uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Shikui Wu

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE / DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Design and implementation of InterNeg Support System in the Invite software platform

TITRE DE LA THÈSE / TITLE OF THESIS

Gregory Kersten

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Morad Benyoucef

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Stan Matwin

Wojtek Michalowski

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

Design and implementation of InterNeg Support System in the Invite software platform

Shikui Wu

A thesis submitted to the Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements for the degree of

Master of Science

Supervisor: Professor Gregory E. Kersten
Professor Morad Benyoucef

Systems Science
University of Ottawa
Ottawa, Ontario, Canada K1N 6N5

Your file Votre référence
ISBN: 0-494-11459-2
Our file Notre référence
ISBN: 0-494-11459-2

Abstract

The rapid development of e-commerce raises the need of systems that support electronic negotiations. Recent work on negotiation support systems concentrates on web-based negotiation support tools.

As part of the InterNeg project, Inspire, which is an e-negotiation system to conduct bilateral negotiations, has been developed and deployed. The system contains a facility for specification of preferences and assessment of offers, an internal messaging system, and graphical displays of the negotiation's progress. Inspire has a number of limitations which are necessary in order to conduct longitudinal experiments but which make the system unusable in real-life situations. These limitations include the fixed number of the negotiation issues and options, and the inability to add more participants to the negotiation table.

INSS (InterNeg Support System) is an e-negotiation system that allows to use different negotiation strategies and tactics, and to negotiate over open and dynamically modifiable problems. It is based on Inspire but enhanced and expanded with some new features for more practical negotiations. The focus of this paper is on the design and implementation of INSS and its key features, including: (1) the construction of a negotiation protocol which allows negotiators to formulate their own negotiation case online, to specify the process and the permissible activities of its participants; (2) the introduction of the problem specification and modification components to construct INSS in the Invite platform; (3) the specification of different types of issues and options which can be introduced by the users and interpolated (or extrapolated) by the system; and (4) the establishment of a negotiation case library.

To implement the system, we apply Fusebox technology prototyping and rapid application development (RAD). The INSS negotiation protocol and all its components are implemented in Invite system with database-driven approach. The system's usability will be tested with a small group of participants in different contexts defined by problem complexity and users' characteristics.

Dedication

To my parents and my wife Cuiping, for their love, encouragements and support.

Acknowledgement

This research could not have been conducted without the support of many people. First of all, I would like to express my appreciation to my supervisors, Dr. Gregory Kersten and Dr. Morad Benyoucef, for their continuous encouragement, helpful advice and considerable patience during my completion of this thesis.

I have to thank all my colleagues at the InterNeg research group, in particular, the Invite development group. I would like to acknowledge the help and contribution of Mr. Ka Pong Law, Dr. Jin Beak Kim and Dr. Stefan Strecker to the development and integration of the Invite system and the INSS negotiation protocol.

I would also thank the two examiners of my thesis defense, Dr. Wojtek Michalowski from School of Management and Dr. Stan Matwin from School of Information Technology and Engineering (SITE), for their great attention and valuable comments and suggestions.

Finally, special thanks to the participants for usability testing of INSS, who spent a lot of time and gave many valuable suggestions.

Table of Contents

List of Figures

List of Tables

Glossary

Alternatives
A generic name for the possible choices that a negotiator may choose from. There may be alternative issues, options and packages. In decision science, alternatives are possible decisions, or elements of the set of alternatives, from which the decision maker may select.

Compromise
A compromise or agreement is the package (combination of options) across all issues that both negotiators jointly agree upon after exchanging a sequence of offers.

Inefficient agreement
An alternative (package) that is accepted by the parties but which is dominated by some other alternative, that is there is an alternative that is better for at least one party than the accepted agreement and not worse for any other party.

Issue
A topic of discussion that is of particular interest in a negotiation. Each issue has a range of alternatives or options, one of which must ultimately be agreed upon by the negotiators in order to achieve a compromise.

NegoDance graph
The exchange of offers and counter-offers can be seen only from one negotiator's perspective (as in the History of Offers graph) or, if both negotiators agree, from the perspective of both of them. One dimension presents ratings of one negotiator and the second dimension those of the second negotiator. With each offer, ratings of both negotiators are associated. Thus you may review the changes in the ratings (concessions) accepted by each negotiator.

Negotiation
Negotiation is a process that involves two or more agents who communicate for the purpose of exchanging some goods and defining the terms of the exchange.

Offer
A combination of options (a package) that is sent by one negotiator to the other. An offer may contain one option for each issue under considerations, or some issues may not be present in the offer.

Option
One of the alternative values that an issue can take. For example, the issue "Tolerable product failure rate" may have the options "3%", "5%" and "10%".

Package
A particular combination of options that has been selected across several or all the negotiated issues. An example of a package is given in the table below:

Price	$3000
Payment	Upon delivery
Failure rate	5%

Pre-negotiation
Pre-negotiation is the first phase of a negotiation. It refers to the initial period (prior to exchange of any offers) when one prepares for the negotiation. Some activities involved in this phase include problem definition, preference elicitation and evaluation of alternative packages.

Post-settlement
A "settlement" is the same as an agreement or compromise, and "post-settlement" refers to the period after the first compromise has been achieved.

Rating
Rating is the core activity of pre-negotiation preparation. The purpose of rating is to use numerical expression to clarify negotiator's preference and utility level of individual negotiation issue, option and package. In INSS, the user is required to specify these three ratings according to personal preference and knowledge. The utility function of individual negotiator is generated not only from the composition of the issue and option ratings but also from the decomposition of package evaluation. According to the utility function, any adjustment of the package evaluation may lead to variation of utility level of other packages.

Utility function
A "utility function" is a subjective measurement that expresses the relative value (worth) of different packages (alternatives) by using a numerical scale. The numerical scale used is arbitrary, typically ranging either from 0 to 1 or from 1 to 100. The minimum number expresses the least desirable and least preferred package. The highest number represents the most desirable and preferred package.

1. Introduction

1.1 Motivation

Information and communication technologies (ICTs) have dramatically altered the way that business is transacted. Particularly, with the adoption of the Internet, many enterprises have established their presence on virtual markets and more and more business transactions have been conducted over the Web. The term *electronic commerce* (e-commerce) has been coined to describe the very large scope of such transactions. An increasing number of other business functions, such as marketing, procurement, planning and coordination, and personnel and human resources, being conducted on the Web the term *electronic business* (e-business) has been introduced to distinguish commercial transactions from other business activities.

Negotiation has traditionally played an important role in all types of commerce activities taking place in physical markets. Computer-based support tools, such as negotiation support systems (NSSs) and decision support systems (DSSs), have been designed to support negotiators, mediators, and arbitrators improve the efficiency and effectiveness of negotiation activities (Bui 1994; Holsapple and Whinston 1996; Kilgour, Fang et al. 1996; Kersten and Lo 2003). These systems provide two types of support in the negotiation process: (1) decision support, and (2) communication support (Benbasat and Lim 1995; Kersten 1997).

With the rapid and significant progress of ICTs and the growth of e-commerce and e-business, it is possible and desirable to support negotiations conducted over the Web – this is commonly called *electronic negotiations* (e-negotiations). The multiple-issue, e-commerce negotiations have been supported by Web-based electronic messaging (EM), Web-based negotiation support systems (NSS), (Kersten and Noronha 1999) and autonomous electronic bargaining agents (EBA) (Chavez and Maes 1996; Oliver 1997). Recent work on NSSs concentrates on Web-based negotiation support tools.

From the perspective of the overall role and behavior in negotiations, Kersten (2004) categorized NSSs into three broad classes: (1) passive systems, (2) active facilitative-mediation systems, and (3) proactive intervention-mediation systems. As part of the InterNeg research program (http://InterNeg.org), several tools which correspond to these classes have been implemented within the SSHRC project on e-negotiation (http://InterNeg.org/enegotiation), that is, respectively, SimpleNS, Inspire, and Aspire.

SimpleNS system provides a virtual negotiation table allowing its users to exchange text-based offers and messages. It facilitates communication among negotiators but it does not provide any type of decision support, e.g. analytical, visualization, and simulation (Kersten, Law et al. 2004).

Inspire, which is an e-negotiation system to conduct bilateral negotiations, has been developed in 1995. The system contains a facility for specification of preferences and assessment of offers, an internal messaging system, and graphical displays of the negotiation's progress (Kersten and Noronha 1999). Inspire has a number of limitations which are necessary in order to conduct longitudinal experiments, but which make the system unusable in real-life situations. These limitations include the pre-setup phase for each negotiation, fixed number of the negotiation issues and options, and the inability to add more participants to the negotiation table.

Aspire is an ENS that combines decision support with software agents, thus providing advice to negotiators (Kersten and Lo 2003). It is an example of proactive intervention-mediation systems which extends Inspire with a software agent called Atin. The Atin negotiation software agent retrieves information from the negotiation knowledge base, and provides advice to the negotiator.

Other academic and commercial systems provide support by addressing specific negotiation problems with various techniques (Shim and Hsiao 1999). For example, WebNS (Yuan, Rose et al. 1998), which is a Java-based Web NSS, mainly facilitates communication during the negotiation process and allows the parties to discuss each issue separately. SmartSettle (www.smartsettle.com), a commercial version of ICANS (Thiessen and Loucks 1994; Thiessen, Loucks et al. 1998), provides extensive decision-analytic support and allows for the negotiation on multiple issues simultaneously. Negoisst (Schoop and Quix 2001), uses a document-oriented approach with semi-structured messages to form the final contract. There are also systems that allow human experts to be involved in claiming and assessing offers (e.g. www.cybertsettle.com, www.electronicourthouse.com).

From the above, it follows that some of the existing Web-based NSSs facilitate communication or document-generation, some also provide analytical support, and others involve software agents or human experts as advisors. Although such literature and experimental works have shown the achievements in e-negotiation support, in terms of the negotiation problem formulation, currently in most of these systems users conduct negotiation using a fixed structure with pre-defined issues. However, in real-life, it is normal for negotiators to bring new issues to the table during the negotiation process. Whereas it is

possible to allow changing issues (e.g. adding a new issue and dropping an existing issue) in a few systems, the intervention of human being (e.g. the system administrator, or a mediator) is required. Thus, there is a lack of a system which is able to support the negotiators directly in their dynamic formulation of the negotiation problems.

Taking into account the number of negotiators involved in the process, Stroebel and Weinhardt (2002)) note that most systems currently only support one-to-one bilateral negotiations. Even though some systems have been designed to support complex negotiations, such as multi-bilateral negotiations and multilateral negotiations, they support only the communication activities. That is, their support is limited to the provision of either dialogue windows or a joined session area. In some cases, the third-party mediators need to participate in the negotiation (Yuan, Rose et al. 1998).

In the situations in which human mediators are not employed, the system acts as a mediator and determines a compromise proposal. It may also, if needed, suggest other alternatives obtained by reducing the same amount of satisfaction (e.g., utility value) for each party (Thiessen, Loucks et al. 1998). These support models may be seen as too mechanistic in that the negotiators cannot participate in the discussion regarding their concessions (e.g., decrease of utility value) and search for innovative solutions. This, in turn, may decrease the effectiveness of negotiation and limit the use of the system. Thus, other approaches need to be considered to cope with complex negotiations.

The definition of e-negotiations has been recently broadened and includes auctions conducted with the use of electronic media (Bichler, Kersten et al. 2003).

E-negotiations are supported by *e-negotiation systems* (ENSs): softwares that employ Internet technologies, are deployed on the Web, and are capable of supporting, aiding or replacing one or more negotiators, mediators or facilitators (Ehtamo, Hamalainen et al. 2004). Kersten (2004) pointed out that these more comprehensive definitions allow researchers to include email, chat and streaming video (Moore, Kurtzberg et al. 1999; Lempereur 2004), and software for automated negotiations and auctions (Zlotkin 1996; Jennings, Faratin et al. 2001). Because of the difficulty in distinguishing between auctions and other, non-auction processes (for instance, those based on the traditional face-to-face negotiations), a distinction is made here between online auctions and e-negotiations. The former are based on auction models and the latter describes online versions of traditional negotiation and dispute resolution processes. Both may be significantly modified and enhanced to account for the specificity of the employed technologies.

A number of different ENSs and models embedded in these systems have been proposed to deal with more complex negotiation scenarios, including multi-bilateral negotiations and multilateral negotiations (Shim and Hsiao 1999; Neumann, Benyoucef et al. 2003). Some of these systems use standard auctions or multi-attribute auctions (e.g., AuctionBot.com, eAuctionHouse.com); some translate negotiation structure into auction platform (Reeves, Wellman et al. 2001); and others combine negotiation and auction mechanisms (GNP, NegotiAuction). There are also systems that provide such support by introducing multi-agents systems (ITA, Transducer, AMTRAS). While all the literature and systems above show the ideas to support users to perform more complex online business transactions, it is a fact that in most of these systems the negotiation problem has to be pre-defined or well-defined. This is because the systems either involve auction protocols or employ agents to act on the behalf of negotiators.

The rapid development of e-commerce makes business transactions directly via the vendor's Websites insufficient because of the very large number of, often unknown, vendors. This led to intermediaries, including *electronic markets* (e-markets). They emerged for the procurement of various kinds of products and services online. Currently auction-based transactions are most popular in e-marketplace, e.g. eBay.com, Yahoo Auctions (http://auctions.yahoo.com/) and Amazon Auctions (http://auctions.amazon.com/). This is due to their implementation and integration in e-commerce applications with several appealing features, such as process efficiency, ease of use, and ability to simultaneously manage very large numbers of bidders (Kersten and Lo 2003).

Most of the auction systems are either based on the standard auctions in which bidding competition is restricted to only the price dimension, or multi-attribute auctions which provide models and tools to solve the limitation of standard auctions (Vulkan and Jennings 1999). Furthermore, auctions use rigid protocols and are limited to one-side communication for business transactions, whereas negotiation is a process to exchange more information with interactions between two (or more) sides and its protocol is more flexible. More importantly, a lasting relationship among negotiators established during the process may contribute to potential business transactions. Thus, several e-marketplaces that provide negotiation mechanisms have been proposed and implemented, such as Kasbah and Tete-a-Tete (Guttman, Moukas et al. 1998; Maes, Guttman et al. 1999).

Online auctions have been considered as an efficient market mechanism in that they allowed for effective matching of many potential buyers and sellers, and for the establishment—through market competition—of prices which matched demand with supply. In contrast, e-negotiations have been seen as a necessary but significantly less efficient mechanism. E-

negotiations are necessary for reasons given above but less efficient than auctions because they involve a small group which may discuss issues only tangentially related to the product and its attributes. Recent studies show however, that in some situations e-negotiation systems may be more efficient market mechanisms than auctions (Thomas and Wilson 2002; Thomas and Wilson 2005).

Chen, Kersten et al. (2005) proposed an e-marketplace for agent-supported commerce negotiations, called eAgora. This e-marketplace allows buyers and sellers to conduct multi-issue negotiations by using a simple bargaining protocol. It allows users to host new negotiations or join existing negotiations in order to pursue their goals, i.e. buying or selling products. eAgora makes it possible for a user to host more than one negotiation to sell the same or different products at the same time. Furthermore, the user can also join other negotiations as buyer to procure a product. The scenarios considered by Chen, Kersten and Vahidov brought e-marketplace concepts that create new opportunities for the implementation of more complex negotiations – for example, those in which one person is engaged in both multi-bilateral and multilateral negotiations.

Users of eAgora are allowed to add new issues and options during the negotiation even though the number of new options and issues is limited. This increases the flexibility of the negotiations. Because eAgora was designed and implemented as an e-marketplace, there are also several limitations, including (1) focus on online product trading; (2) the negotiation problem is not pre-defined, but the flexibility is limited with the fixed maximum number of new issues and options by database design; and (3) simple and fixed negotiation protocol, which makes the system extensibility, as well and its comparability with other e-negotiation systems for further research experiments, difficult.

1.2 Contributions

The purpose of this thesis is to design and implement an e-negotiation system, InterNeg Support System (INSS), which allows negotiators to use different negotiation strategies and tactics, and to negotiate over open and dynamically modifiable problems. The proposed system is based on the Inspire system (Kersten and Noronha 1999), but it is significantly enhanced and expanded with several new features that aim at supporting more practical negotiations, including:

1. The construction of a negotiation protocol which allows negotiators to formulate their own negotiation case online, to specify the process and the permissible activities of its participants;

2. The introduction of the problem specification and modification components to construct INSS in the Invite platform;
3. The specification of different types of issues and options which can be introduced by the users and interpolated (or extrapolated) by the system; and
4. The establishment of a negotiation case library.

To implement the system, we apply Fusebox technology prototyping and rapid application development (RAD) based on current version of Inspire. The INSS negotiation protocol and all of its components are implemented in the Invite system, which is a software platform for multi-protocol e-negotiations (Kersten, Law et al. 2004). Invite contains a database-driven approach for the design of Web-based systems that allow for customizable researcher- or user-definable protocols (Kersten and Lai 2005). The system's usability will be tested with a small group of participants in different contexts defined by problem complexity and users' characteristics.

2. E-commerce and business negotiations

2.1 E-commerce and e-business

The infusion of ICTs into business operations is dramatically changing the way business is conducted. As the Internet has grown to become a business infrastructure, many companies established their Websites to perform the business activities.

Electronic commerce (e-commerce) is an approach to conduct commercial transactions over electronic channels, especially on the Internet. Compared to the traditional business, e-commerce takes advantage of the dramatically growing Internet to enhance the speed and lower the cost of business transactions. Nowadays, e-commerce represents an interdisciplinary field concerning marketing, computer science, consumer behavior and psychology, finance, economic, management information system, accounting and auditing, management, as well as law and ethics (Turban, Lee et al. 1999).

In an e-commerce enabled world, different criteria can be used to classify e-commerce applications. For example, e-commerce is classified by the type of participants in transactions (Schmid and Lindemann 1998) into the three following types:

1. Business-to-business (B2B) refers to all transactions made between organizations;
2. Business-to-consumer (B2C) is a retailing approach to sell to the individuals;
3. Consumer-to-consumer (C2C) makes transactions between consumers.

These transactions are conducted over the Internet in the way which is different from traditional business, such as e-catalogs, e-stores, online auctions, e-payment, etc. However, the key activities of the value-chain in business remained the same. Business deals are often made through negotiations. Negotiation has been seen as a key activity of the business transactions in e-commerce.

Typically, a commerce model includes activities such as advertising, searching, negotiating, ordering, delivering, paying, using, and servicing. The model is shown in Figure 2.1 (Eriksson, Finne et al. 1999). Producers and sellers advertise their goods and services by creating catalogs or stores on the Web to attract customers. On the other hand, buyers and consumers look for a specific product that fulfills their needs by browsing the catalogs or using search engines. Once they "meet each other" on the e-market and the buyers recognize that their demand can be met by the sellers' offering, they negotiate on the terms for exchange, such as price, quantities, delivery days, payment method, etc.

Currently, auctions are a very popular e-commerce market mechanism, especially in B2C and C2C transactions. After the buyers and sellers reach an agreement on the terms (or contract), the buyer pursues orders and payments and the seller processes the orders and delivers the products to the buyer. These activities are supported by ICTs with e-ordering systems, e-payment systems, EDI (electronic data interchange), and so on. Finally, in the post-sale stage, product using and servicing are conducted.

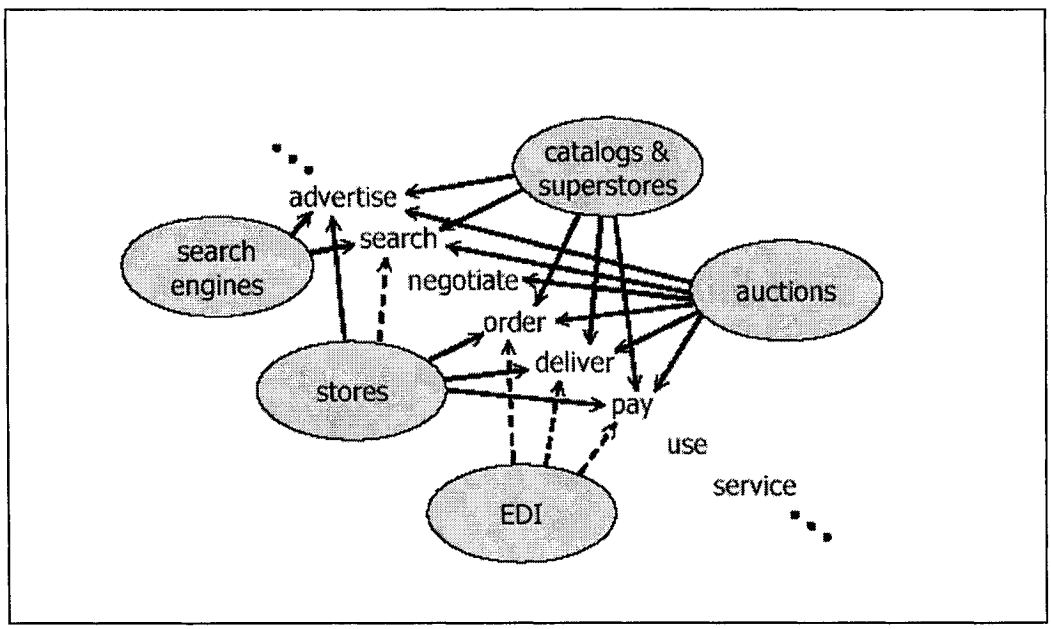

Figure 2.1 Commerce on the Internet (Eriksson, Finne et al. 1999)

Maes, Guttman et al. (1999) put negotiation at the center of the consumer buying behavior model for e-commerce. The model identifies the following six stages in the buying process: (1) need identification, (2) product brokering, which determines what to buy, (3) merchant brokering, which determines from whom to buy, (4) negotiation, where the terms of the transaction are settled, (5) purchase and delivery, and (6) product service and evaluation.

Negotiation is also seen as an important link in the e-commerce value chain. Figure 2.2 shows the value chain model for B2C e-commerce (Feldman 1999). Note that it can be extended for other types of e-commerce. For example, another business can be the consumer to buy and use the product from the producer (B2B), and the producer also can be a seller to sell the product to an individual consumer (C2C).

With the spread of digitization and the Internet, much of the value has shifted from the product itself to information about the product. This information often shapes the business transaction (Wise and Morrison 2000). As more and more information can be separated and exchanged electronically, various applications and tools have been integrated on the e-

commerce platform to facilitate and enhance business transactions online. For more efficient and effective exchange, it is natural to support negotiation on the Internet.

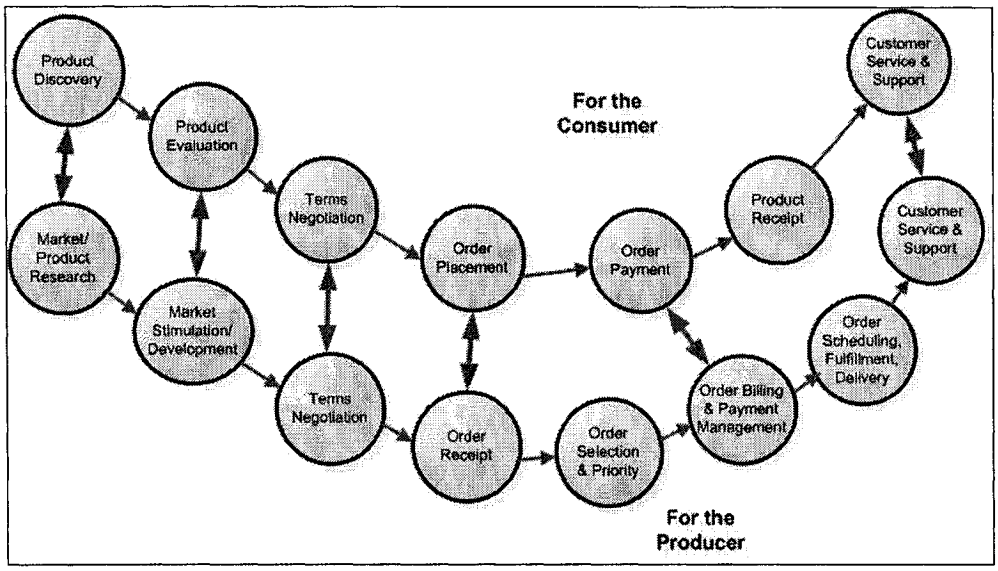

Figure 2.2 E-commerce value chain (Feldman 1999)

By using the financial services industry as an example of advanced B2B solution, Wise and Morrison (2000) identified four major trends of B2B commerce: (1) from simple to complex transactions, (2) from middlemen to speculators, (3) from transactions to solutions, and (4) from buyer-seller exchange transactions to sell-side asset swaps. They pointed out that value tends to accumulate among a diverse group of specialists that focus on such tasks packaging, standard setting, arbitrage, and information management. As the key link in e-commerce value chain and with rich information exchange between buyer and seller, negotiation will play a more and more important role in e-commerce.

2.2 E-marketplace

An e-marketplace is a transaction platform of e-commerce. A traditional marketplace is a physical space in which buyers and seller meet. It is established to facilitate the exchange of tangible and intangible goods and services, such as information, products, and payments. As an extension, e-marketplaces retain all main functions of traditional marketplaces. The key difference is that e-marketplaces provide virtual meeting space and utilize information technologies to enhance all marketing functions: matching demand and supply; assisting the exchange in a business transaction; and providing a legal and regulatory framework for the market (Bakos 1998).

As in the physical world, with the rapid development of e-commerce, business transaction directly via the vendor's Websites is insufficient and an intermediary is required. Thus, e-markets emerged to perform various kinds of product or service online trading. With the aid of ICTs, e-marketplaces have multiplied. In e-marketplace, consumers (companies, individuals and governments) can deal with businesses around the world, evaluate offers, undertake negotiations and realize successful deals.

Recently, the application of online auction mechanisms as a way to resolve the negotiation-phase has attracted numerous companies as they have been able to achieve enormous benefits. In fact, several such commercial systems that provide buyers and sellers with auction mechanisms are well known and widely used, such as eBay, OnSale, Yahoo Auctions, and Amazon Auctions. However, most of these e-marketplaces support only limited negotiation functions using a one-dimensional variable of price.

Wise and Morrison (2000) pointed out the following three flaws of auction and exchange-based B2B markets.

1. The value proposition offered by most exchanges – competitive bidding among suppliers allows buyers to get the lowest possible prices – runs counter to the best recent thinking on buyer-supplier relations, such as joint product-design efforts, integration of complex processes, and long-term service contracts.
2. The exchanges deliver little benefit to sellers. To be successful in the long run, B2B markets need to offer strong incentives to both buyers and sellers; and
3. The business models of most B2B exchanges are "half-baked." In their rush to get online, the companies that run the exchanges haven no study their customers' priorities in-depth, created distinctive offerings, or even map out paths to profitability. They have simply used off-the-shelf software to set up simple auctions as quickly as possible. As the markets mature, these problems have to be fixed and new structures will enable buyers and suppliers to form tight relationships while still enjoying the reach and efficiency of e-commerce.

Applied research and experimental studies apply online multi-attribute auction systems to resolve the restriction of standard auctions. For example, MAMENS (Cho 2001) uses a new multi-attribute auction model with evolutionary negotiation support in the e-marketplace. Considering the differences between negotiation and auctions, including flexible protocols, rich information, effective communication and interaction, and lasting relationship, bargaining and negotiation mechanisms have been proposed to enrich the services provided by e-marketplaces. Kasbah and Tete-a-Tete are two experimental e-markets that allow

customers to negotiate by using software agents on their behalf (Guttman, Moukas et al. 1998; Maes, Guttman et al. 1999). However, these two agent-based e-marketplaces only support negotiating with a fixed-structured and well-defined negotiation problem.

Another example of an e-marketplace supported by negotiation software agents is eAgora (Chen, Kersten et al. 2005). In eAgora, the software agent acts as an advisor to the buyer (seller) and helps identifying and assessing offers. The agent can also provide critique of offers made by the buyer (seller) and their counterparts. The role of the agents available on eAgora is similar to the Atin in Aspire as mentioned in the introduction, in that they support the negotiators rather than negotiate on their behalf, as for example in Kasbah and Tete-a-Tete (Guttman, Moukas et al. 1998; Maes, Guttman et al. 1999).

The eAgora e-marketplace adopted a five-phase negotiation process model proposed by Kersten (1997) and coves four phases: agenda setting, exploring the field, narrowing the differences and searching for agreement, and agreement's assessment (see Figure 2.3). The negotiations in eAgora can involve multiple issues (such as price, delivery, warranty, etc), which may be set before or during the negotiation process.

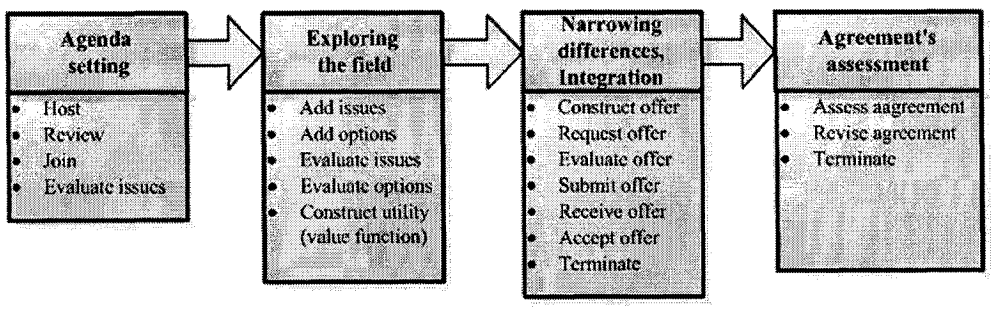

Figure 2.3 eAgora negotiation phase model and activities (Chen, Kersten et al. 2003)

eAgora provides a virtual market for buyers and sellers to meet and negotiate over various products. In fact, the goods listed in the market are negotiating objects. Thus, eAgora members can start by either hosting a new negotiation to sell or buy something, namely host; or joining one existing negotiation to be a guest counterpart (or negotiation partner) if the negotiating object is what the member wants to procure or supply. As a host, the member decides on the required issues, which the negotiation partner bids on. However, the issues, proposed after talks have started, must be accepted by both parties to be added on as required issues. The host always offers the first bid, and every bid is solidified such that it cannot be withdrawn. When an offer is received, the negotiation partner can decide to accept it, make a counter-offer, propose new issues, or even terminate negotiation. An example of the eAgora protocol is given in Figure 2.4; it illustrates a typical one-to-one bargaining.

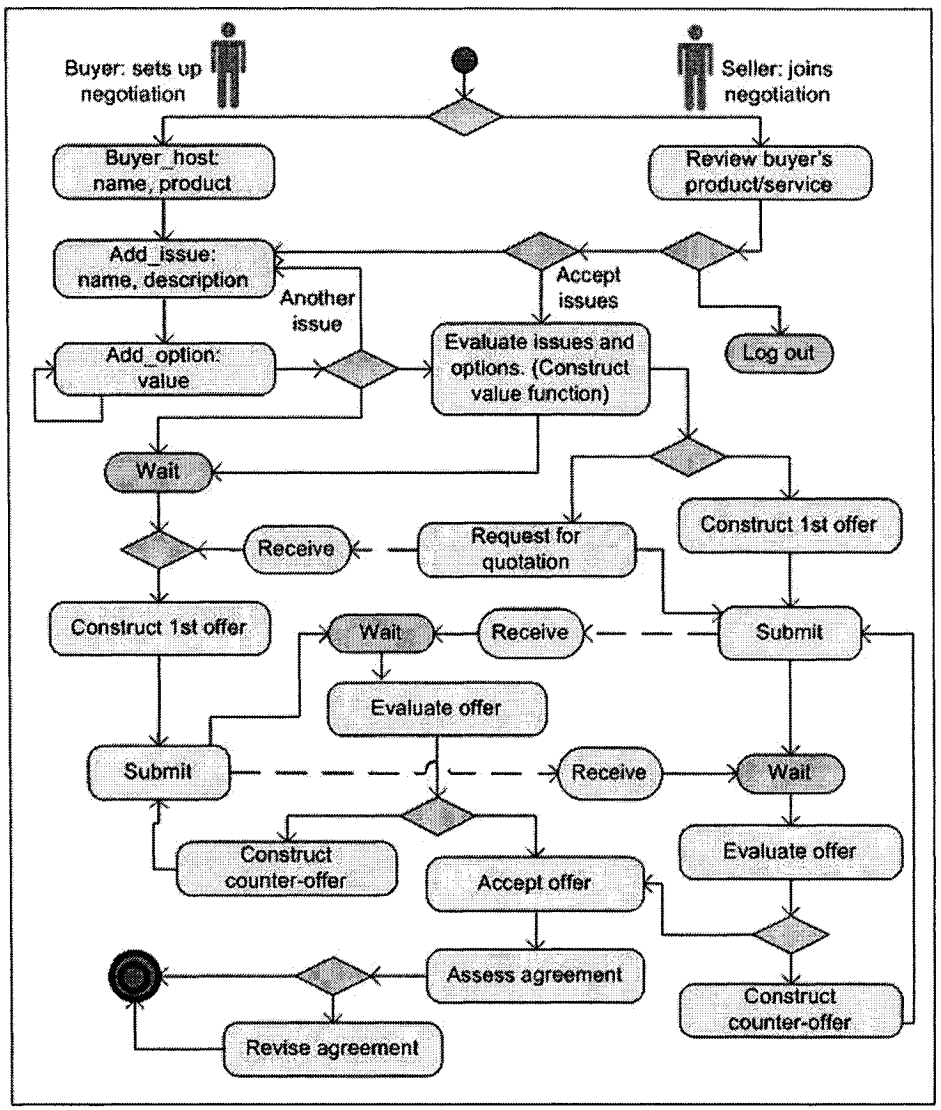

Figure 2.4 An example of the eAgora protocol (Chen, Kersten et al. 2003)

Although based on the bargaining protocol, participants in eAgora only conduct one-to-one bilateral negotiations, the market mechanism and several concepts (e.g. hosting and joining) make it possible to conduct other types of negotiations, such as multi-bilateral and multi-lateral negotiations. More importantly, the negotiators who use eAgora are allowed to add new issues and options during the negotiation even though the number is limited. This also provides a way to increase the flexibility of negotiation systems. However, because eAgora was designed and implemented as an e-marketplace, the limitations are as follows:

1. It is mainly for online product trading which narrows its using in more realistic negotiation fields;

2. The negotiation problem is not pre-defined, but the flexibility is limited with the fixed maximum number of new issues and options by database design;
3. It uses an embedded bargaining negotiation protocol, which limits the system extensibility and makes it is hard to compare with other ENSs for further research.

