environment to provide assistance to the negotiators. We propose – as the first step in the development of an integrated negotiation environment (Aspire) – to aid negotiators with the use of software agents. A negotiation software agent (Atin) is developed in order to provide active support to Inspire users, for web-based negotiation training.

ACKNOWLEDGEMENT

This research could not have been conducted without the support of many people. First of all, I would like to thank Dr. Gregory Kersten for his continuous encouragement, helpful advice and considerable patience through the development of this thesis.

Thanks are extended to my thesis committee, Dr. David Cray, Dr. Alex Ramirez, Dr. Wojtek Michalowski, and Dr. George Haines, for their thoughtful comments and guidance throughout this research.

I have to thank all my colleagues at the InterNeg research group. I would like to acknowledge the help and contribution of Mei Leng to the development of the Aspire system. Also special thanks to Greg Schmidt, who helped a lot in the configuration and installation of software tools.

And finally, thanks to my parents who taught me courage to never give up, without their support this thesis would not be completed.

Table of Contents

1. INTRODUCTION	1
2. BUSINESS NEGOTIATIONS AND E-COMMERCE	5
2.1. E-COMMERCE.....	5
2.2. BUSINESS NEGOTIATIONS.....	8
2.3. ARE AUCTIONS NEGOTIATIONS?.....	11
2.4. WEB-BASED NEGOTIATIONS.....	14
3. COMMON NEGOTIATION PITFALLS	17
3.1. PREPARATION AND PROBLEM ANALYSIS	18
3.2. FOCUS ON POSITIONS.....	19
3.3. INEFFICIENT COMPROMISES.....	19
3.4. SOCIO-PSYCHOLOGICAL FACTOR.....	20
3.5. AVOIDING NEGOTIATION PITFALLS – AGENT-MEDIATED NEGOTIATIONS?	21
4. THE INSPIRE NEGOTIATION SUPPORT SYSTEM.....	23
4.1. THE INTERNEG PROJECT AND INSPIRE.....	23
4.2. INSPIRE SUPPORT METHODOLOGIES.....	24
4.3. INSPIRE NEGOTIATIONS	26
4.4. INSPIRE USE AND ASSESSMENT.....	27
5. WEB-BASED NEGOTIATION TECHNOLOGIES.....	28
5.1. SOFTWARE AGENTS.....	28
5.2. NEGOTIATION SOFTWARE AGENTS	31
5.3. NEGOTIATION SUPPORT SYSTEMS.....	34
6. RESEARCH OBJECTIVE	38
7. METHODOLOGY	42
7.1. SYSTEM DEVELOPMENT METHODOLOGY	42
7.2. SYSTEM ARCHITECTURE.....	49
7.2.1. <i>The web client</i>	51
7.2.2. <i>The web (HTTP) server</i>	51
7.2.3. <i>The application server</i>	52
7.2.4 <i>The database server</i>	53
7.3. ASPIRE FRAMEWORK.....	53
7.4. THE ROLES OF ATIN	60
7.4.1 <i>Pre-negotiation phase</i>	61
7.4.2 <i>Negotiation phase</i>	62
7.4.3 <i>Post-settlement phase</i>	65
7.5. NON-FUNCTIONAL (TECHNICAL) REQUIREMENTS.....	65
7.6. USE CASE MODEL	66
8. DESIGN	71

8.1.	DESIGN ISSUES	71
8.2.	SEQUENTIAL DIAGRAMS	72
8.3.	CLIENT-SIDE USER INTERFACE	77
8.4.	SERVER-SIDE PROCESSING	78
8.5.	DATABASE DEVELOPMENT	78
8.6.	KNOWLEDGE BASE AND RULE IMPLEMENTATION	83
9.	DEVELOPMENT	84
9.1.	TOOLS SELECTION	85
9.2.	HIGH LEVEL PHYSICAL DESIGN	87
9.3.	CODING.....	88
9.4.	KNOWLEDGE BASE IMPLEMENTATION.....	91
10.	TESTING.....	93
10.1.	TEST PLAN	94
10.2.	TESTING ACTIVITIES	96
10.3.	USER ACCEPTANCE TEST PLANNING	101
10.4.	REVIEW AND EVALUATION.....	102
11.	CONCLUSION	104
11.1.	LIMITATIONS OF THIS STUDY.....	105
11.2.	FUTURE WORK	107
	REFERENCES.....	108
	APPENDIX A: PROJECT OUTLINE AND INTENDED DEVELOPMENT PROCESS	117
	APPENDIX B: EXAMPLE OF NEGOTIATION RULES FOR THE KNOWLEDGE BASE.....	118
	APPENDIX C: ASPIRE REVIEW AND EVALUATION FORM	130
	APPENDIX D: SCREENSHOTS	131

List of Figures

FIGURE 1. THE COMMERCE VALUE CHAIN (TRESSE AND STEWART, 1998)	7
FIGURE 2. THE THREE PHASES IN A NEGOTIATION	10
FIGURE 3. INSPIRE'S SUPPORT IN DIFFERENT NEGOTIATION PHASES	25
FIGURE 4. USER ACCEPTANCE OF INSPIRE (1410 RESPONDENTS)	27
FIGURE 5. LIFECYCLE PLANNING FOR ATIN DEVELOPMENT	48
FIGURE 6. N-TIER VIEW OF ASPIRE.	51
FIGURE 7. ASPIRE SUPPORT IN THE THREE NEGOTIATION PHASES.....	54
FIGURE 8 THE ASPIRE ENVIRONMENT: A CONCEPTUAL VIEW (LO AND KERSTEN, 2000) ..	58
FIGURE 9. ATIN ARCHITECTURE	60
FIGURE 10. ATIN'S SUPPORT OF PRE-NEGOTIATION ACTIVITIES	62
FIGURE 11. ATIN'S SUPPORT IN THE CONDUCT OF NEGOTIATION	63
FIGURE 12. ATIN'S SUPPORT OF POST-SETTLEMENT ACTIVITIES	65
FIGURE 13 A USE CASE DIAGRAM FOR ATIN SUPPORT IN PRE-NEGOTIATION PHASE	67
FIGURE 14. A USE CASE DIAGRAM FOR OFFER EXCHANGE SUPPORT	68
FIGURE 15. ATIN SUPPORT USE CASES	69
FIGURE 16. SEQUENCE DIAGRAM FOR USER PREFERENCES SUPPORT	73
FIGURE 17. SEQUENCE DIAGRAM FOR OFFER SUPPORT	74
FIGURE 18. AN EXAMPLE OF ATIN WARNING DURING OFFER SUBMISSION	75
FIGURE 19. SEQUENTIAL DIAGRAM FOR ATIN SUPPORT.....	75
FIGURE 20. ASPIRE DATABASE.....	82
FIGURE 21. HIGH LEVEL DESIGN OF ASPIRE	88
FIGURE 22. ASPIRE FIRST SCREEN	131
FIGURE 23. ATIN ALERTS USER WHEN OPTION RATINGS ARE VIOLATED	132
FIGURE 24. ATIN ALERTS USER WHEN ISSUE RATING IS VIOLATED	133
FIGURE 25. ATIN ALLOWS USER TO ENTER RESERVATION LEVEL	133
FIGURE 26. ATIN ALLOWS USER TO ENTER BATNA	134
FIGURE 27. AN EXAMPLE OF ATIN SUGGESTION DURING NEGOTIATION PHASE	135
FIGURE 28. ATIN ALERTS USER REGARDING LARGE CONCESSIONS BETWEEN OFFERS	136

List of Tables

TABLE 1. DIFFERENCES AND SIMILARITIES AMONG NEGOTIATIONS, AUCTIONS, AND MARKETPLACE	12
TABLE 2. THE FIVE LEVELS OF THE CMM (BEMBERGER, 1997; PAULK, WEBER ET AL., 1995; JALOTE, 2000)	44
TABLE 3. DEVELOPMENT ACTIVITIES	85
TABLE 4. SYSTEM TEST PLAN.....	96
TABLE 5. SYSTEM TEST RESULT	100
TABLE 6. PROJECT SCHEDULE.....	117

1. Introduction

The rapid adoption of the Internet as a commercial medium brought a significant change to the traditional way business is conducted. In order to stay competitive many enterprises established their presence on virtual markets. Kersten (1998a) notes that e-commerce has localized global markets by opening remote markets to retail and to small companies. This allows small companies to look for customers and business partners anywhere in the world over the web. Internet-based tools and other e-commerce technologies (e.g. electronic ordering systems, electronic cash) facilitate and enhance conducting business via the web.

Transactions conducted on the Internet include retail and business-to-business transactions. Both types can be conducted directly on the seller's (or buyer's) web site, or via the use of intermediaries. In the latter case auction based transactions are most popular. Many software tools, search engines and support systems have been developed for both firms and individual customers to facilitate both direct and auction-based transactions.

Electronic shopping baskets that are usually used in a direct transaction are a simple extension of their physical counterparts so that there is little novelty in the shopping experience. Internet Auctions on the other hand, with their wide use and popularity, are a new phenomenon because they are low-cost and as in traditional auctions – they allow for the determination of product price. In traditional markets

auctions are expensive and have high transaction costs. In virtual markets the costs are almost negligible, allowing the use of this form of transactions by any individual or organization.

Auctions conducted on the Internet have gained such popularity that some researchers consider them as the only market-based mechanisms used to determine price and other product attributes (Applegate and Holsapple, 1996; Beam, Segev et al., 1996; Beam and Segev, 1998; Heck and Vervest, 1998; Teich, Wallenius et al., 1998). Auctions, because of their very small transaction costs and their potential to reach worldwide, are now being used to conduct many transactions among business and between businesses and consumers. Electronic commerce negotiations that are typical of business-to-business commerce and other transactions have not yet gained such attention. Recent developers, however, indicate growing interest in web-based business negotiations (Maes, Guttman et al., 1999; Thompson, 1999; Arpinar, Dogac et al., 2000; Benyoucef and Keller, 2000).

Because of the very large and increasing number of vendors on the web who sell products and provide services, as well as many competitive products, buyers' search time and effort also increases. This has led to suggestions that software agents could be used, since they are capable of conducting most of a buyer's or seller's activities. Software agent and decision support are rapidly developing information technologies due to their potential in supporting and conducting electronic transactions and other business activities. These technologies brought e-commerce to the next level – personalization.

Successful examples, such as web-based expert systems (e.g. Personallogic), shopping agents (e.g. BargainFinder, Jango), and recommendation systems (e.g. Amazon.com, InfoScout) have shown support for users can be developed in a personalized way.

Negotiation via the web is currently supported by several technologies, such as negotiation support systems, group decision support systems and negotiation software agents. Typically, and despite the fact that these technologies address different issues and can complement each other, they are used separately. Experiences from the INSPIRE system led us to suggest a web-based integrated software environment to aid negotiators throughout the negotiation process and to provide methodological support and advice (Lo and Kersten, 1999). We propose – as the first step in the development of an integrated negotiation environment – to aid negotiators with the use of software agents. The negotiation agent called Advisor to Inspire Negotiators (Atin), will be designed in order to provide active support to Inspire negotiators.

This work is based on research in three areas: business negotiation and negotiation support, electronic commerce and Internet technologies. We discuss business negotiations via the web in Section 2. Section 3 outlines some common negotiation difficulties and pitfalls. The InterNeg Project and the Inspire Negotiation Support System are introduced in Section 4. The support technologies which are used in the development of the negotiation agent (Atin) will be discussed in Section 5. Section 6 presents the key research objectives of this project. The system architecture and design methodologies are discussed in Section 7. Section 8 outlines the design issues and Section 9 discusses the

development aspects. Testing activities and user evaluation are presented in Section 10. Section 11 concludes this thesis report.

2. Business negotiations and e-commerce

This section discusses the fundamentals of business negotiations that are conducted over the Internet. Section 2.1 introduces electronic commerce (e-commerce), and outlines the generic business model for e-commerce. This is followed by a discussion of business negotiations in Section 2.2. Because of claims that auctions can replace negotiations or they are equivalent to negotiations in e-commerce (Beam, Segev et al., 1996; Kumar and Feldman, 1998; Strobel, 1999), we discuss the similarities and differences between auctions and negotiations in Section 2.3. Electronic negotiations and supported technologies are discussed in Section 2.4, with a brief review of existing systems.

2.1. E-commerce

E-Commerce is a business approach that addresses the needs of organizations, merchants, and consumers to cut costs of while improving the quality of goods and services and increasing the speed of service delivery (Kalakota and Whinston, 1996). Greenstein and Feinman (1999) define e-commerce as:

The use of electronic transmission mediums (telecommunications) to engage in the exchange, including buying and selling of products and services, involving transportation, either physically or digitally, from location to location.

CommerceNet (2000) provides the following definition:

E-commerce is the use of inter-networked computers to create and transform business relationships. The new paradigm of e-commerce is built not only on transactions but on building, sustaining, and improving relationships, both existing and potential.

In simple terms, e-commerce means “doing business electronically”. An early and well-known implementation of electronic commerce has been electronic data interchange (EDI) – a technique that supports the interchange of data between business organizations (Kalakota and Whinston, 1996).

Three types of e-commerce have been identified (Applegate and Holsapple, 1996; Treese and Stewart, 1998; May, 2000). The most popular is business-to-customer (B2C), i.e., electronic shopping which has changed many people’s purchasing habits in recent years. B2C e-commerce focuses on consumer applications that allow transactions and interaction between a company and the consumer over the Internet. The two other types, business-to-business (B2B) and intra-organizational business to administration (B2A), have a stronger impact on business organizations than the B2C type. For example, worldwide B2B commerce totaled more than \$600 billion in 2000, according to Forrester Research, and that total will nearly double in 2001, reaching more than \$1.1 trillion (Arnaut, 2001). Cisco is one of the best examples of successful B2B e-commerce (Cisco Systems, 2000). The company’s web site lets its customers and partners order online, as well as access service and support. Even B2C continues to grow at a fast pace – from \$53 billion last year to \$96 billion in 2001 (Arnaut, 2001).

Although the Internet opened a new channel for doing business that required changes in business models, the key activities of the value-chain remained the same. As illustrated in Figure 1, these activities involve search/attract, interaction, action and reaction.

Figure 1. The Commerce Value Chain (Tresse and Stewart, 1998)

While web and Internet technologies did not change the key activities or introduce new ones, they both have had significant impact on the way these activities are being conducted. For example, the first activity in the value chain is Search/Attract. A buyer identifies a need and he/she looks for a specific product that fulfills his/her need. There are search engines and shopping agents that “do the walking” for the buyer. For example, BargainFinder (Andersen Consulting, 1997) and Jango (2000) are systems that can take a product name as input, obtain price information from the web and perform price comparisons. From a seller’s point of view, the use of Internet marketing is to attract new and keep existing customers.

Once the buyer and seller “meet” on the Internet, they exchange information regarding price and other attributes of the product or service. This refers to the activities performed in the Interact component shown in Figure 1. The seller’s objective in the interaction process is to turn customer interest into orders. The exchange may be routine and not involve any discussion, as is the case with the use of the electronic shopping

baskets with a predefined set of options from which the buyer may choose. This exchange may involve interactions – as in this case of auctions and negotiations – which require both parties being active.

The Act component includes order processing, payment methods, and fulfillment of the order. These activities are supported by Internet commerce technologies such as electronic ordering systems, electronic payment systems, and secure electronic transactions (SET) (Kosiur, 1997; Loshin and Murphy, 1997; Treese and Stewart, 1998). The final component is React, which includes after-sale services, information about new products and customer support.

2.2. Business negotiations

As seen in the previous section, negotiation is a key activity in e-commerce. A negotiation is a decision making process by which two or more parties communicate and exchange ideas and offers with the intention of satisfying their needs by achieving objectives and changing relationships through conferring for agreement (Lax and Sebenius, 1986; Fisher, Ury et al., 1991; Raiffia, 1998). Negotiation is also defined as the process of communication between two or more parties aimed at resolving initial differences in preferences (Pruitt and Lewis, 1977).

There are two main types of negotiation classified by Walton and McKersie (1965), known as distributive and integrative respectively. In general, distributive negotiations mean the parties involved are sharing a “fixed pie”, this is also known as the

“win-lose” situation. The other type, named integrative negotiations, involves the creation of value and openness of information exchange that end up with a “win-win” situation.

Distributive negotiations often involve only one issue. Raiffa (1998) suggests that when we try to convert a single-factor problem into a multi-factor problem, we are moving from distributive negotiations towards integrative negotiations. However, this is not necessarily the case. The Inspire system (2001) uses a simple business negotiation case (Itex-Cypress) which contains multiple issues but we consider it distributive, unless some form of common value could be created by both parties. Kersten and Noronha (1999) argue that the difference between distributive and integrative negotiations lies not in the number of issues, but in the creation of value.

Three sequential phases to study business negotiations have been suggested in the literature (Graham, Mintu et al., 1994). They are the antecedent phase, the concurrent phase, and the consequent phase. These phases roughly correspond to pre-negotiation, conduct of the negotiation, and post-settlement (Kersten and Noronha, 1998a; Raiffa, 1998). The three phases and the key activities are presented in Figure 2.

Figure 2. The three phases in a negotiation

Pre-negotiation is the first phase of a negotiation, also known as the preparation phase. It refers to the initial period (prior to the exchange of any offers) when one prepares for the negotiation. Negotiators specify their needs (i.e. issue identification) and prioritize their goals in this phase. They may also identify their bottom line (i.e. reservation values) and best alternative to a negotiated agreement (BATNA).

The conduct of the negotiation is the process where two parties exchange offers and messages. The parties may have to make concessions and trade-offs, or revise their negotiation objectives. Communication between the parties enhances learning and may also build relationships between them. During this phase both parties learn about their opponent's needs and try to reach a favorable agreement.

The post-settlement phase refers to the period after the first compromise has been achieved. Parties may discuss details about the terms of agreement, such as shipping date and method. Raiffia (1985) suggested “decide first, invent later”, which implies that the parties come to a simple agreement and then a third party suggests outcomes that are better for each party than its initial agreement.