2.3 Auctions

Recently, automated negotiations and auctions have been the subject of the e-negotiation studies (Kersten 2004). This section presents four main approaches to auctions: online auctions, multi-attribute auctions, combination of negotiations and auctions, and automated negotiations with the use of multi-agent systems.

2.3.1 Online auctions

Online auctions have very small transaction costs and therefore they are used to conduct many transactions, especially C2C and B2C. As an effective approach in the buying and selling process, online auctions have been employed in the rapidly emerging Internet-based e-commerce platforms. Vakali and Angelis (2001) made a survey to outline research efforts in relation to online Internet-based auctions modeling and applications.

eMediator (Sandholm 1999) is the first Internet auction to support combinatorial bidding, bidding via price-quantity graphs, and mobile agents. It provides a number of mechanisms, such as eAuctionHouse, a configurable auction platform which determines the winners of the combinatorial auction, identifying profitable contracts for the participating agents.

MarketMaker and AuctionBot are two typical agent-based systems that seek mutual agreements on the terms of transactions that satisfy the participants' predefined constraints, preferences, and objectives. Agents in these systems engage in the information exchange activities that are typical to conventional auctions rather than context rich and complex negotiations (Guttman, Moukas et al. 1998; Maes, Guttman et al. 1999).

MarketMaker is a multi-agent system developed at the MIT Media Lab, which facilitates auctions in an electronic marketplace (Guttman and Maes 1998). 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. As all the auction-based system, it is rigid and limited on single attribute (price) transactions; hence the

communication process is very narrow. Instead of exchanging negotiation offers and information, the agent posts a new bid (upon the approval of the user) once the market information is updated.

The AuctionBot is a flexible, scalable, and robust auction server that supports both software and human agents (Wurman, Wellman et al. 1998). The server manages many simultaneous auctions by separating the interface from the core auction procedures. Its interface consists of two portions: a Web interface for human agents and a software agent API. With a large parameter space, various auctions can be flexibly specified and implemented on the server. As the particularity of auctions, all of these parameters need to be pre-defined. The use of agent also restricts the auctions in AuctionBot, for instance the number of participants, bidding with integer quantities, and new bid dominated previous bid.

2.3.2 Multi-attribute auctions

The advances in information technology allow the use of different and more complex auction mechanisms. To meet the requirements of different and more complicated auctions, some new and mixed protocols are proposed, such as the Anglo-Dutch auction (a hybrid of the sealed-bid and ascending auctions), double auction, and so on. On the other hand, more and more researches and practices on multi-attribute (or multidimensional, multi-issue) auctions provide models and tools to solve the limitation of standard auctions that bidding competition is restricted to the price dimension.

Vulkan and Jennings (1999) show how an English auction can be modified for services, which are multi-dimensional private value objects. Given a service as a pair: (p, S), p is a real number corresponding to the price of the service, and S is a vector of real numbers $S = (S_1, S_2, ..., S_n)$, where $S_1 = q$ is a measurement of the quality of the service, and S_2 to S_n are optional additional dimensions of the same service (such as time before service can be provided, level of support, etc.). Then, each offer or bid in auction can be composed of more than one attribute of the service.

Bichler and Werthner (2000) analyze and classify multi-attribute auctions, where firms bid on a number of attributes, and bids are evaluated by a scoring rule designed by a buyer. Also, they try to broaden the theory and define a general framework for the applicability in both single-unit and multi-unit negotiation situations.

2.3.3 Combination of negotiation and auction systems

While researchers are making great efforts on solving the limitation of traditional auctions above, such as multi-attribute auction, others are attempting to transfer negotiation into auctions, or combine various negotiations together, including auctions.

Reeves and Wellman (2001) proposed an approach for automating the negotiation of business contracts proceeds in the following three broad steps.

1. Determine the structure of the negotiation process by applying general knowledge about auctions and domain-specific knowledge about the contract subject along with preferences from potential buyers and sellers.
2. Translate the determined negotiation structure into an operational specification for an auction platform; and
3. Map the negotiation results to a final contract.

The three steps are implemented in a prototype which employs a declarative specification in Courteous Logic Programs (Reeves, Wellman et al. 2001). The proposed specification allows for the formulation of: (1) high-level knowledge about alternative negotiation structures, (2) general-case rules about auction parameters, (3) rules to map the auction parameters to a specific auction platform, and (4) special-case rules for subject domains. They demonstrated the flexibility of this approach by automatically generating several alternative negotiation structures.

Benyoucef and Alj (2001) devised a Combined Negotiation Support System (CNSS) to help the user conduct all the negotiations at the same time and designed such a CNSS, called CONSENSUS.

The architecture of CONSENSUS relies on workflow technology, negotiating software agents, and rule engine technology. In this architecture, the user models the combined negotiation at build time by using a workflow that captures the sequencing of the individual negotiations and the dependencies between them. At runtime, software agents are assigned to individual negotiations, and they participate in the combined negotiation as actors in the workflow. It relies on rule engine technology to enable the agents to use negotiation strategies. This work was being conducted as part of a project on e-marketplaces.

As a common infrastructure of the project, a generic negotiation platform (GNP) (Benyoucef, Keller et al. 2000) is developed to support various types of negotiations. It requires the negotiation to be described as a JPython23 script and a set of XML files, which together

provide the parameters, and the rules of the game. A formalization of auction rules is addressed to make GNP truly generic. The system provides a toolkit for negotiation designers to specify required information and various negotiation parameters (conditions for ending rounds, phases and the entire negotiation; information to be displayed to or hidden from the players, etc.). However, the negotiation designer should show both programming skills and economic market design knowledge.

Teich and Wallenius (2001) proposed NegotiAuction, an algorithmic Internet-based auction procedure, which combines certain elements of negotiations and auctions. It can be used either in reverse or forward auctions. Three modes of the auction are presented: auto mode, manual mode, and pause mode. The first mode is working in the standard auctions way but with multi-attribute, and the second one allows the auction owner to conduct one on one negotiation. The last option is used to place any or all of the bidders "on hold" during the auction. The author also mentioned that NegotiAuction could be considered to use when the transaction involves an important decision with large dollar stakes dealing with many product attributes in a B2B environment or in repetitive situations.

2.4 Negotiation

2.4.1 Negotiation and negotiation support

As seen in the previous section, negotiation is an important element in business transactions. It is a process in which two or more parties with different criteria, constraints, and preferences, jointly reach an agreement on the terms of a transaction (Walton and Krabbe 1995).

There are two main types of negotiation: *distributive negotiation* and *integrative negotiation* (Walton and McKersie 1965). Distributive negotiation means the parties involved are sharing a "fixed pie," also known as "win-lose" situation. Integrative negotiation involves the creation of value and openness of information exchange which ends up with a "win-win" situation.

The negotiation process can be distinguished as several phases, and several negotiation activities may be conducted in each phase. Based on behavioral studies of negotiations, several negotiation process models have been proposed (Gulliver 1979; Holmes 1992; Ghee-Soon Lim and Murnigham 1994; Schmid and Lechner 1999). Graham and Mintu (1994) suggest that there are three sequential phases in negotiation: the antecedent phase, the concurrent phase, and the consequent phase. These phases roughly correspond to pre-negotiation, negotiation, and post-negotiation (Kersten and Szpakowicz 1998; Raffia 1998).

Pre-negotiation refers to the initial, planning period. Before initiating a negotiation, negotiators prepare and identify their preferences. *Negotiation* phase is where the negotiation takes place, negotiators exchange offers and messages in order to reach an agreement. *Post-settlement* is the process where the agreement (contract) is verified and, if possible, improved, and finally its implementation devised. Note that the negotiators may undertake the activities in each phase many times and similarly loop through the phases themselves, e.g. the negotiator may revise his or her objectives and preference (the activities conducted in the first phase) when preparing and evaluating offers (which are the second phase activities).

A negotiation support system (NSS) integrates the support and media functions, and determines protocols used in the negotiation process. NSSs do not automate the negotiation but support the decision making procedure for reaching joint acceptance by providing communication means and analyzing available information (Bui and Shakun 2002; Kersten 2002; Schoop 2003).

A NSS is built on two supporting blocks, a decision support component and a communication support component (Benbasat and Lim 1995). Decision support components enhance the processing capabilities of information systems used for decision making by the negotiators. These systems help users by structuring negotiation problems through the use of decision analysis methodologies, such as problem analysis and definition, issue consideration, offer construction and its assessment, etc. Communication support components help to construct the support systems for passing, accepting and understanding messages; as well as forming, representing, and analyzing arguments.

Based on the role NSSs can play in the negotiation, Starke and Rangaswamy (1999) distinguished two types: (1) preparation and evaluation systems, and (2) process support systems. While the former mainly provides decision support, the latter focuses on communication support. NSS can be used by a single party, by all participating parties, or by a facilitator or mediator. They can be used in only one phase (e.g., pre-negotiation) or even in the selected activities of one phase. NSSs may also be used in several phases. Taking into account the user group and the phases of the negotiation process, Kersten (2004) proposed the following four functions of NSSs: (1) planning support; (2) assessment; (3) intervention; and (4) process support.

2.4.2 E-negotiation systems

The Internet popularity and the growing number of online business transactions attract more and more Web-based systems developed to facilitate and enhance various functions of e-

commerce. These systems developed for and deployed on the Internet differ from other information systems in the following several key aspects (Kersten 2004):

1. They are network-centric and rely on ever-present Internet connectivity;
2. They allow for tight integration of inter- and intra-enterprise business and a large number of people accessing systems from anyplace; and
3. Their user interface is provided by the Web browsers – it is easy to understand and common to many different applications.

The above characteristics differentiate traditional NSSs from the systems supporting negotiations conducted over the Web. Researchers have constructed NSSs to conduct negotiation on the Internet or to study negotiation models made possible by the Internet. Similarly as commerce and business conducted on the Web are known as e-commerce and e-business Web-based negotiations are called e-negotiations. Typically, the multiple-issue e-negotiations have been supported by several technologies, including: Web-based electronic messaging (EM), Web-based negotiation support systems (NSS) (Kersten and Noronha 1999), and autonomous negotiation software agents (NSA) (Chavez and Maes 1996; Oliver 1997).

In the last years, researchers have put more and more efforts on Web-based negotiation support tools. As introduced before, several such systems have been implemented within the SSHRC project on e-negotiation (http://InterNeg.org/enegotiation), including SimpleNS, Inspire and Aspire. From the perspective of the overall role and behavior in negotiations, Kersten (2004) categorized these systems into three broad classes: passive systems, active facilitative-mediation systems, and proactive intervention-mediation systems. The three classes are compared in Table 2.1.

There are also other systems which provide negotiation support by addressing various negotiation problems (Shim and Hsiao 1999). Here, a brief review of several selected systems is given as following:

WebNS is a Java-based NSS, which provides a structured process for multi-party negotiation (Yuan, Rose et al. 1998). It is based on a negotiation process model derived from Gulliver (1979) dividing the negotiation process into two main phases: preparation and offer exchange. While the system supports multi-party negotiation, it mainly facilitates communication support during negotiation process by using short messages, real-time chat, and video conference within dialogue windows. It does not provide any analytical negotiation support for the negotiators themselves.

Table 2.1 The functions, related activities and examples of ENSs (adapted from Kersten 2004)

	Functions	Activities	Examples
Passive systems	Transport and storage Search and retrieval Formatting, presentation and interaction	Communication, presentation and interaction	SimpleNS (see Section 3.3.4 and 5.3.1)
Active facilitative-mediation systems	Decision problem formulation Decision-maker specification Strategies and tactics	Decision problem and negotiators	Inspire (see Section 3.1 and 3.3.4)
	Offer and message construction and evaluation Counterpart analysis What-if, sensitivity and stability analyses	Process	
Proactive intervention-mediation systems	All above functions	Communication, presentation and interaction Decision problem and negotiators Process	Atin
	Process, history and their assessment Knowledge seeking and use Negotiation protocols Strategies and tactics	Knowledge	

WebNS facilitates multi-issue negotiations with the negotiators setting up issues before starting negotiation. In the offer exchange phase, they discuss each issue separately. When they reach an agreement on one issue, they discuss the next issue, and so on. Thus, it cannot be used in negotiations that evolve in time with issues changing (e.g. add new issue, drop issue) and the negotiations that need to consider the tradeoffs among issues.

SmartSettle (www.smartsettle.com), which is an extended and ported on the Web version of ICANS (Thiessen and Loucks 1994; Thiessen, Loucks et al. 1998), facilities e-negotiation support through the discussion of multiple issues simultaneously. These issues have to be agreed by all parties before conducting the negotiation. In SmartSettle, the negotiation is divided into six phases: prepare for negotiation, qualify interests, quantify satisfaction, establish equity, maximize benefits, and secure commitment (SmartSettle 2004).

Although it is possible to initiate the case for issues changing, a human mediator has to be involved instead of supporting the negotiators themselves. The developers claim that SmartSettle supports complex negotiations (e.g. multilateral negotiation) by using a joint session area to provide support for each party. However, it either requires the intervention of the third-party mediators or the system itself, which acts as a mediator to suggest alternatives by reducing each party's satisfactory. In order to use SmartSettle, a client software is required to be installed on the local computer.

Negoisst is a three-tier Java-based system. It attempts to enable complex negotiation between human negotiators, providing support through the complex negotiation process rather than through the automation of negotiations. A conceptual model called DOC.COM is introduced. It relates the document management component to the communication system (Schoop and Quix 2001). According to this model, each message initiates a new version of documents. Seven message types are specified according to Searle's classification of illocutionary force, including: request, offer, counter-offer, accept, reject, question and clarification.

In Negoisst, message exchanges follow a certain negotiation protocol. The protocol is based on a deterministic finite automaton and constraints are attached on some states. For example, the message will be a type of counter-offer, accept or reject if this message replies to a prior message where the type is offer. Moreover, semi-structured message contents have been obtained. In other words, the system defines categories for parts of the message content as well as a message body with free text remains. The system provides categories that can be selected to form the structure of a contract. Negotiators then choose negotiation issues from these categories before conducting a negotiation. These issues are displayed in the final contract as the contract items (Schoop and Quix 2001; Schoop 2003).

Based on the brief review above, although all the systems support e-negotiations with various techniques, two things are common among these systems limiting their use in real-life negotiations. On the one hand, from the perspective of negotiation problem formulation, the negotiation structure is either defined by the mediator or agreed by all negotiators before the negotiation begins. On the other hand, from the point of view of negotiation types, one-to-one bilateral negotiations can be conducted in most of the existing systems. Other types of negotiations (e.g. multi-bilateral and multilateral) are basically supported with communication technologies in which mediators (human-being or the systems) need to be involved. Hence, there is a need to design and implement an e-negotiation system which is both flexible to dynamically formulate negotiation problem during the process and capable of supporting negotiators to conduct more complex negotiations.

2.4.3 Multi-agent systems for one-to-many negotiations

With the flourishing of the software agent technology, multi-agent system (MAS) has been introduced into the frameworks of negotiation support. ITA (Rahwan, Kowalczyk et al. 2001) and Transducer (Paliwal, Adam et al. 2003) are two frameworks with multi-agents to perform one-to-many negotiations.

The ITA framework has been proposed to implement one-to-many negotiations by means of conducting a number of concurrent coordinated one-to-one negotiations. With several simple negotiation strategies which are similar to standard auctions, such as Dutch auction and English auction, the system coordinates and processes the one-to-one-negotiations. An instance of one-to-many negotiation scenarios is illustrated in Figure 2.5. In this particular scenario, a buyer-agent negotiates a deal with many prospective sellers. The buyer-agent consists of a coordinating agent and number of "sub-negotiators," called sub-buyers. All sub-buyers represent the preferences and constraints of the buyer, but they may use different negotiation strategies (Rahwan, Kowalczyk et al. 2001).

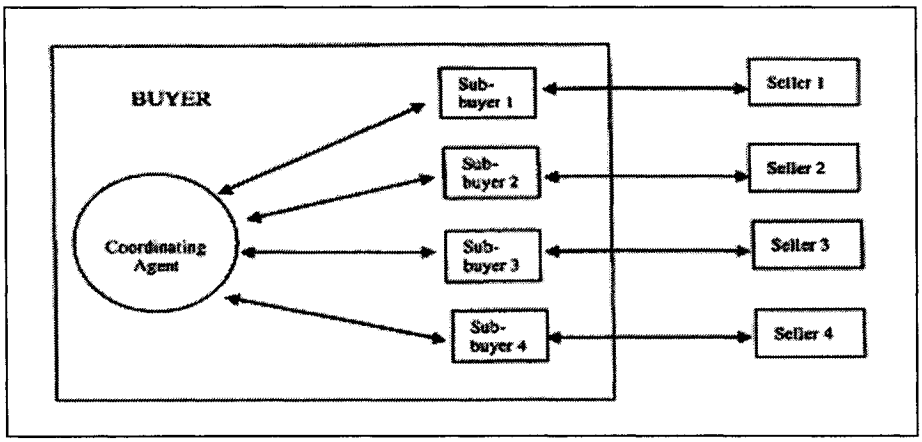

Figure 2.5 One-to-many negotiation (one buyer & many sellers)
(Rahwan, Kowalczyk et al. 2001)

A logic-based approach to multi-agent negotiation is presented by Torroni and Toni (2001). The advantages of such approach stem from the model declarative nature. The model allows to formulate and prove such interesting properties as the termination and convergence of a protocol. It also allows for the identification of agents and their combination or aggregation, implementation of different negotiation policies, and prediction of the behavior of a system with no need for simulation. The authors propose a formal language to describe negotiation that allows catering for two-agent dialogues, in a one-to-one negotiation setting.

Torroni and Toni (2001) consider auctions are an example of one-to-many negotiation mechanisms, where agents try to maximize their profit by buying items in competition with other parties, or selling them to crowds of bidders. Finally, they show the negotiation framework can be extended to accommodate a suitable negotiation language and coordination mechanism (in the form of a shared blackboard) to tackle one-to-many negotiation.

Summarily, in all of the above systems and approaches, some directly use standard auctions or multi-attribute auctions; some translate or combine negotiation and auction; others use multi-agent systems. However, the adoption of auctions and agents makes these systems hard to use for the negotiations in which the problems are ill-defined or evolve in time with issues changing.

2.4.4 Classification of e-negotiation scenarios

As many e-negotiation systems proposed and implemented as above, researchers have attempted to classify them in several ways. Earlier, Beam and Segev (1999) illustrated four representative online trading scenarios: bargaining, online auctions, and online marketplace with and without brokers (see Table 2.2).

Table 2.2 Summary of negotiation models (Beam, Segev et al. 1999)

Feature	Bargaining	Online auction	Online marketplace	Brokered marketplace
Registration	Not required	Required	May be required	Required
Authentication	Not available	Available	Required	Required
Intermediation	Not available	Yes, the auctioneer	No	Yes, the broker(s)
Offers	Not binding (not enforceable)	Binding	Binding	Binding
Number of parties involved	1:1	1:N	1:1	1:N; M:N
Type of goods	Probably easier with hard goods	Both information and hard goods	Both information and hard goods	Both information and hard goods
Attributes supported	All	Price only	Many	Many
Hot spots	Bargaining strategies	Auction rules and type	Bargaining strategies	Brokers deal making algorithms
Flexibility in rules	Much	None	As much as decided by the marketplace	As decided by brokers matchmaking
Optimal strategy	Unknown. Many suggestions available; solutions only for extremely limited problems.	Well-known for major auction types. Monopoly, oligopoly, perfect	Largely unknown. Depends on rules. Solutions available for very limited rules.	Algorithms Unknown. Possibly different for broker, buyer, and seller. Solutions difficult even for very simple situations.
Market structure	Perfect competition, perhaps with a little friction	Competition	Decentralized, perfect competition	Any; possible collection of power in the hands of the broker
Examples	Online flea market: yahoo classified ads	OnSale, eBay, FirstAuction	FastParts, Cypress Cycles[1]	NASDAQ as a continuous double auction; real-estate agents on line

[1] This is the name of a negotiation case used in Inspire.

Recently, a research on the state of the art e-negotiation systems have been conducted (Neumann, Benyoucef et al. 2003). They selected eight commercial and academic ENSs and analyzed by applying the Montreal Taxonomy (MT) (Stroebel and Weinhardt 2002), which is a comprehensive classification scheme for e-negotiations.

Based on Neumann, Benyoucef et al. (2003) comparison of the eight systems three selected concepts have been added and presented in Table 2.3.

Table 2.3 Comparison of ENSs with MT
(adapted from Neumann, Benyoucef et al. 2003)

Systems	**Overview**			**Roles**		**Offer specification**		
	Auction system	Multi-dimensional auction system	Bargaining system	Participation	Agents	Attributes	Values	Structure
eBay	X			Bilateral	Multiple	Single	Single	Fixed
LiveExchange		X		Bilateral	Multiple	Multiple	Single	Flexible
AuctionBot	X	X		Bilateral	Multiple	Single	Single	Fixed
GNP	X	X		Bilateral	One/ multiple	Single	Single	Fixed
Ozro Negotiate			Interactive Bargaining System	Bilateral	One	Multiple	Single	Flexible
Inspire			Mediation Systems	Bilateral	One	Multiple	Multiple	Fixed
AMTRAS	X	X	Interactive Bargaining System	Bilateral	Multiple	Multiple	Multiple	Fixed
eAuctionHouse	X	X		Bilateral	One/ multiple	Single/ multiple	Single/ multiple	
INSS			***Mediation system***	***Bilateral/ multibilateral***	***One/ multiple***	***Multiple***	***Multiple***	***Flexible***

Table 2.3 groups the attributes in three classes: overview, roles and offer specification. The overview group is based on the types of negotiation conducted in the systems; the roles group specifies the number of participants in the negotiation and how they are involved; the offer specification group indicates the number of attributes (issues) for each offer and the number of values (options) for each attribute.

In the process of design and analysis of ENSs, these classifications can be used to address which situation and what problem the system will apply or resolve. As indicated in bold at

the bottom of Table 2.3, the purpose of this paper is to design and implement an ENS, *INSS*, which supports various types of negotiations over open and dynamically modifiable problems. Based on the MT, it is a mediation system, and allows conducting one-to-one bilateral, one-to-many bilateral (multi-bilateral) and, possibly, multilateral negotiations on multi attributes, multi values with a flexible structure.

1. *Overview – mediation bargaining system.* Similar to Inspire, INSS users exchange offers and messages in the system for bargaining and negotiation rather than auctions. This is more practical and flexible for real-life negotiations, in particular, the problem that evolve in time with structures (e.g. issues) changing.
2. *Roles – bilateral/multibilateral participation and one/multiple agents*. By introducing several e-market concepts from eAgora, users in INSS can participate in more than one negotiation with different roles at the same time. For instance, one user hosts a negotiation to sell his or her product, and then the user can invite the target buyer who is already known to negotiate on the product; meanwhile, to decrease the market risk and attract more potential buyers the user may conduct another or more negotiations for the same product for sale. In this case, at first a one-to-one negotiation, and then a one-to-many (multibilateral) negotiation are conducted. If the user who is selling product also wants to buy something (e.g. parts), he or she can also be a buyer in other negotiations. In this supply chain situation, a multilateral negotiation may be performed.
3. *Offer specification – multiple attributes with multiple values and flexible structure.* Same as Inspire, the negotiations in INSS are involving several issues (*attribute*) and each issue contains at least 2 options (*value*). The offers are structured with user-selected options for every issue. As an extended and enhanced version of Inspire, INSS allows adding new issue and options during negotiation process. Thus, the structure of negotiation problem and offers can be changed.

Currently, Inspire has been implemented as one negotiation protocol in the Invite platform. To implement the proposed system, INSS, as another protocol in Invite with the features described above, the related work (Inspire system and Invite platform) and theoretical foundation for negotiation protocol will be introduced in the following chapter.

3. Invite and underlying protocol theory

Advancement of ICTs, on one hand and the popularity of Web-based application on the other hand lead researchers in decision and negotiation support concentrating on the design, implementation and testing of Web-based negotiation support tools. Within the InterNeg Group (InterNeg 2005), several e-negotiation systems have been developed and deployed on the Web (i.e., Inspire, eAgora, SimpleNS, and Aspire). These systems are used to study, teach, simulate, and facilitate e-negotiations. Currently, the Invite system, a software platform for multi-protocol e-negotiations, is under development and testing. Section 3.1 introduces the InterNeg project and Inspire negotiation support system as well as its support methodology. The theoretical foundations for the design of flexible and highly customizable protocol-driven ENSs are given in Section 3.2, and the Invite platform for multi-protocol ENSs is outlined in Section 3.3.

3.1 Inspire negotiation support system

3.1.1 InterNeg and Inspire

InterNeg, a research project set up in 1995, is now growing to be a research group, InterNeg Group (InterNeg 2005). The group is involved with research and training, and the development of materials and systems for decision making and negotiations. The negotiation support systems are used in teaching and training around the world. The group is also involved with the development of systems and agents to support and conduct e-negotiations and negotiations in e-business.

The InterNeg site (http://InterNeg.org) is a virtual organization bringing together people, studies, services, systems and information concerned with decision making and negotiations. It is also a source of, and repository for, negotiation-related resources comprehensively covering topics in negotiation and negotiation support in the international arena.

The Inspire system was developed and deployed in July 1996. Inspire (InterNeg Support Program for International Research Experiments) has been used to study conduct bilateral negotiations and also for training. As of May 2004, 6,126 people from 62 countries negotiated via the Inspire (Kersten 2004).

Most of the Inspire users are students, managers, lawyers, engineers and physicians who came from countries such as Austria, Canada, India, Finland, Korea, Portugal, Russia and U.S. The Inspire system has been used in teaching and training (Madanmohan, Kersten et al. 1999). It is also used to as a research tool for InterNeg Group to study cross-cultural

negotiations over the Web, the impact of decision analysis on the negotiation process, the role of support in negotiation and the role of explanatory and display facilities users' perception and decision-making (Kersten and Szpakowicz 1998; Kersten and Noronha 1998a; Kersten and Noronha 1998b; Kersten and Noronha 1999).

3.1.2 Inspire support methodology

The Inspire system contains a facility for the specification of preferences and assessment of offers, an internal messaging system, and graphical displays of the negotiation's progress. As motioned in 2.3.1, NSSs comprise two components: a decision support component and a communication support component. 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 exchange of structured offers, optionally with accompanying explanatory messages, stand-alone free-text messages and automatic e-mail notification of the opponent's activity.

The Inspire system supports negotiators by a structured negotiation process. The system implements a three-phase model of negotiations: pre-negotiation, negotiation, and post-settlement (see Section 2.4.1). The support activities from each of the three phases are illustrated in Figure 3.1.

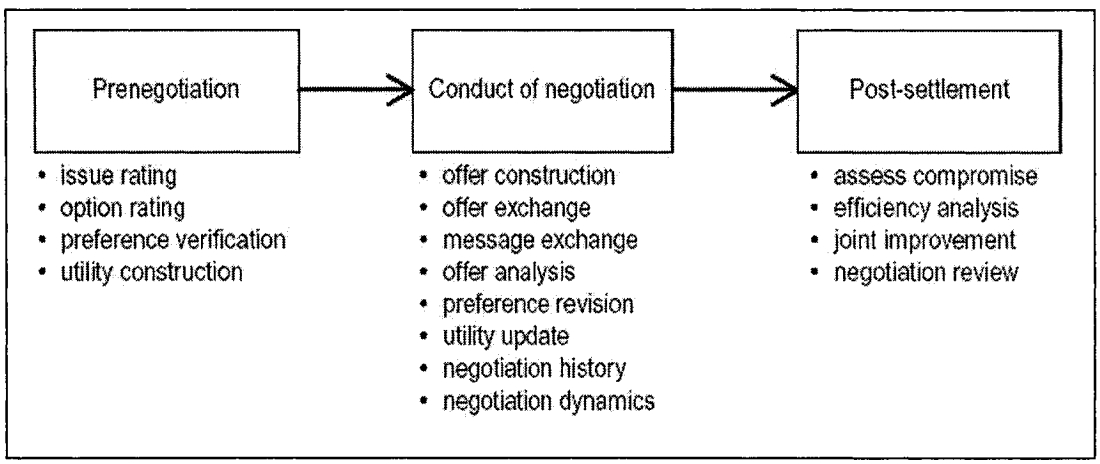

Figure 3.1 The support of Inspire in different negotiation phases (Kersten and Lo 2003)

In the pre-negotiation phase, Inspire helps users to prepare for the negotiation by supporting them to understand the negotiation problem (described in the negotiation case), the main negotiable issues and their options (issue values), and some possible packages (which may form the basis of offers and counter-offers).

The system requests its users to specify the rating of issues and options according to their own preferences for the given negotiation problem. The package rating which is a combination between issue ratings and option ratings enables users to specify the importance of the interrelations within the issues and verify their overall preference. Based on these rating values, the system constructs the value function for each user.

During the negotiation phase, the users begin to negotiate by constructing and exchanging messages and offers. They communicate using free text messages and structured offers - each offer contains user-selected options for every negotiable issue. Based on the elicited preferences fro the user, the system provides values (ratings) of decision alternatives considered by the user and offers submitted by each party. Users also can review their negotiation history in both a text-table list and a graphical dynamics, or review and revise their preference by re-rating issues and options during the negotiation phase.

Once an agreement has been achieved, the Inspire system acts as a mediator and checks for the agreement's efficiency (Pareto-optimality) (Kersten and Noronha 1999). If it is an inefficient compromise, the system presents up to five efficient alternatives and the users may continue their negotiation in the post-settlement phase until they reach an efficient compromise.

3.1.3 Inspire negotiations

Typically, Inspire users conduct bilateral negotiations over a simple case of B2B transactions, called the Itex-Cypress negotiations. The negotiators represent two companies: Itex Manufacturing, a producer of bicycle parts, and Cypress Cycles, which builds bicycles. There are pre-defined four issues: the price of the bicycle components, delivery schedules, payment arrangements and terms for the return of defective parts. Each issue has 3 to 5 predefined options. Thus, there are altogether 180 (3 x 3 x 4 x 5) complete and different packages.

In addition to the Itex-Cypress negotiations, Inspire has been used with more complex cases, such as Union-Management eSchool Negotiation. This describes negotiations between teachers' union and the management of a school district with 7 issues and 8,100 packages, and the negotiation between a health maintenance organization (HMO) and a hospital which has 7 issues and 105,000 packages (Kersten 2004).

The negotiations are pre-setup, the users are selected and they are assigned the case and its sides. Also the deadline is imposed externally on the users. Typically the negotiations take up to three weeks. If both sides agree, they may request the deadline extension. The process

will be illustrated in details in Section 3.3. Depending on the specific situation, the negotiations may result one of the three different outcomes: (1) agreement is reached; (2) the deadline is reached without an agreement; (3) the negotiation is terminated by one side prior the deadline.

After the users learn about the case but before they begin the negotiations, they are requested to fill in a pre-negotiation questionnaire. Upon completion of Inspire negotiations, the users are asked to fill in a post-negotiation questionnaire. One of the questions refers to the users' overall assessment of the system. Generally, users find the system very easy to use, and their evaluation of the overall system is favorable. Over 60 percent of the users stated that they would use a system like Inspire in real-life negotiations and over 80 percent would use such a system to prepare for actual negotiations (Kersten and Noronha 1998a). Considering the fact that Inspire is a fairly simple system with limited capabilities and developed in mid 1990s, the absolute levels of user acceptance of the system are very high. These results lead us to assume that a system like Inspire would be acceptable in e-commerce negotiations.

3.2 Protocol theory

In ICT, a protocol is a set of rules that end points in a telecommunication connection use when they communicate (TechtargetCom 2003). In negotiation, protocol acts as the rules that the negotiation parties flow to interact or communicate when they negotiate. Most of the ENSs are built upon one negotiation protocol to address specific situation (Stroebel 2001). This effectively restricts their use to those problems and interactions that had been assumed a priori by the systems' designers. It has been recognized that the way a problem is represented and the solution process implemented both strongly influence the outcomes of negotiation. Recently, Kersten and Lai (2005) formulated elements of a theory for the design of negotiation protocols. The proposed protocol formalism allows for the construction of models from which users can select a protocol instance that suits them and/or is appropriate for the problem at-hand. Furthermore, this formalism allows for the construction of protocols that can be modified during the user-system interactions. The negotiation protocol of INSS, which is discussed in Chapter 6, is designed and implemented based on this theory.

3.2.1 Negotiation process and protocol

Negotiation support systems integrate the support and media functions, and determine protocols used in the negotiation process. As introduced in Section 2.2.1, several negotiation process models have been proposed for methodological approach to negotiations. Negotiation process models comprise negotiation phases and assign a set of activities to each phase.

The participants of face-to-face (F2F) negotiations follow a certain process model. They select strategies and tactics which then give the basis for undertaking particular negotiation activities. The negotiators decide what strategy (tactic) to follow and when—if at all—the selected strategy (tactic) should be modified or replaced with another one. The negotiators may introduce any change at any time; making a change only depends on them. In contrast, because e-negotiations are conducted with the extensive use of technology, a change can be made if the technology allows it.

The comparison of face-to-face (F2F) negotiation and e-negotiation led Kersten and Lai (2005) to note that process models and strategies and tactics observed in F2F provide an initial basis for the protocols used in e-negotiations. The protocol specifies the negotiation activities and determines whether an activity is undertaken by the user or by the system; it provides a framework for e-negotiations (Kersten and Lo 2003; Kim and Segev 2003).

The relationships between these components of F2F and electronic negotiations are depicted in Figure 3.2.

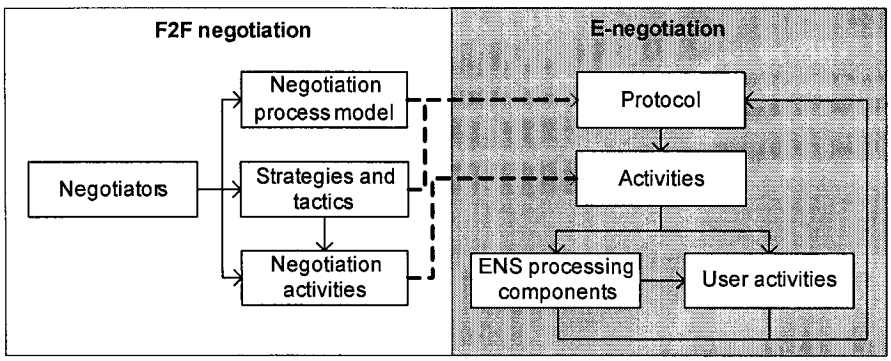

Figure 3.2 Face-to-face and e-negotiation concepts (Kersten and Lai 2005)

The protocol is a formal model, often represented by a set of rules, which govern software processing and communication tasks. It imposes restrictions on the activities undertaken by the ENS and its users through the specification of permissible inputs (Jennings, Faratin et al. 2001). Thus, the protocol controls the ENS and also guides the users' activities. In automated negotiation systems, the negotiation protocol defines the rules under which the interaction between the agents should take place such as what deals can be made and what sequences of offers are allowed (Cranor and Resnick 2000).

Following (Kersten and Lai 2005) the definition of negotiation protocol adopted here is:

Negotiation protocol is a model that guides processing and communication tasks of software and its users, and imposes—explicitly or implicitly—restrictions on their activities through the specification of permissible inputs.

Negotiation protocols in ENSs do not have to be explicitly specified. In the early ENSs in particular, the protocol was embedded in the system in the way that there was only one way to interact with the system. These early systems forced their users to follow a pre-defined negotiation process and in this sense they followed one fixed protocol.

In the consideration of protocols and their comparison it is useful to distinguish between different protocol types. This is because different types may be used in the design of systems targeting different user groups, types of negotiations, and problems. For example, as introduced in previous section, the Inspire system mainly deals with bilateral negotiations with per-defined structure. Thus, the rules of Inspire protocol are defined a priori for the interactions between negotiators. In contrast, the protocol for INSS, the proposed system, may allow adding or modifying rules during negotiation because the system allows negotiators changing the problem during negotiation process. Kersten and Lai (2005) proposed five types listed, together with several examples, in Table 3.1.

Table 3.1 Types of e-negotiation protocols (Kersten and Lai 2005)

Type	Description	Example
Closed	All rules are defined a priori; no rule can be added or modified	Online auction, WebNS, SimpleNS, Inspire, Aspire
Open	Rules may be constructed and added during the negotiation	
Private	Guides the user's activities and defines her valid actions	Aspire
Public	Defines the rules of interactions between the negotiators	WebNS, SimpleNS, Inspire
Comprehensive	Can be used for different types of negotiations	GNP, SilkRoad
Specialized	Applicable to one or a few negotiation types	SmartSettle, WebNS, SimpleNS, Inspire

3.2.1.1 Phases, states, activities

As mentioned in Section 2.4.1, in negotiation process model, a negotiation is composed of several phases and in each phase several activities are undertaken. Undertaking or completing some of the activities leads the negotiator to a particular negotiation situation, e.g. once negotiator accepts the counterpart's offer they reach an agreement, called here a *state*.

At any point of the negotiation, the user and the system are in a particular state; completion of the activities associated with this state moves the negotiation to another state. For example, a negotiator in offer-exchange state could make an offer or read the counterpart's offer, and accept an offer, and then he or she will go to agreement state. Using the negotiation protocol theory, we say that each phase is decomposed into states and with each state one or more activities are associated (see Figure 3.3). Negotiation protocol guides the negotiation process, the moves between states and sequences, and restricts the activity that can be undertaken within the states. The moves between phases and states illustrated in Figure 3.3 represent the process from the perspective of one negotiator.

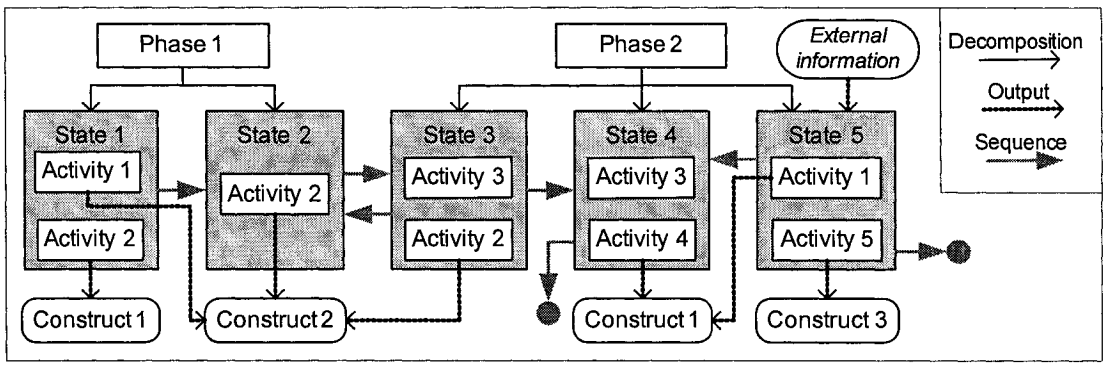

Figure 3.3 Phases, activities and constructs (Kersten and Lai 2005)

In the negotiation process, some activities must be undertaken and some need not. In the same example, the negotiator may just wait for the counterpart's offer rather than make his or her offer, but the negotiator must read the offer and decide to accept or reject it when he or she receives an offer. Also, some activities are conducted by negotiator and some by the negotiator's counterparts. This consideration of states led Kersten and Lai (2005) to propose the following three types of the negotiation states:

1. *Mandatory states* are those which the user or the system has to visit in order to undertake another activity, such as learning about the problem, formulating and proposing offers, assessing counter-offers, and reaching agreement or terminating the negotiation, etc.
2. *Optional states* in which activities may but do not have to be undertaken, such as writing and sending a message attached to the offer as an explanation.
3. *Intervening states* which are activated by information that is external to and not controlled by the user or the system, e.g. an offer submitted by a counterpart invokes state in which the negotiator evaluates this offer (State 5 in Figure 3.3).

The distinction between these three types is context-dependent, i.e. mandatory states in a phase could be optional states in another phase when negotiation moves, and vice versa. For instance, negotiators must learn about the problem in pre-negotiation phase by reading the case (*mandatory state*), while they may review the case (*optional state*) for constructing and evaluating offer in negotiation phase.

3.2.1.2 Sequences, states, exit points

A negotiation can be seen as a sequence of activities which move the negotiator between states. When the negotiator is in any given state, then—according to the typology of states—this negotiator may to the optional sates or to a mandatory state. This allows us to distinguish a *sequence of states*: a *sequence* comprises:

1. One initial state which is the sequence entry;
2. One mandatory state which is the sequence exit state; and
3. Any number of optional states.

The introduction of sequences allows us to describe the negotiation process in terms of sequences: the negotiators and the system moves among sequences comprising negotiations. An example of the moves within and between sequences is depicted in Figure 3.4. The negotiator enters a sequence through its initial state. While in the sequence, the negotiator may undertake activities which move him to an optional state. However, the negotiator can move out of this sequence to another through the exit point(s), only after he or she conducted some activities to move to the mandatory state.

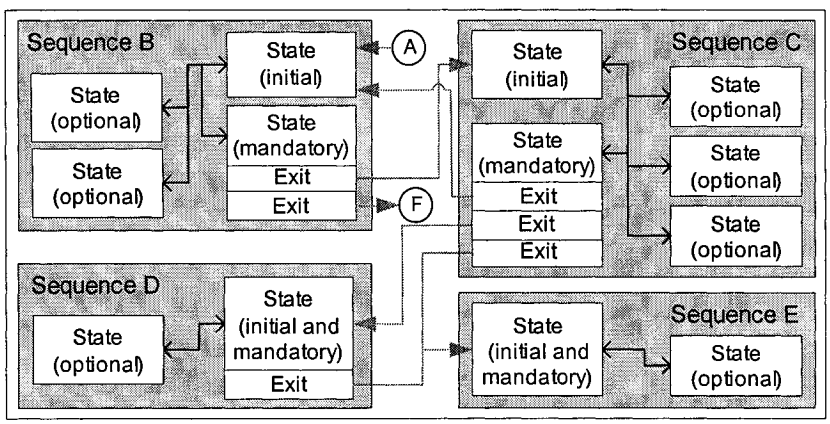

Figure 3.4 Moves between four sequences (Kersten and Lai 2005)

As shown in Figure 3.4, the negotiator starts from A to enter Sequence B in its initial state, and then he or she can move to either one of the two optional states or go to the mandatory

state. In the mandatory state, there are two exit points from which the negotiator could move to Sequence C or F. If the negotiator goes to Sequence C, which is same as in Sequence B but with more optional state and exit points, he or she can move from initial state to optional or mandatory state, and then from the mandatory state the negotiator may go further to Sequence D or E or go back to Sequence B. In Sequence D, the initial state and mandatory state is the same state. Thus, after the negotiator enters into this sequence, he or she can move to the optional state or exit to Sequence E. Note that there is no exit point for Sequence E, which means the negotiator will stay in this sequence.

3.2.1.3 Intervening information

External information is generated by the negotiator's counterpart or the system. For example, after the counterpart sent an offer, the negotiator will be notified by the system that the offer is received. In Figure 3.3, the external information is shown as "entering" the State 5. This information may be received when the negotiator is in this state; it may also be received by the system when the negotiator is in some other state. In this latter case the external information acts as a trigger to activate the State 5, which is then an *intervening state*. This external information, accordingly, is called *intervening information*. Simply, the intervening states are activated by intervening information. In negotiation, there are different types of intervening information, such as received offer/message, reached agreement, terminated by counterpart, etc.

The intervening mechanism is proposed by Kersten and Lai (2005) particularly for protocol modification corresponding to negotiator's action and negotiation situation. According to the theory, intervening states can be added as new states to some sequences, and the existing states in sequence can be replaced with an intervening state as well. Depending on the type of information and the sequence that the negotiator is visiting, the intervening state can be one of the following states:

1. An additional optional state in the sequence which the negotiator is visiting and the sequences that follow this sequence, e.g. a message received from counterpart activates the intervening state "read message" to be added in the current sequence.
2. An updated initial-mandatory state in the sequence in which the negotiator is or one of the subsequent sequences. For example, if a negotiator has not entered into the offer-exchange sequence (e.g. learning problem or eliciting preference) and received intervening information from the system which shows that an offer has been proposed by the counterpart, the negotiator may not be allowed to read the offer. At this moment, the intervening state "read offer" has been added in the negotiator's offer-exchange sequence to replace the original initial state (e.g.

"construct offer") as the new initial state. Thus, once the negotiator enters into the offer-exchange sequence he or she will be shown the counterpart's offer.

3. An initial state in the sequence which the negotiator is forwarded to, e.g. when the negotiator is in offer-exchange sequence and an offer accepted by the counterpart, the negotiator will be forwarded to the next sequence in the initial state "agreement."

The intervening states above, which activate additional states, update initial-mandatory states or forward negotiators, are changing the moves of negotiation. This provides a way to modify the sequences of negotiation protocol during negotiation process.

3.2.2 Negotiation protocol representation

3.2.2.1 Sequence and rules

Negotiation protocol is a formal model, normally defined in terms of "If ... then ..." rules. In this theory, negotiation protocols are represented by sequences. It can be proved that these two representations are equivalent (Kersten and Lai 2005).

According to the protocol theory, there are two types of sequence: the σ-sequence and the ρ-sequence. The σ-sequence describes relationship and moves between states in the same sequence and the ρ-sequence describes the relationship and moves between different sequences, i.e. σ-sequences. Thus, a negotiation protocol can be represented in terms of these two types of sequences, the σ-sequence with the negotiation states and state transitions and, the ρ-sequence with the negotiation sequences and sequence transitions.

This representation allows to distinct two levels of reasoning about protocols, the protocol-level and the sequence-level (Kersten and Lai 2005). This is important for negotiation protocol design and assessment. At the protocol-level, we mainly consider the relationships between σ-sequences; at the sequence-level, we can focus on the composition of the σ-sequence, such as its states and their types (i.e., initial, optional and mandatory).

The two levels of reasoning about protocols are illustrated in Figure 3.4, in which both states and sequences are indicated. At any point of the negotiation process, the sequence and the state where the negotiators are in can be identified, and then the possible activities (optional states) and the next step (exit point) for negotiation are also identified. Once the negotiators complete an activity, they will move to corresponding state or go to the next step from the exit point. In order to illustrate reasoning at the protocol-level a protocol is schematically represented in Figure 3.5. The completion of one sequence will lead the negotiators to move

to one of the reachable sequences. For example, in sequence σ_2 negotiators receive and evaluate the issue proposal from their counterpart. After evaluation, the negotiators may decide to accept, reject or modify the proposal, and then they will move to the corresponding sequence σ_3, σ_6 and σ_7. If there is no reachable sequence for the current sequence, the negotiation ends, e.g. terminated or reached an agreement.

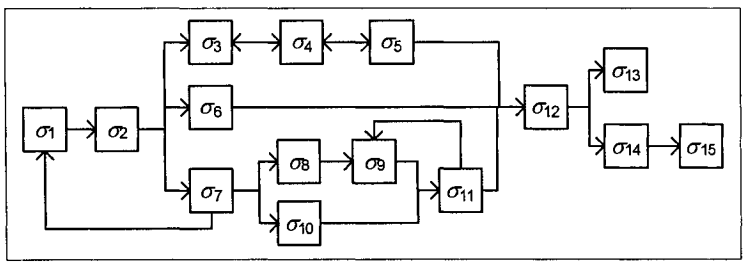

Figure 3.5 Example of a protocol with 15 sequences (Kersten and Lai 2005)

From the perspective of design and implementation of negotiation protocols, Kersten and Lai (2005)) provides an approach to the negotiation as a system. At the protocol level, the σ-sequences are considered as black boxes and only the paths between the sequences are considered. Particularly, the paths from the first sequence of the protocol to each of the negotiation completion sequence showed how users conduct and complete the negotiation. This allows for the break down of the negotiation process and the decomposition of its protocol using a top-down approach (see Figure 3.6). The top-down approach can be used during the design of e-negotiations and the bottom-up in their execution.

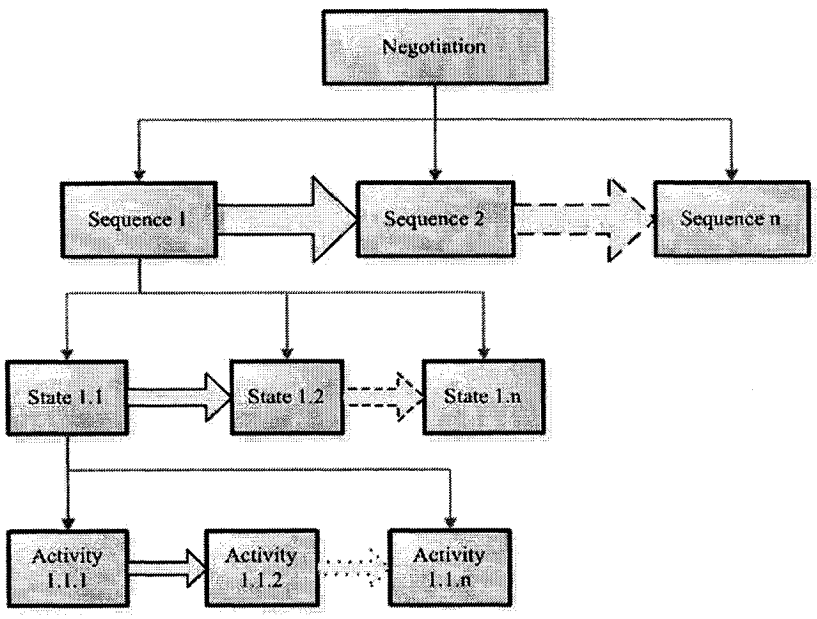

Figure 3.6 Top-down approach to negotiation protocol construction

3.2.2.2 Intervening information and rules

As introduced in previous section, once the system receives intervening information the corresponding intervening states will be activated. What state is activated and to what sequences this state is added depending on the intervening information and the sequence that the negotiator is visiting when this information is received. These are handled by intervening rules.

Intervening state activation involves addition of an optional state and/or the replacement of the existing mandatory and/or initial states. This causes the modification of current sequences (rules). Then, a meta-rule will be generated with the occurrence of the intervening information. Thus, negotiation protocol can be defined by the sequences and intervening rules. For example, in the "Exchange offers" sequence, negotiators initially follow "Construct offer" -> "Read offer" -> "Make counter offer" or "Accept offer". This is part of the initial rules of the negotiation protocol. However, this sequence could be changed if an intervening state was activated by intervening information before the negotiators get into the sequence, e.g. the initial state of the sequence will be changed as "Read offer" instead of "Construct offer" if an offer received from the counterpart. Therefore, the negotiation protocol is composed of the initial sequence and intervening rules.

3.3 Invite platform

3.3.1 Introduction

As discussed in previous sections, currently most of the existing ENSs implement only one negotiation protocol, with a notable exception, the SilkRoad platform designed to support various auction protocols (Stroebel 2003). However, there is no ENS capable of supporting different negotiation protocols, complementary support tools, graphical problem and process representations as well as communication facilities. This restricts the use of ENSs to types of problems and interactions that were assumed and established a priori by its designers. From the research perspective, in particular in experimental studies of the use and adoption of ENSs, the assessment of the impact of different system features on the process and outcomes of negotiations requires the use of systems, whose differences and similarities can be easily controlled by the researcher. In the e-negotiation project, a software platform for multi-protocol negotiation, called Invite, has been implemented as a flexible and highly customizable environment to conduct e-negotiation experiments and field studies (Kersten, Law et al. 2004).

The software platform enables negotiators to map negotiation activities to system components and construct their own protocols by creating a sequence of layout programs invoking components and rules. It provides an approach for the design of Web-based systems that allow for customizable researcher- or user-definable protocols, especially for ENSs.

3.3.2 Theoretical foundations

The use of software in negotiations requires that a process model and a protocol be constructed (Kersten and Law 2003; Kim and Segev 2003). The theoretical foundations of Invite system include the selection of both process model and protocol theory.

As introduced above, several negotiation process models have been proposed based on negotiation behavior study. From the perspective of constructing an ENS, a more detailed model is preferable because it allows for a better (finer) determination of activities and their processing requirements (Chen, Kersten et al. 2003). The Invite system uses a six-phase model presented by Kersten (1997), in which five phases are based on Gulliver's eight-phase model (Gulliver 1979) and a post-settlement phase is added to handle post settlement activities (see Figure 3.7). This model allows for the consideration of a wide range of negotiated decisions, including those which use ENSs.

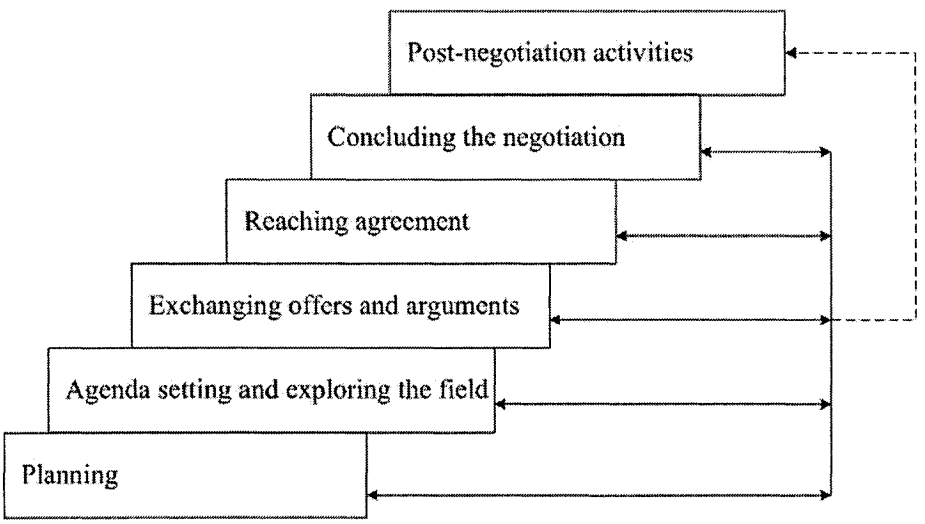

Figure 3.7 Negotiation process model in Invite (Kersten 1997)

Invite system supports multi-protocol negotiations. The negotiation protocols in Invite are based on the theory introduced in the previous section. Corresponding to the decomposing of negotiation process as in Figure 3.7, the system adopts a component-based software development approach. This allows extending or adding components to the component-base

and assembling selected components for the design and deployment of many different protocols depending on the protocol specification. The rules (sequences) are implemented with a database-driven approach. Section 3.3.3.3 will give more details about negotiation protocol database.

3.3.3 Architecture

Invite is designed as a typical three-tier Web application illustrated in Figure 3.8. It is based on a database management system (DBMS) in the persistency layer, a Web-enabled application server in the business logic layer, and Web browser technology in the presentation layer.

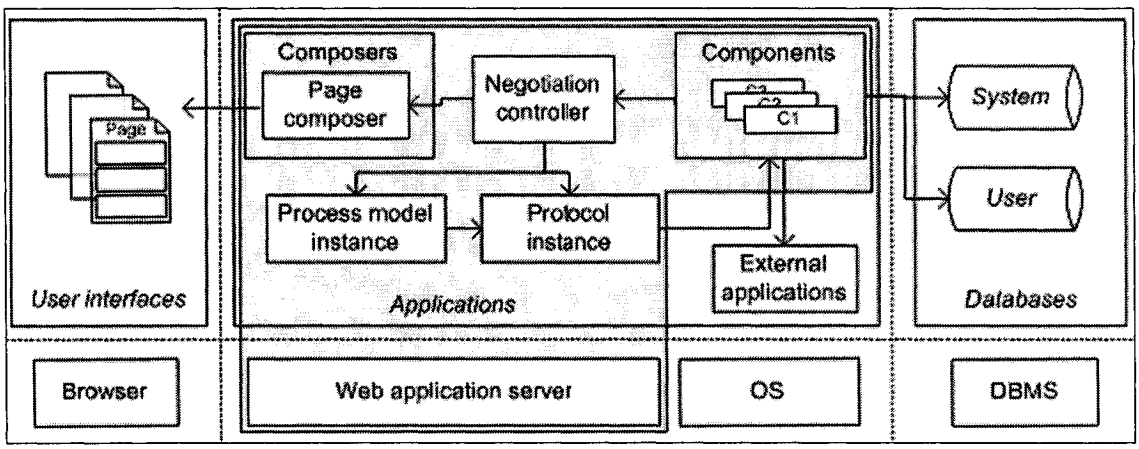

Figure 3.8 Architecture of Invite System (Kersten, Law et al. 2004)

The Invite system consists of the user interface, applications, and the databases. It separates user data from system data into two databases, the system and user database. This facilitates analysis of data collected from negotiators, such as offers and messages.

The user interfaces are comprised of the pages presented to the user. Their main purpose is to display information to the user, receive user input and transmit the user input to the Invite applications. The Invite applications are the core part of the Invite system. They process the requests from the Invite user interfaces. The applications consist of the negotiation controller, the page composers, the components, and the protocol instance as well as external applications.

3.3.3.1 Page composers and components

ENSs, like other Web-based systems, interact with their users through pages. A page as a user-interface either provides information to the user and it may require that the users enter

information. This indicates that one or more activities may be undertaken on one page. For example, a page may both display an offer and ask the user to construct a counter-offer.

Negotiation can be seen as a sequence of activities. From the perspective of the negotiators, activities are the most concrete components of a negotiation. In Invite system, the negotiation activities are represented with components; every activity is associated with at least one component. From the point-view of component-based development, a component is a program that generates output and/or obtains input from the user.

Components are invoked by a page composer program. This program sets the page layout and positions the input/output interface of the components on the page. Since components are reusable, the same set of components can be invoked from different page composers. This is especially useful if the page layout itself is subject to the experimental design (e.g., to study the impact of different positioning of components on a page on the users' performance), but also for the internationalization of e-negotiations (e.g. in different languages and numeric systems).

The page composer provides the link between the negotiation activities and the negotiation protocol. Depending on the protocol, several activities may be positioned on one page or each activity is placed on its own page. For instance, offer receipt, counter offer construction, and message activities may be placed on one or on separate pages.

Several page composers may be used for the same activity and one activity may be used by different composers. Therefore, the system needs to be able to select one composer program that corresponds to the instance of the negotiation protocol and the current state of the negotiation. Process models are not explicitly represented in the Invite system; their role is to provide a framework for the organization of negotiation activities. This framework is implemented in the negotiation protocol which is a sequence of page composer programs and rules imposed on the execution of the sequence. Because each page composer invokes one or more components which uniquely define the negotiation activities, the entire negotiation process can be viewed as a particular sequence of the composers. The protocol rules determine the relationship between page composers; this relationship depends on the context.

3.3.3.2 Controller

In Invite, the core is the negotiation controller. It is a program which consults the protocol, based on the context, updates the protocol instance, and controls the execution of the page composers (see Figure 3.9). The negotiation controller runs a protocol instance assigned to

users' negotiations, then selects and activates the appropriate page composer program according to a protocol instance.

The negotiation controller is independent of the sets of components, page composers and rules. Thus new, additional components, page composers and rules can be introduced into the system without changing the implementation of the negotiation controller. This provides an essential flexibility. The available sets of components, page composers, and the rules constrain the execution of a protocol instance, but do not modify the controller itself.

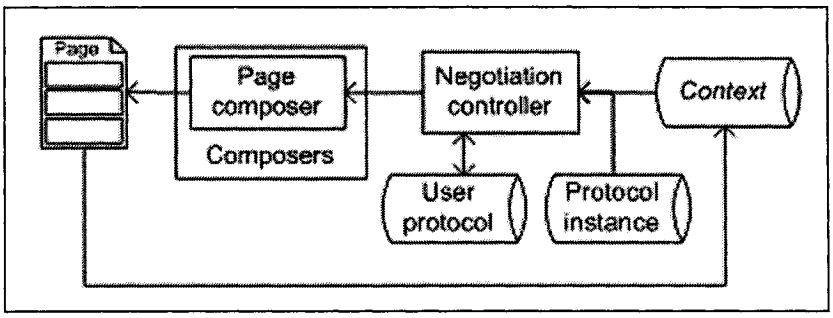

Figure 3.9 Pages, protocols and process model (Kersten, Law et al. 2004)

3.3.3.3 Negotiation protocol database

The negotiation protocol may be implemented in several ways, for instance by a rule-based expert system, or by a database. In Invite, the negotiation protocol is implemented with a database-driven approach. This approach requires that a distinction between a protocol and its protocol instance which is made in run-time.

The protocol is a specification of a single protocol represented with a table in which all page composers are listed and, for each composer, its possible consequent composers. This is done in the protocol designing phase, which includes the initial sequence/state tables and the rule tables that describe rules for handling intervening. The protocol instance is the implementation of the protocol; it is also a table that is being filled in during the negotiation. It includes the negotiation instance/status tables and actual sequence/state tables. The database schema is showed in Figure 3.10.

3.3.3.4 Negotiation content database

The schema of system database and user database are presented in Figure 3.10. They are separate; this facilitates the analysis of data collected from negotiators, such as offers and messages. All the information which related to negotiation are stored in negotiation content

database, such as the negotiation cases and sides, issues and their options, user preferences, offers and messages.

The page composer can either read the data from negotiation content database then show it to users, or provide an interface for user to enter data which then will be stored in negotiation content base.

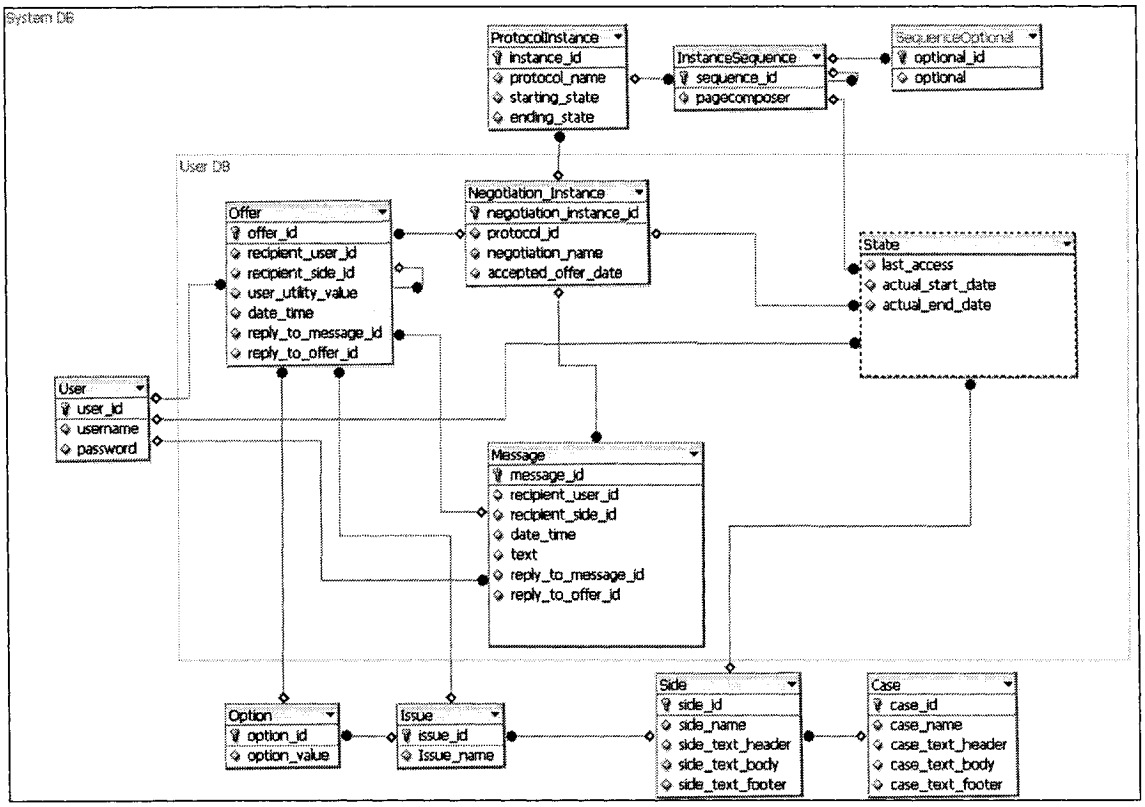

Figure 3.10 Database schema of Invite (Kersten, Law et al. 2004)

3.3.4 Example: the Inspire protocol

Currently, there are two different negotiation protocols deployed in Invite system, Inspire and SimpleNS. The Inspire protocol is derived from the Inspire system introduced in the previous section. The SimpleNS protocol is similar to Inspire but without sequences and states for eliciting negotiators' preference and dealing with post-negotiation activities. In Section 5.3, the SimpleNS protocol will be described in detail to illustrate the database-driven approach for negotiation protocol design. Here, we will just briefly introduce the Inspire protocol with its protocol tables. The protocol tables include initial sequence/states table and intervening rule tables.

3.3.4.1 Initial sequence/states table

This table shows the initial protocol of Inspire (Table 3.2). Each row represents an initial sequence and its states. The grey rows are mapping to negotiation phases. The intervening information generated in the sequence is also shown in parentheses with relevant states.