2.3. Are auctions negotiations?

Buyers and sellers use different transaction and coordination mechanisms to establish communication, exchange information and organize sale/purchase. The use of a particular mechanism depends on the number of buyers and sellers, and the number of attributes they consider in a product or service. In this section we discuss the use of negotiations, auctions, reverse auctions, and the marketplace via the Internet.

One buyer and one seller define a negotiation which may consist of one or more attributes (Teich, Wallenius et al., 1998; Kersten and Noronha, 2000). Computer-aided negotiation systems developed to provide analytical support to negotiators during web-based negotiations are discussed in the literature (Guttman and Maes, 1998a; Guttman and Maes, 1998b; Kersten and Noronha, 1999a). Examples will be discussed in Section 2.4. If there is one buyer and many sellers, then a reverse auction is appropriate, an example being a government auction (Teich, Wallenius et al., 1998). Priceline.com (2000) is another type of reverse auction for unused capacity on airlines. If there is one seller but many buyers, auction is introduced because of its ability to simultaneously manage large numbers of bidders (Teich, Wallenius et al., 1998; Kersten and Noronha, 1999a). There are a large number of well-known intermediaries conducting auctions, e.g., Ebay, Amazon, and Onsale. Buyers place their bids for a unique product, and price (normally the only attribute) is largely unknown. When there are many buyers and sellers, a marketplace is formed (Teich, Wallenius et al., 1998; Kersten and Noronha, 2000). In a marketplace, there are many products available, and buyers choose the

product that interests them. Everyone has at least partial information about price/cost and other attributes, and information is open to public.

	Negotiations	Auctions	Reverse Auctions	Marketplace
No. of Attributes	One or many	One	One	One or Many
No. of Buyers	One	Many	One	Many
No. of Sellers	One	One	Many	Many
Market Efficiency	Very Low	High	High	Very High
Cost	High	Low	Low	Very Low
Relationship	Yes	No	No	No

Table 1. Differences and similarities among negotiations, auctions, and marketplace

The similarities and differences among negotiations, auctions, reverse auctions, and marketplace are summarized in Table 1. Note that a compromise that is inefficient in the marketplace may be locally efficient (Pareto-optimal) in the particular negotiation process. This is because an efficient compromise is one which the parties cannot improve jointly. However, there may be deals on the market which dominate this compromise.

Auctions and electronic negotiations are considered an important part of e-commerce (Kersten and Szpakowicz, 1998; Maes, Guttman et al., 1999; Benoucef, Alj et al., 2001). Popularity, low transaction cost, and ease of use has led some researchers to consider auctions as the coordination mechanism that should replace negotiations, or even that auctions are the same as negotiations (Beam, Segev et al., 1996; Kumar and Feldman, 1998; Sandholm, 1999; Strobel, 1999). This is because Internet auctions provide new efficiencies made possible by virtual markets allowing customers from any place to easily join an auction.

Auctions are usually one-sided: a single seller (or buyer) considers offers (bids) from many buyers (or sellers). Negotiations, on the contrary, are always two-sided: a buyer and a seller exchange offers and information, and each of them considers the offer from the counterpart. Auctions are solely focused on outcomes, therefore the communication utilizes a low-cost process, with the aim at providing the outcome with the best value (Kersten and Noronha, 1999a; Kersten, Noronha et al., 2000). However, they do not assure Pareto-optimal outcome and business relationships. The communication process in negotiations involves both parties learning about each other's needs during their exchange of information. This is important since it may possibly lead to significant outcomes— a long-term relationship between business partners, or between a business and its customers (Kersten and Noronha, 1999a; Lo and Kersten, 1999; Lo and Kersten, 2000).

Another difference between auctions and negotiations may also be noticed when two or more issues (e.g. product attributes) are present (Kersten, Noronha et al., 2000). Auctions typically involve a single issue (i.e. price) while negotiations often involve multiple attributes. Many business transactions require the parties to specify the product or service attributes such as upgrades, return policy, delivery time, and type and quality of components. In these cases, the issue is not only to obtain the best price, but also to establish the terms of the transaction, and the features of the product or service. Negotiations also have the flexibility of adding and removing attributes (negotiable issues). Upon approval of both parties, new issues can always be added to the negotiation

table. Single-issue auctions do not have the flexibility and do not provide feature-rich mechanism required to conduct complex business transactions. There have been efforts in the development of multi-dimensional auction systems (Branco, 1997; Teich, Wallenius et al., 1999; Che, 1992). An example of a two-issue double auction is OptiMark (2000), an electronic stock trading system.

If we consider the time and effort required to achieve a solution, negotiation is a more costly process than an auction. Negotiations require active participation and rich communication between both parties: they involve learning, accommodating of positions, construction of alternatives and modification of constraints (Kersten and Lo, 2001). Low transaction costs, high effectiveness and the possibility of achieving efficient solutions may be the reasons for suggesting that auctions can replace business negotiations. However, it is often the case that the outcome of the negotiation is more than the negotiated product or service. In negotiations, price, and possibly other attributes are largely unknown since these issues remain private. The drawback is in efficiency but the advantages of negotiations lies in the building of relationships between partners and their potential for eventual improvement. Therefore, we suggest that for many transactions auctions cannot replace negotiations.

2.4. Web-based negotiations

The rapid adoption of e-commerce raises the need for systems that support electronic negotiations. The use of the Internet provides businesses and individuals access

to information from anywhere at anytime. This provides the possibility for businesses to globalize, given the opportunity to look for potential business partners all over the world. Although the web has created an efficient channel for business communication, web-based negotiations are considered to be more complex than normal face-to-face negotiations. This stems from the physical distance between the parties, the fact that they do not know each other and may be unable to find common relations, the possibility of different business practices, different cultures — all these contribute to the complexity of the interaction (Kersten and Szpakowicz, 1998).

There are several technologies that may improve the efficiency of electronic negotiations, and help overcome many traditional difficulties (which will be discussed in Section 3) by providing analytical aid to negotiators. These technologies will be outlined in Section 5, and discussed in depth in the rest of this thesis report.

Interestingly, researchers and developers, who consider commerce negotiation as an important form of business transactions, design and implement systems that have a minimal negotiation component and implement auction mechanisms (Guttman and Maes, 1998a; Guttman and Maes, 1998b; Kumar and Feldman, 1998; Sandholm, 1999; Sandholm, 1999; Morris and Maes, 2000). These systems, involving price and other products (or services) attributes which are pre-determined by the seller (or buyer), are more likely to be online auctions than negotiation systems. An example could be the software agent known as MarketMaker (which will be discussed in details in Section 5.2). We argue that this agent supports bidding rather than negotiations.

There are a few web-based systems in practice that support negotiations over the Internet, and provide the parties with tools for conflict resolution. CyberSettle (2000) is an online claim resolution system that supports its users in negotiating insurance claims over the web. LiveExchange (2000) developed by MOA Inc. and the EcommBuilder developed by TradeAccess (2000) provide process-oriented support to electronic negotiations, and are capable of handling multi-issue and multi-stage negotiations. However, all decision-making activities are left to the human negotiator who may want to have methodological support during a negotiation. Frictionless Commerce's PurchaseSource (Frictionless Commerce, 2000) software offers the buyer a personalized decision support system to determine what to buy. These systems provide negotiation support at the computational and cognitive levels.

3. Common negotiation pitfalls

In Section 2 a claim was made that in e-commerce negotiations will continue to be one of the more important mechanisms used in business transactions. With the use of the Internet to conduct value chain activities, the possibility of using negotiation support systems and software agents may increase. Experimental studies conducted in the InterNeg project indicate that there is a strong demand for negotiation support (Kersten, 1998a; Kersten, 1998b). In order to be useful and effective, support tools need to help users overcome the difficulties of negotiation. This is even more important in e-commerce because many people can now negotiate with more partners, including those with whom they previously could not interact, and because many people negotiate without having formal training or experience.

Business negotiations are often very complex, involving diverse issues and multiple options. For example, a case was developed based on a real-life negotiation between a health management organization and a hospital. The scenario contained 7 negotiable issues, 5 options per issue and 16,807 possible complete offers (i.e., offers with every issue present). The complexity increases if the problem is initially ill-defined, if the issues and options are added during the process, or if the negotiators seek integrative compromises. The real-life negotiation on which the case was built, took over 9 months to complete and required almost daily communication (InterNeg, 2000).

The next sections outline several negotiation pitfalls which need to be addressed in support systems and agents. These are some common mistakes negotiators usually make during the negotiation process, which are barriers for them to achieve successful outcomes.

3.1. Preparation and problem analysis

One of the significant pitfalls is the lack of preparation for the negotiation. The Inspire team has received one comment from a business manager:

“When negotiating service contracts with a Taiwanese manufacturer we have great problem. Often he does not understand what I mean and I do not know a clue of what he is saying. The language barrier is one, but the major handicap is lack of preparation. Inspire prepares small time company's like us better for global negotiations...”

Due to a lack of preparation, many negotiators do not understand their preferences and priorities, thus ending up either suffering from a weaker position in the negotiation, or bringing the negotiation to a standstill. When Fisher and Ury (1991) discuss power in negotiations, they suggest the better one's Best Alternative To a Negotiated Agreement (BATNA), the greater one's power is. This is true since when a negotiator prepares him/herself for a negotiation, he/she should understand the problem, identify the goals and defined his/her reservation level and BATNA.

3.2. Focus on positions

Fisher and Ury (1991) state that arguing over positions is inefficient, produces unwise agreements, and endangers an ongoing relationship. Many negotiators think that all negotiations are distributive: whatever one side gains the other loses. If a negotiator focuses too much on positions, he/she may become self-interested and may not be willing to make concessions.

The misperception that negotiation always involves sharing a fixed pie also leads negotiators to interpret a negotiation as a win-lose game. Research suggests many negotiators make such a mistake, and treat negotiation as warfare or a zero-sum game (Lax and Sebenius, 1986; Fisher, Ury et al., 1991; Ury, 1991). If one makes the opponent angry, it not only implies the negotiation will not reach a favorable outcome, but it may also destroy a potential long-term relationship.

3.3. Inefficient compromises

The acceptance of inefficient compromises and the misinterpretation of an opponent's priorities are phenomena that have been observed in many experiments (Kersten and Mallory, 1999). For example, approximately 2000 people have negotiated via the Inspire negotiation support system as of July 1998. Only 59% achieved compromises with their counterpart, and out of these only 41% achieved an efficient (Pareto-optimal) compromise (Kersten and Noronha, 1998a; Kersten and Noronha, 1999a).

The users enter the post-settlement phase when an inefficient (i.e. not Pareto-optimal) compromise is reached. The post-settlement mechanism in Inspire suggests jointly improved agreements, and allows the users to continue the negotiation in order to end up with higher satisfaction. Out of those who achieved an inefficient compromise, only 18% wish to continue and improve their results. In other words, 82% of the people who achieved an inefficient compromise did not want to improve it.

3.4. Socio-psychological factor

Although electronic negotiations are mostly conducted via electronic media (such as the Internet), they are not computer tasks but human and interpersonal activities. A negotiator's personality may cause biases and illusions during a negotiation, thus imposing obstacles to reaching a successful outcome.

Thompson (1998) suggests a few socio-psychological factors such as self-enhancement and egocentric judgment that are detrimental to successful negotiations. These factors may lead to different reactions (e.g. overconfidence, face-saving) that result in the failure of the negotiation. Research shows that negotiators tend to be overconfident that their positions will prevail if they do not "give in" (Walton and McKersie, 1965; Bazerman and Neale, 1982). For example, Bazerman and Neale (1982) show that negotiators in final-offer arbitration consistently overestimate the probability that their final offer will be accepted.

Bazerman (1985) found that the frame of buyers and sellers systematically affected their negotiation behaviour. Consider an example in which a negotiator defines his reservation value at price x , and aspiration level at price z . Suppose this person receives an offer at price y , that is between x and z . If the negotiator has a positive frame he would view the offer as a gain from his reservation value. If the person has a negative frame he would view it as a loss from his target price (aspiration value). Bazermann (1998) suggests that negotiators with a positive frame reach more compromises and obtain greater overall satisfaction than negotiators with a negative frame.

3.5. Avoiding negotiation pitfalls – agent-mediated negotiations?

In the preceding sections we outlined some of the common pitfalls in negotiations. These pitfalls may often be avoided if the negotiators are provided with help and support. People may make mistakes in a negotiation due to many factors, such as carelessness, inadequate experience, and over-confidence. This leads to the need to have a mechanism that is capable of providing extensive assistance to the negotiators, helping them to understand the situation and problem, their priorities, and to negotiate rationally.

Rubin and Sander (1988) suggest the use of skillful human agents in representative negotiations. One of the reasons to engage in this type of negotiation is that the agents have expertise that the principals lack, and they are more likely to make more favorable agreements. The agent can be a consultant or an advisor, who provides strategic advice and assists the principal during the negotiation.

There are negotiation software agents that represent the user and do the negotiation autonomously on behalf of the user over the Internet (Sandholm and Lesser, 1995; Guttman and Maes, 1998a; Guttman, Moukas et al., 1998; Moukas, Guttman et al., 1998; Maes, Guttman et al., 1999; Sandholm, 1999; Thompson, 1999; Morris and Maes, 2000). However, we argue that in such cases the learning opportunity for the principal is diminished, or an integrative solution is not likely to happen since it is difficult to develop a relationship between the two principals.

Therefore, instead of representative negotiations, we suggest the use of agents in direct negotiations – known as agent-assisted negotiations. The principals negotiate directly with each other, and each of them is supported by an agent. The agent can be a consultant or an advisor, who provides strategic advice and assists the principal during the negotiation. This led us to consider a system that would guide negotiators throughout the whole progress of the negotiation, and provide extensive support and advice whenever appropriate.

4. The Inspire negotiation support system

Recent work on negotiation support systems concentrates on web-based negotiation support tools (Kersten, 1998a; Kersten and Noronha, 1998a; Guttman and Maes, 1998a; Benbasat, Lim et al., 1995). Within the InterNeg project (InterNeg, 2000) two negotiation support systems called Inspire and INSS were developed and deployed on the web. The systems were used to study, teach, simulate and facilitate electronic negotiations. Section 4.1 introduces the InterNeg project and the Inspire negotiation support system. The supporting methodologies in Inspire are outlined in Section 4.2 and aspects of Inspire negotiations are discussed in Section 4.3. Section 4.4 concludes the discussion by studying the users' feedback and assessment.

4.1. The InterNeg project and Inspire

InterNeg (2001) is a research project that builds upon three emerging technologies: net-centric computing, decision and negotiation support, and software agents. It is designed to develop an environment that supports electronic negotiations over simple or complex problems, including real-life situations.

InterNeg allows negotiations in multiple countries by many people at the same time. The InterNeg site, and a prototype of one of its web-based negotiation systems, Inspire, have been operational since July 1996. As of September 2000, over 4000 people from sixty countries have negotiated through Inspire. The majority of the users came from countries such as Austria, Canada, India, Finland, Korea, Portugal, Russia, and U.S.

Because it is web-based, inter-cultural negotiations are carried out as easily as intra-cultural negotiations. The system can also be used at any place that has an Internet connection

The Inspire system (2001) has been used as a research tool for the InterNeg group to study cross-cultural negotiations over the web (Kersten and Szpakowicz, 1998; Kersten and Noronha, 1998a; Kersten and Noronha, 1998b; Kersten and Noronha, 1999a). It is also used to study the impact of decision analysis on the negotiation process, the role of support in negotiation and the role of explanatory and display facilities on users' perception and decision-making.

4.2. Inspire support methodologies

The decision support functions implemented in Inspire include preference elicitation, construction of the utility function, quantitative evaluation of offers, maintenance of the negotiation history and graphical representation of the negotiation dynamics. The communication support functions include the exchange of structured offers with accompanying arguments, free-text messages and automatic email notification of the opponent's activity.

An important feature of negotiations with the Inspire system is the structure of the process. Inspire supports the three stages of negotiation illustrated in the Figure 2. The negotiation progresses through three distinct phases: pre-negotiation analysis, conduct of

the negotiation, and post-settlement as discussed in Section 2.2. The support of Inspire in the three phases is illustrated in Figure 3:

Figure 3. Inspire's support in different negotiation phases

During the pre-negotiation phase, Inspire helps the user to better prepare for the negotiation. The activities include helping the user to understand the negotiation problem, the main negotiable issues and offers, and some possible combinations (which may form the basis of offers and counter-offers). The user defines his/her own preferences and the system takes the input from the user to construct the utility function.

The negotiation phase in Inspire may begin with the construction of an opening offer. There is a pre-defined format for offers – each offer contains user-selected options (issue values) for each of the negotiable issues. An offer may be accompanied with a free-text message, which allows the users to communicate directly. Inspire provides a numeric rating for each offer sent/received, which represents the “goodness” of the offer. This rating is calculated based on the user's utility function. Users may also review their negotiation history, or review and revise their preference ratings during the negotiation phase. A graph displaying the dynamics of the negotiation is also available.

Once a compromise is achieved, the Inspire system acts as a mediator and checks for its efficiency (Pareto-optimality) (Kersten and Noronha, 1999a). The system takes into account both users' utility functions, and determining if any further improvement can be made to the agreement. If the compromise is inefficient, the system computes efficient packages and displays a few to the both users, which allows them to re-negotiate.

4.3. Inspire negotiations

The Inspire users negotiate with anonymous partners over the web with the system providing support such as graphical and numeric analysis. Participants can remain anonymous, not knowing their counterpart's identity, or may reveal their identity through the message facility.

Inspire bilateral negotiations are conducted over a simple case of business-to-business transactions. Typically they take up to three weeks. The process may result either in a compromise, the parties may choose to conclude the negotiation with no compromise, or one party may terminate negotiations at any time. This last possibility allows users to reject offers that are bad for the companies they represent. The user who terminates his/her negotiation may immediately request a new negotiation and is assigned another partner.