Table 3.2 Initial sequence/states table of Inspire protocol

	Sequence Name	Initial State	Mandatory State	Optional States (Intervening Information)	Exit Points (Intervening Information)
			Phase 1		
1	Start Negotiation	ReadPublicCase	ReadPublicCase	ReadNegotiationDetails ReadPublicCase	Read Private Information
2	Read Private Information	ReadPrivateCase	ReadPrivateCase	ReadNegotiationDetails ReadPublicCase	Rate Issues
			Phase 2		
3	Rate Issues	RateIssue	RateIssue	ReadPublicCase ReadPrivateCase RateIssue	Rate Options
4	Rate Options	RateOption	RateOption	ReadPublicCase ReadPrivateCase RateOption	Rate Packages
5	Rate Packages	RatePackage	RatePackage	ReadPublicCase ReadPrivateCase RatePackage	Exchange Offer
			Phase 3		
6	Exchange Offer	ConstructOffer	ReadOffer	ReadPublicCase ReadPrivateCase WriteMessage (Message) History ConstructOffer (Offer)	Agreement (Agreement) Rate Issues Rate Packages
			Phase 4		
7	Agreement	ReadAgreement	ReadAgreement	History ReadAgreement	Post Settlement Terminate Negotiation (Termination)
			Phase 5		
8	Post Settlement	ConstructPSOffer	ReadPSOffer	ReadPublicCase ReadPrivateCase History NegoDance ConstructPSMessage (PSMessage) ConstructPSOffer (PSOffer)	Post Settlement Agreement (PSAgreement)
			Phase 6		
9	Post Settlement Agreement	ReadPSAgreement	ReadPSAgreement	ReadPSAgreement NegoDance History	Post Questionnaire Terminate Negotiation (Termination)
			Phase 7		
10	Terminate Negotiation	TerminateNegotiation	TerminateNegotiation	History	Post Questionnaire
			Phase 8		
11	Post Questionnaire	PostQuestionnaire	PostQuestionnaire	PostQuestionnaire History	

3.3.4.2 Intervening rule tables

There are three tables to implement intervening rules, respectively, "Addition of optional states" table (Table 3.3), "Update initial states" table (Table 3.4) and "Forwarding" table (Table 3.5). All intervening rules are based on the information type, such as offer, message, agreement, etc.

Table 3.3 Addition of optional states table of Inspire protocol

Information Type	Sequence	Optional state to add
Offer	Exchange offer	Read offer
Message	Exchange offer	Read message
Agreement	-	-
Termination	-	-
PSOffer	Post settlement	Read PS offer
PSMessage	Post settlement	Read PS message
PSAgreement	-	-

Table 3.3 shows the optional states are added in the sequence when system received the relevant information from negotiators. Table 3.4 illustrates the initial state of the sequence are updated according to the information.

Table 3.4 Update initial states table of Inspire protocol

Information Type	Sequence	Initial state to set
Offer	Exchange offer	Read offer
Message	Exchange offer	Read message
Agreement	-	-
Termination	-	-
PSOffer	Post settlement	Read PS offer
PSMessage	Post settlement	Read PS message
PSAgreement	-	-

In Table 3.5, negotiators are allowed to move from the original sequence to the target sequence. For example, when a negotiator received an offer from the counterpart during negotiation process, whether the negotiator can read this offer depends on which sequence he or she is in. If the negotiator has not entered into the "exchange offer" sequence, he or she cannot read the offer. Similarly, negotiators can reach agreement only after they are in "exchange offer" sequence.

Table 3.5 Forwarding table of Inspire protocol

Information Type	Allowed origination (from sequence)	To sequence
Offer	Exchange offer	Exchange offer
Message	Exchange offer	Exchange offer
Agreement	Exchange offer	Agreement
Termination	Any	Termination
PSOffer	Post settlement	Post settlement
PSMessage	Post settlement	Post settlement
PSAgreement	Post settlement	Post settlement agreement

4. Research objectives

4.1 Problem statement

As discussed above, most current academic and commercial ENSs support traditional one-to-one bilateral negotiations with fixed-structure. While some researchers put efforts on conducting more complex types of negotiation by using standard-auctions and multi-attribute auctions, others combine negotiation and auctions or employ multi-agent systems. However, involving auction mechanism reduced the flexibility and effectiveness of e-negotiations and the delegations of the negotiation activities to agents require that the negotiation problem is well-defined. Thus, on one hand, there is a lack of system which is able to support the negotiators themselves to dynamically formulate negotiation problems to meet their specific needs. On the other hand, taking into account the e-marketplace mechanism with some concepts in eAgora, more complex types of negotiation may be easily performed.

The related work of ENS in e-negotiation project includes the Inspire system and the Invite platform. The former is a structured ENS which supports one-to-one bilateral negotiations; the latter provides an environment for designing, deploying and testing multi-protocol negotiations. Currently, the Inspire system is migrating into the Invite platform.

INSS (InterNeg Support System) is an ENS developed as the part of the InterNeg project. The system is based on Inspire with some additional features; it was designed and developed in 1998 (http://InterNeg.org). INSS is now defunct and has not been used since 1999 because of its design flaws. The system became non-operational because of update of the Solaris operating system and associated libraries which it required to do numerical computation. Furthermore, according to Kersten (2005), the system maintenance and upgrade was not possible due to the code complexity and lack of documentation.

The purpose of the early version of the INSS was to provide a customizable software capable of supporting different types of negotiations. The technology available at that time made this too difficult. With the Invite software platform being available, it is now possible to achieve this purpose. Based on the previous work and current Inspire system, the new INSS system can be designed and implemented using rapid application development (RAD) approach in the Invite platform.

Obviously, there is a gap between our objective, i.e., INSS, and the current version of Inspire system due to its limitations as following:

1. The negotiation problem is pre-setup by system administer or instructor, including the case and sides;
2. The negotiation structure is pre-defined by the problem with fixed number of issues and options;
3. The negotiation participants are pre-assigned with specific side and additional participants are not allowed;
4. It only conducts one-to-one bilateral negotiations.

Accordingly, to solve the limitations above, the Inspire needs to be enhanced and expanded to cover following functions:

1. The negotiators can select one of the current five cases or formulate a new case;
2. New issues and options can be added during negotiation process;
3. Additional participants can be involved;
4. A new protocol can be deployed in Invite to conduct various types of negotiations.

4.2 Research objectives

The purpose of this research is:

"To design and implement INSS in the Invite software platform, and test the resulting system by a small group of participants"

This purpose can be decomposed into the following objectives:

(1) The establishment of a case library, including typical cases, user defined cases, and the structure of these cases (sides, public information, private information, issues, options);

(2) The introduction of the problem specification and modification components which define the use case, the role of user, the deadline and other specifications of negotiation;

(3) The introduction of a negotiation protocol which allows negotiators to formulate their own negotiation case online, to specify the process and the permissible activities of its participants;

(4) The specification of different types of issues and options which can be introduced by the users and interpolated (or extrapolated) by the system;

(5) The application of the e-marketplace concepts to provide a platform to conduct multi-bilateral negotiations;

(6) The testing of the system, including verification and validation, by a small group of participants in different contexts defined by use-case models.

With the new features, the INSS system will allow to use different negotiation strategies and tactics, and to negotiate over open and dynamically modifiable problems. Thus, the system can be used in more real-life negotiations.

5. Methodology

To implement INSS within the Invite platform, the phases of software development life cycle (SDLC) will be followed in order to develop the system. The conventional waterfall model will be integrated with prototyping and rapid application development (RAD). Because all the related systems have been developed using Fusebox technology and because the Invite platform provides an environment that contains a database-driven approach for the design of ENSs with customizable protocols, these two approaches will be applied in designing and implementing INSS in Invite.

The system development methodology of the whole project is presented in Section 5.1. The Fusebox technology for developing components and integrating them into the Invite platform is introduced in Section 5.2. In Section 5.3, the database-driven approach for negotiation protocol design is discussed.

5.1 Systems development life cycle

5.1.1 Overview

The most common way to develop software systems is through the systems development life cycle (SDLC). The conventional SDLC, also known as waterfall model, is a structured phase-by-phase approach to developing systems. Broadly speaking, it includes the following phases (Vliet 2000):

1. Problem analysis
2. Requirements engineering
3. Design
4. Implementation
5. Testing
6. Maintenance

The conventional SDLC model for the phased development of software is, to a large extent, "document-driven" (Vliet 2000). The pile of paper that is produced in the course of the project guides the development process.

Furthermore, explicit feedback to earlier phases occurs once after errors have been detected. Thus, the way in which conventional SDLC views the development process does not sufficiently fit reality. Therefore, prototyping techniques will be also applied in our system implementation.

5.1.2 Prototyping

A prototype is a working model of a software system (or possibly parts of a software system) that emphasizes certain aspects. The aim of developing a software prototype relatively cheaply can be achieved through, for example (Vliet 2000):

1. The use of very high-level languages, in which an executable version can be created quickly. This executable but probably rather inefficient version can be used to test the usability of the proposed system;
2. The development of a system with less functionality, in particular as regards quality attributes such as speed and robustness.

Prototyping is particularly useful for situations in which user requirements are unclear or ambiguous, as well as for systems with a considerable emphasis on the user interface that show a high degree of user interaction. Since INSS is an ENS that enhances and extends the Inspire system, we consider the Inspire system a prototype of our final system. When this methodology is applied, a version of INSS is developed with more functions based on user's requirements within the initial phases, as well as within the feedback during the development process.

On the other hand, users must be aware of the pitfalls of the prototyping approach, particularly that it is not all that easy to change software, and that a prototype is a prototype and not a production-quality system. Therefore, prototyping must be planned and controlled. In this case, the number of iterations must be limited and the traditional SDLC approach should be applied to make the process manageable and controllable (Vliet 2000). Therefore, the number of phases will be predefined (see the research plan in Appendix H).

5.1.3 Rapid application development

Rapid application development (RAD) emphasizes user involvement, prototyping, reuse and revision of other systems, the use of automated tools, small development teams, and a fixed time frame within which activities are done (Vliet 2000). The RAD life cycle consists of four phases (Martin 1991): requirements engineering, user design, construction and cutover.

Based on the current implementation of the Inspire system in the Invite platform, the focus is on developing additional components for INSS and expanding the Inspire protocol using rapid application development. Two RAD techniques can be applied (McConnell 1996): evolutionary prototyping and staged delivery. Evolutionary prototyping is a cycle in which the system is developed in increments so that it can readily be modified in response to end-

user and customer feedback. Staged delivery refers to the fact that the software is developed in stages, usually with the most important capabilities being developed first. System development can and will be performed in several stages. In each stage, a part of the INSS negotiation protocol containing some new functions will be implemented, tested and later integrated (see the research plan in Appendix H).

5.2 Fusebox development methodology

The Inspire and Invite systems are both implemented using the Fusebox development methodology, which provides a high level of modularity and flexibility (Peters and Papovich 2002). The Fusebox methodology views Web systems development in terms of components (known as fuses) that are assembled to create an entire Web page, a fuseaction. The fuses are linked together using the Fusebox engine, metaphorically similar to the electrical fuse box in homes. The Fusebox engine, which consists of the Fusebox core files, also parses elements of Web pages and passes them to the ColdFusion server, which sends them to the HTTP server.

As previously introduced, in Invite, the negotiation controller, depending on the negotiation protocol, invokes the page composers, which link the components. Accordingly, the controller is working with the Fusebox core files as the engine of the Invite system, while the page composers link the fuses using fuseactions. To implement the INSS system, some existing fuses can be reused and more new fuses need to be developed; then, the page composers for INSS can be defined based on the requirements and negotiation protocol.

5.2.1 Introduction

Fusebox is a standard framework and methodology for building Web-based applications. Fusebox attempts to reduce the software failure rate by creating a standard framework and methodology for writing Web applications and managing Web development projects. It makes use of some of the strengths of ColdFusion MX - particularly the use of ColdFusion components (Peters and Papovich 2002).

Briefly, the process of building a Fusebox application can be described in three sentences: A developer breaks his or her application down into actions (fuseactions) and then puts pages and scripts (fuses) together in a directory (circuit). The HTTP request will call specific actions like "index.cfm?fuseaction=mycircuit.myfuseaction," and the associated fuses are then executed by the circuit controller. The core files enable an API and some accompanying utilities, which execute the sequence of files.

As a framework and methodology, Fusebox addresses development problems such as unmanageable complexity, wasteful redundancy of effort, time-consuming code maintenance, and slow development.

Because the techniques used in Fusebox are supported on many different platforms, the fundamental principles that the Fusebox framework is based on work with almost any Web application server. Right now, there are versions of the Fusebox core files for: ColdFusion, PHP, J2EE, ASP and Lasso. Currently, the Invite system is developed with Fusebox 4.0 and deployed on a ColdFusion Web application server. Thus, we intend to apply the Fusebox technology with the same version for our system.

5.2.2 Fusebox methodology

The Fusebox methodology known as the Fusebox Lifecycle Process (FLiP) introduces an elegant new way of managing the software development process. The goal of FLiP is to reduce the atrocious software failure rate. Through a very specific process, FLiP helps developers by providing them with a series of steps for developing a Web application, including everything from when and how to talk with clients to architecting and coding the application using the Fusebox framework.

There are two fundamental ideas behind FLiP. The first is to use a process that is, at all times, closely tied to client feedback. The second is to encourage inexpensive changes in the design early in the process, resulting in a reduced need for changes later in the process, when those changes become progressively more expensive. It is also important to note that FLiP is designed for the technical aspects of a project, starting at a point when the developer is ready to begin building an application.

Generally, the steps in FLiP are as follows (Peters and Papovich 2002):

1. Wireframing: a wireframe is an initial model of the business process for which the application is being built. This step aims to identify the key activities in the system.
2. Prototyping: it is the creation of the HTML representation of the final application, i.e. user interface (UI) deign of the system.
3. Architecture and fuse coding: based on the wireframing and prototyping, the system requirements can be organized into circuits, fuseactions and fuses. Then, the fuses are written in ColdFusion Markup Language (CFML).
4. Unit testing: a unit test in Fusebox is simply a test run on a single fuse.
5. Integration and deployment: this includes integrating fuses into fuseactions, assembling fuseactions under circuits, testing the application and deploying it on the

server. For some cases, the deployment may involve four separated environments, which are development, testing, staging, and production server.

5.3 Database-driven negotiation protocol approach

ENSs are built upon negotiation process models and protocols, which together provide the framework to guide users in conducting negotiation activities. The e-negotiation protocol theory introduced in Section 3.2 provides the theoretical foundation for negotiation protocol design. The Invite platform, which is based on this theory and a six-phase process model, provides an environment for testing negotiation protocols and deploying multi-protocol ENSs. There are several ways to design and implement negotiation protocols, such as through rule-based expert systems, logic programming and databases. In our case, we will choose a database-driven approach.

5.3.1 Negotiation protocol representation

A negotiation protocol is normally a formal model, which can be represented in several ways. Based on the protocol theory, negotiation protocols are represented by sequences. There are two types of sequences and two levels of reasoning within protocols: the protocol-level and the sequence-level. Thus, from the perspectives of design and implementation, we can follow a top-down approach to design a protocol with these two levels (see Figure 3.6). Firstly, from the protocol-level, we decompose the negotiation process into several sequences and specify the relationships and moves between these sequences; secondly, from the sequence-level, these sequences are further decomposed into several states, after which the relationships and moves between the states within each sequence can be defined. Taking into account the distinction between these two levels of reasoning and these two design steps, we represent the protocol in two different ways, using diagrams and tables, which finally translates into a database for negotiation protocol in Invite.

5.3.1.1 Diagrams

State diagrams, also called state-transition diagrams or state machine diagrams, depict various states as well as the transitions between those states (Ambler 2004). In the diagrams, a state represents a stage in the behavior pattern of an object and a transition is a progression from one state to another and will be triggered by an event that is either internal or external to the object. Like Figure 3.4 (in Section 3.2.1.2), which describes the moves between four sequences, state diagrams can be used to determine the phases, sequences, states and transitions (or moves) in a negotiation protocol.

A state diagram is an easily maintainable and easy to understand graphical representation of negotiation protocol analysis and design. It shows both a static overview, as well as the dynamic changes of the negotiation protocol.

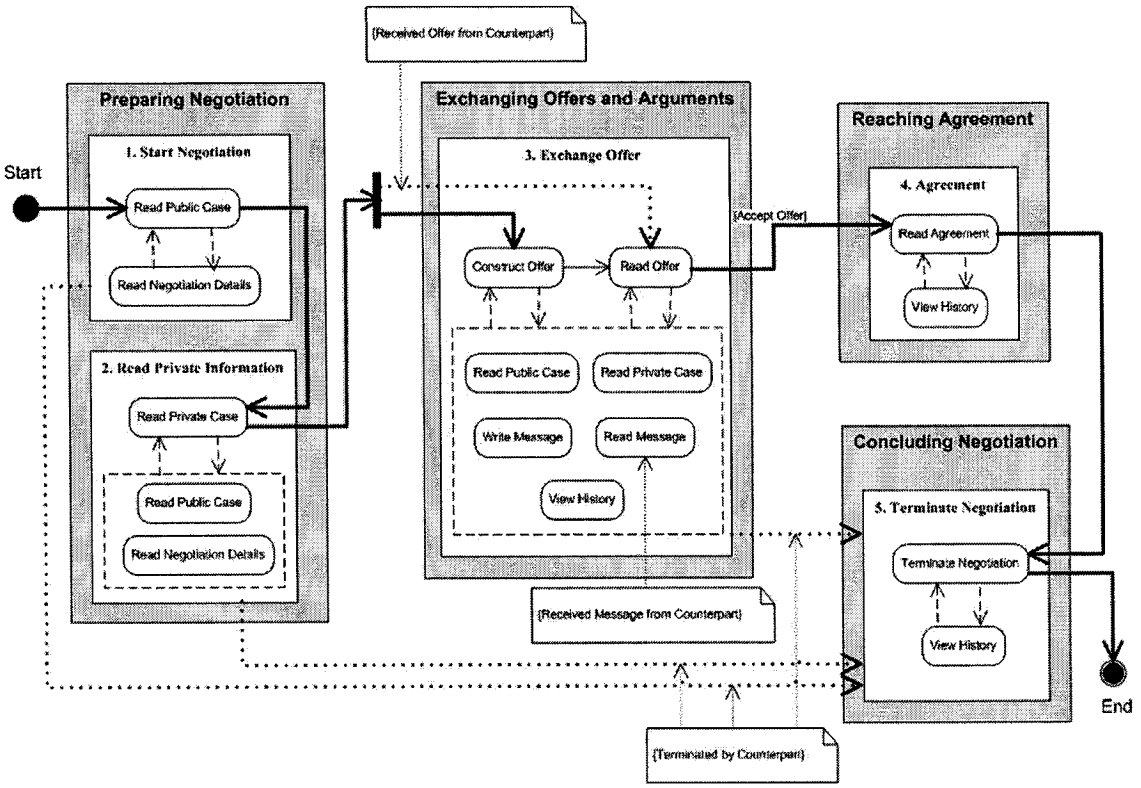

Figure 5.1 State diagram for SimpleNS protocol

Figure 5.1 shows an example of a state diagram of SimpleNS protocol. Each grey block represents a negotiation phase, while the white blocks represent sequences. The rounded rectangles represent states, including initial, mandatory and optional states. There are three types of arrows in the diagram to illustrate the relationships among sequences and states: the wide solid arrow is used in the transition between sequences, which starts in the mandatory state (exit point) of a current sequence and ends up in the initial state of next sequence, the thin solid arrow is used in the transition from an initial state to a mandatory state in the same sequence, and the dashed arrow shows the transition from an initial/mandatory state to optional state(s). Note that some optional states are grouped in a dashed rectangle, which means negotiators can move from an initial/mandatory state to any one of the optional states as well as move freely between the optional states. Furthermore, the dotted arrows and the three notes represent intervening events and rules, e.g. receiving offers and messages from the counterpart activates the intervening states "read offer" and "read message," negotiation terminated by the counterpart leads the negotiator to conclude negotiation directly.

From the static overview, we see that there are a total of four phases and five sequences in SimpleNS protocol. Each sequence has one or more states. In some sequences, the initial state and the mandatory state are the same state, which is represented by a wide solid arrow originating from the mandatory state (exit point) of a previous sequence and going through to the initial state of the next sequence (e.g. sequence 1, 2, 4, and 5). In sequence 3, the initial state is "construct offer" and the mandatory state is "read offer," which means that once the negotiators enter into the "exchange offer" sequence they will first construct an offer and only after receiving and reading an offer from their counterpart will they be able to leave this sequence by accepting an offer to reach a negotiation agreement.

Also, the diagram illustrates the dynamic change within the protocol due to intervening events and rules, such as updating initial states, adding optional states and so on. For example, the default initial state in sequence 3 is "construct offer," but if the negotiator receives an offer from his or her counterpart before entering this sequence, the initial state will become "read offer," i.e. the negotiator must first read the offer, instead of constructing an offer. During the exchanging of offers, if the negotiator receives a message from his or her counterpart, the "read message" optional state will be added.

5.3.1.2 Tables

In Invite, each negotiation protocol is represented with several tables, including the "Initial sequence/states table" and the intervening rules tables. We take the SimpleNS protocol as an example to illustrate the representation of negotiation protocol using tables.

The "Initial sequence/states table" represents the phases, sequences, states as well as the relationships among them (see Table 5.1). In this table, each grey block represents a phase and each row represents a sequence, including: sequence name, initial states, mandatory states, optional states and exit point(s). By default, negotiators will be in the initial state once they enter into a sequence. They can then move from the initial state to any optional state and only once they enter into the mandatory state can they leave this sequence via exit points. This illustrates the moves in the same sequence between states and the moves between sequences via exit points. Also, it shows the intervening information, such as offer (submit), message (sent), etc.

The intervening rules tables include three tables: "Addition of optional states" (Table 5.2), "Update initial states" (Table 5.3), and "Forwarding" (Table 5.4). These tables represent the intervening rules, such as adding optional states, updating initial states and forwarding, based on various intervening information.

Table 5.1 Initial sequence/states table of SimpleNS protocol

	Sequence Name	Initial State	Mandatory State	Optional States (Intervening Information)	Exit Points (Intervening Information)
Phase 1					
1	Start Negotiation	ReadPublicCase	ReadPublicCase	ReadNegotiationDetails ReadPublicCase	Read Private Information
2	Read Private Information	ReadPrivateCase	ReadPrivateCase	ReadNegotiationDetails ReadPublicCase	Exchange Offer
Phase 2					
3	Exchange Offer	ConstructOffer	ReadOffer	ReadPublicCase ReadPrivateCase WriteMessage ***(Message)*** History ConstructOffer (***Offer***)	Agreement ***(Agreement)***
Phase 3					
4	Agreement	ReadAgreement	ReadAgreement	History ReadAgreement	Terminate Negotiation (***Termination***)
Phase 4					
5	Terminate Negotiation	TerminateNegotiation	TerminateNegotiation	History	Post Questionnaire
Phase 5					
6	Post Questionnaire	PostQuestionnaire	PostQuestionnaire	History	

Table 5.2 shows the optional states that can be added to specific sequences during the negotiation process. For example, if a negotiator submits an offer in the sequence "exchange offer," the optional state "read offer" will be added to the sequence for the counterpart.

Table 5.2 Addition of optional states table of SimpleNS protocol

Information Type	Sequence	Optional state to add
Offer	Exchange offer	Read offer
Message	Exchange offer	Read message

As described before, in some situations, the intervening information can activate certain rules to replace the default initial state of a sequence. In Table 5.3, two such rules have been defined for offer and message respectively.

Table 5.3 Update initial states table of SimpleNS protocol

Information Type	Sequence	Initial state to set
Offer	Exchange offer	Read offer
Message	Exchange offer	Read message

In addition, some intervening information, such as offer and message, only lead negotiators to move between the states in the "exchange offer" sequence; others may result in moves between sequences, such as agreement and termination. Considering the distinction between these two types of transitions, the forwarding intervening rules have to be determined (see Table 5.4).

Table 5.4 Forwarding table of SimpleNS protocol

Information Type	Allowed origination (from sequence)	To sequence
Offer	Exchange offer	Exchange offer
Message	Exchange offer	Exchange offer
Agreement	Exchange offer	Agreement
Termination	Any	Termination

In comparing diagrams and tables as ways to represent a negotiation protocol, it seems that tables provide a more detailed view of the composition of the protocol, as well as a more specific description of intervening rules, while diagrams provide an overview of the protocol and show the transitions (or moves) between sequences and states more clearly. Moreover, the protocol tables can be easily transferred into a database schema. Thus, the second representation, tables, will be applied in the analysis and design of INSS negotiation protocol.

5.3.2 Database-driven approach

Based on the protocol analysis and design utilizing the two types of representation detailed above, especially tables, an e-negotiation protocol can be designed and implemented using a database-driven approach that is adopted by the Invite platform. This database-driven approach is the process to establish the negotiation protocol database, which is represented in Table 5.5.

Table 5.5 Steps and tables for negotiation protocol

	Step	Example	Table Name
1	To create states which are used in the protocol	ConstructOffer, ReadOffer	availableStates
2	To create sequences with initial state and mandatory state	Sequence: ExchangeOffer Initial state: ConstructOffer Mandatory state: ReadOffer	initialSequence
3	To set up the optional state(s) for each sequence	Sequence: ExchangeOffer Optional states: ReadPublicCase, ReadPrivateCase WriteMessage	initialOptional
4	To set up the exit point(s) for each sequence	Sequence: Exchange Offer Exit points: Agreement, TerminateNegotiation	initialExitpoint
5	To distinguish the different types of intervening information	Received offer (or message, agreement), Sent offer (or message, agreement)	informationType
6	To set up the intervening rules to update initial/mandatory state	Intervening: ReceivedOffer Sequence: ExchangeOffer Rule: update initial state ContructOffer to ReadOffer	incomingIntervening-InitialMandatoryUpdate
7	To set up the incoming intervening rules to add optional state	Intervening: ReceivedOffer Sequence: ExchangeOffer Rule: add optional state ReadOffer and ViewHistory	incomingAdditionalOptional
8	To set up the outgoing intervening rules to add optional state	Intervening: SentOffer Sequence: ExchangeOffer Rule: add optional state ViewHistory	outgoingAdditionalOptional
9	To set up the intervening rules to forward sequence	Intervening: ReceivedAgreement Sequence: ExchangeOffer Rule: forward to sequence Agreement	incomingInterveningForwarding
10	To set up rules to permit the access of intervening information	Intervening: ReceivedOffer Sequence: ReadPublicCase Rule: cannot read the offer (until enter into sequence ExchangeOffer)	incomingPermissibleIntervening

5.3.2.1 Initial protocol tables

In initial sequence/states table (e.g. Table 5.1), a negotiation protocol is decomposed into several sequences, and each phase includes different types of states. These sequences are

related to each other through exit points. In Table 5.5, following steps 1 to 4 the initial protocol can be established based on the initial/sequence table. It includes three parts: initial sequence, initial optional states and initial exit points. The database schema is shown in Figure 5.2.

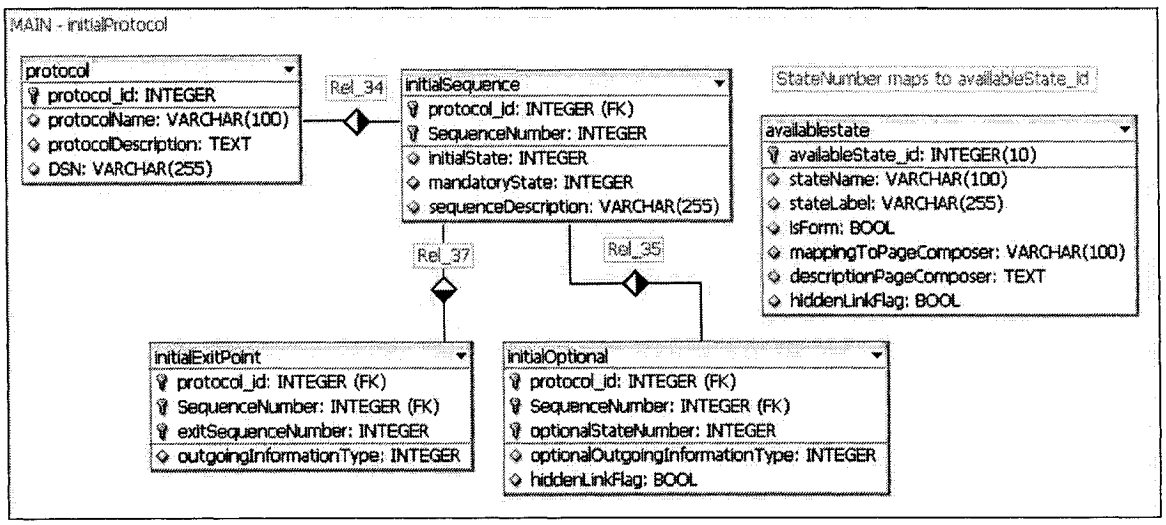

Figure 5.2 Database schema for initial protocol

5.3.2.2 Intervening rule tables

To represent intervening rules in a negotiation protocol, three types of intervening rule tables have been illustrated in previous sections (see Tables 5.2, 5.3 and 5.4). Steps 5 to 10 in Table 5.5 illustrate how to implement these tables in a negotiation protocol database. Figure 5.3 shows the database schema for intervening rules in negotiation protocol.

In some situations, the negotiator may simply be informed with a notice, such as "you got an offer," but the negotiator cannot read the offer until he or she enters into the "exchange offer" sequence. These types of rules relate to the permission of intervening, which is set up in the "incomingPermissibleIntervening" table (see Figure 5.3).

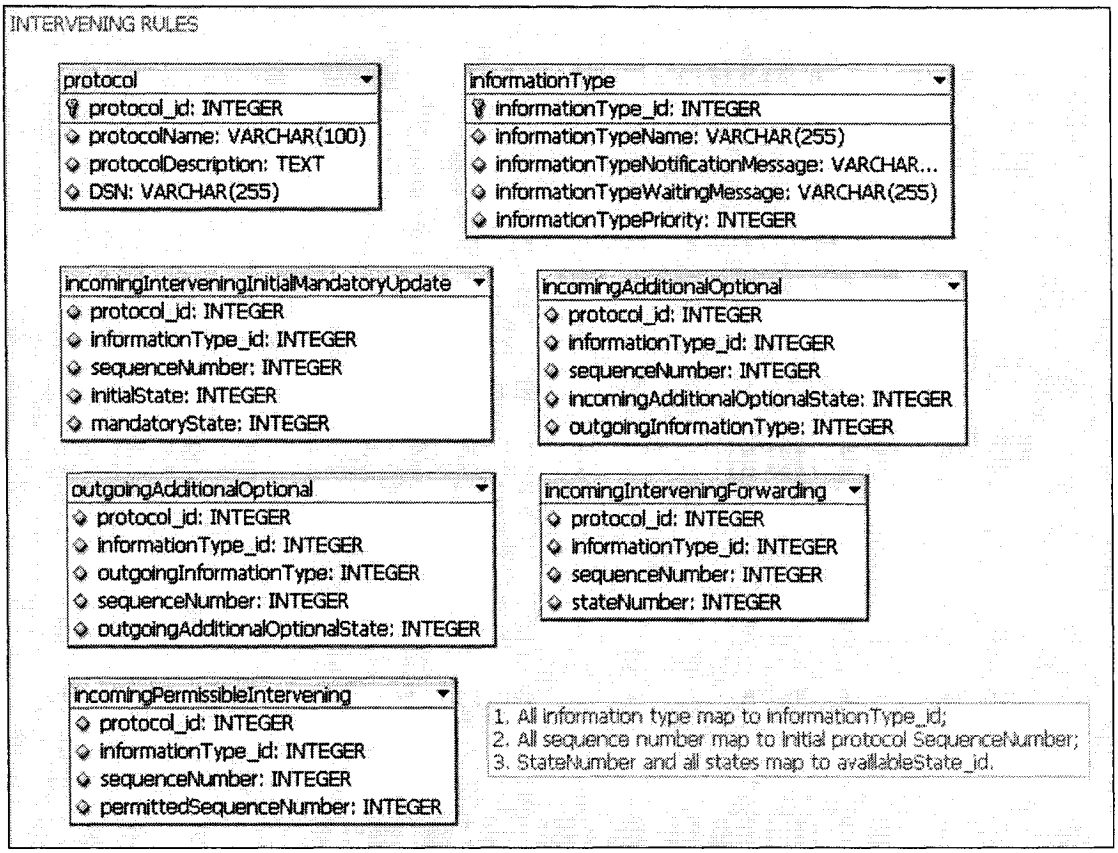

Figure 5.3 Database schema for intervening rules

5.3.3.3 Actual sequence/state tables

In Invite, once users select an ENS (SimpleNS, Inspire) with which to start negotiation, a negotiation instance will be generated in the system. Then, the negotiators and the system follow the pre-defined initial protocol to negotiate with each other. During the negotiation process, the initial protocol can, according to specific rules, be dynamically modified by the intervening events (or information), results in forwarding a negotiator to another sequence, adding optional state(s) and updating the initial/mandatory state. Thus, the actual sequence/state may be different from the initial protocol.

The actual protocol that negotiators followed in the real negotiation process is recorded in the actual sequence/state tables, including: actual sequence, actual optional states, actual exit points and actual incoming intervening information. Note that the outgoing intervening is ignored because it rarely occurs and is normally the same as incoming intervening information. The database schema for actual negotiation protocol is depicted in Figure 5.4.