4.4. Inspire use and assessment

Upon completion of Inspire negotiations, users are requested to fill in a post-negotiation questionnaire. One of the questions refers to the users' overall assessment of the system; they are asked if they would use a similar system in real-life negotiations, to prepare for a real-life negotiation, or as a practice tool to improve their skills. The results are shown in Figure 4. Generally, users find the system very easy to use, and their evaluation of the overall system is favorable. Over 60% of Inspire users stated that they would use a system like Inspire in real-life negotiations and over 80% would use such a system to prepare themselves to conduct actual negotiations (Kersten, 1998a; Kersten and Noronha, 1998a). While the feedback on the Inspire system conforms to our expectation, the absolute levels of user acceptance of the system are surprisingly high. These results led us to assume that a system like Inspire would be acceptable in e-commerce negotiations.

Figure 4. User Acceptance of Inspire (1410 respondents)

5. Web-based negotiation technologies

We discussed web-based negotiation systems in Section 2.4 and introduced one of the first web-based negotiation support systems, Inspire, in the previous section. Negotiation via the web is currently supported by several technologies, such as negotiation software agents, group decision support systems, and negotiation support systems.

This section discusses some of the above technologies that are relevant to this project. We introduce software agents in Section 5.1, followed by a discussion on negotiation software agents in Section 5.2. Negotiation support systems will be introduced in Section 5.3.

5.1. Software agents

Software agents have become one of the hottest research topics in the fields of artificial intelligence, software development and electronic commerce, because of its potential to automate previously manual operations. Agent research and development has provided many definitions of these systems. These agents are programs to which one can delegate a task. Maes (1998) provides the following definition:

Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed.

The IBM Glossary of Computing Terms provides the following definition (IBM, 2001):

Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires.

Although there is no a universally accepted definition of the term “software agent”, there is a general consensus about some of the attributes of a software agent. In general, an agent is a computer program that is situated in some environment; it is continuously active, capable of autonomous action (either proactive or reactive), and work on tasks on behalf of its user (Caglayan and Harrison, 1997; Gazis, 1998; Jennings and Wooldridge, 1998; Wooldridge, 1999). These programs differ from regular software since they are personalized, continuously running, and autonomous to an extent.

There are different characteristics of software agents. A software agent system may consist of one agent or more, depending on the requirements of the environment. A single agent system is one in which the problem domain is encapsulated within the system, and is not designed to interfere with other agents (Greenstein and Feinman, 1999). However, a multi-agent environment necessitates cooperation between agents (Haugeneder and Steiner, 1997).

The state from which an agent executes its commands — i.e. either from a single host or from a remote site — distinguishes agents as being either mobile or static. Static agents use embedded knowledge to assist in filtering incoming information (Bradshaw, 1997; Caglayan and Harrison, 1997). Examples of this are e-mail agents that parse only

relevant information from a message to the user. Mobile agents are able to communicate between other agents and the host (or user) and execute various commands (or perform tasks) to meet their objectives. Examples of this could be shopping agents that go to different web sites and perform price comparisons for the users.

The reasoning mechanisms of software agents can range from a set of simple “if-then” rules to sophisticated machine learning algorithms such as neural networks or Bayesian networks (Bradshaw, 1997; Caglayan and Harrison, 1997). Rule-based agents have internal knowledge bases that contain information and rules they should understand in order to perform certain tasks on behalf of users. The degree of personalization of agents depends on the database. The more information the agent receives from the user, the more personalized it can be.

According to Thompson (1999), software companies are revolutionizing e-commerce by creating agents that compare, buy and, sometimes, even sell products. Wang (1999) discusses the use of software agents in electronic commerce. He suggests software agents may have roles in assisting a buyer in purchasing, including a purchasing rule agent, merchandise search agent, negotiation agent, distributed inventory control agent, and accounts payable agent. From a seller’s perspective, agents may also match the customer’s profile with catalog offerings and to provide personalized experiences.

Notification agents alert users of significant events where an event is defined as a change in the state of information (Caglayan and Harrison, 1997). For example,

Amazon.com (2000) uses notification agents to inform users when new books become available. URL-minder (2000) is another notification agent that retrieves web resources periodically to detect changes from the last retrieval.

Recommendation agents are used to make predictions and recommendations based on user profiles and business intelligence. For example, Barnes&Nobles.com uses an agent to provide recommendations to customers based on the users' previous purchases (Barnes&Nobles, 2000).

Andersen Consulting developed a shopping agent called BargainFinder, where agents could comparison-shop for music CDs from among eight on-line CD retailers that had agreed to participate (Andersen Consulting, 1997). However, the agents were blocked by the retailers who they realized that using price as the only criterion could be hazardous to their business (Thompson, 1999). This raised the attention of multi-criteria business transactions, such as negotiations. Frictionless Commerce (2000) introduced a software agent that compares different attributes of a product.

5.2. Negotiation software agents

As discussed in the previous section, software agents can be used to automate different negotiation tasks including those involved in buying and selling products over the Internet (Guttman, Moukas et al., 1998; Maes, Guttman et al., 1999; Thompson, 1999; Morris and Maes, 2000). Business negotiations are a more complex process and the capabilities of agents to perform negotiation tasks are being researched (Beam and Segev,

1997; Faratin, Sierra et al., 1998; Kersten, 1998a; Kersten, 1998b; Kersten and Noronha, 1999b; Lo and Kersten, 1999; Maes, Guttman et al., 1999; Sandholm, 1999).

At any given time organizations and customers may be engaged in several different negotiations. The effort and time required to conduct negotiations led to the development of software agents that are capable of automating a significant part of the process. These agents, known as negotiation software agents (NSA), engage in a bargaining process that is characterized by several parameters.

Examples of negotiation-based systems are MarketMaker (previously known as Kasbah) (2000), AuctionBot (2000), and Tete-a-tete (2000). The purpose of these agent-systems is to find mutual agreements on the terms of transactions that satisfy the parties' constraints, preferences and objectives. However, some of these systems are based on theoretical frameworks that are only partially relevant to negotiations. Despite the claims made by the developers, the use of negotiation methodologies is often over-simplified and reduced to one form of negotiation or auction (Kersten and Noronha, 1999a). These "auction" agents might provide a wrong perception to users that NSA are not capable of engaging in, and supporting, context rich and complex negotiations.

One of the better-known systems, MarketMaker, is a multi-agent system developed at the MIT Media Lab, which facilitates auctions in an electronic marketplace. A seller may post a product for sale through the selling agent. Interested buyers post their bids with the help of their buying agents. Both parties define their desired and worst

acceptable price, as well as the slope for making concessions to their agents at the initiation stage. The agents submit bids and monitor the negotiation process, however, the human user makes the final decision. MarketMaker supports web auctions rather than negotiations; the system is rigid and allows for only single attribute transactions – price. The communication process is thus very narrow. Instead of exchange of negotiation offers and information, the agent posts a new bid (upon the approval of the user) once the market information is updated.

From our point of view, negotiation software agents may take over some activities in a negotiation but it not necessary for agents to handle all the tasks. For example, the agent may either negotiate on behalf of the negotiator (based on pre-determined user specifications) or assist the negotiator throughout the negotiation. We suggest that the negotiation software agent (NSA)'s negotiation strategy should not be eliminated or neglected, as this is the key aspect which differentiates among NSAs and other e-commerce agents.

Kersten and Noronha (2000) propose negotiation software agents to provide information and knowledge (e.g., statistics and inferences) about past negotiations, search through the negotiation transcripts and other process descriptions, and compare situations, interests and issues of past problems against the current problem. These agents may also receive knowledge from various sources, such as other agents, the environment, user input and databases, then interpret and understand that knowledge and intelligently

used this information to assist the negotiator throughout the negotiation processes (Torsun, 1995).

The NSA represents its user and makes offers and counter-offers based on parameters values supplied by the user, i.e., the agent's principal (Beam and Segev, 1997). The possible functions of such agents largely depend on their degree of autonomy, the type of the negotiation, and the specificity of the principal's directives. The functions depend also on the agent's interactions with other systems and agents. The agent may be highly specialized and co-operate with other agents, interact directly with the principal, or it may communicate via a decision support system (DSS) or a negotiation support system (NSS) that supports the negotiators in the construction of problem representations, in their assessment and modification, suggest new issues/options, and innovative (for the principal) approaches to cope with conflict.

5.3. Negotiation support systems

A negotiation support system (NSS) is software designed to support complex negotiation activities, improve negotiation efficiency and effectiveness (Bui, 1994; Holsapple and Whinston, 1996; Kilgour, 1996; Kersten, 1998a). NSS support ranges from systems that help negotiators prepare for a negotiation, to mediation and interactive systems that restructure the way negotiations usually take place (Rangaswamy and Shell, 1997). The foundation of NSS is decision and negotiation analysis (Lax and Sebenius, 1986; Raiffa, 1998). Negotiation analysis integrates decision analysis and game theory in

order to provide methodological support to users. Negotiation analysis is aimed at bridging the gap between descriptive qualitative models and normative formal models of bargaining. This approach adopted a number of behavioral concepts (e.g., reservation values, BATNA, integrative/distributive negotiations and principled negotiations) and incorporated them into quantitative models (Kersten, 2000). This allowed advisors to conduct formal analysis of negotiations and provide support.

Rangaswamy (1997) introduces two types of NSSs, they are preparation and evaluation systems and process support systems. Preparation and evaluation systems operate away from the bargaining table to help individuals privately organize information, develop preferences, refine pre-negotiation strategies, or evaluate negotiation offers. Process support systems operate at the bargaining table; the systems are designed not only to assist parties in gaining a subjective representation, but also to help negotiators move toward more integrative settlements (Thiesen and Loucks, 1992). Process support systems could be further broken down into two types: mediation systems and interactive bargaining systems. Mediation systems act as a mediator to prompt parties towards jointly optimal agreements. Interactive bargaining systems simultaneously support the negotiation process of all parties and enable the parties to communicate directly with each other over computer networks. Interactive bargaining systems may also have a mediation function. The Inspire system (discussed in Section 4) is an example of interactive bargaining system. As discussed in Section 4, the system supports three

distinct phases in a negotiation, and when the parties enter post-settlement the system will act as a mediator to seek opportunities for joint improvements.

An NSS comprises two components: a *decision support* component and *communication support* component (Benbasat, Lim et al., 1995). The decision support component enhances the information processing capabilities of the negotiators. These systems help the user to structure ill-defined problems through decision analysis methodologies. The communication support component facilitates the exchange of offers and arguments thus decreasing the time to settlement and increasing satisfaction with the results.

NSSs typically emphasize support, rather than automation. For example, Inspire elicits user preferences and allows negotiators to send offers electronically via the web. During the negotiation, Inspire aids the negotiator by representing their offer as a numeric value – which is the utility of the proposed offer. This numeric value or rating, gives a general overview of an offer being sent by giving the negotiator a quantitative representation of how “good” the offer is. During the post-settlement phase, Inspire can suggest alternatives which are mutually beneficial to both parties.

NSSs are designed to facilitate the various phases of the negotiation process such as understanding the negotiation case, assigning preference ratings for negotiable issues and options, and setting the reservation level before the negotiation begins. The tools for support are varied and they include decision science methods (e.g., decision tables,

decision trees, multi-attribute utility theory), statistical methods (e.g., forecasting, regression analysis), and game theory.

Increasing attention has been paid to systems that support web-based negotiations in recent years, although most of them have not yet been deployed. The Mediator (2000) uses case retrieval and adaptation to propose solutions (best possible agreements) to international disputes. PERSUADER (Robotics Institute, 2000) integrates case-based reasoning and game theory to provide support to industrial disputes. CyberSettle (2000) supports simple disputes that involve only one single issue and two parties. Bellucci and Zeleznikow (1998) have also introduced a negotiation decision support system (NDSS) which supports negotiation in a similar way to that of NSS. NDSS can also propose sample settlements by interpreting the goals and needs of the parties and by analyzing offer history.

6. Research objective

As discussed in the preceding sections, negotiations can be simple and routine, but often they are novel, complex and difficult. Therefore different approaches and architectures may be required to provide negotiators with support and advice. Some negotiations may be fully automated and conducted by an NSA, other supported with an NSS, and yet in other negotiations an NSS+NSA environment may be necessary. This research falls into the integrated NSS+NSA environment.

Some researchers, as we discussed in Section 5.4, foresee complete automation of negotiation processes over the web. Based on the review of both behavioral research and e-commerce literature, we believe that software agents are needed to support web-based negotiations and that there will be a need for human intervention. The agents will be required to provide advice to the negotiator and to help them achieving favorable outcomes. Certain routine activities may be automated, but not the key negotiation activities. This is because with complete automation human and organizational learning may disappear.

On the extreme spectrum of negotiation support is complete automation. The use of software agents for this purpose has been proposed (Maes, 1998; Maes, Guttman et al, 1999; Wang, 1999). A less extreme and more realistic approach would be using NSA to provide a differing level and scope of advice. These agents could also conduct autonomously such activities as alerting users, providing suggestions, based on

negotiation methodology. However they would be acting within the boundaries defined by the negotiator.

The rationale for this research is based on the following five observations discussed in detail in the preceding sections.

1. Business negotiations are already conducted over the Internet and more negotiations will be conducted by managers who are not professional negotiators.
2. The increased use of Internet and web-based technologies for communication and business transactions make web-based support easier than ever before.
3. An increasing number of business negotiators is willing to use support tools.
4. Available technologies, especially those utilized in NSS and NSA allow for provision of comprehensive support and advice.
5. NSS and NSA can effectively help negotiators in handling complex and difficult processes and help the negotiator to negotiate rationally.

The goal of this research is to:

Develop a software agent, integrate it in a negotiation support environment, and test its use in negotiations conducted by people via a web-based negotiation support system for training purposes.

This goal can be decomposed into the following four objectives:

1. Formulation of a framework for an interactive computing environment comprised of software agents and negotiation support systems;
2. Specification of an architecture for the agent development;
3. Construction of the negotiation software agent; and
4. In-house testing of the integrated environment.

The proposed framework will integrate knowledge-based systems, databases and web-based decision support within a network-centric computing paradigm. The negotiation agent, called *Advisor To Inspire Negotiators* (Atin), will be interacting with an appropriately modified Inspire negotiation support system. We call this new integrated Atin+Inspire environment *Aspire* (Atin-Supported Program for Intercultural Research Experiments). *Aspire* will allow us to test Atin's usefulness and to conduct preliminary comparisons with Inspire negotiations. In-house testing of *Aspire* will be carried out to ensure the quality level and the requirements are met.

To achieve the above objectives different architectures and technologies used for the development of software agents will be compared leading to the selection and development of e-commerce negotiation workbench. The results of this research will allow for detailed experimental studies on negotiators' behavior and the comparison of the effectiveness of different support environments. Although these studies are not in the scope of this project, in Section 11 we suggest several research experiments which can be

undertaken with the use of Aspire. This can be done by measuring the impact of the agents on the achievement of compromises and the efficiency of the achieved compromises.

Comparative studies can be carried out using both the Inspire and the Aspire support environments and comparing the impact of the Atin agent on negotiation effectiveness and users' satisfaction with the process and results. The evaluation can be done with the use of online questionnaires, similar to those currently used in the Inspire negotiations.

7. Methodology

The focus of the project described in this thesis is the development of Atin, a negotiation software agent. This will require modification of the existing Inspire system so that the negotiator's activity is accessible by both Atin and Inspire. In this section, we discuss the development methodology for the negotiation software agent (Atin). In some cases, when we refer to the whole system architecture, we consider Atin as a component of the whole system, Aspire.

The system development methodology and approaches are discussed in Section 7.1. The Aspire system architecture is discussed in Section 7.2. Section 7.3 gives an overview of the new integrated system (Aspire) framework. Section 7.4 outlines the functional requirements of Atin and non-function requirements are listed in Section 7.5. Section 7.6 presents examples of use case models.

7.1. System development methodology

System development methodology provides guideline to follow for completing every activity in the system development life cycle (Satzinger, Jackson et al., 2000; Fournier, 1998; OOTC, 1997; Jalote, 2000). However, no single methodology can satisfy all software development projects. The methodologies used in this project are being custom-tailored to accommodate the system's needs.

Several system development methodologies were reviewed when we considered alternatives. After considering the scope of this project, we utilized and customized the IBM work product and workbook-centered development approach (OOTC, 1997), Capability Maturity Model (Bemberger, 1997; Paulk, Weber et al., 1995; Jalote, 2000), and rapid application development methodology (McConnell, 1996) for this project.

IBM's Object-Oriented Technology Center uses an iterative and incremental process, but organizes it around work products such as documents, models, and software. It also uses a structured workbook to ensure that the work products are available and organized for access by the entire project. Work products are seen as the primary means of project communication. The reasons for selecting the IBM approach for this project are: (1) its support of object oriented development (Inspire's implementation is object-oriented), (2) flexibility and extensibility of the method, (3) easy to learn and intuitive, and (4) the InterNeg team has followed this development methodology since 1997 and is thus familiar with the approach.

Every software development project goes through a project lifecycle, which consists of stages such as requirement analysis, design, prototypes, implementations, and testing. The IBM approach takes the results of above planned project activities, and considers them as work products (OOTC, 1997).

The Capability Maturity Model (CMM) for software developed by the Software Engineering Institute (SEI) is also reviewed (SEI, 2001). CMM is a framework that

focuses on processes for software development, and describes the principles and practices underlying software process maturity. The five levels defined in CMM are summarized in Table 2 (Jalote, 2000):

LEVEL	FOCUS	DESCRIPTION
1. Initial	Heroics	Ad hoc activities; dependence on the heroic efforts and skills of key individuals.
2. Repeatable	Project Management	Each project has a well-defined software life cycle, but different models are used for different projects; success is predictable for similar projects.
3. Defined	Engineering Process	Uses a documented model for all activities; model is customized at the beginning of each project
4. Managed	Product and Process Quality	Metrics are defined for activities and deliverables; data is collected during the project to quantify progress
5. Optimized	Continuous Improvement	Measurement data are used to improve the model

Table 2. The five levels of the CMM (Bemberger, 1997; Paulk, Weber et al., 1995; Jalote, 2000)

In level 1, a project is executed in a manner that the team and the project manager see fit. Level 2 applies to scenarios where project management practices are established. The software processes for the organization is precisely defined and regularly followed in level 3. Quantitative understanding of the process capability in level 4 enhances the prediction and control of the process performance. At level 5 focuses on continuous improvement of the software process, including process change management, technology change management, and defect prevention (Jalote, 2000).