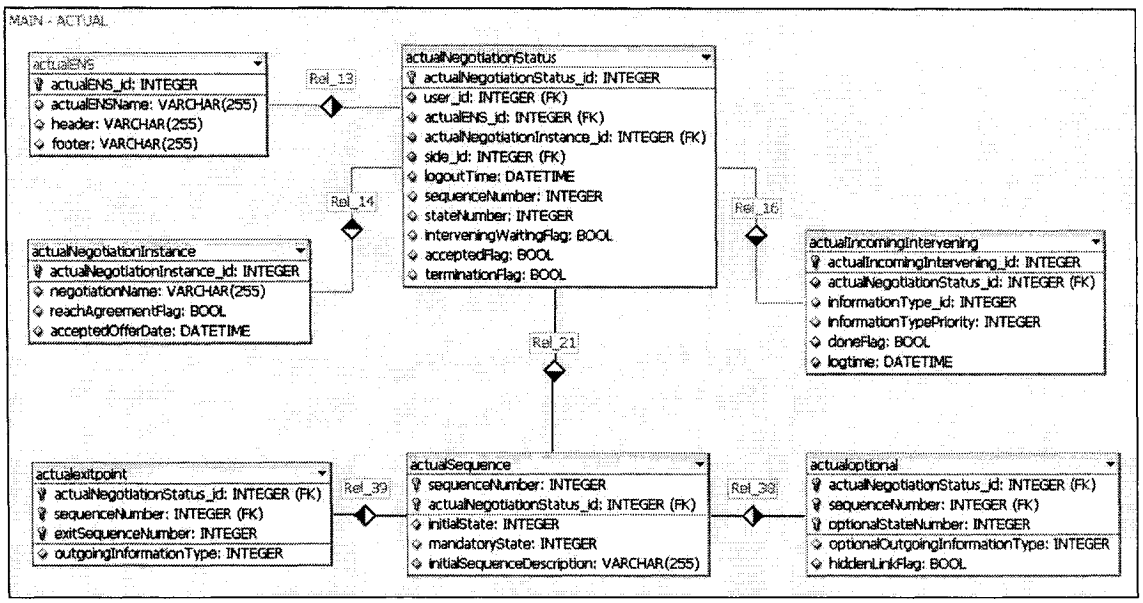

Figure 5.4 Database schema for actual negotiation instance and protocol

6. Analysis and design

Following the SDLC methodology, this chapter mainly discusses the analysis and design of INSS system. Section 6.1 focuses on the INSS functionality based on requirement analysis and related work. The INSS negotiation methodology, which includes the negotiation process model and negotiation protocol, is illustrated in Section 6.2 and the system architecture is outlined in Section 6.3.

6.1 INSS functionality

6.1.1 INSS and Inspire

As discussed in the problem statement, INSS is based on the Inspire system but includes some enhanced and extended features. Furthermore, the Invite platform provides an environment for multi-protocol negotiations, in which the components of one protocol can be reused in another. Thus, the system analysis and design is based on the comparison of functionality between INSS and Inspire. To fill the gap that is the system requirement, the prototype and RAD methodology are applied based on the implemented functionality of the Inspire system.

Essentially, there are three foundations for negotiation support in the Inspire system: the 3-phase negotiation process model (see Figure 3.1), the Inspire negotiation protocol (see the protocol tables in Section 3.3.4) and the analytical support. In INSS, we maintain the analytical support process. Currently, the Inspire system has been integrated into the Invite platform; hence a 6-phase process model (see Figure 3.7) is employed. INSS will also be implemented in Invite based on the same process model. However, more negotiation activities will be conducted in INSS utilizing a more complex protocol. Therefore, the analysis and design of INSS focus on its process model and negotiation protocol.

6.1.2 Main functionality of INSS

To achieve the objectives of our project, the functionality of INSS should cover the following tasks:

1. **To perform bilateral, multi-bilateral and multi-lateral e-negotiations:**
   (1) An individual user can be involved in several negotiations at the same time
   (2) An individual user can have multiple roles in different negotiations
   (3) Exchange messages and offers in real time and dynamically
   (4) Phase control of the negotiation process

2. **To provide the tools for decision-making support:**
   (1) Set up preferences and rating on issues/options/packages
   (2) Calculate the utility value of negotiation
   (3) Draw a graphical overview of the history of offers
   (4) Make suggestion and give advice during the negotiation process
3. **To support negotiating with various cases:**
   (1) Create and set up a case library for registered users to negotiate
   (2) Allow users to create and set up their very own cases for specific problems
4. **To allow adding issues and options before or during negotiation:**
   (1) Allow identifying issues and options for a new case before negotiation
   (2) Allow proposing/adding issues and options for an existing case during negotiation
5. **To deal with different type of issues and options:**
   (1) Support different type of issues (e.g. quantitative, qualitative and categorical)
   (2) Deal with preferences (rating values) on different type of issues and options

6.2 INSS negotiation methodology

To perform the above functions, INSS provides a framework to conduct e-negotiations by extending the process model in Inspire and creating a new negotiation protocol to handle this process. These two parts compose the negotiation methodology of INSS.

6.2.1 INSS negotiation process model

To be implemented in Invite, INSS also follows the 6-phase negotiation process model but involves more activities (see Figure 6.1).

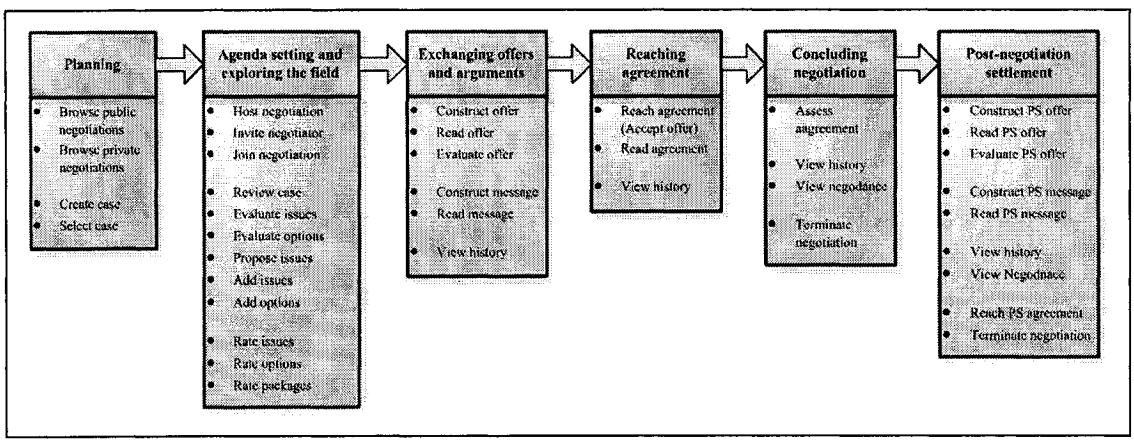

Figure 6.1 INSS negotiation process model

1. *Planning.* In INSS, user may first browse public negotiations in the system in order to search an object (e.g. product, service, job, contract) that meets his or her needs as a shopper in a marketplace. If the user finds something interesting he or she may then join and negotiate with the host; if not or if the user wants to, he or she can negotiate with others by creating a new case or selecting a case from the case library based on the user's needs. In this phase, users may also browse their private negotiations in order to check the status of each negotiation or to invite potential negotiators (e.g. buyer, agency, employer, etc.).
2. *Agenda setting and exploring the field.* The user can set up a negotiation agenda by hosting a negotiation, including: the negotiation case and the sides of case, starting date and deadline, and negotiation protocol, etc. Then, the user can invite other users to be negotiators, or other users may join the negotiation as negotiators. By reviewing the case, the user may identify several basic issues before starting negotiation, and then, if needed, propose other issues during the negotiation process. Once all negotiators agree upon an issue proposal, later negotiations must include this issue. Normally, at least one issue and two options are needed to start a negotiation. Once an issue and its options are added into a negotiation, negotiators can identify their preferences by filling up a series of rating tables for issues, options and packages. This is the process by which all negotiators may explore the negotiation field, and by which each negotiator may explore their utility function. The utility function will be used to provide decision support when negotiators construct an offer, evaluate counter-offers and assess an agreement during the following phases.
3. *Exchanging offers and arguments.* This is the most interactive phase of a negotiation. The negotiator constructs an offer by selecting a package of issues and its options, indicating a preferred utility value, and then sending the offer to the other side(s) in the negotiation. After they have received an offer from their counterpart, negotiators can read the offer and evaluate it with their utility function in order to decide whether to construct a counter-offer or accept the offer. Meanwhile, they may construct messages to communicate with other negotiators in order to explain or argue an offer. Negotiators can also review the negotiation history, which has recorded all the offers and messages made in the negotiation.
4. *Reaching an agreement.* Once all negotiators accept an offer, the negotiation has reached an agreement. Negotiators may review the agreement and the negotiation history.
5. *Concluding the negotiation.* In this phase, negotiators assess the agreement with the support of the system. For example, the system may suggest Pareto optimal solutions for the negotiation based on the utility function of all negotiators, display the negotiation history with both text and graph, and show the joint utility for offer exchanging in a NegoDance graph. If all negotiators agree to close the negotiation with this agreement,

the negotiation is terminated; if they intend to improve the negotiation outcome, the negotiation enters the post-negotiation phase.

6. *Post-negotiation*. The activities in this phase are similar to those in phase 3 – exchanging offers and arguments. However, the purpose of the post-negotiation phase is to improve the outcome of the negotiation. Negotiators can either construct Post-negotiation Settlement (PS) offers and arguments based on the Pareto optimal solutions. The negotiation will be automatically terminated after a PS agreement is reached.

Note that it is not necessary for any one negotiation to include all the phases above. For example, the negotiation could be terminated when exchanging offers during Phase 3. Also, these phases can be overlapped, i.e. the negotiators may go back to revise their preferences when constructing and evaluating offers. Particularly, INSS allows negotiators to propose new issues and add options during the negotiation process, which could occur at any point from Phase 1 to Phase 3 until reaching an agreement.

6.2.2 INSS negotiation protocol

INSS negotiation protocol is based on the negotiation protocol theory and the negotiation process model and is designed and represented with tables in Appendix A. The initial protocol of INSS is illustrated in Table A.1. The grey row represents a high level of sequence, called *a block*, which is corresponding to one of the main activities in the negotiation process model of INSS. Each of the other rows shows a sequence and its initial state, mandatory state, initial optional states and exit points. The intervening information generated in the sequence is also shown in parentheses with the relevant states. Some of the sequences have no exit points in the initial protocol table because the negotiators are forwarded to other sequences by intervening information (see Table A.4). The initial protocol and intervening rules of INSS negotiation protocol are described as follows:

Block 1: Browse negotiations (100 –103)

This block is mainly for the user to browse his or her private negotiations and all public negotiations, as well as the detailed information, so that the user has an overview of the current negotiations and can search for negotiations he or she is interested in joining.

Once a user logs into the system, he or she will enter into sequence 101 and the system will show all the negotiations that the user has participated in, along with the main information concerning each negotiation. From this point, the user has three exit points by which to move into another sequence: 1) sequence 102 - to get more detailed negotiation information,

2) sequence 100 - to list all the available negotiations for all users in the system; 3) block 2 - to create a new negotiation to address specific needs.

In a similar process as in sequence 101 above, in sequence 100 the user can move to sequence 102 or sequence 110 in block 2 or return to sequence 101.

In sequence 102, the user can move back to sequence 100 or 101. If the user is the host of the negotiation, the user can go further, to block 4, and invite other users to join the negotiation as negotiators, or directly add a selected user as a negotiator; in both cases, a new instance of this negotiation is set up. If there are any instances already set up, the user can, for further information, select one from the list to enter into sequence 103. If the user is not the host and if he or she is interested in the negotiation, the user can go to block 5 and apply to join the negotiation.

For each established negotiation instance, the user can select it in order to view its details and check the negotiation status in sequence 103. Following this, the user can decide to start or continue the negotiation.

Block 2: Host new negotiation (110 –111)

This block is mainly for the user to host new negotiations that meet his or her specific needs if there is no appropriate negotiation currently in the system. The user can also set up instances for this negotiation.

Hosting a negotiation involves three steps: writing down negotiation details, selecting an existing case and specifying the sides of case for negotiation. The detailed information for the new negotiation includes a name, description, deadline, type, etc. The system will initially list all existing cases, from which the user can select. In the meantime, the user can read a brief introduction of each case. If the user decides to use an existing case, then he or she must specify the side for which the user will stand; otherwise, if there is not an appropriate case for the user to use, he or she can go to block 3 in order to write a new case for this negotiation. Once a side has been specified, the new negotiation is hosted and the host can either enter sequence 111 in order to set up an instance immediately, or go to block 4 in order to invite another user to join the negotiation as a negotiator.

In sequence 111, the host sets up an instance by adding a negotiator (a selected user), specifying the side for this negotiator and selecting a negotiation protocol (currently only INSS protocol). For each instance, a copy of the initial protocol will be added to the

negotiation protocol database in Invite for both parties. They can then select this instance to start negotiation.

Block 3: Create new negotiation case (120 –124)

Currently, there are five typical negotiation cases in our case library. This block is mainly for the user to create new negotiation cases if no appropriate case currently exists in the library for the user's specific negotiation. Each negotiation case includes public information (the public case) for all sides, private information (the private case) for each side, negotiation issues and their options. Therefore, writing a new case includes four steps: writing the public case, writing the private case, entering issues and entering options for each issue. There are three main issue types: quantitative, qualitative and categorical. The quantitative issue can have numerical discrete or continuous options, while the qualitative and categorical issues include ordinal options.

Firstly, the user must write the public case in sequence 120, and then write the private case in sequence 121. During the written process, the user can read and revise these cases at any time. For each case, the number of issues is unlimited and each issue must have at least two options. Note that having too many issues may influence the efficiency and performance of the negotiation. The user can enter each issue and some information (e.g. name, description, type) in sequence 122, enter its options in sequence 123, and then go back to enter another issue. Once all issues and their options have been entered, the user will move to sequence 124 in order to complete the case.

After completing the new case, the user can add it to the case library to be reused in other negotiations. Afterwards, the user can go to block 1 to browse negotiations or go to block 4 to invite another negotiator to join the negotiation.

If there is already at least one negotiator who has joined the negotiation, the user can start negotiation directly. This happens when the user enters this sequence from block 7 (sequence 213 or 215) instead of block 1. In this case, the system will automatically update the existing case if the user wants to add it to the library.

Block 4: Invite negotiator (130 –134)

This block is mainly for the user who hosted the negotiation, i.e. the host, to invite other users to join his or her negotiation as negotiators.

In sequence 130, the host can choose another user to add to the negotiation. In the meantime, the host also can review the negotiation's detailed information and read the public case as well as the private case. Once the host has invited a user, the "invitation" intervening information will be generated and sent to the selected user. The host then enters into sequence 131 to wait for the selected user's response, which can either be to join or to reject. If the host receives the "join notification" intervening information, he or she will be forwarded to block 6 in order to start the negotiation with the negotiator; if the host receives the "non-join notification" intervening information, which means the selected user rejected the invitation, the host will be forwarded to sequence 134 in order to read the rejection and invite someone else.

When the selected user receives an invitation to join a negotiation, he or she may read the negotiation details and the public case to decide whether or not to join. If the user accepts the invitation, the "join notification" intervening information will be generated and sent back to the negotiation host; if the user rejects the invitation, the "non-join notification" intervening information will be generated and sent back to the host, and the selected user will move to sequence 133, in which the user can enter into sequence 100 to browse public negotiations.

Block 5: Join negotiation (140 –144)

This block is mainly for the user to join a negotiation that is hosted by other users acting as negotiators.

In sequence 140, the user can choose an available side of the negotiation case to join and review negotiation's detailed information or read the public case. At the same time, the "join application" intervening information will be generated and sent to the host. Then, the user enters into sequence 141 to wait for the host's approval to join. If the user receives the "join approval" intervening information, the user will be forwarded to block 6 to start negotiation with the host acting as negotiator; if the user receives the "join rejection" intervening information, which means the host has rejected the user's application to join, the user will be forwarded to sequence 144 to read the rejection. If the user wants to look for other negotiations to join, he or she can move to sequence 100 in block 1 to browse public negotiations.

When the host receives the user's application to join as a negotiator, the host will be forwarded to sequence 142. The host may read the negotiation details and the public case in order to decide whether to approve or reject. If he or she accepts the application, the "join approval" intervening information will be generated and sent back to the applicant, and the host will be forwarded to block 6 to start negotiation (instance) with the user.

If the host rejects the application, the "join rejection" intervening information will be generated and sent back to the applicant, and then the host will be forwarded to his or her previous state (the negotiation state the host was in before he or she received the application to join and entered into sequence 142 in this block).

Block 6: Start negotiation (200 –201)

This block is the beginning point of a negotiation instance, and is mainly for negotiators to review the negotiation (negotiation details, the public case and the private case). In the meantime, negotiators can add more options to current negotiation issues or go to block 7 to propose new issues.

Negotiators start negotiation in sequence 200 by reading the public case and then read the private case in sequence 201. In both sequences, the negotiator can also review the negotiation details. If the negotiator wants to add more issues to the negotiation he or she can go to block 7 to propose new issues; if more options are needed, the negotiator can add options to current issues; otherwise, the negotiator will go to block 9 to set up his or her preference by rating issues and options.

Depending on the issue type, there are two different ways to add options. For quantitative issues, the options can be added directly and the "new quantitative option" intervening information will be generated and sent to other negotiators. If the negotiator has already rated current issues and options, the system can automatically calculate the new quantitative option's rating value by interpolation or extrapolation based on the current options' rating, so that negotiator can use it for later negotiation; otherwise, the negotiator will set the quantitative option's rating value in block 9. The negotiator can add this type of option as an optional state without leaving the current sequence. Once the negotiator adds a quantitative option, "read new quantitative option" will be added as an additional optional state for the counterpart. However, for qualitative issues, the negotiator must go to block 8 in order to add options and then rate them manually in block 9.

Block 7: Propose new issue (210 –215)

This block is for negotiators to propose new issues that will extend the negotiation. Issue proposal includes an explanation (or description), the issue type and at least two options. Once a new issue has been proposed, it has to be accepted by other negotiators and then added to the negotiation. If other negotiators disagree with adding the new issue, they can either reject or modify it. This can be done either at the beginning of negotiation or during negotiation prior to reaching an agreement.

In sequence 210, the negotiator can propose a new issue and review the negotiation case. If the negotiator wants to discuss certain issues or exchange ideas, he or she can write a message and then send it to the other negotiators. This will generate the "message" intervening information, which will prompt the message receiver to read the message. On the other hand, if the negotiator receives the "message" intervening information from other negotiators, he or she can read it immediately. After a new issue has been proposed, the "issue proposal" intervening information will be generated and sent to all other negotiators. The user who proposed the new issue will then wait for the other negotiators' response, which can be an acceptance or rejection of the current proposal, or a new proposal. If the negotiator receives the "issue approval" intervening information from all negotiators, he or she will be forwarded to sequence 204 in block 3 to complete (update) the case. If the negotiator does not want to wait, the negotiator can leave this sequence from sequence 211 in order to continue his or her previous negotiation state. The previous state is the sequence and state from which the negotiator comes to propose an issue, e.g. reading the public/private case, rating issues/options/packages, exchanging offer and so on.

Once a negotiator has received an issue proposal, he or she will be forwarded into sequence 212 to decide whether to accept, reject, or modify the proposal. In the meantime, as in sequence 210, the negotiator can review the case and write or read a message.

If the negotiator accepts the issue proposal, he or she will move to sequence 213 and the "issue approval" intervening information will be generated and sent back to the user who proposed the issue. Afterwards, all the negotiators must return to sequence 204 in block 3 to update the case in order to restart the negotiation.

If the receiver rejects the proposal, he or she will move to sequence 214 and the "issue reject" intervening information will be generated and sent back to the user who proposed the issue. At the same time, the negotiator will be forwarded to his or her previous state (before the negotiator received the issue proposal), and the user will moved into sequence 212 in order to read the rejection.

Alternatively, in sequence 215, the receiver may modify any information included in the issue proposal (e.g. explanation, type and options) to make it acceptable. The previously proposed issue (the original issue proposal) will be replaced by the modified issue proposal and the "issue proposal" intervening information will be generated and sent to the other negotiators.

Block 8: Add qualitative option (220)

This block is specifically for negotiators to add options for qualitative or categorical issues. As previously mentioned, based on the issue type, there are two different ways to add options. For quantitative issues, adding options can be seen as an optional state in the current sequence; for qualitative or categorical issues, the negotiator must exit the current sequence to move into this block.

In sequence 220, the negotiator can also write messages for discussing and exchanging ideas. Once a new option has been added, the "new qualitative option" intervening information will be generated and sent to other negotiators. After adding all the options, the negotiator will be forward to sequence 204 in block 3 to update the case.

Block 9: Set up preference (230 –232)

This block is mainly for negotiators to setup their preferences by rating issues, options and packages. This will create the negotiator's utilities for analytical support during negotiation as well as for evaluation of the outcome of negotiation.

Firstly, the negotiator must rate all the issues that were accepted by all negotiators in sequence 230. The negotiator also can review the negotiation case and write/read messages at any time. If the negotiator wants to negotiate additional issues with other negotiators, he or she has to enter into block 7 in order to propose new issues; adding more options to current issues is a similar process as in previous blocks.

After finished rating the issues, the negotiator will enter into sequence 231 in order to rate the options and then enter into sequence 232 in order to rate the packages. These two sequences will generate option-rating values that will be used to calculate the negotiator's utility of negotiation offers. The option-rating values can also be used to calculate the new quantitative option's rating value by interpolation or extrapolation.

If the negotiators want to review or revise their preferences, they can re-rate issues, options and packages sequentially. This means that the negotiators must start by rating the issues, then the options, and end by rating the packages. After rating packages, the negotiators can move to block 10 to exchange offers.

Block 10: Exchange offer (300)

This block is mainly for negotiators to exchange offers. The negotiator can construct offers, read offers and view the negotiation history. Additionally, the negotiator can review the case, propose new issues, add options, and read/write messages.

A negotiation offer is composed of one option from each issue. When the negotiator constructs an offer, he or she has to select an option from each issue so that the system can calculate the utility of the offer based on the negotiator's preference, which was provided in block 9. If the negotiator decides to submit the offer, the "offer" intervening information will be generated and sent to the other negotiators. When the negotiator receives an offer, he or she can read it and the system will also calculate the utility based on the negotiator's preferences. If the negotiator is not satisfied with the offer received, he or she can make a counter-offer with a higher utility. Once all negotiators agree with an offer, they have reached a negotiation agreement and the "agreement" intervening information will be generated and sent to all negotiators, after which they will be forwarded to block 11.

Block 11: Reach agreement (400)

This block is mainly for negotiators to reach and review a negotiation agreement. In some negotiations, it is possible that negotiators have reached an agreement but that there is still one or more offers with a higher utility than the agreement reached by the negotiators. To improve the outcome of the negotiation, the Pareto efficient method is applied in this block. Based on the utility of all negotiators, the system can calculate and determine whether or not the agreement is a Pareto optimal solution. If it is not, the system will list some possible alternative offers that are Pareto efficient solutions with a higher utility for all negotiators, and the negotiator may move to block 13 to proceed with the post-negotiation settlement. Otherwise, the negotiator will move to block 14 to terminate the negotiation and the "termination" intervening information will be generated and sent to the other negotiators.

Block 12: Improve agreement (500)

This block is mainly for negotiators to proceed with the post-negotiation settlement in order to improve the negotiation outcome. Similar to exchanging offers in block 10, the negotiator can construct/read post-negotiation settlement offers, write/read post-negotiation settlement messages and view the negotiation history. The post-negotiation settlement offer can be selected from the Pareto efficient solutions. Once all negotiators are satisfied with a post-negotiation settlement offer, they have reached the post settlement agreement and the "post-negotiation settlement agreement" intervening information will be generated and sent to all negotiators, who will then be forwarded to block 13.

Block 13: Reach post-negotiation settlement agreement (600)

This block is mainly for negotiators to reach and review the post-negotiation settlement agreement. Besides viewing the negotiation history, the system integrates in one graph all

the offers from negotiators from the point of view of the current negotiator, called *NegoDance*. This allows negotiators to review the negotiation process with utility movement, so that they can learn more about their negotiation strategies and tactics. After reaching a post-negotiation settlement agreement, the negotiation will be automatically terminated and negotiators will move to block 14.

Block 14: Terminate negotiation (700)

This block is mainly for negotiators to terminate the negotiation and view negotiation history. It is a default exit point for all sequences, which means the negotiator can terminate the negotiation at any time. This will generate the "termination" intervening information and send it to all negotiators.

6.3 System architecture

INSS is designed as a typical four-tier Web application, including a Web browser client, a Web server, a Web application server and a database server (see Figure 6.2).

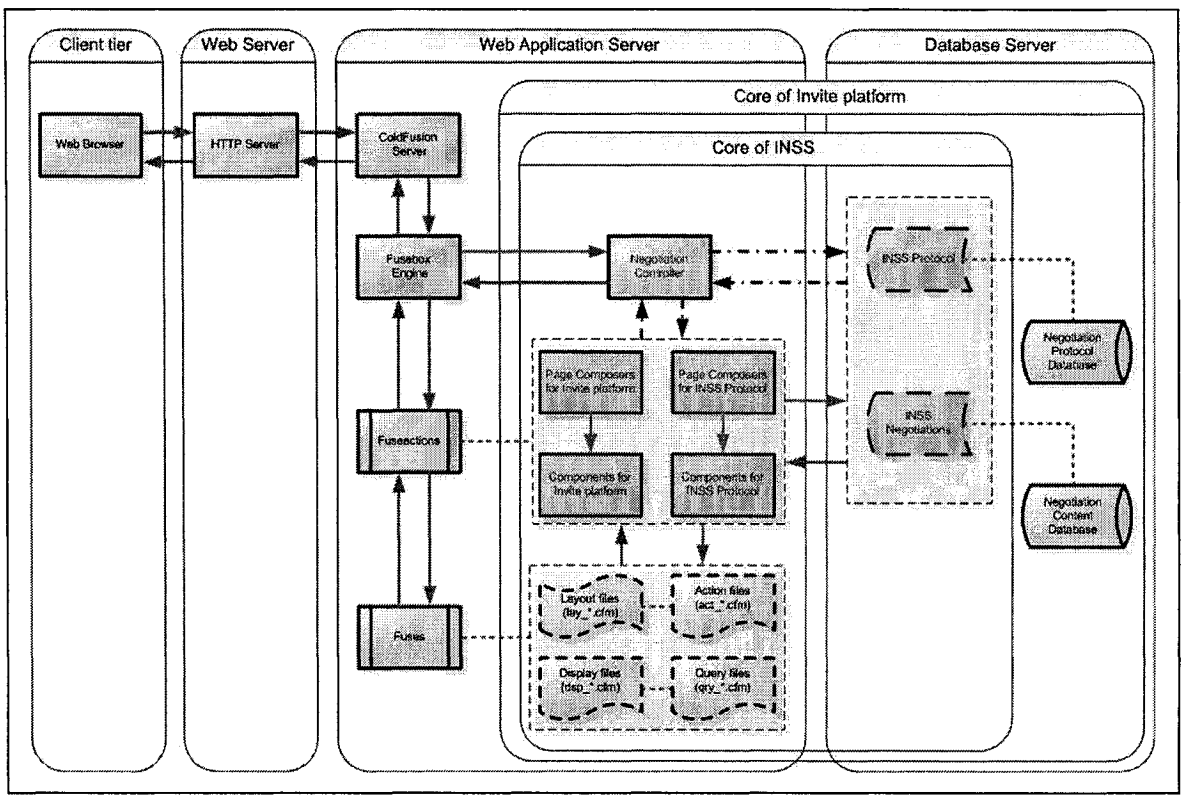

Figure 6.2 System architecture of INSS

The core of Invite is implemented on a ColdFusion application server. The negotiation controller works with the Fusebox engine to invoke page composers according to the negotiation protocol. The core of INSS includes the page composers and components for INSS negotiation protocol as well as the databases. There are two databases on the database server: the negotiation protocol database (NPDB) and the negotiation content database (NCDB). The INSS negotiation protocol tables are deployed in the NPDB and the data about negotiations using INSS protocol are stored in NCDB.

6.3.1 Invite platform

In Invite, all negotiation protocols are deployed in NPDB and the information about the negotiations is stored in NCDB. INSS protocol and negotiations can be seen as an instance of these two databases (as illustrated in Figure 6.2 by the two dashed arrows on the right hand side).

There are two types of page composers and components, one for the Invite system and another for different protocols. The negotiation controller in Invite is composed of the system's page composers and components, and invokes and activates the protocol's page composers and components based on the negotiation protocol and the negotiation status. The negotiation controller is independent from the protocols. Thus, INSS retains all the system's page composers and components. Since the Invite platform is component-based, many components, even page composers, can be reused for different negotiation protocols, despite being designed and developed for a specific protocol.

The Invite system is deployed with Fusebox technology on the ColdFusion Web application server. The negotiation controller is working with the Fusebox engine to interact with the page composers and components, which are fuseactions. Each fuseaction is composed of one or more fuses. There are four types of fuses in Fusebox, each of which has a different prefix: for display (dsp_*.cfm), layouts (lay_*.cfm), queries (qry_*.cfm) and actions (act_*.cfm). The first two types are mainly for the user interface and the others are to deal with form actions and database queries.

6.3.2 Page composers for INSS

In Invite, the page composers and their components are corresponding to negotiation states and negotiation activities. They are both implemented with fuseactions but on different levels. At the lower level, each component is a set of fuses used to generate output and/or obtain input from the user by reading/updating the negotiation databases; at the higher level, each page composer is a set of components, i.e. fuseactions. Because INSS is an extended

version of Inspire, several page composers that have been developed for Inspire can be reused in INSS. In addition, new page composers and components have been designed and developed for INSS in order to implement new features of the system. Table B.1 in Appendix B shows all the page composers and their components of INSS. Each page composer includes one or more fuseactions, i.e. components. Some of the fuseactions, which are for displaying information to provide the input/output user interface, are indicated with "V"; others, which are mainly for manipulating data, are indicated with "M."

6.3.3 INSS negotiation protocol database

As illustrated previously, the NPDB in Invite includes several tables to store the initiation protocol, the intervening rules and actual protocol instances (see Figure 5.2, 5.3 and 5.4). The database schema of NPDB for INSS negotiation protocol is same as in Invite. The tables described in Section 6.2.2 are designed for the initial protocol and intervening rules of INSS. For each negotiation instance that uses INSS protocol, Invite will copy the initiation protocol tables as an actual instance and update the data based on the intervening rules and negotiation status.

6.3.4 INSS negotiation content database

In Invite, NCDB stores all the information about negotiations, including negotiation cases, negotiation issues/options, user preferences (issues/options rating), and negotiation offers/messages. To implement the six-phase negotiation process model in INSS (see Figure 6.1), several tables need to be added to the NCDB in Invite:

1. Tables for creating and setting up negotiation cases: case_issue, case_options
2. Tables for dealing with different type of issues in proposing and adding issues/options: issue_type, issue_unit
3. Tables for conducting bilateral negotiations: negotiation, negotiator

The resulting database includes five groups of tables, which are as follows:

1. Tables for negotiation case: negotiation_case, case_side, case_issue, case_options
2. Tables for user's role in negotiation: negotiator
3. Tables for negotiation: negotiation, issue, options, issue_type, issue_unit
4. Tables for user's preference: issue_rating, option_rating
5. Tables for offer/massage: offersmessages, offers, messages

The database schema of NCDB in INSS is shown in figure 6.3.