Although CMM is aimed to support the software process of an organization and to plan process improvements, the methodology is applicable to this software development project since this proposed project management processes broadly cover its

essential aspects. The approach consists of 3 stages: (1) planning, (2) execution, and (3) closure. Project planning entails all activities that must be completed before the execution of the project, such as requirement analysis, project estimation, and risk management. The project execution phase focuses on monitoring and controlling the execution, such as high level design, development, system testing, peer review, and project activity tracking. The project closure phase serves as a final review, ensures that experiences and lessons learned in the current project can be properly captured for future projects.

Based on the above methodologies, we have identified the six key processes: (1) project planning, (2) requirement specification and analysis, (3) design, (4) implementation, (5) testing, and (6) project closure.

Project planning includes identifying a sequence of necessary steps that a project can follow, such as project milestones. Rapid application development methodology (McConnell, 1996) is reviewed, and our project planning approach is discussed at the end of this section.

Requirement specification and analysis represents the application requirements of the system. Requirements can be categorized as functional or non-functional (OOTC, 1997). Functional requirements express action the system should perform, and the non-functional requirements are specifications that do not relate to the system's features but relate to its compatibility, usability, performance, etc (Conallen, 2000). The requirement specifications attempt to describe the system to be built. Functional requirements are

discussed in Section 7.4 and Non-functional requirements are outlined in Section 7.5. Use case modeling is used to represent top-level functional requirements, and is presented in Section 7.6.

Design describes the structure of the system to be built. According to Conallen (2000), two major activities of designing web-based systems are different from designing other software systems, which are: (1) partitioning of objects onto the client or server side, and (2) defining web page user interfaces. Proper positioning of the business objects is critical and highly depends on the architecture. The N-tier system architecture is presented in Section 7.2 and the conceptual system framework is discussed in Section 7.3. Detail design aspects are discussed in Section 8; such as the user interface model that documents the design of the application's user interface, sequence diagrams, and database structure design.

Implementation is the working applications and work products required to build them. In other words, it is the process of mapping requirements and design into codes and components. This phase consists of work products such as tools selection, coding guidelines, development environment, and source code. The detailed development process is outlined in Section 9.

Testing is an integral part of the project lifecycle. The IBM approach suggests the use of test cases to assert and verify the correctness of code to ensure its required quality level. Jalote (2000) proposed to use two steps of testing in the Capability Maturity Model

(CMM): (1) system test planning, and (2) system testing activity. The work product for system test planning is the test plan (listed in Section 10.1) that consists of the test environment and features to be tested (test cases). A test case gives the condition that needs to be tested as well as the expected results. The work product for system testing activity is a system test report that specifies whether the system behaved as expected.

Once system testing is completed, we arrange user-acceptance testing before project closure in order to receive feedback from end-users. Project closure is the stage where the system is accepted by end-users. Closure analysis is an important activity since it is the key for process improvement in the future. It consists of a review of lessons learned in the existing projects.

The rapid application development (RAD) approach is also used in this project. We put together two RAD techniques, (1) evolutionary prototyping, and (2) staged delivery, to obtain the project lifecycle development plan (Figure 5). Evolutionary prototyping is a life cycle, which the system is developed in increments so that it can readily be modified in response to end-user and customer feedback (McConnell, 1996). Staged delivery refers to the fact that the software is developed in stages, usually with the most important capabilities being developed first (opt. cit).

The integration of the above techniques is suitable for our development needs because the specifications of each stage are affected by the contents in the knowledge bases (which is another research project, discussed in Section 11.1). Using this approach

allows the developer to (1) put useful functionality into the system and test it before it is completed (2) receive feedback early enough to make necessary changes in the stages, (3) reduce the estimation error by allowing for recalibration and re-estimation after the prototype delivery at the end of each stage, and (4) minimize the final integration problems

Figure 5. Lifecycle planning for Atin development

Figure 5 presents the lifecycle planning for the development. Once the problem statement (i.e., to construct a negotiation agent) has been identified, the preliminary concept of the system can be formulated. This allows defining the requirements of the system, such as the roles of Atin. Requirements are important because they provide a scope and boundary for the later activities. During the requirement analysis phase specifications of the features of the system, its components, and most suitable

development tools and supporting technologies are specified. It is then followed by the architectural design of the system.

After completion of the analysis and design phases, implementation and prototype delivery is undertaken in three stages. Each stage consists of detailed design, coding, and testing. The black box testing methodology will be used since it focuses on the functional requirements of the system (Pressman, 1992). This technique attempts to find (1) incorrect or missing functions, (2) interface errors, (3) errors in data structure or database access, (4) performance errors, and (5) initialization and termination errors. These will be assigned as test cases in the test plan (Section 10.1). Once all the stages are completed, test cases are used to assert and verify the correctness of code, and the fulfillment of the system requirements. The three delivery stages follow Atin's required functionality in the pre-negotiation, conduct of negotiation, and post-settlement phase.

7.2. System architecture

Traditional client/server systems consist of three parts: a client, a server, and a protocol that bridges the gap between the client and the server tiers (Chaffee, 2000; Fournier, 1998). A three-tier architecture consists of a client tier that handles frontend interactions with the user, a backend tier which provides access to dedicated components (e.g., database server), and a middle-tier component (e.g., HTTP server) that allows users to share and control business logics by isolating it from the actual application (Fournier, 1998; Orfali, Harkey et al., 1999; Chang and Harkey, 1998). Conceptually, the three-tier

architecture can be extended to n-tiers (Chaffee, 2000). This requires the expansion of the middle-tier into several separate servers which provide connections to various types of services, integrating and coupling them to the client, and to each other. In other words, n-tier design separates the architecture of the application into as many tiers as one would need.

N-tier architecture provides the following advantages (Chaffee, 2000):

1. Clear separation of user-interface-control and data presentation from the application logic.
2. Support multiple applications more easily
3. Common protocol/API
4. Modular design therefore components can be re-used

The Aspire integrated negotiation support environment comprises the modified Inspire system, agent application development, knowledge bases (KB) and databases. The conceptual design follows the n-tier architecture. It involves: the web client, the web server, the application server (consists of the NSS, NSA, and KB), and the database server, as shown in Figure 6.

Figure 6. N-tier view of Aspire.

7.2.1. The web client

The first tier is also known as the client tier, which is basically the web browser. This tier is responsible for the presentation of information, receiving user instructions and controlling the end user interface. This comprises the interface of the Aspire system (NSS + NSA), implemented through web pages.

7.2.2. The web (HTTP) server

The HTTP server tier consists of an Apache web server with server-side technologies such as PHP (PHP hypertext pre-processor) and JavaScript, and CGI/Perl. This tier includes server-side scripts to process the application requests sent from the browser.

7.2.3. The application server

The NSS, NSA, and knowledge base together form the application tier. The NSS server consists of the Inspire engine, which is a collection of programs in C++ and are executed on the Inspire host itself and invoked via the web server and the CGI protocol.

The NSA server contains all the components of the Atin – a reasoning engine, a parser, processing component, and application programming interface (API) to the database server. The reasoning engine contains reasoning logic used to process the rules (from the knowledge base) and data (from the database) (Bigus and Bigus, 1998). The parser is used to convert the incoming information to the agent-readable format. The processing component consists of applications that process user request, update user information, etc. API acts as intermediaries between the NSA and the database server.

Atin needs to access facts and rules through its knowledge bases. Negotiations rules in the knowledge bases are generated from the user database (the database is discussed in 7.2.4.), literature research, experienced negotiators, and the feedback from Inspire users. Data are made available through 2500 usable cases of Inspire negotiations. The knowledge base built from the data is used to teach the agents how to negotiate, by giving the intermediate offers sent for successful and unsuccessful negotiations. Wirth and Shearer (1997) proposed that knowledge discovery is used to extract new rules and patterns from a database. Knowledge is then derived from Inspire users' experiences (Mahadevan, 1999) and literature on negotiations (Lax and Sebenius, 1986; Ury, 1991).

7.2.4 The database server

The database server tier consists of a relational database server responsible for data storage and retrieval. During a negotiation, all the user activities (e.g., offer submission) and information (e.g., reservation level, BATNA) will be stored in the user database. These facts will be useful when the agent respond to a user query or provides the user with strategic moves. The agent will acquire information about the current user from the database, and search the entire meta-knowledge base for appropriate rules. The agent may then either perform negotiation analysis or make suggestions.

7.3. Aspire framework

The Aspire system is an integration of a negotiation support system (Inspire) and a negotiation software agent (Atin). As discussed in Section 4.2, the Inspire system provides quantitative negotiation analysis and communication support for the negotiators. Section 6 explained there are some missing support features in Inspire which could be fulfilled by the use of negotiation software agents. Figure 7 presents the complete support in Aspire.

Figure 7. Aspire support in the three negotiation phases

The information described in Figure 7 is similar to Figure 3 (p. 25), except by the addition of the Atin support. As seen in the figure, the Inspire system emphasizes on negotiation analysis and numerical support – e.g., a numeric rating that represents user's utility value of an offer, the NSA support focuses on negotiation methodology and explanation - e.g., allow the user to select his preferred negotiation style and explain how he/she should approach the negotiation process.

The negotiation process in Aspire consists of the following aspects:

1. User

Aspire negotiations involve two users who need not to be physically present in the same location. Each of them negotiates using Aspire via a web browser. User

involvement is implemented through the user interface, and assisted by the negotiation software agent.

2. Negotiation Content

Negotiation content includes issues, options, possible packages, reservation values, BATNA, and utility ratings.

3. Negotiation Outcome

The negotiation result can be divided into 3 categories: (1) no agreement reached before deadline, (2) efficient agreement, and (3) inefficient agreement. If the users reached an inefficient agreement, they have the option to improve the achieved compromise in the post-settlement phase. Aspire's post-settlement analysis will present a list of better packages for them to continue the negotiation.

In the previous sections, we discussed the nature of negotiations and some of the common pitfalls. In Section 2.2, we stated that creation of values between the parties lead to integrative negotiation outcomes. Creation of values requires rich communication and learning (better understand the product/situation), where both parties have to discuss openly to achieve a win-win solution. The message engine in the Aspire system supports communications between the parties, negotiators may send an offer accompanied with a message, or a standalone message to their partners. Furthermore, depending on the user's

selected negotiation style, Atin may suggest users with the context of message during the negotiation.

In Section 3.1, we discussed some negotiations failed due to lack of preparation. Since the preparation (pre-negotiation) phase is very important, Aspire provides support to the negotiator and guides him/her throughout the process, from problem identification, issue and option rating, to reservation values and BATNA settings. This process (with agent support) helps the negotiator to better understand the negotiation problem and identify the priorities. If the negotiator makes a mistake during the process, Atin will alert the user with the appropriate advice.

Section 3.2 and 3.4 outlines some human factors that are detrimental to the negotiation outcome, such as being over-focused on positions, self-enhancement, and egocentric judgment. Negotiations that are brought to a standstill due to face-saving of both sides were one of the typical examples in Inspire (2001). Some negotiators refuse to make concessions during the negotiation and make their partners angry. We attempt to avoid this occurrence by using Atin as an advisor – when Atin notices that a user has no change in position in consecutive offers, it suggests the user that to make a small concession and also explain the rationale behind the suggestion.

We discussed that some users who reached inefficient compromises refused to use Inspire's post-settlement mechanism in Section 3.3. We attempt to increase the use of post-settlement mechanism with the aid of Atin. If a user reaches an inefficient

compromise and refuses to use post-settlement, Atin will explain to the user about this feature and suggest him/her to jointly improve the achieved compromise with his/her counterpart via the post-settlement mechanism.

In Section 3.5, we suggested the use of agents in direct negotiations. This forms the concept of the Aspire integrated negotiation environment, which consists a negotiation software agent (Atin), the Inspire negotiation support system, and other systems (Lo and Kersten, 1999; Lo and Kersten, 2000). Inspire plays the role of the frontend, interacting with the user as well as supporting information exchange (e.g., offers, messages) with the counterpart. Atin is designed to provide the negotiator with a full range of methodological support, such as assessment of the negotiator's activity, suggestion of possible moves, and answers to the negotiator's questions. Atin acts as an advisor throughout the negotiation process, while the user has full control on the negotiation. If the user makes a mistake, Atin interfere and warns the user. On the other hand, the user may ask Atin a question during the negotiation.

From the user's point of view, Atin may be considered as a newly added feature to the INSPIRE system because of its extension of Inspire's functionality. The agent acts as a negotiation assistant (advisor) to Inspire negotiations and adds value to the current system. However, from the developer's perspective, Atin is a standalone system that continuously interacts with the Inspire system and the user via the exchange of information. This loosely coupled architecture provides flexibility, allowing the

replacement of Inspire with a different NSS, also addition of more NSAs, and changes in the scope of NSAs activity and their level of independence without affecting the NSSs.

Figure 8 The Aspire environment: A conceptual view (Lo and Kersten, 2000)

The conceptual view of the whole system, including two users negotiating via the Aspire system is presented in Figure 8. The Aspire system resides on the InterNeg web server, and the users communicate with their own copies of Aspire (frontend) via a web browser. The Aspire system consists of (1) an individual negotiation support component, (2) a joint negotiation support component, (3) a message engine, (4) a database, (5) knowledge bases, and (6) a negotiation assistant.

The individual and joint negotiation support components, the message engine, and the user database all-together form the regular version of Inspire. The individual support component includes the Inspire features that support the individual negotiator. These

include preference specification, utility construction, and negotiation history representation. The joint negotiation support component is the post-settlement mechanism in Inspire. This component handles the mediation features, and therefore it is assessed by components handling the individual negotiator's activities. The message engine handles communication and information exchanges between the two negotiation parties. An email is sent to the negotiator's email account once he/she receives an offer/message from the opponent.

The negotiation assistant presented in figure 8 is the Atin negotiation software agent, which retrieves information from the database and knowledge base, and provides advice to the negotiator. Atin provides suggestions to the users based on its knowledge base and the database. The knowledge bases consist of rules about negotiation strategies and tactics. The database stores all activities of each negotiator (e.g., preference ratings, offers and messages sent, etc.) and will be used by both the NSA and NSS. At certain stages in the negotiation, the negotiator may request support from Atin by asking questions. In order to provide suggestions, Atin may request some additional information from the user (e.g., negotiation strategy, willingness to make concession, etc.). These inputs from the user will help the agent to filter out irrelevant information, and display the most appropriate advice.

Information submitted by the negotiator (e.g., an offer or a message) is passed to the Inspire engine. Inspire handles communication between users (in this case, sends an offer to the counterpart via the message engine), saves the user activity to the database, as

well as performing decision support activities (e.g., returning the numeric utility value to the user after computation). It also invokes the user's negotiation assistant – Atin. The agent receives the user input, collects relevant information from the databases, searches the knowledge base, and returns appropriate suggestions (if any) to the user's web browser.

7.4. The roles of Atin

The architecture of the negotiation agent is presented in Figure 9.

Figure 9. Atin Architecture

Information is being passed from Inspire to Atin, and a parser is used to parse the information in the format that Atin understands. Atin receives the information and searches knowledge bases and databases for rules and historical data respectively, and parses the relevant suggestions to Inspire.

The Atin agent is invoked when the user logs in to the Aspire system. It stands by and receives notifications about user activity from Inspire. For certain activities, Atin

searches its knowledge bases and checks if there is any appropriate advice for the user. It searches the database, and performs analysis on the user's past activities. At any point of the negotiation, the user may seek help from the agent by asking questions. For example, based on a set of predetermined formulas the agent is able to compute the value of an offer and performs evaluation. However, it is not compulsory for the negotiators to seek help from Atin. Atin is seen as an optional feature capable of providing both active and passive support to the user. For this prototype, users may disable Atin when they log in to the Aspire system. If the user disables Atin's support, it continues to collect negotiation activities and relevant information of the user, but it will not proactively interfere the negotiation until the user activates the agent in a later time.

If the user enables Atin's support, the agent must be active throughout all the phases: preparation, conduct of negotiation, and post-settlement – although the level of support may vary in each phase (Lo and Kersten, 1999; Lo and Kersten, 2000).

7.4.1 Pre-negotiation phase

During the preparation phase, Atin assists the negotiator in structuring the problem. Atin helps the negotiator in the preference elicitation and utility construction steps by giving comments on his/her action. Similarly it may help in setting the BATNA and reservation values. The pre-negotiation interactions between Inspire and the user, and Atin's activities are illustrated in Figure 10.

an offer that is below his/her BATNA, Atin will alert the user with a warning message before the offer is being sent to the counterpart. At any time, as indicated in Figure 11, the user may seek advice from the agent regarding tactics, counter-offers, and so on.

Figure 11. Atin's support in the conduct of negotiation

Upon request Atin may propose structured offers, which are based on previous exchanges of offers and the level of concession made by the user. For example, the agent may first ask the user to define a negotiation strategy (hard and positional bargaining, accommodating, or process and relationship oriented). When the user receives an offer from the opponent, the agent may offer an assessment of the offer to the negotiator while the NSS provides a quantitative evaluation (i.e., numeric utility rating).

An important activity of the agent is to access the activities undertaken by the user and the counter-offers sent by the opponents and to provide interpretations. For example, one can judge the negotiator's range of flexibility based on the differences between the utility value of BATNA, and the reservation values of these issues. The differences between the aspiration values and the highest utility value may indicate the strength

(power) the negotiator perceives. These values also allow the evaluation of offers that the negotiator wants to submit.