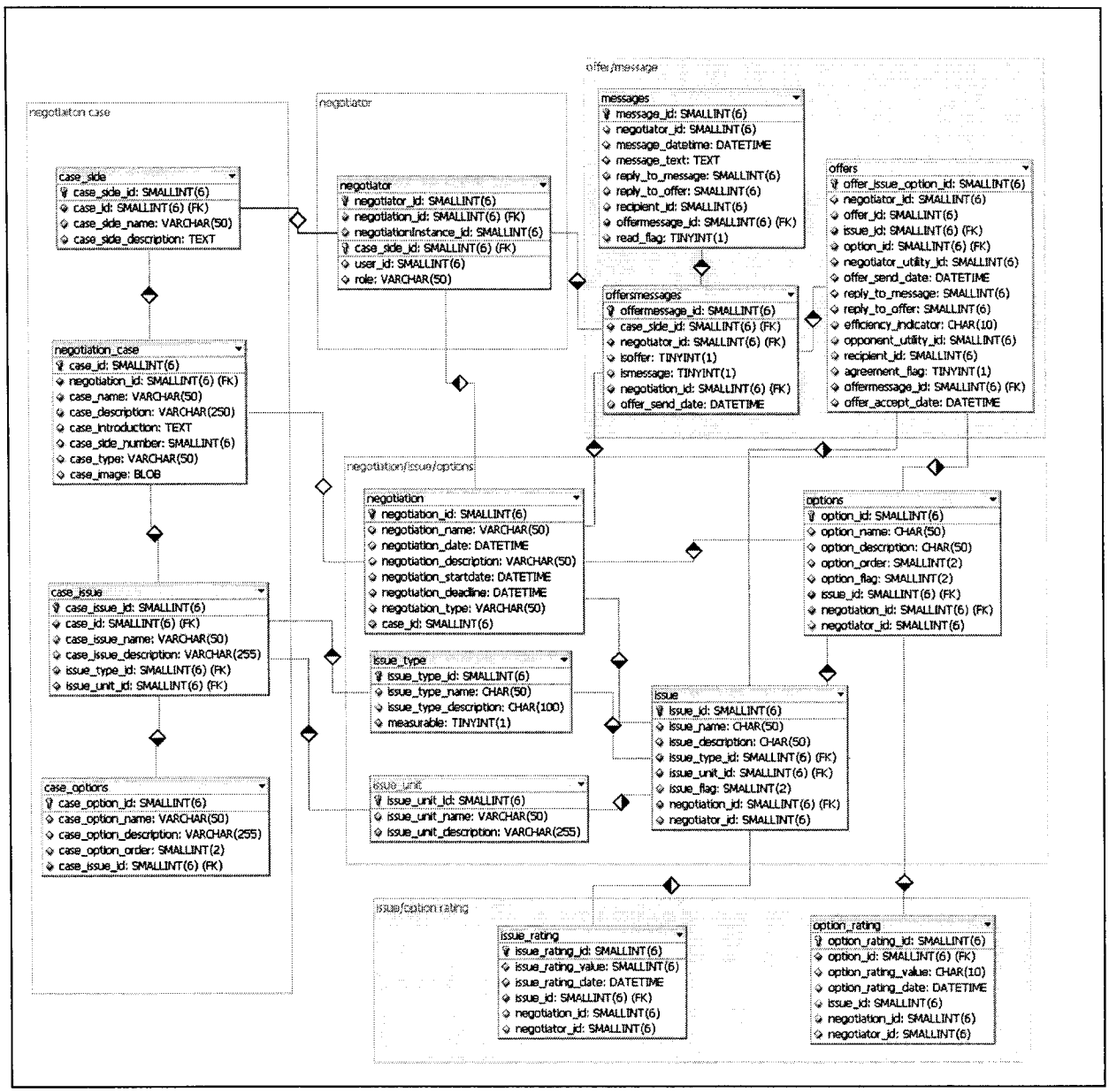

Figure 6.3 Database schema of NCDB in INSS

7. Development and implementation

Based on the system analysis and design above, INSS negotiation protocol can be developed and implemented in Invite by following these five steps:

1. Translate negotiation process to protocol. This includes selecting a negotiation process model that meets the specific protocol or negotiation requirements, then decomposing the process into states and constructs, and decomposing states into activities. A six-phase negotiation process model is used in INSS (see Figure 6.1), which is decomposed by following the top-down approach (see Figure 3.6).
2. Design negotiation protocol. Negotiation protocol can be designed and represented by using either state diagrams or tables. INSS protocol is illustrated with tables in Section 6.2.2.
3. Construct negotiation protocol database and negotiation content database. Following the database-driven approach, the INSS protocol tables can be transferred into the NPDB in Invite. The NCDB schema is designed based on the process model and the features of INSS (see Figure 6.3).
4. Develop page composers and components. Corresponding to the states/activities in INSS protocol, new page composers and components can be identified, and then developed and implemented one by one following FLiP; others can be implemented by reusing those for Inspire protocol.
5. Deploy page composers on Invite platform. The new fuseactions and fuses developed for INSS protocol are set up in the circuit.xml.cfm files, in which the fuses are linked as fuseactions. According to the circuits defined in Invite, the developed files, i.e. fuses, are uploaded to corresponding directories on the server. As a result, the new page composers and components can be integrated in Invite.

Steps 1 and 2 have been completed in previous sections. Thus, the rest of the work is based on the system analysis and design necessary to perform database construction (Section 7.1), page composer development (Section 7.2) and system deployment (Section 7.3).

7.1 Database construction

Currently, MySQL database system is used in Invite to build and manage tables, fields and relationships.

To implement INSS negotiation protocol in Invite, the protocol tables need to be transferred into the NPDB and represented as the initial protocol and intervening rules. Following the ten steps of database-driven approach (see Table 5.5), all the associated tables can be

constructed. Currently, these tables have to be created manually by using MySQL-Front. The actual protocol and instance tables can be automatically set up and updated in the system.

The NCDB in Invite contains several tables for negotiations with other protocols. For INSS, several new tables need to be added. In the new tables, the fields that have been used in current tables are retained (e.g. case_id, issue_id, etc.). The relationships between these new tables and the current tables need to be specified (see Figure 6.3).

The finished databases are uploaded to the database server, and the databases are specified on the ColdFusion application server using Data Source Names (DSNs). When called by Invite, the DSNs connect ColdFusion applications to the database, once access permission is granted.

7.2 Page composer development

The page composers for INSS protocol, which have been identified in Table B.1 in Appendix B, can be developed following the FLiP. First, each page composer is decomposed into one or more components (fuseactions) by wireframing, and then the page composers as well as the components indicated with "V" can be designed with prototyping. Normally, these components are mainly for displaying information or providing user interface with specific layouts and templates. Afterwards, each fuseaction can be decomposed into a set of fuses. Table C.1 in Appendix C shows all the fuseactions and fuses in INSS. These fuseactions and fuses need to be set up in the circuit.xml.cfm files. The fuses are coded with CFML.

Once a fuse is developed, unit testing can be conducted. As a component-based system, each fuseaction is tested as well after all fuses of the fuseaction are implemented. ColdFusion provides several debugging and testing tools for Web application development. For example, all run-time information, such as data activity, form and session variables, and exception information, can be traced. The system performance can also be monitored.

Once fuseactions have passed the testing, they can be integrated in building the system. There are two considerations for integration. On the page composer and components level, fuses and fuseactions are integrated to perform the activities and states of INSS protocol; on the system level, all page composers are integrated with the negotiation controller for execution of the protocol. These two levels of integration are both implemented by setting up the circuit.xml.cfm files. The organization of the Fusebox environment makes unit testing

and integration a structured process, leaving only minor glitches to be resolved using overall testing.

7.3 System deployment

System deployment refers to the assertion of the proper execution of the application in the production environment. For INSS, there are two main jobs for system deployment: the deployment of the INSS protocol used to build the INSS system in the Invite platform and the deployment of the INSS system on the server.

Basically, setting up INSS protocol in the Invite platform, includes the following steps:

1. Add INSS negotiation protocol in the NPDB of Invite;
2. Add required tables in the NCDB of Invite;
3. Create Data Sources for INSS on ColdFusion Server;
4. Upload developed fuses to respective directory on the server;
5. Update circuit.xml.cfm files to add fuseactions for INSS.

Once INSS protocol has been deployed in Invite, the INSS system is built for deployment on the server. Since INSS is programmed with Fusebox technology and integrated with the Invite platform, there are very few changes required after being deployed to the public server from the development environment. Mainly, the changes requires concern configuring the ColdFusion Server, which includes creating DSNs and setting up the system path for Java classes.

The resulting INSS is found at: http://137.122.146.100/negotiations/Invite_INSS/index.cfm. See Appendix D for screenshots of INSS.

8. Testing

The final testing of INSS consisted of a usability test based on a convenient sample. People from various backgrounds and age groups were invited to use INSS in two main scenarios to verify the actual functionality of the system's concepts in a hands-on approach (Whitten and Bentley 1998). The main focus was to determine whether or not the design and implementation of INSS in the Invite platform was ideal for Web-based negotiation. Furthermore, the main features of INSS were also examined to confirm that it succeeded in supporting negotiators. Section 8.1 provides the testing case and two testing scenarios, Section 8.2 describes the manner in which the usability tests were conducted, and Section 8.3 presents the results of the testing.

8.1 Testing case and scenarios

INSS can be seen as an extended and expanded version of Inspire. Thus, the Itex-Cypress negotiation used in Inspire was selected to be the basic testing case of INSS. The negotiators represent two companies: Itex Manufacturing, a producer of bicycle parts, and Cypress Cycles, which builds bicycles. To test the functionality and features of INSS, two scenarios were designed:

Scenario 1: adding new issues and options during the negotiation process

This scenario is similar to the scenario in Inspire, but there are only two pre-defined issues: the price of the bicycle components and the delivery schedules. Each issue also has 3 to 5 predefined options. More importantly, negotiators are allowed to propose/add other issues and options during the negotiation process.

Scenario 2: hosting negotiations and instances

In this situation, users can host negotiations using the basic case above or create a new case if they need to do so. Then, they can invite or join other users to negotiate on the case. In addition, users can try to make a better gain by conducting more than one negotiation at the same time if they set up additional negotiation instances within the same case.

8.2 Usability testing

The entire evaluation was comprised of the two negotiation scenarios above and a three-part questionnaire (pre-test, scenario 1post-test and scenario 2 post-test). See Appendix E for the questionnaire based on Chen's work (Chen, Kersten et al. 2003) and Invite's current research

experiments. The questions were read and explained to the users before they answered the questionnaire.

Fourteen individuals who had volunteered to test INSS were divided into seven pairs. In order to evaluate the system, one volunteer in a pair of served as Itex Manufacturing and another served as Cypress Cycles. They started by completing the pre-test part of the questionnaire to determine their expertise level on negotiation and Internet usage. A demonstration of INSS was given to show all the features and functionalities of the system, during which user comments and questions were recorded. The demonstration served to teach the volunteers how to use the system and the negotiation process. They were then directed into different rooms to start negotiations.

The first negotiation scenario involved adding new issues and options to the negotiation, and began by reading the public information concerning the negotiation. After reading their private information, negotiators could review the pre-defined issues and options. The users were asked to negotiate at least three issues and at least three options per issue. Thus, one or more issues needed to be proposed and added during the negotiation. Once an issue was added, all the negotiators indicated their preferences on the issues and options. Following this, they started exchanging offers and messages until the negotiation was terminated (either because an agreement was reached or because one party walked away). Finally, the volunteers were asked to respond to the second part of the questionnaire. This measured how well the INSS (for adding issues and options during the negotiation process) serves to support Web-based negotiations.

The second scenario involved hosting negotiations and instances. This time, the volunteers were asked to negotiate just two issues with at least three options per issue. The scenario started when one user hosted a new negotiation. This user could either use the given case or create a new case for the negotiation. If a new case was created, a minimum of two issues and at least three options for each issue needed to be added. Then, the user was required to set up one instance with another user for this negotiation. For each instance, the two users followed the same negotiation process as in the first scenario, except that they did not need to add new issues or options during negotiation. If users wanted, they could set up more than one instance with different users. During negotiation, one user could switch to another instance at anytime in order to view its status, make offers, read offers, etc. Thus, the user could gain a better result in the same negotiation case by comparing offers from different users. Once negotiations in the second scenario were completed, volunteers filled out the final part of the questionnaire, to determine how well the INSS (to negotiate on user-defined problems and perform multi-bilateral negotiations) performed from their perspective.

8.3 Results

The responses provided by the users were collected and calculated to present the overall picture of INSS's usability. See Appendix F for the table of results. The average participants access the Internet at least several times a week and are familiar with buying/selling something over the Web. Most of the respondents had some experience at negotiating, whereas only three of these had used negotiation support system before.

Within the first scenario, the general feedback from the user showed that all of the respondents are in favor of employing INSS to conduct or prepare negotiations. The average user claimed that they were satisfied with the usability of the system. They rated their experience as between good and great.

The respondents liked the fact that INSS is easy to use and issues/options can be added during negotiation. In particular, they thought it is helpful to propose new issues and add options when they felt it is hard to make compromise on the existing negotiation issues and options. They also commended the function by which the rating value for a new option to quantitative issues can be calculated automatically. As an improvement, they suggested that more instructions are needed to explain the terminologies on some pages. In addition, the users noted that the more issues added to the negotiation the longer the negotiation would be. They also mentioned that the e-mail notification is not necessary for online negotiators, because they can check incoming information in the system.

In terms of the hosting negotiation and instances, most of the volunteers responded that the system provided a marketplace for users easily to negotiate with many people for the same case. Almost one-third of them created their very own case for negotiation, and then 15% more of the participants reached an agreement. In addition, 93% of the participants agreed conducting more negotiation instances is helpful to improve their negotiation outcome. They rated their satisfaction level with this feature as between good and very good. Most individuals said that they would use INSS to practice or conduct negotiations in the future.

Some respondents expressed that the current features of the system to conduct multi-bilateral or multilateral negotiations are limited. For example, at one time users can only view the current negotiating instance's information. If the users want to check the negotiation status of another instance, they have to switch to the instance. The system could improve by showing some important information of all instances together. In particular, if the system could show all the offers from different negotiators on one page, users can easily compare these offers to make a final decision.

Moreover, from the negotiation history graphs and NegoDance graphs which were generated by the system for these negotiations, we do see that users conducted negotiation using different strategies and tactics. In Figure 8.1, both of the two negotiators started from a high-rating value initial offer, and then exchanged offers with a lower rating value. Others were negotiating by rising the offers' rating value (see Figure 8.2).

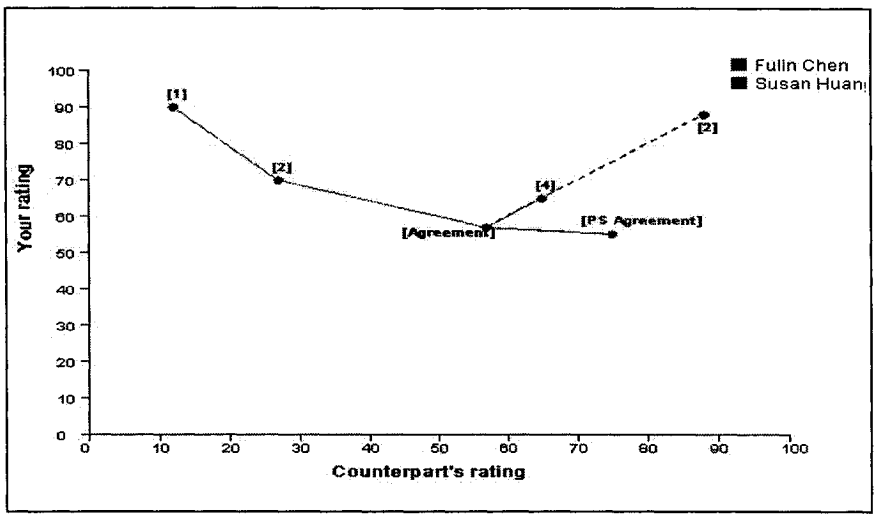

Figure 8.1 NegoDance graph 1

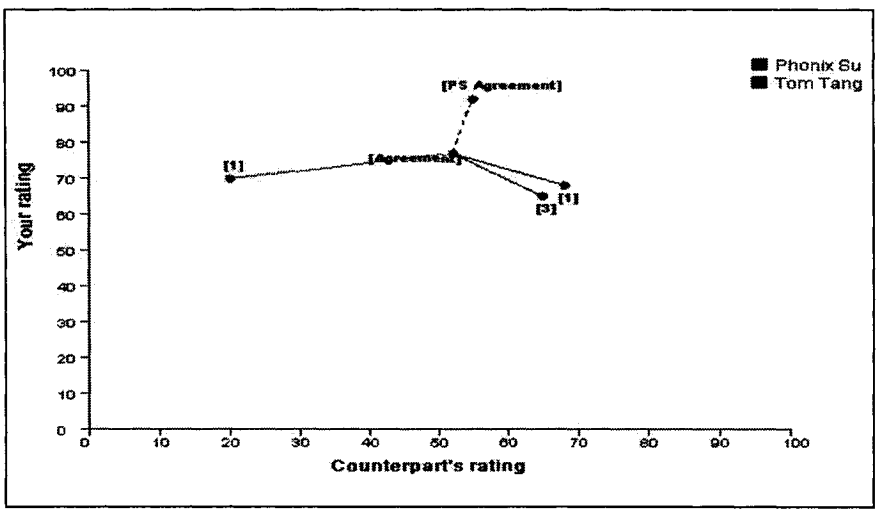

Figure 8.2 NegoDance graph 2

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

9. Conclusion

The literature review demonstrated that as e-commerce and e-business grow and e-markets mature negotiation will play a more important role in these fields. Moreover, in real-life negotiations it is required to support negotiators to deal with various complex issues, such as negotiation case management, negotiation structure changing, different types of negotiations, etc.

The contributions of this thesis are: (1) proof of concepts to construct a protocol for realistic negotiation over open and dynamically modifiable problems; (2) construction of INSS in the Invite software platform; (3) development of page composers and components for case creation and management, hosting of negotiations and their instances, and addition issue/options; and (4) Usability testing and verification.

The methodological choices to develop and implement the ENS in the Invite software platform consist of utilizing the main stages of the SDLC integrated with prototyping and RAD to apply the Fusebox methodology and a database-driven approach in relation to the system components and a negotiation protocol. The resulting system, INSS, is evaluated through usability testing by a small group, which showed that it is well received. 93% of the participants said that they would use INSS to conduct or prepare negotiations, and most of them found the features of the system useful in negotiations.

9.1 Limitations

The limitations to this work are due mostly to time and budget constraints as well as the current design of the Invite platform, such that restrictions are places on the functional and technical aspects of INSS, concerning both the negotiation protocol and types of negotiations. Several hardwired components for INSS protocol implementation in Invite are listed in Appendix G.

Two functional flaws that respondents in the evaluation found are the lack of sufficient instructions to support usage of the system and the difficulty to view the information of all negotiation instances at one time. If INSS is to progress to public testing, then a tutorial and help features will be necessary to teach and assist users. Although the solution for the second problem could be to provide some important information of all instances on one page, it is still limited by the current design of Invite which mainly supports bilateral negotiations.

Also, the system should be able to identify whether a negotiator is negotiating online or not, and then only send the e-mail notification to those offline. Another technical shortcoming of

the system is that the graphs are showing negotiation history with limited features. They were implemented with Java classes. Currently, the Invite development group is upgrading the system to ColdFusion 7, in which several more functional graph classes are provided.

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

9.2 Future work

In order to determine the usefulness of the system and its impact on the negotiation outcomes as well as its possible extensions, further testing in a large scale is necessary to be conducted. This may be done by the experiments of using e-negotiation systems and the comparison to other ENSs (different protocols) in Invite.

In the meantime, more features may be added on the system to support real-life negotiations. For example, partial offers which may not contain all negotiation issues as one package can be exchanged in the system. In addition, some contract templates could be introduced to the case library both for creating negotiation case and eliciting negotiation issues. As more functions are built in Invite, other types of negotiations can be supported in INSS, such as multi-lateral negotiations, group negotiations, etc.

10. References

Ambler, S. W. (2004). The Object Primer: Agile Model Driven Development with UML 2, Cambridge University Press.

Bakos, Y. (1998). "The emerging role of electronic marketplaces on the Internet." Communications of the ACM 41(8): 35-42.

Beam, C., A. Segev, M. Bichler and R. Krishnan (1999). "On Negotiations and Deal Making in Electronic Markets." Information Systems Frontiers 1(3): 241-258.

Benbasat, I. and F. Lim (1995). "A Framework for Communication Support in Group Work with Special Reference to Negotiation Systems." Group Decision and Negotiation 4(2): 135-158.

Benyoucef, M., H. Alj, M. Vezeau and R. K. Keller (2001). "Combined Negotiations in E-Commerce: Concepts and Architecture." Electronic Commerce Research 1(3): 277-299.

Benyoucef, M., R. K. Keller, S. Lamouroux, J. Robert and V. Trussart (2000). Towards a Generic E-Negotiation Platform. Sixth International Conference on Re-Technologies for Information Systems, Zurich, Switzerland.

Bichler, M., G. E. Kersten and C. Weinhardt (2003). "Special Issue on E-Negotiations." Group Decision and Negotiation 12(2 and 4).

Bichler, M. and H. Werthner (2000). "A Classification Framework of Multidimensional, Multi-Unit Procurement Negotiations." 11th International Workshop on Database and Expert Systems Applications (DEXA'00).

Bui, T. (1994). "Evaluating Negotiation Support Systems: A Conceptualization." 27th Annual Hawaii International Conference on System Science.

Bui, T. and M. F. Shakun (2002). Negotiation Support Systems Minitrack. The 36th Hawaii International Conference on System Sciences, Hawaii, IEEE Computer Society.

Chavez, A. and P. Maes (1996). "Kasbah: An Agent Marketplace for Buying and Selling Goods." Proceedings of the First International Conference on the Practical Applications of Intelligent Agents and Multi-agent Technology: 75-90.

Chen, E., G. E. Kersten and R. Vahidov (2003). An e-marketplace for agent-supported commerce negotiations, InterNeg Research Papers.

Chen, E., G. E. Kersten and R. Vahidov (2005). "Agent-supported Negotiations on E-marketplace." International Journal of Electronic Business 3(1): 28-49.

Cho, S. (2001). "A Framework for the Evaluation of an Electronic Marketplace Design with Evolutionary Negotiation Support." IEEE (HICSS-34) 5.

Cranor, L. F. and P. Resnick (2000). "Protocols for Automated Negotiations with Buyer Anonymity and Seller Reputations." NetNomics 2(1): 1-24.

Ehtamo, H., R. P. Hamalainen and V. Koskinen (2004). An e-learning module on negotiation analysis. Hawaii International Conference on System Sciences, Hawaii, IEEE Computer Society Press.

Eriksson, J., N. Finne and S. Janson (1999). "SICS MarketSpace - An Agent-Based Market Infrastructure." Proceedings of the 1998 Workshop on Agent-Mediated Electronic Trading.

Feldman, S. (1999). The Objects of E-Commerce, Keynote speech at ACM 1999 Conference on OOPLSA. 2004.

Ghee-Soon Lim, S. and J. K. Murnigham (1994). "Phases, Deadlines, and the Bargaining Process." Organizational Behavior & Human Decision Processes 58(2): 153-171.

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

Gulliver, P. H. (1979). Disputes and Negotiations: A Cross-Cultural Perspective. Orlando, FL, Academic Press.

Guttman, R., A. Moukas and P. Maes (1998). "Agents as Mediators in Electronic Commerce." International Journal of Electronic Markets 8(1).

Guttman, R. H. and P. Maes (1998). "Agent-mediated Integrative Negotiation for Retail Electronic Commerce." Workshop on Agent Mediated Electronic Trading (AMET'98).

Holmes, M. (1992). Phase Structures in Negotiation. Communication and Negotiation, Sage: Newbury Park, CA.

Holsapple, C. W. and A. B. Whinston (1996). Decision Support Systems: A Knowledge-Based Approach, West Publishing Company.

InterNeg (2005). http://InterNeg.org, InterNeg Group.

Jennings, N. R., P. Faratin, A. R. Lomuscio, S. Parsons, M. J. Wooldridge and C. Sierra (2001). "Automated Negotiations: Prospects, Methods and Challenges." Group Decision and Negotiation 10(2): 199-215.

Kersten, G. E., Ed. (1997). Support for Group Decision and Negotiations: An Overview. Multicriteria Analysis, Springer Verlag: Heidelberg.

Kersten, G. E. (2002). "The Science and Engineering of E-negotiation: Review of the Emerging Field." InterNeg Research Papers.

Kersten, G. E. (2004). E-negotiation systems: Interaction of people and technologies to resolve conflicts. UNESCAP Third Annual Forum on Online Dispute Resolution, Melbourne, Australia.

Kersten, G. E. (2005). Personal communication. S. Wu. Ottawa.

Kersten, G. E. and H. Lai (2005). Satisfiability and Completeness of Protocols for Electronic Negotiations, InterNeg Research Papers.

Kersten, G. E. and K. P. Law (2003). SimpleNS. Montreal. 2004.

Kersten, G. E., K. P. Law and S. E. Strecker (2004). A Software Platform for Multiprotocol E-Negotiations, InterNeg Research Papers.

Kersten, G. E. and G. Lo (2003). "Aspire: Integration of Negotiation Support System and Software Agents for E-Business Negotiation." International Journal of Internet and Enterprise Management 3.

Kersten, G. E. and S. J. Noronha (1998a). Negotiations and the Internet: Users' Expectations and Acceptance, InterNeg Research Papers.

Kersten, G. E. and S. J. Noronha (1998b). "Negotiations via the World Wide Web: A Cross-culture Study of Decision Making." Group Decision and Negotiation 8(3): 252-279.

Kersten, G. E. and S. J. Noronha (1999). "WWW-based Negotiation Support: Design, Implementation, and Use." Decision Support Systems 25: 135-154.

Kersten, G. E. and S. Szpakowicz (1998). Modelling Business Negotiations for Electronic Commerce. Intelligent Information Systems: Proceedings of the 7th Workshop, Malbork, Poland, IPI PAN.

Kilgour, D. M., L. Fang and K. W. Hipel (1996). "Negotiation Support Using the Decision-Support System GMCR." Group Decision and Negotiation 5: 371-383.

Kim, J. and A. Segev (2003). A Framework for Dynamic eBusiness Negotiation Processes. IEEE Conference on E-Commerce, CITM.

Lempereur, A. (2004). "Updating Negotiation Teaching Through The Use of Multimedia Tolls." International Negotiations Journal 9(1).

Madanmohan, T. R., G. E. Kersten, S. J. Noronha, M. Kersten and D. Cray (1999). Learning Negotiations with Web-based Systems. A Resource Book of Methods and Applications, Kluwer Academic.

Maes, P., R. H. Guttman and A.G. Moukas (1999). "Agents that Buy and Sell: Transforming Commerce as we Know It." Communications of the ACM 42(3): 81-91.

Martin, J. (1991). Rapid Application Development, MacMillan.

McConnell, S. (1996). Rapid Development: Taming Wild Software Schedules, Microsoft Press.

Moore, D., T. Kurtzberg, L. Thompson and M. Morris (1999). "Long and Short Routes to Success in Electronically Mediated Negotiations: Group Affiliations and Good Vibrations." Organizational Behavior and Human Decision Processes 77(1): 22-43.

Neumann, D., M. Benyoucef, S. Bassil and J. Vachon (2003). "Applying the Montreal taxonomy to State of the Art E-Negotiation Systems." Group Decision and Negotiation 12(4): 287-310.

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

Paliwal, A. V., N. Adam, V. Atluri and Y. Yesha (2003). "Electronic Negotiation of Government Contracts through Transducers." CIMIC Technical Report TR-2003-01.

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

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

Rahwan, I., R. Kowalczyk and H. H. Pham (2001). "Intelligent Agents for Automated One-to-Many e-Commerce Negotiation." Proceedings of the 25th Australasian Computer Science Conference: 197-204.

Reeves, D. M., M. P. Wellman and B. N. Grosof (2001). "Automated Negotiation from Declarative Contract Descriptions." ACM(AGENTS'01).

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

Schmid, B. F. and U. Lechner (1999). Logic for Media - The Computational Media Metaphor. The 32nd Annual Hawaii International Conference on System Sciences, Hawaii.

Schmid, B. F. and M. A. Lindemann (1998). "Elements of a Reference Model for Electronic Markets." IEEE Computer Society Press 4: 193-200.

Schoop, M. (2003). "Negoisst: A Negotiation Support System for Electronic Business-to-Business Negotiations in E-Commerce." Data and Knowledge Engineering 47(3): 371-401.

Schoop, M. and C. Quix (2001). "DOC.COM: A Framework for Effective Negotiation Support in Electronic Marketplaces." Computer Networks 37(2): 153-170.

Shim, J. and N. Hsiao (1999). "A Literature Review on Web-Based Negotiation Support System." Documentation for Web-based Negotiation Training System (WNTS).

SmartSettle (2004). http://www.smartsettle.com.

Starke, K. and A. Rangaswamy (1999). Computer-Mediated Negotiations: Review and Research Opportunities, University Park, eBusiness Research Centre.

Stroebel, M. (2001). "Design of Roles and Protocols for Electronic Negotiations." Electronic Commerce Research 1(3): 335-353.

Stroebel, M. (2003). Engineering Electronic Negotiations. New York, Kluwer.

Stroebel, M. and C. Weinhardt (2002). "The Montreal Taxonomy for Electronic Negotiations." Group Decision and Negotiation.

TechtargetCom (2003). http://www.whatis.com.

Teich, J., H. Wallenius, J. Wallenius and A. Zaitsev (2001). "Designing Electronic Auctions: An Internet-Based Hybrid Procedure Combining Aspects of Negotiations and Auctions." Electronic Commerce Research(1): 301-314.

Thiessen, E. M. and D. P. Loucks, Eds. (1994). ICANS: Interactive Computer Assisted Negotiation Support. Computer Assisted Negotiation and Mediation: Prospects and Limits. Cambridge, MA, Harvard, Harvard Law School.

Thiessen, E. M., D. P. Loucks and J. R. Stedinger (1998). "Computer-Assisted Negotiations of Water Resources Conflict." Group Decision and Negotiation 7(2): 109-129.

Thomas, C. J. and B. J. Wilson (2002). A Comparison of Auctions and Multilateral Negotiations. The RAND Journal of Economics. 33: 140-155.

Thomas, C. J. and B. J. Wilson (2005). Verifiable Offers and the Relationship Between Auctions and Multilateral Negotiations. Economic Journal: forthcoming.

Torroni, P. and F. Toni (2001). "Extending a Logic Based One-to-One Negotiation Framework to One-to-Many Negotiation." Engineering Societies in the Agents World II: Second International Workshop.

Turban, E., J. Lee, D. King and H. M. Chung (1999). Electronic commerce: a managerial perspective. Upper Saddle River, N.J., Prentice Hall.

Vakali, A., L. Angelis and D. Pournara (2001). "Internet Based Auctions: A Survey on Models and Applications." ACM SIGecom Exchanges 2(2): 6-15.

Vliet, H. v. (2000). Software engineering: principles and practice, John Wiley.

Vulkan, N. and N. R. Jennings (1999). "Efficient Mechanisms for the Supply of Services in Multi-Agent Environments." Proceedings of 1st International Conference on Information and Computation Economies.

Walton, D. N. and E. C. W. Krabbe (1995). Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning, SUNY Press.

Walton, R. E. and R. B. McKersie (1965). A behavioral theory of labor negotiations: an analysis of a social interaction system. New York, McGraw-Hill.

Whitten, J. and L. Bentley (1998). Systems Analysis and Design Methods. Boston, McGraw-Hill.

Wise, R. and D. Morrison (2000). "Beyond the Exchange: The Future of B2B." Harvard Business Review.

Wurman, P., M. P. Wellman and W. E. Walsh (1998). The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents. Second International Conference on Autonomous Agents, Minneapolis.

Yuan, Y., J. B. Rose and N. Archer (1998). "A Web-Based Negotiation Support System." Electronic Markets 8(3).

Zlotkin, G. (1996). "Mechanisms for Automated Negotiation in State Oriented Domains." Journal of Artificial Intelligence Research 5: 163-238.

Appendix A: INSS negotiation protocol

1. Initial protocol

Table A.1 Initial sequence/states table of INSS protocol

	Sequence Name	Initial State	Mandatory State	Initial Optional States (Intervening Information)	Exit Points (Intervening Information)
			Block 1		
100	Browse Public Negotiation	BrowsePublicNegotiation	BrowsePublicNegotiation	BrowsePublicNegotiation	Browse Private Negotiation Host New Negotiation Read Negotiation Details
101	Browse Private Negotiation	BrowsePrivateNegotiation	BrowsePrivateNegotiation	BrowsePrivateNegotiation	Browse Public Negotiation Host New Negotiation Read Negotiation Details
102	Read Negotiation Details	ReadNegotiationDetails	ReadNegotiationDetails	ReadNegotiationDetails	Browse Public Negotiation Browse Private Negotiation Host New Negotiation Setup Negotiation Join Negotiation
103	Select Negotiation Instance	SelectNegotiationInstance	SelectNegotiationInstance	SelectNegotiationInstance ReadInstanceDetails Check Negotiation Status	Start Negotiation Continue Negotiation
			Block 2		
110	Host New Negotiation	WriteNegotiationDetails	WriteNegotiationDetails	WriteNegotiationDetails Select Existing Case	Browse Public Negotiation Browse Private Negotiation Write New Case

	Sequence Name	Initial State	Mandatory State	Initial Optional States (Intervening Information)	Exit Points (Intervening Information)
111	Setup Negotiation	SetupNegotiation	SetupNegotiation	SetupNegotiation	Browse Public Negotiation Browse Private Negotiation Host New Negotiation Select Negotiation Instance
			Block 3		
120	Write New Case	WritePublicCase	WritePublicCase	WritePublicCase	Write Private Case
121	Write Private Case	WritePrivateCase	WritePrivateCase	ReadPublicCase RevisePublicCase ReadPrivateCase RevisePrivateCase WritePrivateCase	Enter Issue
122	Enter Issue	EnterIssue	EnterIssue	EnterIssue ReadPublicCase RevisePublicCase ReadPrivateCase RevisePrivateCase	Enter Option
123	Enter Option	EnterOption	EnterOption	EnterOption ReadPublicCase RevisePublicCase ReadPrivateCase RevisePrivateCase	Enter Issue Complete Case
124	Complete Case	CompleteCase	CompleteCase	ReadPublicCase RevisePublicCase ReadPrivateCase RevisePrivateCase AddCaseToLibrary	Browse Public Negotiation Browse Private Negotiation Invite Negotiator Start Negotiation
			Block 4		
130	Invite Negotiator	AddNegotiator	AddNegotiator	ReadNegotiationDetails ReadPublicCase ReadPrivateCase AddNegotiator (***Invitation***)	Wait for Joining

	Sequence Name	Initial State	Mandatory State	Initial Optional States (Intervening Information)	Exit Points (Intervening Information)
131	Wait for Joining	WaitForJoining	WaitForJoining	WaitForJoining ReadNegotiationDetails ReadPublicCase	
132	Receive Invitation	ReadInvitation	ReadInvitation	ReadNegotiationDetails ReadPublicCase ReadInvitation	Start Negotiation (***JoinNotification***) Reject Negotiation Invitation
133	Reject Negotiation Invitation	RejectInvitation	RejectInvitation	RejectInvitation (***NonJoinNotification***)	Browse Public Negotiation
134	Read Rejection to Invitation	ReadRejection	ReadRejection	ReadRejection	Invite Negotiator
			Block 5		
140	Join Negotiation	JoinNegotiation	JoinNegotiation	ReadNegotiationDetails ReadPublicCase JoinNegotiation (***JoinApplication***)	Wait for Join Approval
141	Wait for Join Approval	WaitForJoinApproval	WaitForJoinApproval	WaitForJoinApproval ReadNegotiationDetails ReadPublicCase	Browse Public Negotiation Browse Private Negotiation Host New Negotiation
142	Receive Join Application	ReadJoinApplication	ReadJoinApplication	ReadNegotiationDetails ReadPublicCase ReadJoinApplication	Start Negotiation (***JoinApproval***) Reject Join Application
143	Reject Join Application	RejectJoinApplication	RejectJoinApplication	RejectJoinApplication (***JoinRejection***)	
144	Read Rejection to Join Application	ReadJoinRejection	ReadJoinRejection	ReadJoinRejection	Browse Public Negotiation
			Block 6		
200	Start Negotiation	ReadPublicCase	ReadPublicCase	ReadPublicCase	Read Private Case

	Sequence Name	Initial State	Mandatory State	Initial Optional States (Intervening Information)	Exit Points (Intervening Information)
201	Read Private Case	ReadPrivateCase	ReadPrivateCase	ReadPublicCase ReadPrivateCase ReviewIssues AddQuantitativeOption (***NewQuantitativeOption***)	Propose New Issue Add Qualitative Option Rate Issues
			Block 7		
210	Propose New Issue	ProposeNewIssue	ProposeNewIssue	ReadPublicCase ReadPrivateCase WriteMessage (***Message***) ReviewIssues ProposeNewIssue (***IssueProposal***) AddQuantitativeOption (***NewQuantitativeOption***)	Continue Previous Sequence Add Qualitative Option
211	Continue Previous Sequence	PreviousState	PreviousMandatoryState		
212	Receive Issue Proposal	ReceiveIssueProposal	ReadIssueProposal	ReadPublicCase ReadPrivateCase WriteMessage (***Message***) ReviewIssues ReceiveIssueProposal	Accept Issue Proposal Reject Issue Proposal Modify Issue Proposal
213	Accept Issue Proposal	AcceptIssueProposal	AcceptIssueProposal	AcceptIssueProposal (***IssueApproval***)	
214	Reject Issue Proposal	RejectIssueProposal	RejectIssueProposal	RejectIssueProposal (***Issue Reject***)	
215	Modify Issue Proposal	ModifyIssue	ModifyIssue	ReadPublicCase ReadPrivateCase WriteMessage (***Message***) ModifyIssue (***Issue Proposal***)	Continue Previous Sequence

	Sequence Name	Initial State	Mandatory State	Initial Optional States (Intervening Information)	Exit Points (Intervening Information)
			Block 8		
220	Add Qualitative Option	AddQualitativeOption	AddQualitativeOption	ReadPublicCase ReadPrivateCase WriteMessage **(Message)** AddQualitativeOption **(NewQualitativeOption)**	
			Block 9		
230	Rate Issues	RateIssue	RateIssue	ReadPublicCase ReadPrivateCase RateIssue WriteMessage ***(Message)*** ReviewIssues AddQuantitativeOption ***(NewQuantitativeOption)***	Rate Options Propose New Issue Add Qualitative Option
231	Rate Options	RateOption	RateOption	ReadPublicCase ReadPrivateCase RateIssue RateOption WriteMessage ***(Message)*** ReviewIssues AddQuantitativeOption ***(NewQuantitativeOption)***	Rate Packages Propose New Issue Add Qualitative Option
232	Rate Packages	RatePackage	RatePackage	ReadPublicCase ReadPrivateCase RateIssue RateOption RatePackage WriteMessage ***(Message)*** ReviewIssues AddQuantitativeOption ***(NewQuantitativeOption)***	Exchange Offer Propose New Issue Add Qualitative Option

	Sequence Name	Initial State	Mandatory State	Initial Optional States (Intervening Information)	Exit Points (Intervening Information)
			Block 10		
300	Exchange Offer	ConstructOffer	ReadOffer	ReadPublicCase ReadPrivateCase RateIssue RateOption RatePackage WriteMessage ***(Message)*** ConstructOffer ***(Offer)*** ReviewIssues AddQuantitativeOption ***(NewQuantitativeOption)***	Agreement ***(Agreement)*** Propose New Issue Add Qualitative Option
			Block 11		
400	Agreement	ReadAgreement	ReadAgreement	History ReadAgreement	Improve agreement Terminate Negotiation ***(Termination)***
			Block 12		
500	Improve agreement	NegoDance	ReadPSOffer	ConstructPSOffer ***(PSOffer)*** ConstructPSMessage ***(PSMessage)*** History NegoDance	Reach Post-negotiation Settlement Agreement ***(PSAgreement)***
			Block 13		
600	Reach Post-negotiation Settlement Agreement	ReadPSAgreement	ReadPSAgreement	History NegoDance ReadPSAgreement	Terminate Negotiation ***(Termination)***
			Block 14		
700	Terminate Negotiation	TerminateNegotiation	TerminateNegotiation	History	

2. INSS intervening rule

Table A.2 Addition of optional states table of INSS protocol

Information Type	Sequence	Optional state to add
Invitation	100 Browse Public Negotiation 101 Browse Private Negotiation 102 Read Negotiation Details 110 Host New Negotiation	ReadInvitation
JoinApplication	100 Browse Public Negotiation 101 Browse Private Negotiation 102 Read Negotiation Details 110 Host New Negotiation	ReadJoinApplication
IssueProposal	201 Read Private Case 210 Propose New Issue 212 Receive Issue Proposal 215 Modify Issue Proposal 230 Rate Issues 231 Rate Options 232 Rate Packages 300 Exchange Offer 500 Post Settlement	ReadIssueProposal
Offer	300 Exchange Offer	ReadOffer History
Message	210 Propose New Issue 212 Receive Issue Proposal 215 Modify New Issue 220 Add Qualitative Option 230 Rate Issues 231 Rate Options 232 Rate Packages	ReadMessage
	300 Exchange Offer	ReadMessage History
PSOffer	500 Improve Agreement	ReadPSOffer
PSMessage	500 Improve Agreement	ReadPSMessage

Table A.3 Update initial states table of INSS protocol

Information Type	Sequence	Initial state to set
Offer	300 Exchange Offer	ReadOffer
Message	300 Exchange Offer	ReadMessage
PSOffer	500 Improve Agreement	ReadPSOffer
PSMessage	500 Improve Agreement	ReadPSMessage

Table A.4 Forwarding table of INSS protocol

Information Type	Allowed origination (From sequence)	Destination (To sequence)	Destination (To State)
JoinNotification	100 Browse Public Negotiation 101 Browse Private Negotiation 102 Read Negotiation Details 110 Host New Negotiation	200 Start Negotiation	ReadPublicCase
NonJoinNotification	100 Browse Public Negotiation 101 Browse Private Negotiation 102 Read Negotiation Details 110 Host New Negotiation	133 Reject Negotiation Invitation	RejectInvitation
JoinApproval	141 Wait for Join Approval	200 Start Negotiation	
JoinRejection	141 Wait for Join Approval	144 Read Rejection to Join Application	
IssueProposal	201 Read Private Case 230 Rate Issues 231 Rate Options 232 Rate Packages 300 Exchange Offer 500 Post Settlement	212 Receive Issue Proposal	ListIssueProposal
IssueApproval	210 Propose New Issue	201 Read Private Case	ReadPrivateCase
IssueReject	210 Propose New Issue	210 Propose New Issue	ListIssueProposal
NewQualitativeOption	231 Rate Options	231 Rate Options	RateOption

Information Type	Allowed origination (From sequence)	Destination (To sequence)	Destination (To State)
	232 Rate Packages 300 Exchange Offer 500 Improve Agreement		
Offer	300 Exchange Offer	300 Exchange Offer	ReadOffer
Message	210 Propose New Issue 212 Receive Issue Proposal 215 Modify New Issue 220 Add Qualitative Option 230 Rate Issues 231 Rate Options 232 Rate Packages 300 Exchange Offer	210 Propose New Issue 212 Receive Issue Proposal 215 Modify New Issue 220 Add Qualitative Option 230 Rate Issues 231 Rate Options 232 Rate Packages 300 Exchange Offer	ReadMessage
Agreement	300 Exchange Offer	400 Agreement	ReadAgreement
PSOffer	500 Improve Agreement	500 Improve Agreement	ReadPSOffer
PSMessage	500 Improve Agreement	500 Improve Agreement	ReadPSMessage
PSAgreement	500 Improve Agreement	600 Post-negotiation Settlement Agreement	ReadPSAgreement
Termination	*Current sequence (any)*	999 Termination	History

Appendix B: Page composers and components of INSS

Table B.1 Page composers and components of INSS

Phase	Activity/State	Page Composer	Component (Fuseaction)
Initiation	Login	Login	DspLoginForm (V) ValidateLogin (M) CheckAuthentication (M)
Planning	Browse public negotiations	BrowsePublicNegotiation	ReadPublicNegotiation (M) ShowPublicNegotiation (V)
		ViewPublicNegotiationDetails	UpdateSession (M) ReadPublicNegotiationDetails (M) ReadNegotiationInstance (M) ShowPublicNegotiationDetails (V)
	Browse private negotiations	BrowseMyNegotiation	ReadPrivateNegotiation (M) ShowPrivateNegotiation (V)
		ViewNegotiationDetails	UpdateSession (M) ReadNegotiationDetails (M) ReadNegotiationInstance (M) ShowNegotiationDetails (V)
		CheckNegotiationStatus	ProcessNegotiationInstance (M) ReadNegotiationDetails (M) ShowNegotiationStatus (V)
	Browse case library	BrowseNegotiationCase	ReadCaseList (M) ReadCaseInfo (M) ShowNegotiationCase (V)
		ViewNegotiationCaseSide	ReadCaseSideInfo (M) ShowCaseSide (V)
		ViewNegotiationCaseDetails	ReadCaseDetails (M) ShowCaseDetails (V)
	Create new case	NewNegotiationCase	ShowNegotiationCaseForm (V)
		AddNegotiationCase	InsertNewCase (V) ConfirmInsertNewCase (V)
		AddNegotiationCaseSide	InsertCaseSide (M) ConfirmInsertCaseSide (V)
Agenda setting and exploring the field	Host new negotiation	HostNewNegotiation	ReadCaseList (M) ReadCaseInfo (M) ShowNegotiationForm (V)
		PostNegotiation	InsertNewNegotiation (M) ConfirmInsertNegotiation (V)
	Setup negotiation	SetupNegotiation	ReadNegotiationInfo (M) SelectNegotiator (M) SelectCaseSide (M) SelectProtocol (M) ShowSetupNegotiation (V)
		AddNegotiationInstance	ReadNegotiationInfo (M) InsertNegotiationInstance (M) StartingNegotiation (V)

Phase	Activity/State	Page Composer	Component (Fuseaction)
	Join negotiation	JoinNegotiation	ReadNegotiationInfo (M) SelectNegotiator (M) SelectCaseSide (M) SelectProtocol (M) ShowJoinNegotiation (V)
		ConfirmJoinNegotiation	JoinNegotiation (M) ShowConfirmJoinNegotiation (V)
	Review case	ReviewPublicCase	ReadPublicCase (M) ShowPublicCase (V)
		ReviewPrivateCase	ReadPrivateCase (M) ShowPrivateCase (V)
	Review issue/options	ListIssue	ReadIssueList (M) IssueList (V)
		ReadIssueDetails	ReadIssueDetails (M) IssueDetails (V)
	Propose new issue	NewIssueProposal	ReadIssueList (M) ReadIssueType (M) ReadIssueUnit (M) NewIssueProposal (V)
		ProposeIssue	ProposeIssue (M)
		ConfirmProposeIssue	ReadIssueProposalDetails (M) ConfirmProposeIssue (V)
	Evaluate issue/proposal	ListIssueProposal	ReadIssueProposalList (M) IssueProposalList (V)
		ReadIssueProposalDetails	ReadIssueProposalDetails (M) IssueProposalDetails (V)
	Accept issue proposal	AcceptIssueProposal	AcceptIssueProposal (M)
	Modify issue	ModifyIssue	ReadIssueDetails (M) ReadIssueType (M) ReadIssueUnit (M) ModifyIssue (V)
		ProcessModifyIssue	ModifyIssue (M)
		ConfirmModifyIssue	ReadIssueProposalDetails (M) ConfirmModifyIssue (V)
	Reject issue proposal	RejectIssueProposal	RejectIssueProposal (M)
	Add new option	NewOption	ReadIssueDetails (M) NewOption (V)
		AddOption	AddOption (M)
		ConfirmAddOption	ReadIssueDetails (M) ConfirmAddOption (V)
	Rate issue	RateIssue	ReadIssueInfo (M) RateIssue (V)
		ConfirmRateIssue	confirmRateIssue (M)
	Rate option	RateOption	ReadOptionInfo (M) RateOption (V)
		ConfirmRateOption	ConfirmRateOption (M)
	Rate package	RatePackage	ReadPackageInfo (M) RatePackage (V)

Phase	Activity/State	Page Composer	Component (Fuseaction)
		ConfirmRatePackage	ConfirmRatePackage (M) ConfirmRatePackage (V)
Exchanging offers and arguments	Make offer and Send offer	ConstructOffer	ReadOffer (M) ReadIssueInfo (M) ConstructOffer (M) ConstructOffer (V)
		ProcessOffer	ProcessOffer (M)
		ConfirmOffer	ConfirmOffer (M) ConfirmOffer (V)
	Get offer and Read offer	ReadOffer	ReadOffer (M) ReadOffer (V)
	Make message and Send message	ConstructMessage	ConstructMessage (V)
		ProcessMessage	ProcessMessage (M)
		ConfirmMessage	ConfirmMessage (M) ConfirmMessage (V)
	Get message and Read message	ReadMessage	ReadMessage (M) ReadMessage (V)
	View negotiation history	HistoryInspire	HistoryInspire (M) HistoryInspire (V)
Reaching agreement	Reach agreement (accept offer)	AgreementInspire	ReadOffer (M) AgreementInspire (M) ReadAgreementInfo (M) HistoryInspire (M) AgreementInspire (V)
	Assess agreement	ReadAgreement	ReadAgreementInfo (M) AgreementInspire (V)
		HistoryInspire	HistoryInspire (M) HistoryInspire (V)
		NegoDance	NegoDance (M) NegoDance (V)
	Accept agreement	Termination	Termination (M) Termination (V)
Post-negotiation settlement	Make PS offer and Send PS offer	PSConstructOffer	ReadAgreementInfo (M) PSReadIssueInfo (M) PSConstructOffer (M) PSConstructOffer (V)
		PSProcessOffer	PSProcessOffer (M)
		PSConfirmOffer	PSConfirmOffer (M) PSConfirmOffer (V)
	Get PS offer and Read PS offer	PSReadOffer	PSReadOffer (M) PSReadOffer (V)
	Make PS message and Send PS message	PSConstructMessage	PSConstructMessage (V)
		PSProcessMessage	PSProcessMessage (M)
		PSConfirmMessage	PSConfirmMessage (M) PSConfirmMessage (V)
	Get PS message and Read PS message	PSReadMessage	PSReadMessage (M) PSReadMessage (V)

Phase	Activity/State	Page Composer	Component (Fuseaction)
Reaching PS agreement	Reach PS agreement (accept PS offer)	PSAgreementInspire	PSReadOffer (M) PSAgreementInspire (M) PSReadAgreementInfo (M) HistoryInspire (M) PSAgreementInspire (V)
Concluding negotiation	Close Negotiation	Termination	Termination (M) Termination (V)
	Review negotiation	HistoryInspire	HistoryInspire (M) HistoryInspire (V)

Appendix C: Fuseactions and fuses of INSS

Table C.1 Fuseactions and fuses for INSS

	Fuseaction	Fuse
	DspLoginForm (V)	dspLoginForm.cfm
	ValidateLogin (M)	act_validateuser.cfm
	CheckAuthentication (M)	qry_authenticateUser.cfm
	ReadPublicNegotiation (M)	qry_getPublicNegotiation.cfm
	ShowPublicNegotiation (V)	dsp_publicNegotiation.cfm
	UpdateSession (M)	act_updateSession.cfm
	ReadPublicNegotiationDetails (M)	qry_getPublicNegotiationDetails.cfm
	ShowPublicNegotiationDetails (V)	dsp_publicNegotiationDetails.cfm
	ReadPrivateNegotiation (M)	qry_getMyNegotiation.cfm
	ShowPrivateNegotiation (V)	dsp_myNegotiation.cfm
	ReadNegotiationDetails (M)	qry_getNegotiationDetails.cfm
	ReadNegotiationInstance (M)	qry_getNegotiationInstance.cfm
	ShowNegotiationDetails (V)	dsp_negotiationDetails.cfm
	ProcessNegotiationInstance (M)	qry_getOpponentNegoInfo.cfm qry_getUserNegoInfo.cfm act_setNegotiationInstance.cfm
	ShowNegotiationStatus (V)	dsp_negotiationStatus.cfm
	ReadCaseList (M)	qry_getNegotiationCase.cfm
	ReadCaseInfo (M)	qry_getNegotiationCaseInfo.cfm
	ShowNegotiationCase (V)	dsp_browseNegotiationCase.cfm
	ReadCaseDetails (M)	qry_getNegotiationCaseDetails.cfm
	ShowCaseDetails (V)	dsp_negotiationCaseDetails.cfm
	ReadCaseSideInfo (M)	qry_getNegotiationCaseSideInfo.cfm
	ShowCaseSide (V)	dsp_negotiationCaseSide.cfm
	ShowNegotiationCaseForm (V)	dsp_addNewNegotiationCase.cfm
	InsertNewCase (M)	qry_insertNegotiationCase.cfm
	ConfirmInsertNewCase (V)	dsp_confirmAddNegotiationCase.cfm
	NewNegotiationCaseSide (V)	dsp_addNewNegotiationCaseSide.cfm
	InsertCaseSide (M)	qry_insertNegotiationCaseSide.cfm
	ConfirmInsertCaseSide (V)	dsp_confirmAddNegotiationCaseSide.cfm
	ShowNegotiationForm (V)	dsp_hostNegotiation.cfm
	InsertNewNegotiation (M)	act_postNegotiation.cfm qry_insertNegotiation.cfm
	ConfirmInsertNegotiation (V)	dsp_confirmPostNegotiation.cfm
	ReadNegotiationInfo (M)	qry_readNegotiationInfo.cfm
	SelectNegotiator (M)	qry_selectNegotiator.cfm
	SelectCaseSide (M)	qry_selectNegotiationCaseSide.cfm
	SelectProtocol (M)	qry_selectProtocol.cfm
	ShowSetupNegotiation (V)	dsp_setupNegotiation.cfm
	InsertNegotiationInstance	qry_insertNegotiationInstance.cfm
	StartingNegotiation (V)	dsp_startingNegotiation.cfm
	ShowJoinNegotiation (V)	dsp_joinNegotiation.cfm
	JoinNegotiation (M)	qry_joinNegotiation.cfm

	Fuseaction	Fuse
	ShowConfirmJoinNegotiation (V)	dsp_confirmJoinNegotiation.cfm
	ReadPublicCase (M)	qry_readPublicCase.cfm
	ShowPublicCase (V)	dsp_readPublicCase.cfm
	ReadPrivateCase (M)	qry_readPrivateCase.cfm
	ShowPrivateCase (V)	dsp_readPrivateCase.cfm
	ReadIssueList (M)	qry_getNegoIssues.cfm
	IssueList (V)	dsp_listIssues.cfm
	ReadIssueDetails (M)	qry_getIssueDetails.cfm
	IssueDetails (V)	dsp_issueDetails.cfm
	ReadIssueType (M)	qry_getIssueType.cfm
	ReadIssueUnit (M)	qry_getIssueUnit.cfm
	NewIssueProposal (V)	dsp_propIssueForm.cfm
	ProposeIssue (M)	qry_insertIssueProposal.cfm
	ConfirmProposeIssue (V)	dsp_confirmProposeIssue.cfm
	ReadIssueProposalList	qry_getPropIssues.cfm
	IssueProposalList (V)	dsp_listIssueProposals.cfm
	ReadIssueProposalDetails (M)	qry_getIssueProposalDetails.cfm
	IssueProposalDetails (V)	dsp_issueProposalDetails.cfm
	AcceptIssueProposal (M)	qry_acceptIssueProposal.cfm
	ModifyIssue (V)	dsp_modifyIssueForm.cfm
	ModifyIssue (M)	qry_modifyIssue.cfm
	ConfirmModifyIssue (V)	dsp_confirmModifyIssue.cfm
	RejectIssueProposal (M)	qry_rejectIssueProposal.cfm
	NewOption (V)	dsp_addOptionForm.cfm
	AddOption (M)	qry_insertOptions.cfm
	ConfirmAddOption (V)	dsp_confirmAddOption.cfm
	ReadIssueInfo (M)	qry_readIssueInfo.cfm
	RateIssue (V)	dsp_rateIssue.cfm
	ConfirmRateIssue (M)	qry_updateIssueInfo.cfm
	ReadOptionInfo (M)	qry_readIssueOptionInfo.cfm
	RateOption (V)	dsp_rateOption.cfm
	ConfirmRateOption (M)	qry_updateOptionInfo.cfm
	ReadPackageInfo (M)	qry_getPackageInfo.cfm
	RatePackage (V)	dsp_ratePackage.cfm
	ConfirmRatePackage (M)	qry_updatePackageInfo.cfm
	ConfirmRatePackage (V)	dsp_confirmRatePackage.cfm
	ConstructOffer (M)	qry_actConstructOffer.cfm
	ConstructOffer (V)	dsp_constructOffer.cfm
	ProcessOffer (M)	act_insertOffer.cfm
	ConfirmOffer (M)	qry_getOfferInfo.cfm qry_getOfferID.cfm qry_getOfferMessage.cfm qry_getTotalUtility.cfm qry_getPhase.cfm
	ConfirmOffer (V)	dsp_confirmOffer.cfm
	ReadOffer (M)	qry_readLastOfferInfo.cfm
	ReadOffer (V)	dsp_readLastOffer.cfm

	Fuseaction	Fuse
	ConstructMessage (V)	dsp_constructMessage.cfm
	ProcessMessage (M)	qry_insertMessage.cfm
	ConfirmMessage (M)	qry_getPhase.cfm qry_getLastMessage.cfm
	ConfirmMessage (V)	dsp_confirmMessage.cfm
	ReadMessage (M)	qry_readMessageInfo.cfm
	ReadMessage (V)	dsp_readMessage.cfm
	AgreementInspire (M)	qry_agreementInspire.cfm qry_getIssueInfo.cfm qry_getOpponentTotalUtility.cfm qry_getUserOffer.cfm qry_getNumOptions qry_getOpponentOffer.cfm
	ReadAgreementInfo (M)	qry_readAgreementInfo.cfm
	AgreementInspire (V)	dsp_agreementInspire.cfm
	PSReadIssueInfo (M)	qry_PSgetIssueInfo.cfm
	PSConstructOffer (M)	qry_getOpponentTotalUtility.cfm qry_actPSConstructOffer.cfm
	PSConstructOffer (V)	dsp_psConstructOffer.cfm
	PSProcessOffer (M)	act_PSinsertOffer.cfm
	PSConfirmOffer (M)	qry_getOfferInfo.cfm qry_getOfferID.cfm qry_getOfferMessage.cfm qry_getTotalUtility.cfm qry_getPhase.cfm
	PSConfirmOffer (V)	dsp_psConfirmOffer.cfm
	PSReadOffer (M)	qry_psReadLastOfferInfo.cfm
	PSReadOffer (V)	dsp_psReadLastOffer.cfm
	PSConstructMessage (V)	dsp_psConstructMessage.cfm
	PSProcessMessage (M)	qry_psInsertMessage.cfm
	PSConfirmMessage (M)	qry_getPhase.cfm qry_getLastPSMessage.cfm
	PSConfirmMessage (V)	dsp_psConfirmMessage.cfm
	PSReadMessage (M)	qry_psReadMessageInfo.cfm
	PSReadMessage (V)	dsp_psReadMessage.cfm
	PSAgreementInspire (M)	qry_psAgreementInspire.cfm qry_getUserOffer.cfm qry_getOpponentOffer.cfm
	PSReadAgreementInfo	qry_psReadAgreementInfo.cfm
	PSAgreementInspire (V)	dsp_psAgreementInspire.cfm
	HistoryInspire (M)	qry_getHistoryInspire.cfm qry_getUserOffer.cfm qry_getOpponentOffer.cfm
	HistoryInspire (V)	dsp_historyInspire.cfm
	NegoDance (M)	qry_getNegoDance.cfm
	NegoDance (V)	dsp_negoDance.cfm
	Termination (M)	qry_Termination.cfm
	Termination (V)	dsp_Termination.cfm