Atin also compares the user's current activities with his/her negotiation history and provides overall assessment based on DANA's (David's Automated Negotiation Analyser) knowledge base (Cray, 1999). For example, if a user makes significant concession (a huge drop of utility) between offers, the agent makes the following suggestion (Cray, 1999):

"It is generally unwise to make very large changes in your position between offers. There was a difference of [x] points between your recent two offers. As a tactic multiple small concessions provide your counterpart with a sense that you are being cooperative while, in fact, you are giving away little. Where the number of issues and options are limited, as in this case, you may want to make concessions by reducing your request on an important issue while increasing it on a one of less importance. This would minimize the erosion of your position when making a concession. If the cause of your large drop was a lack of time it means that you have not planned far enough ahead in your negotiation strategy".

If a user does not make any concession between offers (no change between consecutive offers), the agent may provide the following advice (Cray, 1999):

"I notice that there is no change of positions in your recent offers. While remaining firm is certainly an acceptable and common tactic in bargaining, small concessions will often keep the process moving. This is especially important when your counterpart has other possible suppliers/buyers. A minor concession signals your willingness to negotiate as well as your general adherence to your current position."

7.4.3 Post-settlement phase

Once a compromise has been achieved during the negotiation phase, Inspire checks it for efficiency (Pareto-optimality) and presents possible alternatives for joint improvement. Inspire takes into consideration the utilities of both parties and computes the efficient packages (alternatives) for the users. The agent provides an explanation of why the user should seek efficient compromises and suggests that the user continue the negotiation. If both parties agree to continue the negotiation, Atin continues to support with features as in the negotiation phase. Figure 12 illustrates Atin's support in the post-settlement phase.

Figure 12. Atin's support of post-settlement activities

7.5. Non-functional (technical) requirements

The non-functional requirements are system requirements that are not directly related to what the system should do (OOTC, 1997). In other words, while the functional

requirements determine the analysis process, the non-functional requirements drive the design of the system. Non-functional requirements for Atin include the following:

- (a) The system has to support IE 5.0 and above.
- (b) The system should support concurrent users.
- (c) Since Inspire users are worldwide; connection response time should be taken into consideration.
- (d) The software agent runs on a local host, which is a Sun OS Solaris platform.

7.6. Use case model

Use case modeling uses a special sequence of transactions performed by the system interacting with a user in dialogue (Sodhi and Sodhi, 1996; Satzinger, Jackson et al., 2000; Conallen, 2000). The complete collection of use cases, actors, and diagrams forms a use case model. This modeling methodology will be used for describing the requirements of Atin. The Inspire features are encapsulated in the diagrams since the goal of this project is to deliver a negotiation software agent. To illustrate the application of the use case model we present three cases.

Figure 13 A Use Case Diagram for Atin support in pre-negotiation phase

Figure 13 presents the use cases in the pre-negotiation phase. There are two actors. “Negotiator” represents a role that a user can play in the Aspire system. “Atin” represents the role of the negotiation agent. In the pre-negotiation phase, the user submits preference ratings (i.e., issue, option, package ratings), and the agent retrieves information about the user and updates the database. Based on the information, the agent searches its knowledge bases and sees if there are any appropriate suggestions. If the condition matches, the agent processes the advice and provides the appropriate suggestions to the user. For example, when the user submits issue ratings, Atin is triggered by Inspire and checks if the user’s input violates any negotiation rules in the knowledge base. The user will be alerted if his/her activity violates the negotiation rules

in the knowledge base. The agent also assists the user in better prepared for the negotiation phase, such as defining the bottom line (reservation level) and BATNA.

Figure 14. A Use Case Diagram for offer exchange support

Figure 14 illustrates the use case diagram for a user during the negotiation phase. The user invokes Atin and selects his/her preferred negotiation strategy. Atin updates the user profile (in the database) and provides some instructions for the user's selected strategy by search its knowledge bases. The agent analyzes the "goodness" of an offer/counteroffer. For example, when a user submits/accepts an offer, Atin retrieves the user's information and check if the offer he/she is trying to send/receive violates his/her pre-defined preference settings, such as reservation level and BATNA. Atin also analyzes

the user's activities (and compare with relevant information in the user database) and interfere if there are any suggestions.

Figure 15. Atin support Use Cases

Figure 15 displays the use cases when a user requests Atin's support. The user requests support from the agent, and the agent retrieves the user's information from Inspire and the database. Depending on the status of the user (since the support level varies), a menu is displayed and user may select from the available options. Once the user selects an option, Atin searches database and knowledge bases to process user's request. Atin will determine additional information is required from the user, will ask him/her to

enter the necessary information. Atin updates the user's profile with the information and also searches the knowledge bases to generate the appropriate advice.

8. Design

Section 8.1 discusses general design issues for this project. Since this project is developed based on the Inspire system, the design guidelines of the InterNeg development team are followed. Sequential diagrams are presented in Section 8.2. CMM suggests the main output of design to be the functional design and database design documents. Client interface design and server-side processing are discussed in Sections 8.3 and 8.4 respectively. The database design and table structure are presented in Section 8.5.

8.1. Design issues

Rapid prototyping, simplicity, and extensibility are among the most important design criteria in building our integrated software environment. Therefore, in the design of the components of the integrated negotiation software environment, the object-oriented and rule-based methodology that we have used to develop the Inspire and INSS systems will also be used (Kersten and Noronha, 1999b). The use of object-oriented techniques can benefit the developers through code reusability, since a design pattern is a set of co-operating objects or classes in a particular structural pattern that reappears in many implementations. A rule-based methodology is easy to understand; each rule can be viewed as a unit of information in a knowledge base, which can be easily added or removed (Bigus and Bigus, 1998).

The proposed architecture is modular so that all the components and their objects can be developed independently. The system requires nothing more than a web browser and an Internet connection thus enhancing portability for our end-users (Kersten and Noronha, 1999b). In addition to the non-functional requirements (Section 7.5) we have identified a list of major design considerations, based on the proposed architecture:

1. A simple and user-friendly interface for user input. We do not want users to spend too much time learning how to use the system. The GUI design of Aspire should be similar to the current Inspire interface.
2. Redundant error handling. Verification of user input (e.g. such as preferences) is required.
3. Flexibility of the system to allow future upgrades.

8.2. Sequential Diagrams

The different stages of Atin support can be identified by pre-negotiation, conduct of negotiation, and post-settlement phases. These phases roughly correspond to Atin's support in different phases: preference and strategic advice (pre-negotiation), offer exchange support and negotiation assistance (conduct of negotiation), and post-settlement advice (post-settlement).

During the detail design phase, the view of the system developed during high-level design is broken down into modules and programs, and will be discussed in the following with the aid of sequential diagrams in this section.

Figure 16. Sequence Diagram for user preferences support

Figure 16 presents the sequence diagram of user preferences definition, such as issue, option, and package ratings. Although preferences are defined in the pre-negotiation phase, the system goes through the same process if the user changes his/her preferences during the negotiation phase. A user submits his/her preferences based on the negotiated issues and options, which activates the preference validation function of the agent. The function sends a query to the negotiation preference knowledge base and searches to determine if any rule(s) is violated and displays the appropriate advice to the user. The save rating function inserts (or updates) user preferences into the database table.

Figure 17. Sequence Diagram for Offer support

Figure 17 illustrates the sequence diagram when a user submits an offer. When an offer is submitted, Atin's preference validation function calculates the offer utility and checks if the offer has violated any predetermined preferences, such as reservation values or BATNA. For example, if any offer is lower than the reservation value, or the package utility rating is lower than the BATNA utility. If any of the conditions are violated, it displays a warning message to the user before processing the offer submission further. Since we do not want Atin to make negotiation decisions on behalf of the user, the negotiator may ignore the agent's warning and continue the submission of the offer (see Figure 18). The user may either cancel the offer submission, or continue. The offer will then be inserted into the user_offer table in the database and sent to the counterpart.

Figure 18. An example of Atin Warning during offer submission

Figure 19. Sequential Diagram for Atin support

Figure 19 shows the sequential diagram of Atin's interactive support. As stated in the previous sections, a user may ask Atin for advice during the negotiation. The user may select from a list of available options. Atin provides solutions and suggestions by searching the knowledge bases and retrieving user and negotiation data from the Inspire engine and databases. In order to provide more personalized assistance, depending on the question, the agent may need some additional information from the negotiator (if the

information is not readily available from the database). The requested information helps the agent to justify and display the most appropriate advice for the user.

When a user invokes Atin, Atin checks to see if the user has entered all the necessary preference-related information (e.g. reservation level and BATNA). Since this information determines the agent's response, Atin may not be able to provide assistance without certain essential information. The agent asks the user to enter the required information if he/she has not entered any of the necessary information.

The agent also requests the user to select a negotiation style, since Atin may provide suggestions based on user's selected style. Once this information is entered, it is stored in the database and the agent searches over the knowledge base to provide the appropriate suggestions. Once the negotiation style is defined, the agent provides advice based on the information provided by the user and from the database. The user may select options from a list of available choices and the agent processes the user's request and retrieves the relevant information from the database and the knowledge base. In order for the agent to become more personalized, the agent may request further inputs from the user before making a suggestion.

Due to time limitation in development, Atin will provide basic support to the users during the post-settlement phase. As discussed in Section 3.3, out of the Inspire users who achieved an inefficient compromise, only 18% wished to continue and improve their results via Inspire's post-settlement mechanism. If both users reach an

inefficient compromise and do not wish to use our post-settlement mechanism, Atin will display a message to explain the benefits of using such mechanism and suggest them to continue the negotiation in the post-settlement phase.

8.3. Client-side user interface

The mappings between client page model elements and codes are essentially the same for all server-side enabling technologies. The Atin frontend consists of web pages and graphic files (see Appendix D).

The user interface screen flow describes the sequence of screens, which the user expects to see from their browser. For example, in Inspire the standard screen flows consist of: Introduction → Case description → Preparation → Rate issues → Rate options → Rating packages → Make offer (send message). Since the Atin interface is embedded within Inspire, the screen flow is relatively simpler: Ask Atin Now → Menu selection → Request information → Suggestions → Explanations.

The user interface is designed based on two criteria, ease of use and load time. The layout of buttons is simple and easy to understand. Pop-up windows are used because it is easier for the user to compare/refer to information on the negotiation screen. As mentioned in Section 4.3, the Inspire system has been used by over 4,000 people from different countries. Due to this reason, load time is a major consideration especially for countries that do not have stable and fast internet connections. Therefore we do not consider putting large graphic files on the pages, as well as Java applets.

8.4. Server-side processing

The backend processing consists of object-oriented programming tools, server-side scripting languages, and API to databases. Since the Inspire system is built in C++, the language is used for exchanging and passing information between the NSS and the NSA. The `inasa_advice` function in Inspire calls the agent and passes the parameters in order to retrieve advice from the inputs (e.g. issues, options, offers) submitted by the user. The agent is invoked and returns a message to the NSS regarding further processing instructions, and displays appropriate messages to the user.

8.5. Database development

The original Inspire system stores user information and activities in a user log file. Since the agent requires frequent access to user information, there is a need to implement a relational database for the integrated system. The database system in Aspire stores information to be used by both the NSS and the NSA. Other irrelevant information is stored separately with the original Inspire system.

The database design indicates the tables needed in the system, the attributes of each table, the primary key (PK), and the foreign keys, etc. Mysql was selected to be the relational database server, but it does not fully support the foreign key feature. Therefore we cross-reference the tables and join them before making queries. Nine tables are defined for the Aspire database (Figure 20, p.82) and discussed below:

User:

This table stores key information about the negotiator, including **userName (PK)**, **session_directory (PK)**, **negoName**, **case**, and **status**. The **userName** is the negotiator's login name to the Aspire system, and the **negoName** is the Negotiation Name. **Session_directory** is the path location where the negotiation files and information are stored, which is a unique identifier for each negotiation. The **case** field states the negotiation case (e.g. Itex or Cypress) that is presented to the user. The **status** field represents the status of the user's negotiation agent, since the user may turn the agent on or off.

User_style:

The **user_style** table consists of the **userName (PK)**, **session_directory (PK)**, and **style**. As discussed above, **session_directory** is a unique identifier for each negotiation, and therefore **userName** and **session_directory** together become a unique identifier for each user. User defines his/her negotiation style and the NSA stores/updates it in the **style** field.

BATNA

The **BATNA** table consists of **batnaId (PK)**, **userName (PK)**, **session_directory (PK)**, **type**, **time**, **issue1**, **issue2**, **issue3**, **issue4**, **batna_description**, **rating**. **BatnaId** is an auto-increment field, since we record all of the changes a user makes during a

negotiation. Each batnaId is also associated with a timestamp and when the NSA checks for BATNA validation, it searches for the latest (most recently) BATNA information. There are two ways BATNA can be determined: either 1). Defined it on the current negotiated issues or 2). Defined on an alternative that is different then the currently negotiated. The *type* field records the type of the BATNA based on the above. If the first type is entered, the option for each issue is inserted and the NSA computes the BATNA utility (rating) based on the selected options. If the latter type is chosen, the user enters a description (a situation in which he/she is able to obtain something very different that he/she will use instead) and his/her utility (rating) on the BATNA.

Reservation level

The fields in the reservation_level table are very similar to the BATNA table. It consists of reservationlevelId (PK), userName (PK), session_directory (PK), time, issue1, issue2, issue3, issue4, and rating. reservationlevelId is an auto-increment field, since we record the changes a user makes during a negotiation. Each reservationlevelId is also associated with a timestamp and when the NSA checks for reservation values validation, it searches for the latest (most recent) reservation levels.

User_offer

The user_offer table consists of offerId (PK), UserName (PK), session_directory (PK), stage, time_sent, issue1, issue2, issue3, issue4, rating and Opponent_rating. OfferId is an auto-increment field as well as a unique identifier of the offer package. The stage

field represents the stage of the negotiation when the offer is submitted (conduct of negotiation, or post-settlement). The time_sent field stores the time when the offer is made. The rating field stores the utility rating of the sender, while the Opponent_rating field stores the opponent's utility rating (for the same offer).

Issue_name

The issue_name table consists of the issueId (PK) and issueName. The purpose of the table is to identify the available issues in a negotiation case. IssueId is an auto-increment field while issueName is the associated name of the issue. For example, issueId = 1 and issueName = Price.

Issue_rating

The issue names are defined in the issue_name table and it is related to the issue_rating table. The issue_rating table consists of issueRatingId (PK), userName (PK), session_directory (PK), time, issueId, weight. The issueId and weight reflects the weight assigned by the user to a specific issue during the issue preference activity.

Issue_options

The issue_options table consists of issueId (PK), optionId (PK), and optionName. This table is used for identifying the available options of each negotiated issue in a selected case. For example, issueId = 1, optionId = 1, optionName = \$3.47.

Option_rating

The option rating table consists of optionRatingId (PK), userName (PK), session_directory (PK), time, issueRatingId, optionId, and rating. This table stores the option ratings submitted by the user, based on each issue (identified by issueRatingId).

Figure 20. Aspire database

Underlined - Primary key (PK)

Bolded – Required fields (NOT NULL)

8.6. Knowledge base and rule implementation

The knowledge base consists of sets of negotiation rules retrieved from previous Inspire data and literature. The rules can be identified as two types: 1) qualitative, and 2) quantitative. The qualitative rules involve support for the negotiator's tactics, such as suggestions containing negotiation style and strategy. The quantitative rules involve computation of offer utility and comparison with user preferences.

Since the Aspire prototype attempts to avoid some common negotiation pitfalls presented in Section 3, most of the rules are developed to address those issues. A set of rules is defined in order to guide the users during the pre-negotiation phase, such as how to assign weight to negotiation issues, how to prioritize the options, and how to define one's bottom line. There is another set of rules designed for the negotiation phase, addressing common and popular mistakes that negotiators may make during a negotiation. There are also other sets of rules about negotiation style and strategic advice.

The current knowledge bases are developed using the Jess expert system shell. Information is passed from the NSS engine and Jess searches its knowledge bases. Since Jess is developed in Java, a JDBC (Java DataBase Connectivity) driver is installed and acts as one of the APIs to the database (refer to Figure 21 in Section 9.1). JDBC accomplishes its goals through a set of Java interfaces, each of which are implemented separately in order to support the exchange of data between knowledge bases and the database.

9. Development

The implementation of the Atin system is more than just writing and compiling the code, although that is a large part of the overall workflow. The development process is the process by which user requirements are elicited and software satisfying these requirements is designed, built, tested, and delivered (Jalote, 2000). The development of Atin is considered as a new application but also a major enhancement for the existing Inspire system.

Problem specification, requirement analysis and high-level design have been discussed in Section 7 and 8. Problem specification states the problem we are trying to solve with the development of this new system. Requirement analysis outlines the required functionalities to solve the problem. High-level design gives the solution at a high level of abstraction – contains a functional architecture of the system, and also the database design. High-level architecture design has been completed prior to development. Once the required tools are installed, low-level detail design has been carried out, with specification of each module. Use cases diagrams and sequential diagrams (see Section 8.2) were developed to aid the development.

Table 3 presents the major tasks/activities that are conducted during the development, along with a list of the anticipated technical deliverables. Our first task was to install a web server that will allow us to host sites, that would be secure for e-commerce solutions, and that could be driven via scripts to connect to a database server

and extract its data. These development activities are further discussed in the rest of this section.

ACTIVITY	DELIVERABLE (S)	Discussed in
Tools Selection and installation	Identified the tools to be used. Installed Apache 1.2.5, PHP 4.0.12, Mysql 3.22.22, Jess50b3, JDBC.	Section 9.1
Physical design specifications	High level design, UML use cases, sequential diagrams, database design	Section 9.2
Coding	Modification of inspire, frontend development, PHP scripts, SQL programming, JavaScript validation, Java data conversion program, etc.	Section 9.3
Knowledge base development	Negotiation rules, JESS knowledge base	Section 9.4
Testing	System test plan, test cases, acceptance tests	Section 10

Table 3. Development activities

9.1. Tools Selection

As discussed in Section 7, web applications development involves a number of the web-enabling technologies that need to be managed. These technologies are mechanisms by which web pages become dynamic and able to respond to user input. The choice of technologies always depends on the nature of the application, tools used in the existing system, and even the knowledge of the developer.

The tools selection for this project includes client-side development languages such as HTML, JavaScript, and some distributed object technologies. The languages and technologies on the server-side include C++, Java, CGI/Perl and PHP. The server side also deals with the Inspire engine, MySql database server, the Jess expert system shell, and knowledge bases.

Due to legacy reasons the Common Gateway Interface (CGI) is used for data transfer between the Aspire frontend and the NSS but not for the NSA. Although it is true that Servlets and CGI programs could meet the same requirements, scripting language (i.e., PHP) is preferred due to performance and efficiency (Castagnetto, Rawat et al., 1999). PHP is also a simple, yet powerful scripting language that integrates easily with Mysql, a powerful relational database server. The original Inspire system runs on a NCSA web server, but the Apache web server is required in order to run PHP.

Jess (2000) is used as the inference engine of the agent. It is primarily an expert system development environment (which has the capacity to “reason” using knowledge supplied in the form of declarative rules), but components of JESS can be used independently for a rule-based agent (Friedman-Hill, 1997). Another reason for selecting JESS is because of its most prominent feature – it is written in Java which is platform independent and easy to integrate with other applications.

As mentioned in the previous sections, software agents are continuously active programs - The (*run-until-halt*) function in Jess is applicable since it will run until requested to terminate (i.e. the halt function is called). If there are no rules to fire, *run-until-halt* suspends the calling thread until new rules are activated.

The scripted web pages are invoked by the web server after it detects the page consists of scripts to interpret (e.g. PHP scripts). This is indicated by the file name extension (.php). When the web server receives a request for one of these pages, it first

locates the page and then passes the page to the appropriate application to process. The application server processes the page, interpreting any server-side script in the page, and interacting with server side resources. The result is a web page that is sent to the client's browser.

9.2. High Level Physical Design

The interface receives all user requests and queries and passes the information to the inference engine. Java programs act intermediaries between the databases and the agent.

Figure 21 illustrates a high level view of the system. There are two ways for passing information: (1) via NSS or (2) via NSA. The first situation submits information to the NSS (Inspire), and NSS passes the information to a data conversion program. It sends information to the NSA inference engine and updates the database. The database conversion program is written in Java and is used for converting the parameters to a database/knowledge base readable format. The second scenario is where the user requests advice from the agent, the information is sent via PHP to activate the processing files. The programs retrieve information from the database and knowledge bases and return advice to the user.

Figure 21. High level design of Aspire

Design issues and aspects are discussed in the Section 8. Sequential diagrams are used to give a visual interaction of the scenarios. These diagrams are listed in Section 8.5. A normalized database design is also developed (see Section 8.6), for the data storage in the relational database system.

9.3. Coding

The following coding guidelines have been followed (OOTC, 1997):

- a. File naming conventions
- b. File structure – separation of frontend and processing files
- c. Identifier naming conventions
- d. Global variables
- e. Exception and error handling

f. Calling conventions and return types

Based on each specification, the original Inspire backend system has been modified. The Inspire engine is written in C++, and it is responsible for authentication of the user, displaying appropriate information regarding the user's negotiation status, providing persistent saving of negotiation activity, and email notifications. The main programs under the Inspire engine are being modified: several methods are being added, in order to pass information to the agent, as well as to invoke some of the agent's features.

The Atin interface consists of web pages that dynamically display appropriate messages to the user, resides at the top of each page (see Appendix D). The interface is also used for handling user interaction, allowing the user to input information as well as display messages to the user. The user may select their requests and enter any information to the agent. PHP scripts runs on the server side for processing help and validation features. User input validation is handled by JavaScript. It is necessary to use this kind of error checking to reduce the possibility of invalid input.

Mysql supports clients' application programming interfaces (API), which are required for accessing data from the database such as PHP and database generic middleware such as ODBC and JDBC. JDBC (Java Data Base Connectivity) provides for uniform object-oriented access to a relational database, which may be local or located on

a remote server (Schmid, 1999). This is used as an API for Atin to connect to the Mysql database server.

The NSA uses PHP as one of the API(s) for connecting to the database server, as well as for backend processing of information. When certain information is passed to the NSA, it calls the `connect_db.php` program which initiates the connection to the database server. After connecting to the database, information can be retrieved/inserted from/into the tables. PHP also handles processing of information of the NSA, such as choosing what to display to the user. Switch statements are used in the main program and it redirects to other PHP applications based on the incoming information.

Using the `SQLSource` class, the program is able to interact with the Mysql database and perform standard database operations such as `SELECT`, `UPDATE`, and `INSERT`.

For example, let us assume that user “Gregory” asks the agent to suggest a counter-offer. Assume that Atin requires all the previous offers from the use in order to make the suggestion. The request is then processed by a PHP script that decodes the values sent by the client’s browser to the NSA. The agent receives the request and generates the following SQL query and sends it to the database:

```
$result = mysql_query("select * from offer where userName =
'Gregory' and negoName = 'test' ") or die(mysql_error());
```

The agent then retrieves the result (if there is no result then return a standard error message) from the query, processes the data and searches its knowledge bases. The final output (i.e. the suggestion for a counter-offer) will be formatted for displaying in the client's browser.

Once the Atin development is completed, the Inspire makesession program has been modified. The original *makesession* program is used to set up an Inspire negotiation between two users. A new program called *makesession_atin* has been developed for creating negotiation sessions for Aspire. The program calls a PHP script and inserts user information into the users table in the Aspire database before creating the negotiation session.

9.4. Knowledge base implementation

The knowledge base consists of sample rules derived from previous Inspire data, the literature, and our experiences. The rules have been listed in the form of declarative knowledge and some of them are based on certain conditions. Each rule has antecedent and consequent clauses, which represents the IF x THEN y condition.

The rules are implemented in knowledge bases and access via the Jess inference engine and PHP programs. Negotiation activities are passed from Inspire engine to Jess (Java programs), and retrieve any relevant information from the database via JDBC. When negotiation data is passed to Atin, it searches the knowledge bases by examining

the antecedent clauses of each rule to determine which ones can be triggered and return suggestions to the user.

10. Testing

The test workflow focuses on the evaluation of the executable artifacts of the system. Many different tests are made. Functional tests determine whether specific functions, as defined in the requirements specification, have been implemented properly. Integration tests validate the individual component interfaces and their ability to work with one another. The acceptance test is another formal test that the user community performs on the system.

As discussed in Section 7.1, preliminary testing is performed at the end of each increment. As the development of the Aspire system gradually evolves through the various stages (see Figure 5, p.48), more thorough and systematic testing is required before system deployment. We have identified four testing stages: (1) unit testing, (2) integration testing, (3) system testing, and (4) user acceptance testing.

Unit testing is used to verify the smallest logical units of application code work as the design specifications. It is a highly iterative process which starts at the end of each development stage illustrated in Figure 5 (p.48). Integration testing is to ensure that the various modules work properly according to the specifications after integration (combining products in different stages). System testing is to verify that the final system performs as listed in the functional requirements. User acceptance testing validates the functionality of the entire system, including both technical issues and other factors, such as the level of user friendliness.

Unit testing and integration testing are performed during the system built phase, regardless of the percentage of the overall system completed. Most of the functional bugs are fixed during the two testing stages. This section focuses on the discussion of system testing and user acceptance testing. Section 10.1 presents the system test plan and the required testing activities and a test report is presented in Section 10.2. The planning of user acceptance testing is discussed in 10.3, and Section 10.4 presents the results and a summary of user feedback.

10.1. Test Plan

During system testing, various hardware/software/platform components that make up the client server architecture are jointly tested to verify their level of interoperability. The output after the test planning stage is a test plan in Table 4 below.

System Test Plan for Atin	
Test Environment	
Server: SunOS 5.7	
Client: Windows 98/2000/ME, Internet Explorer 5.0 and above	
Features to be tested	
Condition to be tested	Expected Result
1. User Interfaces	
a). Navigational interfaces	Should be consistent in all screens
b). Alignment of interface elements	
c). Resolution and scrolling	
2. Browsers/Operating System in which all the Test Conditions are to be tested	
a). Browsers	Internet Explorer 5.0 and above
b). Operating System	Windows 98/2000/ME
3. Generic Test Conditions	
a). Click on all links	System should display appropriate screen. It should not display "Page Not found"
b). Use of empty values in mandatory fields	Appropriate error message should be displayed
4. Security Check	
a). Login with incorrect info	Aspire should give an appropriate message and should not allow the user to enter.
5. Business Processes	
5.1 Atin Intro Screen	
a). User click	Should display an intro to Atin when clicked
5.2 Atin Proactive Support	

a). Issue Rating	Control of issue weight allocation
b). Option Rating	Consistency with option rating
c). Package Rating	Consistency with option rating
d). Violation of reservation level	Warning message display if offer violates
e). Violation of BATNA	Warning message display if offer violates
5.3 User Information Entry	
a). Display appropriate screen	Screens and messages during different stages
b). Reservation Level Input	Store the reservation level to database
c). BATNA Input	Store BATNA value to database
d). Define Strategy	Display instructions and suggestions
e). Revise previously entered info	Display and update info upon request
5.4 Proactive features	
	For certain user request, the agent should be able to check the user information and automatically provide advice.
a). Appropriate advise at certain stage	Suggestions varies upon scenario and matches with the knowledge base
b). Logical	Not applicable in this version
c). User feedback	Not applicable in this version
5.5 Negotiation Terminology	
a). Terminology Search	Display the appropriate explanation of the term
b). Terminology requested not available	Display a message stating that the terminology is not available

Table 4. System test plan

10.2. Testing Activities

The test plan in Section 10.1 identified features to be tested, as well as the proposed test environments, test parameters, test procedures, and stopping criteria.

Testing activities follow the test cases designed in the test plan (Table 4). The system test report provides the outcome of executing the system test plan. For each test case, it specifies whether the system behaved as expected. Once a bug is found in the system testing, the problems are logged in the test result document, and later fixed.

As discussed in Section 7.1, we followed the black box testing methodology for validating the system. The testing activity involves two people: (1) the developer, and (2) a tester who does not know how the program works. The reason to involve a third-party tester is to avoid developer's bias during the testing process. The test cases are being examined carefully in order to identify all possible errors. The process includes (1) searching for incorrect or missing functions, (2) identifying user interface problems (e.g., problems with screen resolution), (3) testing the rules in knowledge bases, and (4) checking compatibility issues with different browsers and platforms. The testers compared the actual result of each test case to the expected result, and report if bugs are discovered.

System testing begins with the set up of Aspire negotiations sessions. The negotiation set up program (i.e., `makesession_atin`) is tested to ensure that all the information is inserted into the database. The testers play the role of end-users, negotiate via the Aspire system and make use of all the new features. The testers also attempt to make unwise decisions during the negotiation in order to test Atin's alert and warnings. Destructive testing, which tries to determine if the system does things that it should not do, is also performed by the testers. An example of destructive testing is clicking on the

submit button for ten times and check if the database will record ten entries of the same information.

The system testing activities were carried out successfully as expected (although a few minor bugs are identified) since most of the defects are removed during unit testing (where we test each piece of code against its designed functionality) and integration testing (after all different modules are combined into the final system).

During system testing we have identified several bugs in the Aspire system. In general the bugs can be categorized into two types. The first type is errors relating to database access. For example, the BATNA value was not updated in the database table. We recognize the problem based on the error message:

“PHP Warning: Supplied argument is not a valid MySQL result resource in update_batna.php on line 121”

The above error message indicates that there is an invalid Mysql statement in the program file update_batna.php. The detail error description has been documented and the bug was solved.

The second type of error relates to missing functions. For example, in the drop down menu which the user may ask question, if the user selects nothing (i.e., default selection) and click on the submit button, it returns a blank page display. This is caused by missing input validation functions. After reviewing the source code, JavaScript data validation is added.

Since several defects were discovered during the system testing, we carry out a second cycle using the same test cases. The second test cycle emphasizes on checking the bugs identified in the previous round, to ensure the known bugs have been fixed. Table 5 presents a summary of the test results. Please note that the test case number in the table is referenced to the numbers in the test plan (Table 4).

Test case	Test Description	Actual Output	Result
1a	Browse through all available screens and checks for consistency	All screens displayed as expected	PASS
1b	Browse through all available screens and checks alignment	All pages content aligned	PASS
1c	Browse the system with 800*600, 1024*768 resolution	Pages look o.k. Scroll bar available for pop up windows	PASS
2a	View the application with IE 5, IE 5.5	Pages look o.k.	PASS
2b	Testing via Windows 98/2000/ME	Pages look o.k. and applications run properly	PASS
3a	Clicks on all links & icons	Bug: Some error displayed if click on Ask Atin Now with no selection	FIXED
3b	Submit empty mandatory fields	Bug: the system accepted empty mandatory fields.	FIXED
4a	Login with incorrect info	Error message display: No such negotiation	PASS
5.1a	Click on Atin Intro screen	Atin introduces its features	PASS
5.1b	Switch off Atin	Atin does not interfere my negotiation process	PASS
5.1c	Activate Atin	Atin resumes support and proactive advice	PASS
5.2a	Login with incorrect info	Error message display: No such negotiation	PASS
5.2b	Allocate over 90 points to a single issue	Atin alerts and displays explanation	PASS
5.2c	Assign option rating in an incorrect way	Atin alerts and explains what I did wrong	PASS
5.2d	Submit an offer below reservation values	Atin alerts user without submitting the offer	PASS
5.2e	Submit an offer below BATNA value	Atin alerts user without submitting the offer	PASS
5.3a	Ask Atin to suggest style during pre-negotiation	Atin response the support is not available until pre-negotiation phase is completed	PASS
5.3b	Enter reservation level	Stored in database table	PASS
5.3c	Enter BATNA	Bug: BATNA utility not calculated correctly	FIXED
5.3d	Select competitive style	Atin displays how to approach the opponent based on competitive style	PASS
5.3e	Revise BATNA	Bug: Does not get updated in database	FIXED
5.4a	Refuse to use post-settlement	Atin explains the benefits of using such mechanism to achieve jointly improved outcome	PASS
5.5a	Search for the term "BATNA"	Atin displays the definition and an example of BATNA	PASS
5.5b	Search for the term "happy"	Atin displays "terminology is not available in database"	PASS

Table 5. System test result

10.3. User acceptance test planning

Once the system testing is completed the final step is user acceptance testing. End users are involved in this activity, since we need them to play the role of the negotiator who verifies whether the constructed system satisfy its requirements or not. Since our target is to investigate if Atin can be a helpful tool during web based negotiations, we have invited two groups of users with different backgrounds to test the system.

The first group of the users has used the original Inspire system within the past 12 months and the other group has never used any web-based negotiation support systems before. The rationale behind the above selection is because we are interested in having feedback from both previous Inspire users, and new users. We would like to know from previous Inspire users whether (1) the Aspire system provides more extensive support to users than Inspire, (2) web-based negotiations become easier with the aid of an agent, and (3) the features they considered most/least useful during the negotiation. We also would like to know from novice negotiators concerning (1) their experience with Aspire, (2) if a NSS-NSA integrated environment provide adequate support for web-based negotiations, and (3) the features they consider most/least helpful during the negotiation. A simple survey was developed and we requested that each user to fill in a copy after they completed a negotiation (see Appendix C).

Negotiation cases are set up for the users and each of them plays the role of either a buyer or seller (depending on their case scenario description). The two groups of users

are paired among each other randomly. Our aim is to receive feedback from these users in order to improve the existing features, support methodologies, and user-friendliness for the next version of Aspire.

10.4. Review and evaluation

We invited 18 users to use the Aspire prototype. These users negotiate with unknown opponents for a week and are requested fill in a review and evaluation survey (see Appendix C).

Generally the feedback from the users has been favorable. For those people who have used the Inspire system before, they found that the Aspire system has provided them more support, as well as been much easier to use compared to the original Inspire system. One user stated that “the new features in Aspire provide the user a better idea of what’s going on and it helps me to clarify a few issues which I might have misunderstood during my last use”. This conforms to our expectation that web-based negotiation becomes easier with the aid of an agent.

Users claim that the pop-up warnings from Atin play a significant role in both their decisions and their assessment of their own negotiation strategy. This indicates that such a feature could reduce the occurrence of certain negotiation pitfalls. One user made the following comment:

“The pop-up warnings not only alerts the user on an unreasonable action they have made, but also reminds the user

some of the previously identified parameters in the pre-negotiation phase. I was too focused on my rating value and did not realize my offer violates one of the bottom line values. These alerts also prevent me from overlook of important issues during the negotiation."

The users are also asked in the survey whether they feel in control during the negotiation. Over 78% stated that they are in control of the negotiation process and feel that Atin assists the negotiation without taking over the control from them. This conforms to our expectations, since we would like the agent to assist the user whenever required, but not taking over the negotiator's control.

Other comments from the users can be categorized into two types: (1) functional and (2) technical. The first type relates to the features and functionality of the system. Some users stated that the current features of Atin are quite limited, but the system has the potential to provide further support during the negotiation process. There are also suggestions related to the enhancement of the existing features, such as a "wizard-like" process that guides novice users through the negotiation process, and step-by-step instructions in how to define BATNA values. The second type of comments relates to some technical issues. Some users from other countries stated the load time for some pages was slow. One user suggested making Atin an applet which can interact with the user in real time, without reloading the page.

11. Conclusion

Based on the literature review we assumed that there would be demand for a NSS+NSA system. According to our review such an environment has not yet been developed. We developed Atin and an integrative negotiation support environment (Aspire) based on the existing negotiation support system (Inspire). The development of Atin also involves other relevant components such as databases and sample knowledge bases.

During the progress of this thesis and development, papers have been written and also presented in research seminars and conferences (Lo and Kersten, 1999; Lo and Kersten, 2000). Comments and various suggestions have been received from attendees. A popular question was: Is Atin a software agent? Attendees have made critiques that Atin should be considered as an online help system or a web-based expert system. There have been ongoing discussions on the definition of software agents. As discussed in Section 5.1, the common characteristics of software agents are: (1) continuously active, (2) degree of autonomy (proactive or reactive), and (3) capable of working on behalf of the user. We see Atin as a software agent since (1) it is invoked upon user login, (2) it may proactively advise the user or react based on user activity, and (3) it works on behalf of the user to analyze the negotiation data (based on a set of pre-determined rules).

The focus of this project is on (1) development of a negotiation agent, (2) integration of the software agent with a web-based negotiation support system, and (3) provision of a platform for future research.

The negotiation software agent is portable and can be easily implemented with similar NSSs (e.g. the INSS system). Portability and flexibility was an essential issue during our design considerations, since the agent should be adaptable to other versions of Inspire cases.

The current Aspire prototype consists of a limited number of rules, thus the negotiation support is quite rigid. However, this has proven that the use of software agent technologies may have a positive impact on web-based negotiations. This could be tested further with other experiments when further rules are added to the knowledge base.

Section 11.1 presents the limitations of these research studies. A few issues restricted the research process, as well as the development of the system. Future directions of work are discussed in Section 11.2.

11.1. Limitations of this study

As with many research projects, there are always time and budget constraints. The same applies when we select our development tools, we tried as much as possible to use freeware and open source software. For this purpose, we have selected Jess and Mysql. There are better technologies but they are not readily available and often very expensive

(for example IBM WebSphere costs app. \$40,000, Oracle, DB2 for database, IBM Blaze for rule processing). Even if these tools become available, time restriction would be another issue due to the compatibility with the existing systems and the learning curve of these new technologies.

Another technological restriction is related to the existing Inspire system. Since Inspire was developed in 1996 and the engine is mainly written in C++, there have been difficulties with communication between the Inspire system and Atin. Another issue is the consistency of information processes. The original HTML form in Inspire submits information via the Common Gateway Interface (CGI), while the new ones use PHP. Furthermore, since most of the Inspire dynamic pages are piped together by the engine, PHP scripts were unable to run on certain pages.

In this study we concentrated on development and implementation issues. We will conduct preliminary testing of the software. Further, because of the focus on the design and development of an integrated environment, less focus is expected on the development and testing of a comprehensive knowledge base for Atin. Such a knowledge base is a large research project on its own, since it requires literature studies, and data mining on available negotiation data. Due to time constraints, we have reasonably decided to limit the scope of this project. As discussed in Section 6, the main deliverable for this project will be a working prototype for the Aspire system. There are also various research topics and studies that could be carried out after the completion of this project, however these are not within the scope of this study.

11.2. Future work

The development of the Atin prototype, and the general feedbacks from the users confirmed our assumption that, a negotiation software agent will be a useful feature to support web-based negotiations. At present Atin does not have adequate knowledge to provide further support to the users. We continue working on making the negotiation knowledge base more comprehensive.

In the next version of the agent we will emphasis knowledge base development and variable degrees of autonomy. Several levels of autonomy would allow the user to choose from various assistance levels, ranging from inactive to fully autonomous. We also plan to revise the Inspire system to accept reservation level, aspiration level, and BATNA values. Although Atin requests such information from the user, it would be more logical for the NSS to request such information so that Atin can access them too. The rationale behind this alternation is that users will then be required to define their bottom lines and possible alternatives (even without Atin support), thus making the goal of the negotiation clearer.

References

- Amazon, <http://www.amazon.com>, October 2000.
- Andersen Consulting, <http://bf.cstar.ac.com/bf>, December 1997.
- Applegate, L. M. and C. W. Holsapple (1996). "Electronic Commerce: Building Blocks and New Business Opportunity." Journal of Organizational Computing and Electronic Commerce 6(1): 1-10.
- Arnaut, G. (2001). "Getting back to Business." Report on Business. July.
- Arpinar, S., A. Dogac, et al. (2000). "An Open Electronic Marketplace through Agent-based Workflows: MOPPET".
- Downloaded from <http://www.metu.edu.tr/papers/mopson.ps>
- AuctionBot, <http://auction.eecs.umich.edu>, October 2000.
- Barnes&Nobles, <http://www.barnesandnoble.com>, November, 2000.
- Bazerman, M. H., T. Magliozzi, et al. (1985). "The Acquisition of an Integrative Response in a Competitive Market." Organizational Behaviour and Human Performance 34: 294-313.
- Bazerman, M. H. and M. A. Neale (1982). "Improving Negotiation Effectiveness under final Offer Arbitration: The role of selection and training." Journal of Applied Psychology 67: 543-548.
- Bazermann, M. H. (1998). Judgement in Managerial Decision Making. New York, John Wiley & Sons Inc.
- Beam, C. and A. Segev (1997). Automated Negotiations: A Survey of the State of the Art. Berkeley, CA, Haas School of Business.
- Beam, C. and A. Segev (1998). Auctions on the Internet: A Field Study. Berkeley, University of California.
- Beam, C., A. Segev, et al. (1996). Electronic Negotiation through Internet-based Auctions. Berkeley, University of California.

- Bellucci, E. and J. Zeleznikow (1998). A Comparative Study of Negotiation Decision Support Systems. Bundoora, Victoria, Database Research Laboratory, Applied Computing Research Institute, La Trobe University.
- Bemberger, J (1997). Essence of the Capability Maturity Model. IEEE Computer, June: p. 112-114.
- Benbasat, I., F. Lim, et al. (1995). "A Framework for Communication Support in Group Work with Special Reference to Negotiation Systems." Group Decision and Negotiation 4(2): 135-158.
- Benoucef, M., H. Alj, et al. (2001). "Combined Negotiations in E-Commerce: Concepts and Architecture." Electronic Commerce Research Journal 1(3): 277-299.
- Benyoucef, M. and R. Keller (2000). "A Conceptual Architecture for a Combined Negotiation Support System". Downloaded from
<http://iro.umontreal.ca/~labgelo/Publications/Papers/dexa-2000.pdf>
- Bigus, J. P. and J. Bigus (1998). Constructing Intelligent Agents with Java. New York, Wiley Computer Publishing.
- Bradshaw, J. M. (1997). "An Introduction to Software Agents." in Software Agents. J. M. Bradshaw (Ed.), Cambridge, AAAI Press/The MIT Press.
- Branco, F. (1997). "The Design of Multidimensional Auctions." Rand Journal of Economics. 28: 63-81.
- Bui, T. (1994). Evaluating Negotiation Support Systems: A Conceptualization. 27 th Annual Hawaii International Conference on System Science, Hawaii.
- Caglayan, A. and C. Harrison (1997). Agent Sourcebook. New York, John Wiley & Sons, Inc.
- Castagnetto, J., H. Rawat, et al. (1999). Professional PHP Programming. Birmingham, Wrox.
- Chaffee, A. (2000). "One, two, three or n tiers?" Java World. Jan: 1-6.
- Chang, D. and D. Harkey (1998). Client/Server Data Access with Java and XML. New York, John Wiley & Sons, Inc.

Che, Y. (1992). "Design Competition through Multidimensional Auctions." Rand Journal of Economics 24: 668-679.

Cisco Systems Inc, <http://www.cisco.com>, November 2000.

CommerceNet, <http://www.commerce.net>, November 2000.

Conallen, J. (2000). Building Web Applications with UML. Upper Saddle River, Addison-Wesley.

Cray, D (1999). David's Automatic Negotiation Analyser (DANA). InterNeg. Carleton University, Ottawa.

CyberSettle, <http://www.cybersettle.com>, November 2000.

Faratin, P., C. Sierra, et al. (1998). "Negotiation Decision Functions for Autonomous Agents."

Robotics and Autonomous Systems 24: 159-182.

Fisher, R., W. Ury, et al. (1991). Getting to Yes: Negotiating Agreement Without Giving In. New York, Penguin Books.

Fournier, R. (1998). A Methodology for Client/Server and Web Application Development. Upper Saddle River, Prentice Hall.

Frictionless Commerce, <http://frictionlesscommerce.com>, November 2000.

Friedman-Hill, E. J. (1997). Jess, The Java Expert System Shell. Livemore, Sandia National

Laboratories. Downloaded from <http://herzberg.ca.sandia.gov/jess/docs/index.html>.

Gazis, D. (1998). PASHAs: Advanced Intelligent Agents in the Service of Electronic Commerce. in The Future of the Electronic Marketplace. D. Leebaert (Ed.), Cambridge, The MIT Press: 145-176.

Graham, J. L., A. T. Mintu, et al. (1994). "Explorations of Negotiation Behaviors in Ten Foreign Cultures Using a Model Developed in the United States." Management Science 40(1): 72-95.

Greenstein, M. and T. Feinman (1999). Electronic Commerce: Security Risk Management and Control. New York, McGraw-Hill.

- Guttman, R. H. and P. Maes (1998a). Agent -mediated Integratize Negotiation for Retail Electronic Commerce. Workshop on Agent Mediated Electronic Trading (AMET'98), Minneapolis, MN.
- Guttman, R. H. and P. Maes (1998b). Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce. Second International Workshop on Cooperative Information Agents (CIA'98), Paris.
- Guttman, R. H., A. G. Moukas, et al. (1998). "Agent-mediated Electronic Commerce: A Survey." Knowledge Engineering Review 13(3).
- Haugender, H. and D. Steiner (1997). "Co-operating Agents: Concepts and Applications", in N. R. Jennings and M. J. Wooldridge (Ed.), Agent Technology: Foundations, Applications, and Markets, Springer, 175-202.
- Heck, E. v. and P. Vervest (1998). "How Should CIOs Deal with Web-Based Auction?" Communications of the ACM 41(7): 99-100.
- Holsapple, C. W. and A. B. Winston (1996). Decision Support Systems: A Knowledge-Based Approach, West Publishing Company.
- IBM , <http://www-3.ibm.com/ibm/terminology/goc/gocmain.htm>, Apr 2001.
- Inspire, <http://www.interneg.org/inspire>, November 2000.
- InterNeg, <http://www.interneg.org>, November 2000.
- Jalote, P (2000). CMM in Practice. Processes for Executing Software Projects at InfoSys. Reading, Addison Wesley Longman Inc.
- Jango, <http://www.jango.com>, November 2000.
- Jennings, N. R. and M. J. Wooldridge (1998). Applications of Intelligent Agents. Agent Technology: Foundations, Applications, and Markets. N. R. Jennings and M. J. Wooldridge. Berlin, Springer-Verlag: 3-28.
- Jess, <http://herzberg.ca.sandia.gov/jess/>, November 2000.
- Kalakota, R. and A. B. Winston (1996). Frontiers of Electronic Commerce. New York, Addison-Wesley Publishing Company, Inc.
- Kersten, G. E. (1998a). Automated Negotiation via an Intelligent Agent. <http://www.iiasa.ac.at/Research/DAS/research/ecom/ecneg/>.

- Kersten, G. E. (1998b). Negotiation Support Systems and Negotiating Agents. Modèles et Systèmes Multi-Agents pour la Gestion de l'Environnement et des Territoires, Cemagref, ENGREF, Clermont-Ferrand, France.
- Kersten, G. E. (2000). Modeling Distributive and Integrative Negotiations. Review and Revised Characterization, InterNeg. Available at:
<http://interneg.org/interneg/research/papers/2000/02.pdf>.
- Kersten, G. E. and G. Lo (2001). Negotiation Support Systems and Software Agents in E-business Negotiations. Proceedings of the First International Conference on Electronic Business, Hong Kong. Available at:
<http://interneg.org/interneg/research/papers/2001/05.pdf>.
- Kersten, G. E. and G. R. Mallory (1999). "Rational Inefficient Compromises in Negotiation." Journal of Multicriteria Decision Analysis 8(2): 106-112.
- Kersten, G. E. and S. Noronha (1998a). Negotiations and the Internet: Users' Expectations and Acceptance, InterNeg. Available at:
<http://interneg.org/interneg/research/papers/1998/01.pdf>
- Kersten, G. E. and S. Noronha (1998b). "Negotiation via the World Wide Web: A Cross-cultural Study of Decision Making." Group Decision and Negotiation 8(3): 252-279.
- Kersten, G. E. and S. J. Noronha (1999a). Negotiations in Electronic Commerce: Methodological Misconceptions and a Resolution. InterNeg. Available at
<http://interneg.org/interneg/research/papers/1999/02.pdf>.
- Kersten, G. E. and S. J. Noronha (1999b). "Supporting International Negotiation with a WWW-Based System." Decision Support Systems 25: 135-154.
- Kersten, G. E., S. J. Noronha, et al. (2000). Are All E-Commerce Negotiations Auctions? COOP'2000: Fourth International Conference on the Design of Cooperative Systems, Sophia-Antipolis, France.
- Kersten, G. E. and S. Szpakowicz (1998). Modelling Business Negotiations for Electronic Commerce. The 7th Workshop on Intelligent Information Systems, Warsaw.
- Kilgour, D. M. (1996). "Negotiation Support Using The Decision-Support System GMCR." Group Decision and Negotiation 5(4-6): 371-383.
- Kosiur, D. (1997). Understanding Electronic Commerce. Redmond, Microsoft Press.

- Kumar, M. and S. I. Feldman (1998). Business Negotiations on the Internet. Yorktown Heights, IBM Institute of Advanced Commerce.
- Lax, D. and J. Sebenius (1986). The Manager As Negotiator. New York, The Free Press.
- LiveExchange, <http://www.moai.com/solutions/LiveExchange>, November 2000.
- Lo, G. and G. E. Kersten (1999). Negotiation in E-Commerce: Integrating Negotiation Support and Software Agent Technologies. Proceedings of the 29th Atlantic Schools of Business Conference, Halifax, (CDROM).
- Lo, G. and G. E. Kersten (2000). Conducting Negotiations via Web-based Systems: Experiences and New Developments. 42nd National Conference of the Canadian Operational Research Society, Edmonton, Alberta.
- Loshin, P. and P. Murphy (1997). Electronic Commerce: On-line Ordering and Digital Money. Rockland, Charles River Media.
- Maes, P. (1998). "Artificial Life meets entertainment: Life like autonomous agents." Communications of the ACM 38(11): 108-114.
- Maes, P., R. H. Guttman, et al. (1999). "Agents that Buy and Sell: Transforming Commerce as we Know it." Communication of the ACM 42(3): 81-91.
- Mahadevan, A. (1999). Using Data Mining to describe the outcome of Web-based Negotiations. School of Business. Ottawa, Carleton University.
- MarketMaker, <http://ecommerce.media.mit.edu/maker/maker.htm>, November 2000.
- May, P. (2000). The Business of ECommerce: From Corporate Strategy to Technology. Cambridge, Cambridge University Press.
- McConnell, S. (1996). Rapid Development. Redmond, Microsoft Press.
- Morris, J. and P. Maes (2000). Negotiating Beyond the Bid Price. CHI 2000 Workshop Proceedings: Designing Interactive Systems for 1-to-1 E-Commerce, The Hague.
- Moukas, A., R. H. Guttman, et al. (1998). Agent-mediated Electronic Commerce: An MIT Media Laboratory Perspective. First International Conference on Electronic Commerce (ICEC'98), Seoul, Korea.
- Mysql, <http://www.mysql.org>, November 2000.

OOTC (1997). Developing Object-Oriented Software: An Experience-Based Approach. Upper Saddle River, Prentice Hall.

Optimark, <http://optimark.com>, November 2000.

Orfali, R., D. Harkey, et al. (1999). Client/Server Survival Guide. New York, John Wiley & Sons, Inc.

Paulk, M., C. Weber, et al (1995). The Capability Maturity Model: Guidelines for Improving the Software Process (The SEI Series). Reading, Addison-Wesley: p.15-17.

PHP, <http://www.php.net>, November 2000.

Pressman, R. (1992). Software Engineering: A Practitioner's Approach. New York, McGraw-Hill, Inc.

Priceline, <http://www.priceline.com>, November 2000.

Pruitt, D. G. and S. A. Lewis (1977). "The Psychology of Integrative Bargaining." in Negotiations: Social-Psychological Perspectives. D. Druckman (Ed.), London, Sage Publications: 416.

Raiffa, H. (1985). "Post-Settlement Settlements." Negotiation Journal 1: 9-12.

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

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

Robotics Institute, <http://www.cs.cmu.edu/afs/cs/usr/katia/www/persuader.html>, November 2000.

Rubin, J. Z. and F. E. A. Sander (1988). "When Should We Use Agents? Direct vs. Representative Negotiation." Negotiation Journal 4(4): 395-401.

Sandholm, T. (1999). "Automated Negotiation." Communications of the ACM 42(3): 84-85.

- Sandholm, T. and V. Lesser (1995). Issue in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework. First International Conference on Multi-Agent Systems, San Francisco, California, AAAI Press/The MIT Press.
- Sandholm, T. W. (1999). Distributed Rational Decision Making. Mutliagent Systems: A Modern Approach to Distributed Artificial Intelligence. G. Weiss. Cambridge, The MIT Press.
- Satzinger, J. W., R. B. Jackson, et al. (2000). Systems Analysis and Design in a Changing World. Cambridge, Course Technology.
- Schmid, H. A. (1999). "An Architecture for Internet Business Applications with Business Components." in Software Architectures: Advances and Application. L. Barroca, J. Hall and P. Hall (Ed.), Milton Keynes, Springer-Verlag.
- SEI, <http://www.sei.cmu.edu/cmm/cmms/cmms.html>, July 2001.
- Sodhi, J. and P. Sodhi (1996). Object-Oriented Methods for Software Development. New York, McGraw-Hill.
- Strobel, M. (1999). On Auctions as the Negotiation Paradigm of Electronic Markets. Ruschlikon, IBM Zurich Research Laboratory: 7.
- Tedeshi, B. (1999). "Real Force in E-Commerce is Business-toBusiness sales." New York Times, E-Commerce Report.
- Teich, J., H. Wallenius, et al. (1998). "Multiple Issue Auction and Market Algorithms for the World Wide Web." Decision Support Systems, 1999. 26: 49-66.
- Tete-a-tete, <http://ecommerce.media.mit.edu/Tete-a-Tete>, November 2000.
- The Mediator, <http://www.mcn.org/c/rsurratt/conflict.html>, November 2000.
- Thieseen, E. M. and D. P. Loucks (1992). Computer Assisted Negotiation of Multiobjective Water Resources Conflicts. Ithaca, NY, Cornell University.
- Thompson, C. (1999). "Agents and E-commerce." Report on Business. September.
- Thompson, L. (1998). The Mind and Heart of the Negotiator. Upper Saddle River, Prentice-Hall Inc.
- Torsun, I. S. (1995). Foundations of Intelligent Knowledge-Based Systems. Bradford, Academic Press Inc.

TradeAccess, <http://www.tradeaccess.com>, November 2000.

Treese, G. W. and L. C. Stewart (1998). Designing Systems For Internet Commerce. Reading, Addison Wesley Longman.

URL-Minder, <http://www.netmind.com/url-minder>, November 2000.

Ury, W. (1991). Getting Past No. New York, Bantam Books.

Walton, R. E. and R. B. McKersie (1965). A Behavioral Theory of Labour Negotiations: An Analysis of a Social Interaction System. New York, McGraw-Hill.

Wang, S. (1999). "Analyzing Agents for Electronic Commerce." Information Systems Management Winter: 40-47.

Wirth, R. and C. Shearer (1997). "Towards process-oriented tool support for Knowledge Discovery in databases." in Principles of Data Mining and Knowledge Discovery: First European Symposium. J. Komorowski and J. Zytkow (Ed.), Springer-Verlag.

Wooldridge, M. (1999). "Intelligent Agents." in Muliagent Systems: A Modern Approach to Distributed Artificial Intelligence. G. Weiss. Cambridge (Ed.), The MIT Press: 27-78.

APPENDIX A: Project outline and intended development process

Please note that Table 6 documents the broad outlines of the development process that is to be used for this project only.

Phase/Activity	(Expected) Start Date	(Expected) Completion Date	Notes
Software Concept	Sept 1, 1999	Oct 31, 1999	Develop a software agent to support Inspire
General System Requirements	Nov 1, 1999	Jan 31, 2000	Agent support user during web-based negotiations
Study Inspire Architecture	Feb 20, 2000	Mar 15, 2000	Identify the components needed to be modify
Define Requirement	Mar 20, 2000	Apr 15, 2000	Provide methodological support to users in different phases
Requirement Analysis	Apr 16, 2000	Apr 30, 2000	Use cases, and object interaction diagrams
Research on Development Tools (technologies)	Apr 16, 2000	Jun 30, 2000	PHP, Mysql, Apache, Java
System analysis	Apr 16, 2000	May 15, 2000	Defined roles of NSS & NSA
Architectural Design	May 1, 2000	July 30, 2000	Architectural diagram
Modification of Inspire	Aug 1, 2000	Aug 10, 2000	Re-compiled Inspire-A
Prototype Development	Aug 11, 2000	Sep 10, 2000	Tested
Frontend design	Sep 1, 2000	Sep 15, 2000	Atin interface developed.
Database design	Sep 16, 2000	Oct 10, 2000	User database & negotiation database
Phase 1 design specification	Oct 11, 2000	Oct 20, 2000	Sample knowledge base
Phase 1 Implementation	Nov 20, 2000	Jan 10, 2001	Completed
Phase 1 testing	Jan 18, 2001	Jan 27, 2001	Testing passed.
Phase 1 Wrap Up	Jan 28, 2001	Feb 2, 2001	Done.
Phase 2 design specification	Feb 4, 2001	Feb 28, 2001	Discussion and rules implementation.
Phase 2 implementation	Mar 19, 2001	Mar 28, 2001	Coding
Phase 2 testing	Apr 10, 2001	May 20, 2001	Debug – problem with Inspire pages. Solved.
Phase 2 Wrap Up	May 28, 2001	Jun 10, 2001	Done.
Phase 3 design specification	Jun 12, 2001	Jun 20, 2001	Discussion with Gregory
Phase 3 implementation	Jun 25, 2001	July 17, 2001	Coding
Phase 3 testing	July 23, 2001	July 27, 2001	Testing passed.
Overall refinement	July 27, 2001	July 29, 2001	FE touch up
Final System Integration	July 30, 2001	Aug 5, 2001	
Final Testing	Aug 8, 2001	Aug 18, 2001	Tested
Deliver Aspire system	Aug 29, 2001		Trial negotiations

Table 6. Project schedule

APPENDIX B: Example of negotiation rules for the knowledge base

The sample rules below are designed for the pre-negotiation phase, for prototyping purpose only.

1. Issue rating

Let w, x, y, z be the weight assigned to the four negotiable issues respectively.

IF $w=x=y=z$

Then print

“When assigning weights for issues in this case you judged them all to be equally important. You may well have thought that this was so although the case indicates that some issues are more important than the others. One of the most important steps in successful negotiation is to understand your own preferences. You may want re-assign the weightings by considering which issues are more important and which can be treated more flexibly.”

IF w or x or y or $z > 70$

Then print

“When indicating your preferences on the issues in this case you placed a weight of $[x]$ on $[]$. This may, of course, reflect your view of the case material but the various issues were designed to be more evenly weighted even though some are clearly more than others. For both strategic and tactical reasons most experienced negotiators try not to tie themselves to a single issue. Doing so leaves them vulnerable to a counterpart who recognize

2. Option rating

Each side is told the preferred direction, e.g., if Cypress then “the lower price the better”. This means that the maximum value should be assigned to the lowest price value.

Let x be the value assigned to the issue.

And let $y =$ *Your preferences appear to be inconsistent with the preferred values of the company which you represent in this negotiation. (More)*

Cypress:

if rating(3.47) $\neq x$ then y

if rating(4.37) $\neq 0$ then y

if rating(20) $\neq x$ then y

if rating(60) $\neq 0$ then y

if rating(60 days after delivery) $\neq x$ then y

if rating(Upon delivery) $\neq 0$ then y

if rating(Full price) $\neq x$ then say y

if rating(Ten percent) $\neq 0$ then y

Itex

if rating(3.47) $\neq 0$ then y

if rating(4.37) $\neq x$ then y

if rating(20) $\neq 0$ then y

if rating(60) $\neq x$ then y

if rating(60 days after delivery) $\neq$ 0 then y

if rating(Upon delivery) $\neq$ x then y

if rating(Full price) $\neq$ 0 then say y

if rating(Ten percent) $\neq$ x then y

Additional explanation.

More: The best (worst) option should have the highest (lowest) value reflecting the option's preference. Some negotiators do not assign such a value because they do not think this option can be included in the compromise or because they negotiate to achieve a high rating (score) rather than a compromise that is good for the company they represent. Note that the rating is solely for your own use and the score of the compromise you achieve should not be used to evaluate your negotiation.

3. Package Ratings:

After packages' ratings was computed:

Let y = This the best possible package for your company therefore it's rating should be maximum, that is 100. (More1)

Let z = This the worst possible package for your company therefore it's rating should be minimum, that is 0. (More2)

Check if:

Cypress:

If

Rating for package (3.47 \$; 20 days; 60 days after delivery; Full price) $\neq$ 100

Then y

If

Rating for package (4.37\$; 60 days; Upon delivery; ten percent) $\neq 0$

Then z

Itex:

If

Rating for package (4.37\$; 60 days; Upon delivery; ten percent) $\neq 0$

Then y

If

Rating for package (3.47 \$; 20 days; 60 days after delivery; Full price) $\neq 100$

Then z

More1: The best package must have the highest rating. If not then there should be another offer that is better for the company.

More2: The worst package must have the highest rating. If not then there should be another offer that is worse for the company.

After user modified the package ratings:

Check if min value for the package rating < -5 .

Option 1. min value = 0 – continue

Option 2. min value ≥ -10 and less than zero, then recalculate the ratings using the following formula:

Old rating = a, Min rating = x ($-10 < z < 0$) then

New rating = $(a - x) * 100 / (100 + x)$ (note that x is negative)

Option 2. min value < - 10 then

Rule:

If min rating value < - 10

Then

You made significant changes of the ratings for the displayed packages. These changes contradict the ratings you entered for the issues and options and now the worst package rating is smaller than -10 (and not zero). Please revise ratings that you have entered. ([More](#))

More. Package ratings are displayed to help you verify if the issue and option ratings you have done earlier produces correct overall rating for the displayed packages. You may want to modify some ratings and then ratings of all the packages are recalculated. Your changes may be drastic. If, for example, you changed a rating from 80 to 10) then some packages that had low ratings might have now negative rating. This should be modified so that all ratings of packages fall between 0 and 100. This is because your initial rating was done through distribution of positive values and the negative values indicate inconsistency.

If the user ignores Agent's suggestion, i.e., resubmits rating and again the value is less than -10, then the following message is displayed:

In the negotiation supported with Inspire ratings plays an important role and it cannot negative. While some error is allowable (app. 105) the Inspire system cannot continue if it detects strong contradiction between the ratings entered for the issues and options and ratings that you have entered for some of packages. ([More](#))

More. In order to reduce the inconsistency you may modify the ratings for individual issues and/or options so that the package ratings reflect your preferences. Alternatively, you may make the changes in the package-rating table less drastic.

The above message keeps popping up until the min package rating is ≥ -10 .

About the negotiator and process

1. How would you describe your negotiation style and approach to conflict? (Select one that best applies)

Competitive OR

Accommodating OR

Collaborating OR

Avoiding

If *competitive* then:

You may consider making the first offer AND

Your offer may be extreme (giving app. 100 utility value) AND

You should explain in the accompanying message the rationale for such a high offer.

2a. If the opponent has already sent first offer then

send an extreme offer AND a message

clarifying why the offer is not acceptable if the opponent's first offer yield's very low utility value (less than 10) AND

sending most extreme offer you have.

OR

thanking for the offer AND

indicating your strong willingness to continue negotiations AND

explaining and justifying your extreme offer (if utility is between 10 and 30).

OR

thanking for the offer AND

indicating to continue negotiations AND

asking for general improvement AND

explaining why these improvements are necessary (if utility is over 30).

If accommodating then:

You represent accompany and should not give up potentially very good opportunity. Consider using either the collaborating or competitive style.

If avoiding then:

You represent accompany and should not give up potentially very good opportunity. Consider using either the collaborating or competitive style.

If collaborating then:

Begin with a message rather than an offer AND

Indicate your willingness to discuss the cooperation between both organizations
AND

Indicate that you represent interests of the company AND

**State what is for you very important, but to not provide too much information
AND**

Ask what is more important and what less to your counterpart.

5a. If the opponent has already sent first offer then send a message

**IF the opponent's first offer yield's very low utility value (between 10 and 30)
thank for the offer clarifying why the offer is not acceptable AND**

ask the counterpart for the statement about their interests AND

mention your willingness to collaborate and achieve a good compromise.

OR

**IF the opponent's first offer yield's medium utility value (less than 10) thank for
the offer AND**

clarifying why the offer is not acceptable AND

state what is for you very important, AND

do not provide too much information AND

state your willingness to collaborate and achieve a good compromise AND

ask the counterpart for the statement about their interests

OR

IF the opponent's first offer yield's medium utility value (more than 30) thank for the offer AND

clarifying why the offer is not acceptable AND

state what is for you very important AND

consider giving some specific information AND

ask the counterpart for the statement about their interests AND

stress your willingness to collaborate and achieve a good compromise for both parties AND

you may want to consider sending an offer with the utility value between 90 and 70.

General rule for competitive negotiations

Discuss the problems of the counterpart's offer AND

Reply to the counterpart's concession with a concession as small as possible without making your counterpart angry AND

Highlight the importance of your concession AND

Indicate your willingness to continue negotiations

General rule for collaborative negotiations

Be open and request the counterpart's open discussion of their interests and preferences AND

Try to clarify the interests and preferences AND

Search for common ground, for the issues which two parties share AND

**Look for possibility to add issues or add new offers which are better for both sides
AND**

**Make significant concessions if required, match or even better the counterpart's
concessions but expect the same from them.**

First message – a general rule may contain these points (or some of them)

Introduce yourself and your company

Discuss the process

Deadline

Who is going to begin first

Frequency of offers and messages

Ask about counterpart's expectations

Emphasize the importance of getting agreement together

Give your general views on the negotiations

Mention your underlying concerns

Every message – a general rule

Summarize what the counterpart stated AND

State your point of view on the counterparts' statement

Message accompanying offer – a general rule

Justify your offer AND

Show how your offer is an improvement (if possible) AND

Be positive

Concession making and a message accompanying offer – a general rule

State your reasoning for making a concession AND

State that you make conditional concessions AND

State what you expect for your concession (“We agreed on ... and expect that you accept ...”)

The agent has always check if BATNA and reservation levels are violated

IF your counterpart does not reply

THEN send a message AND

ask if there is any problem AND

indicate how long you are willing to wait.

IF your counterpart does not reply AND

you had sent a message (rule 13) AND

you can set new negotiations

THEN send a message AND

ask if there is any problem AND

give firm deadline explaining why you cannot wait

IF you perceive a deadlock (other than no replies)

THEN summarize the achieved progress AND

reiterate your willingness to achieve a compromise AND
ask your counterpart for their suggestions.

Quantitative rules for concession making and round t :

IF $\text{Change}(t) = \text{Utility}(\text{counterpart offer}, t) - \text{Utility}(\text{counterpart offer}, t-1)$ AND
 $\text{Change}(t) > 0$

THEN Select an offer for which:

$\text{Utility}(\text{your offer}, t) \geq \text{Utility}(\text{your offer}, t-1) - \text{Change}(t)$.

IF negotiations are competitive AND
 $\text{Change}(t) = \text{Utility}(\text{counterpart offer}, t) - \text{Utility}(\text{counterpart offer}, t-1)$ AND
 $\text{Change}(t) < 0$

THEN do not make an offer AND

ask your counterpart if she/he indeed made a concession AND
consider collaborative negotiations BECAUSE
there is a possibility that some of the parties interests overlap.

IF negotiations are collaborative AND
 $\text{Change}(t) = \text{Utility}(\text{counterpart offer}, t) - \text{Utility}(\text{counterpart offer}, t-1)$ AND
 $\text{Change}(t) < 0$
THEN do not make an offer AND
explain your counterpart that the current offer is worse for you than the previous offer
AND
state why the current offer is worse AND
ask your counterpart for the reasons behind the new offer.

APPENDIX C: Aspire Review and Evaluation Form

1. Did you use the Inspire system before using Aspire? (Yes/No)

If no, please proceed to question 2.

If yes:

- a). How many times did you use Inspire?
- b). When was the last time you use Inspire?
- c). Did you achieve an agreement?
- d). If you achieved an agreement, were you satisfied with it?
- e). Comparing Aspire to the original Inspire system, you find Aspire:
(Easier to use/More difficult to use)
(Providing more support/less support)

2. Did you receive any warning/alert from Atin during the negotiation?
If yes, were the warnings useful? Please explain.

3. Have you used any features of Atin? (Yes/No)

If yes:

- a). Did Atin's suggestions affect your decision? Please explain.
- b). What is your overall comment on Atin's features?

4. Were you satisfied with the way this negotiation proceeded? Please explain.

5. Did you feel being in control of the negotiation?

6. Did you reach an agreement with your opponent in this negotiation?

7. Did you use post-settlement?

8. Which feature(s) of Aspire do you like most?

9. Which feature(s) of Aspire do you like least?

10. What is your overall comment of Aspire?

APPENDIX D: Screenshots

Figure 22. Aspire first screen

Figure 23. Atin alerts user when option ratings are violated

Figure 24. Atin alerts user when issue rating is violated

Figure 25. Atin allows user to enter reservation level

The screenshot shows a web browser window with the address bar displaying "http://atinsys.org/inspire-new/inspire/inspire/2.4/select...". The main content area has a title "Define the best Alternative that is different to the negotiated package." and a prompt "Please Enter alternative here." followed by a large text input field. Below this, there is a label "BATNA Rating for the Alternative" and a small input field. At the bottom right of the form are two buttons: "Submit" and "Close".

Figure 26. Atin allows user to enter BATNA

https://internet.org/cgi-bin/inspire/2.2b10atin/save/package_ratings...
 File Edit View Favorites Tools Help
 Back Forward Stop Search Favorites History
 Address http://internet.org/cgi-bin/inspire/2.2b10atin/save/package_ratings

INSPIRE INSPIRE Questionnaire Evaluation Information Package Rating Package Rating

Status of your soso negotiation with mark
 with deadline Wed, 5 Sep 2001 23:59:59 GMT

Atin Advisor to INSPIRE Negotiator

1. Negotiation case ✓
 2. Preparation ✓
 - Issue assessment ✓
 - Option assessment ✓
 - Package evaluation ✓
 3. Negotiation
 - Ask Atin for advice.
 - Make an offer
 - Evaluate your offer
 4. Post-settlement

Hello soso!

[Ask Atin Now!](#)

Make an offer

Welcome to the negotiation phase submit your reservation and BAT to help you in the negotiation, but

You can now send an offer to you from the table below. If you wish, Please remember that each offer counterpart chooses to accept it. If you do not wish to make an offer

Price
 Delivery
 Payment

Atin suggests:

You have selected Competitive negotiation style
 You may consider making the first offer AND
 Your offer may be extreme (giving app. 100 utility value) AND
 You should explain in the accompanying message the rationale for such a higher offer.

General rule for competitive negotiations
 Discuss the problems of the counterpart's offer AND
 Reply to the counterpart's concession with a concession as small as possible without making your counterpart angry AND
 Highlight the importance of your concession AND
 Indicate your willingness to continue negotiations

Internet

Figure 27. An example of Atin suggestion during negotiation phase

Figure 28. Atin alerts user regarding large concessions between offers