Appendix D: Screenshots of INSS

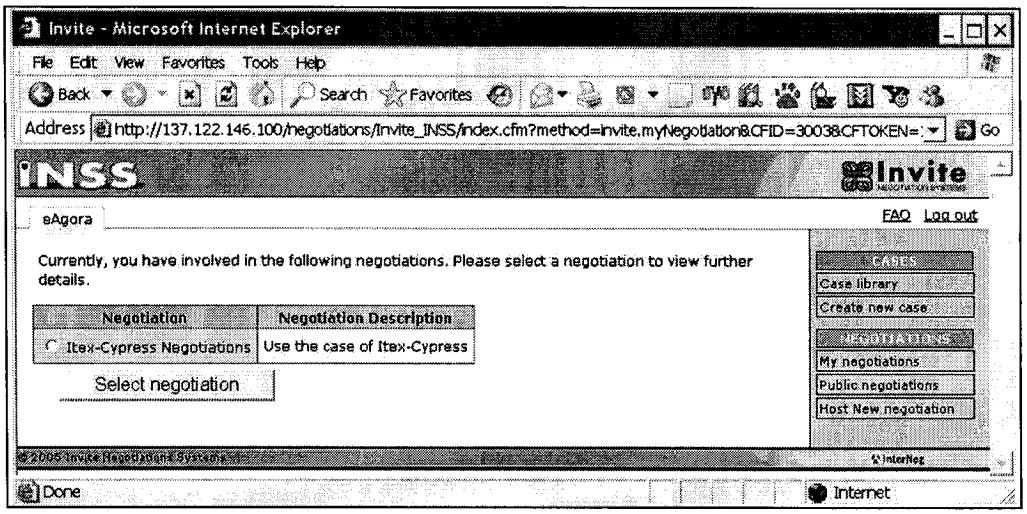

Figure D.1 Current negotiation(s) of one user

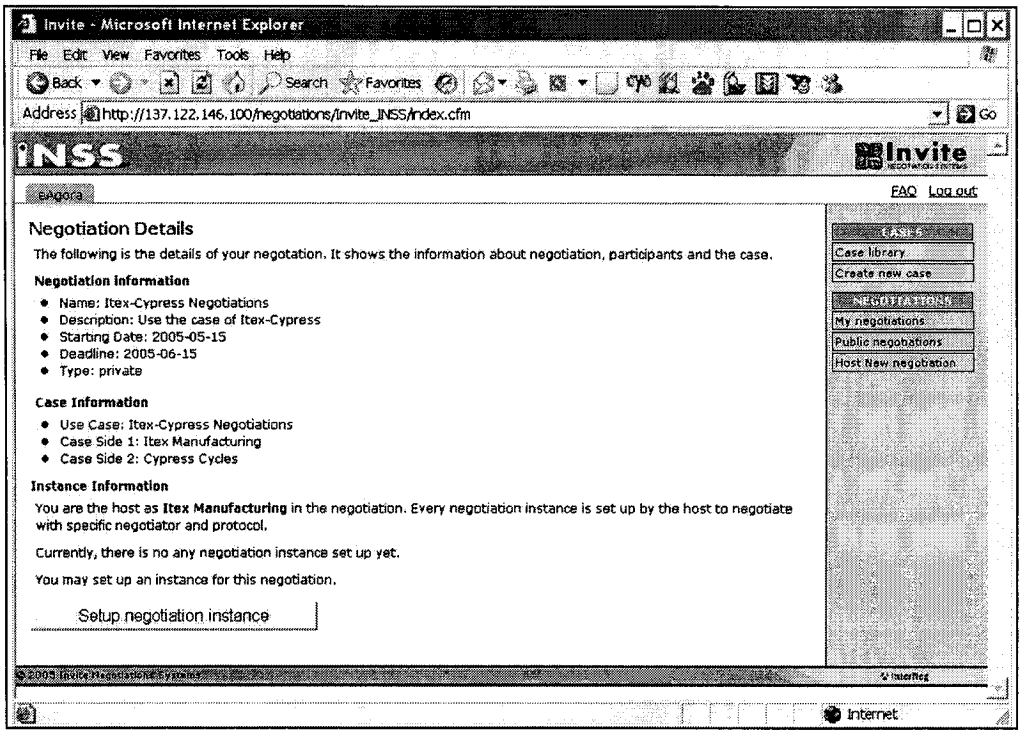

Figure D.2 Negotiation details

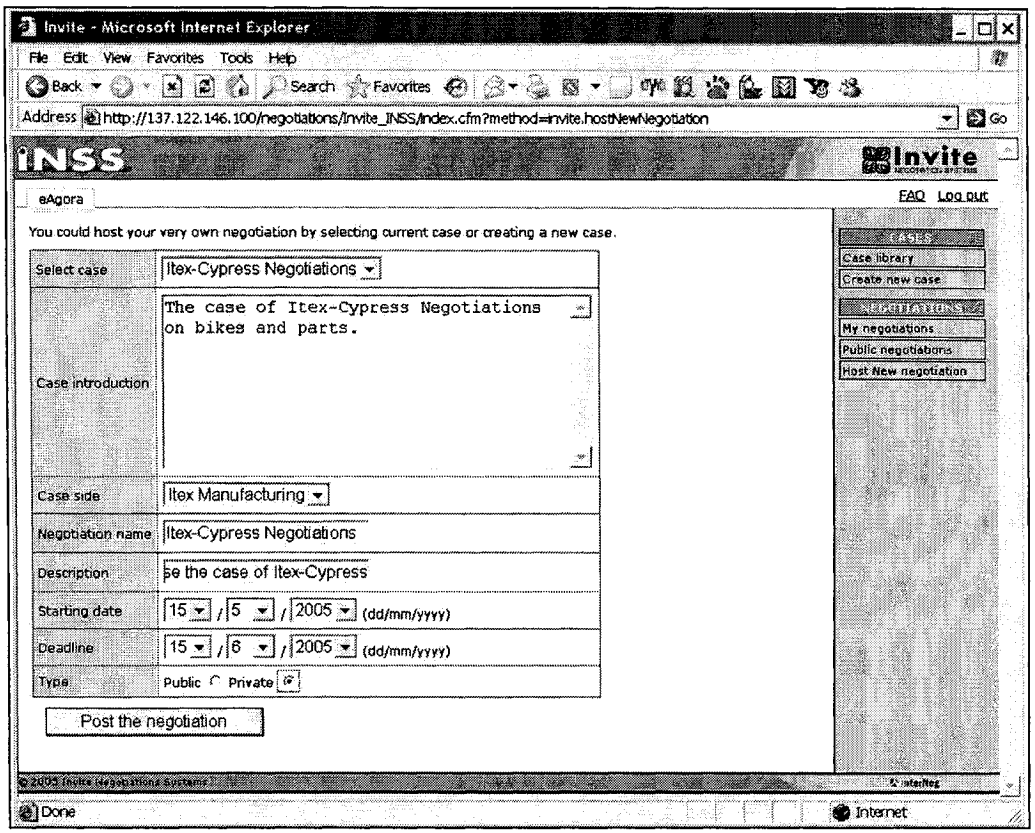

Figure D.3 Hosting a negotiation

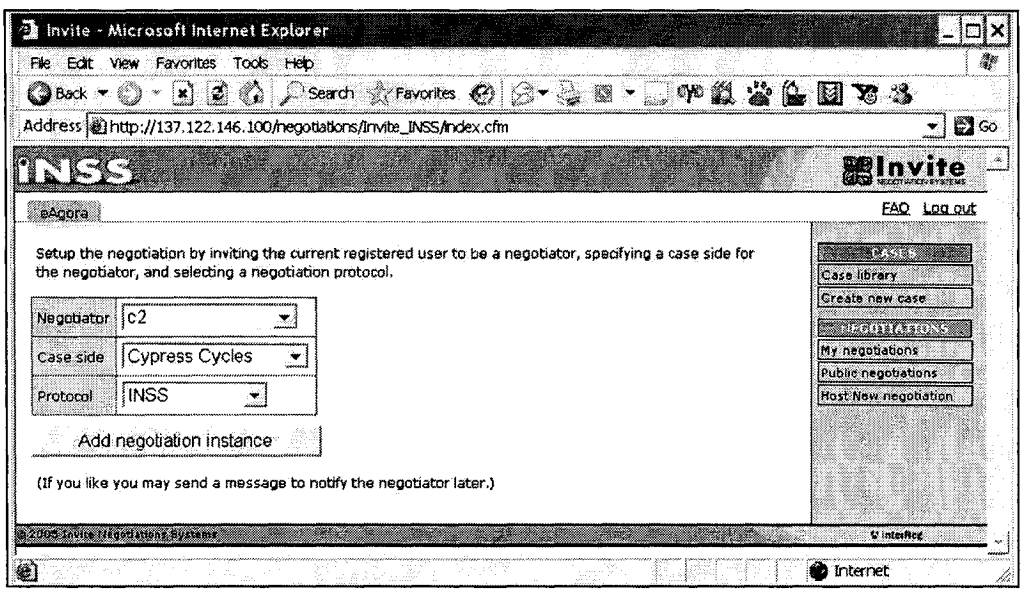

Figure D.4 Setting up a negotiation instance

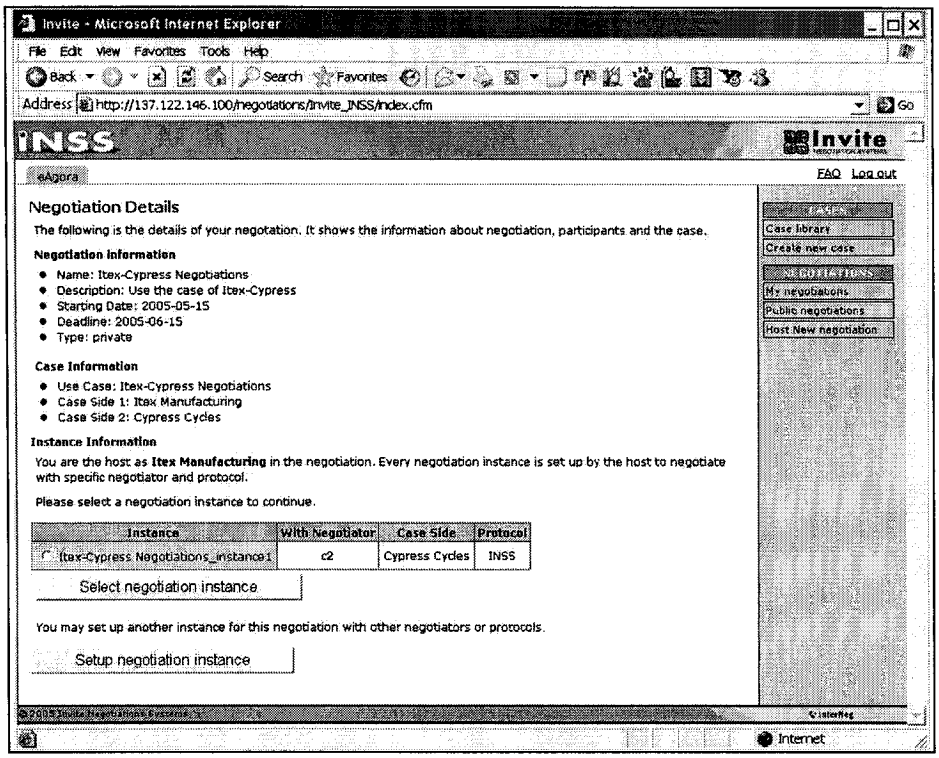

Figure D.5 Selecting a negotiation instance to start

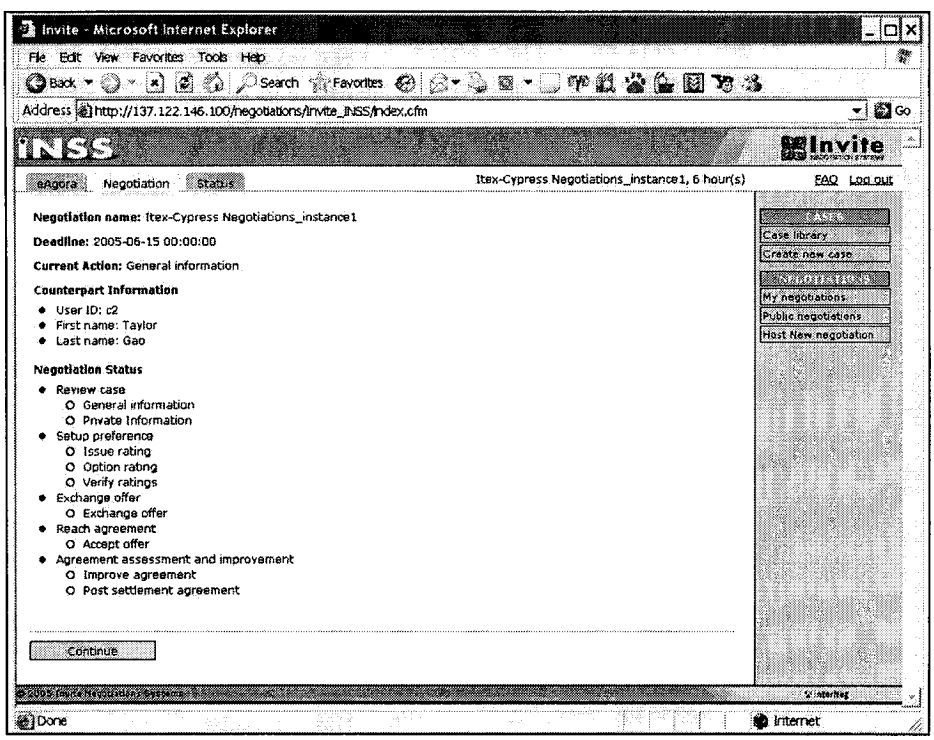

Figure D.6 Starting or continuing negotiation

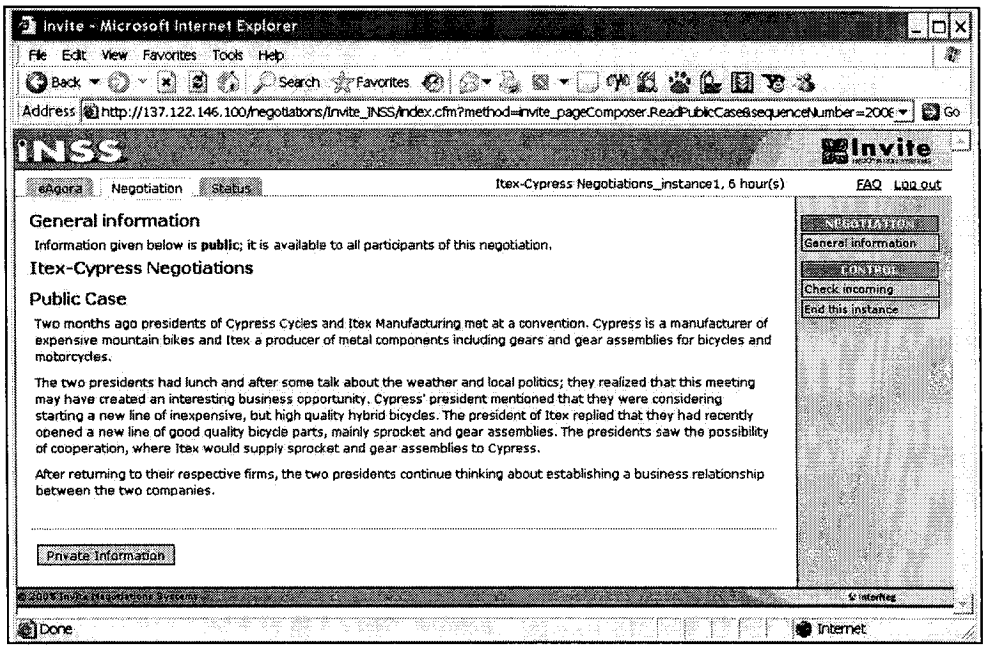

Figure D.7 Reading general information (public case)

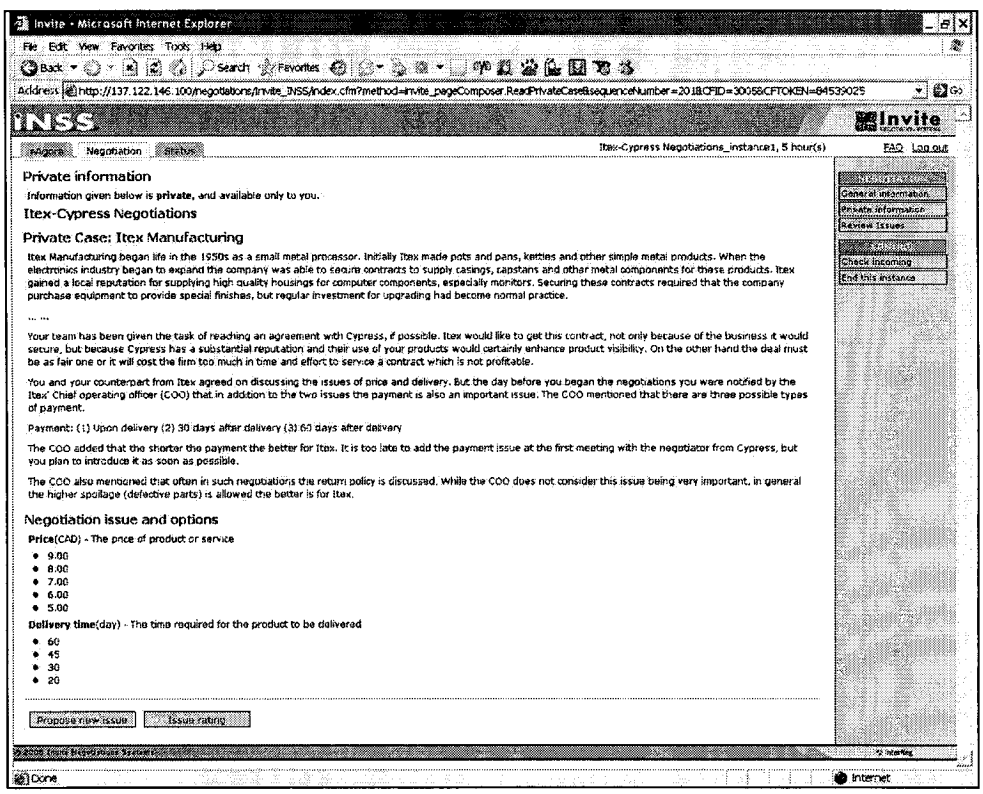

Figure D.8 Reading private information (private case)

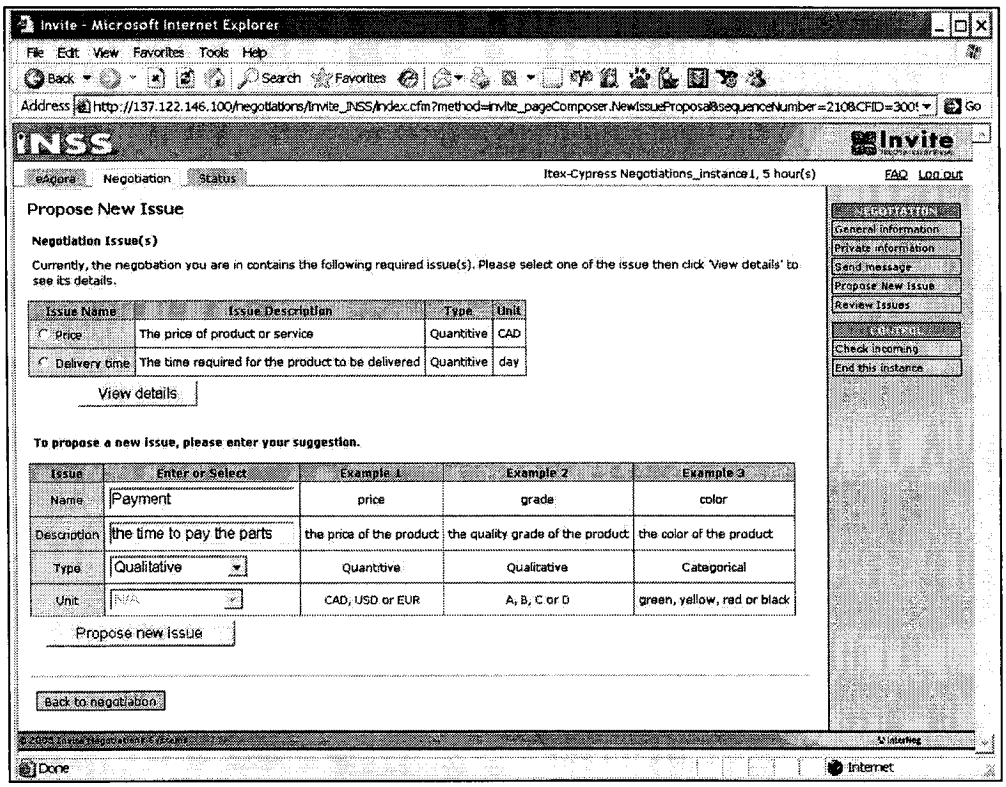

Figure D.9 Proposing a new issue during negotiation

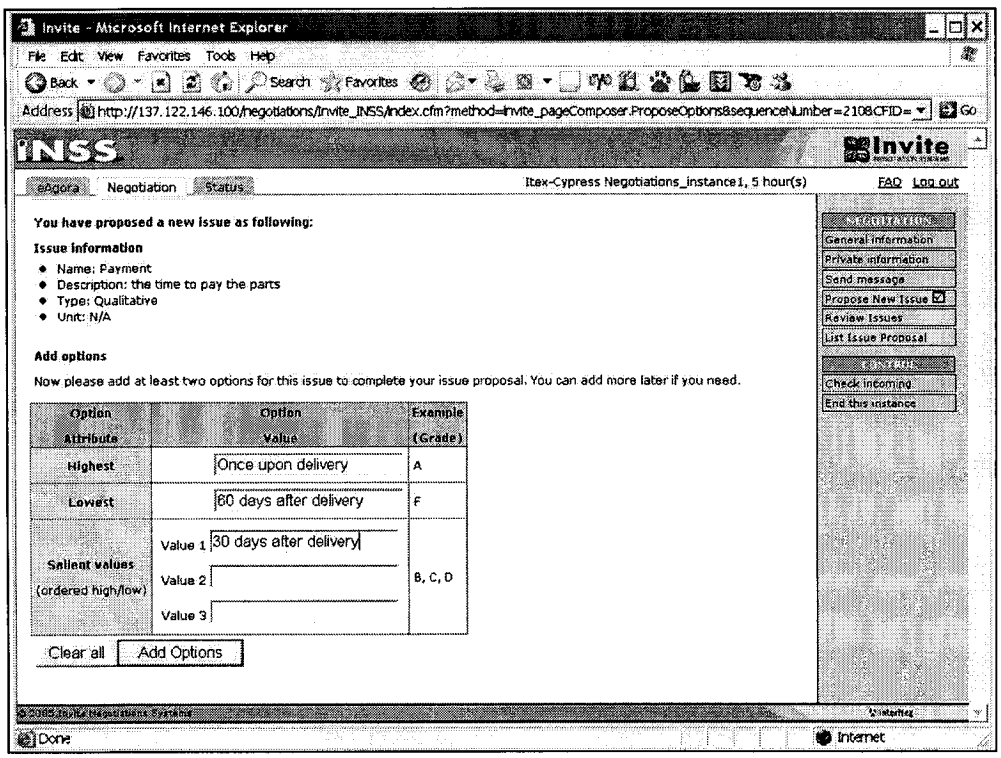

Figure D.10 Adding options to complete an issue proposal

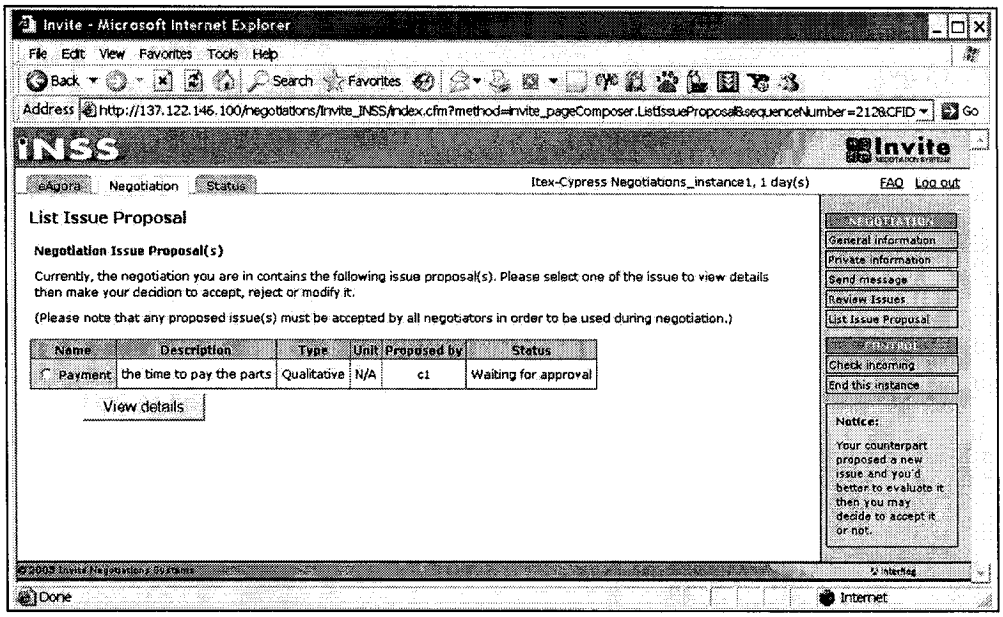

Figure D.11 Receiving counterpart's issue proposal

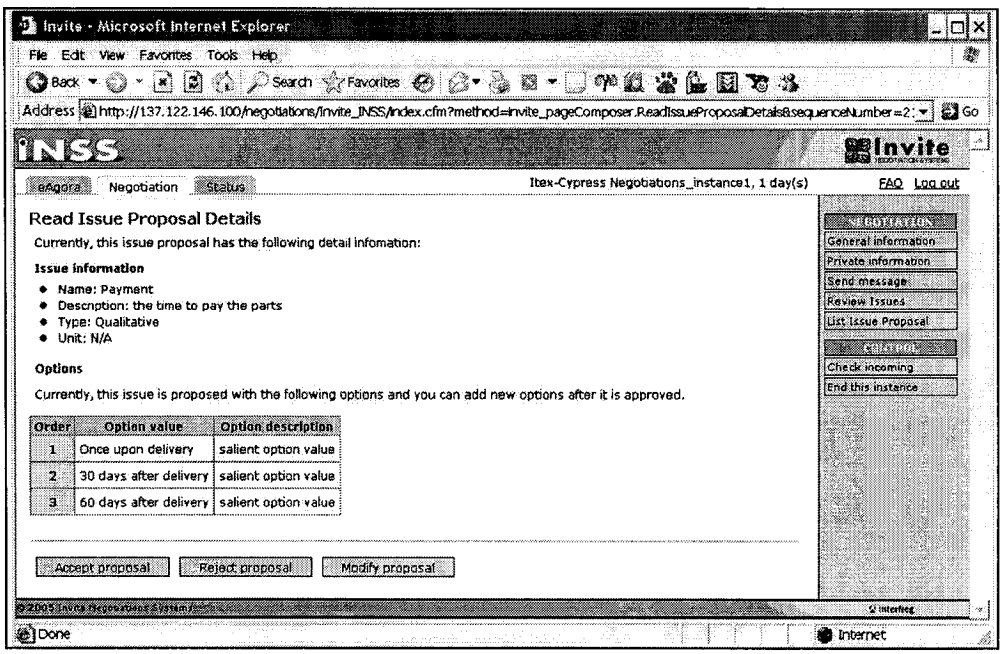

Figure D.12 Evaluating counterpart's issue proposal

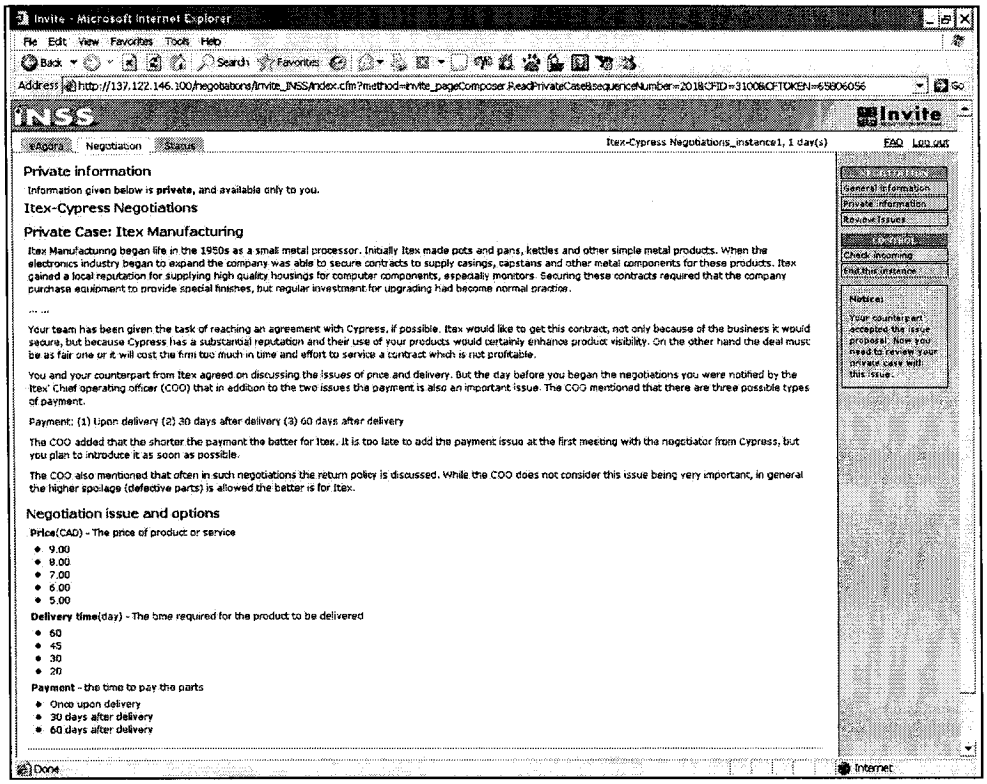

Figure D.13 Accepting counterpart's issue proposal

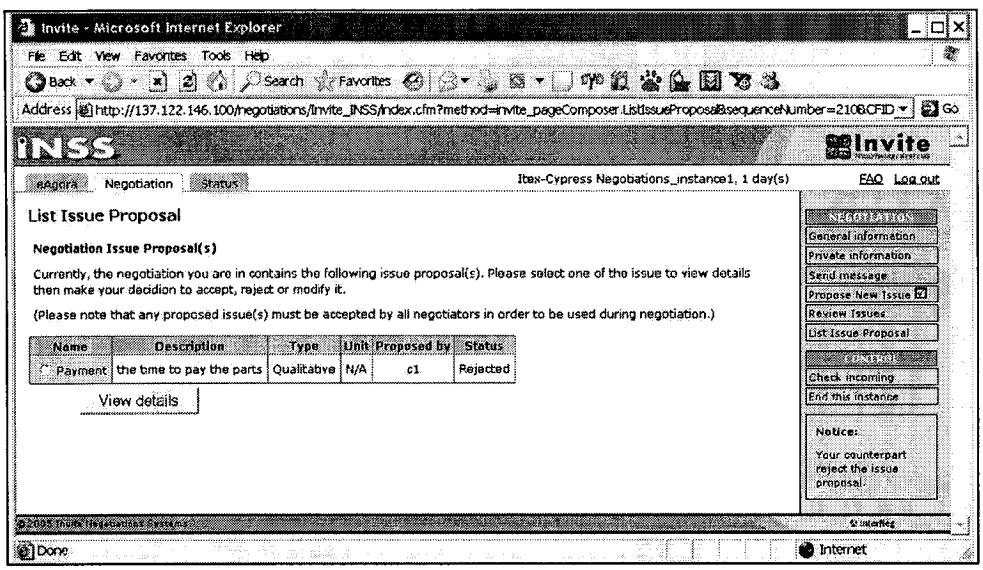

Figure D.14 Rejecting counterpart's issue proposal

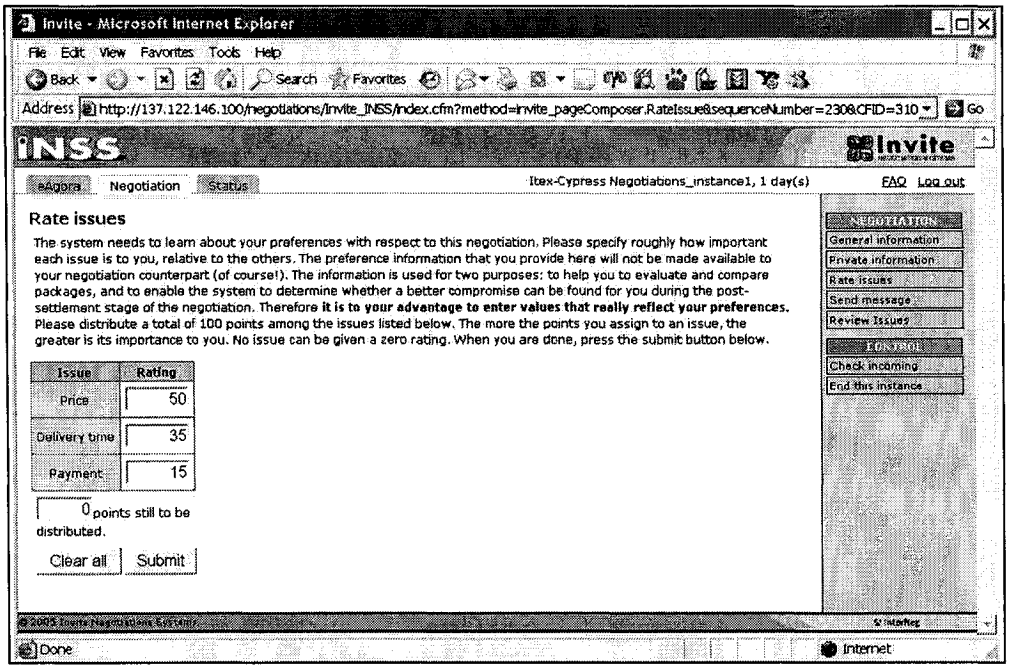

Figure D.15 Rating negotiation issues

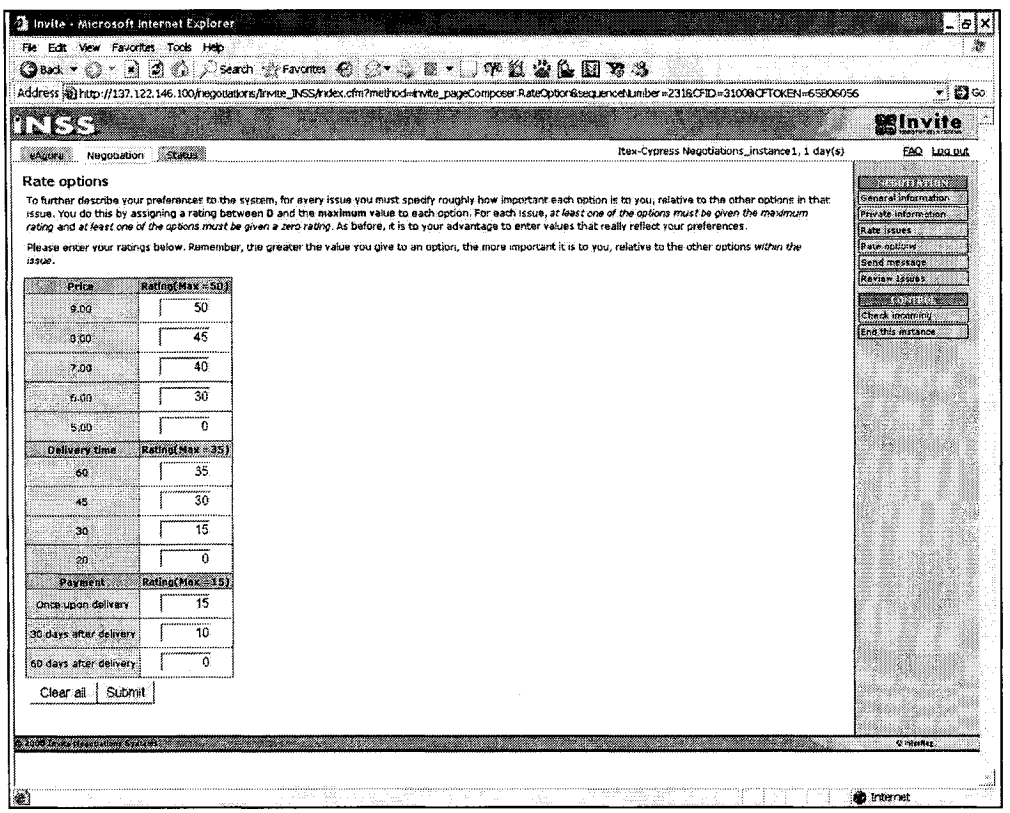

Figure D.16 Rating options

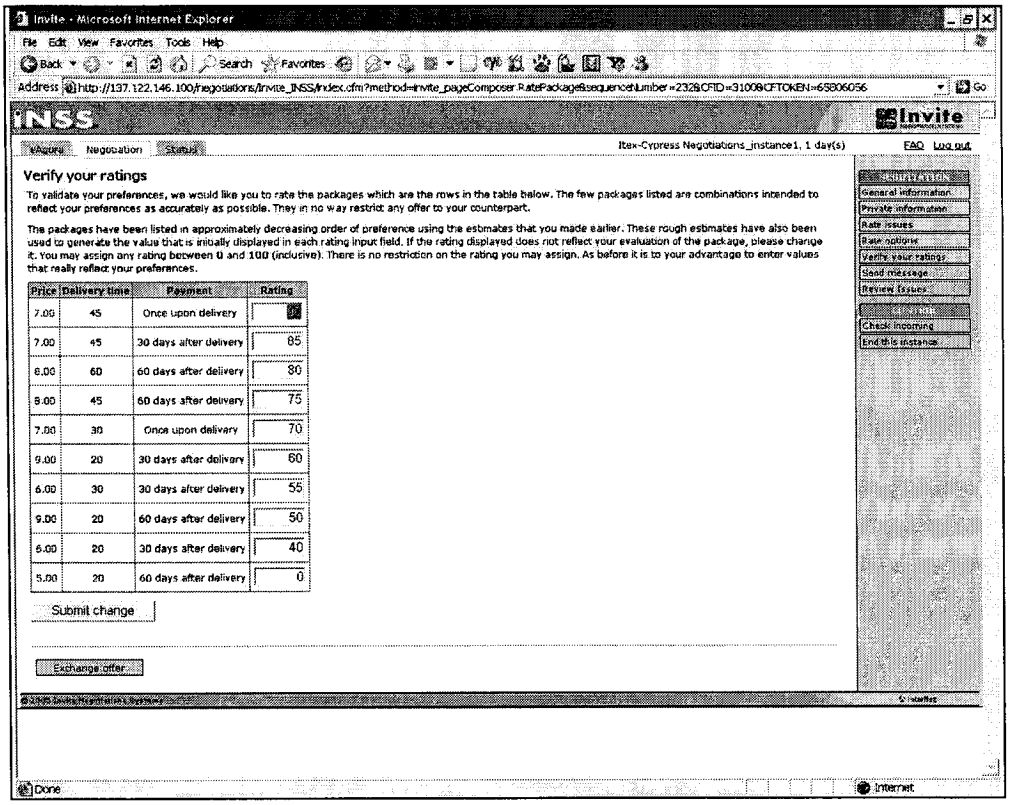

Figure D.17 Verifying ratings (rating packages)

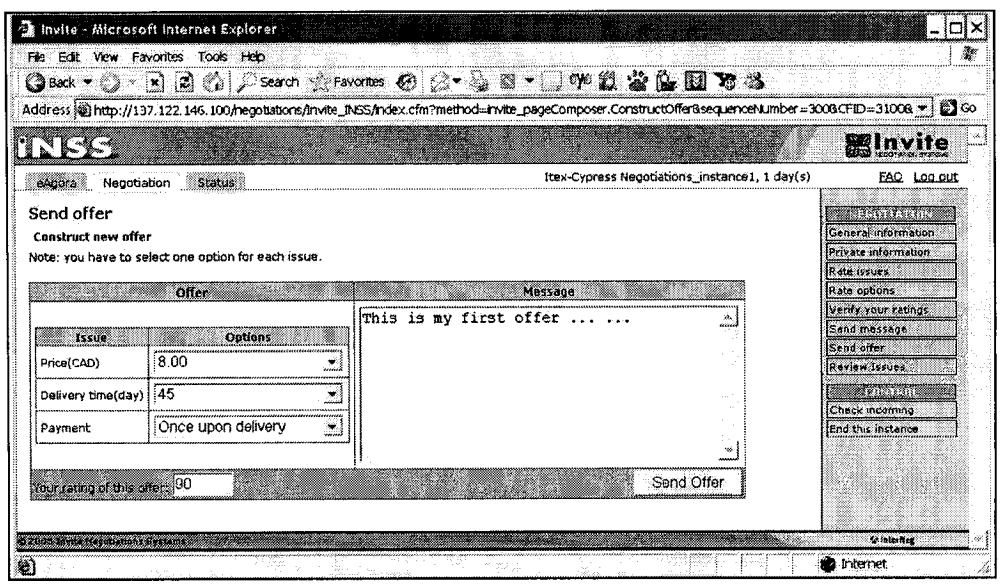

Figure D.18 Constructing an offer (sending offer)

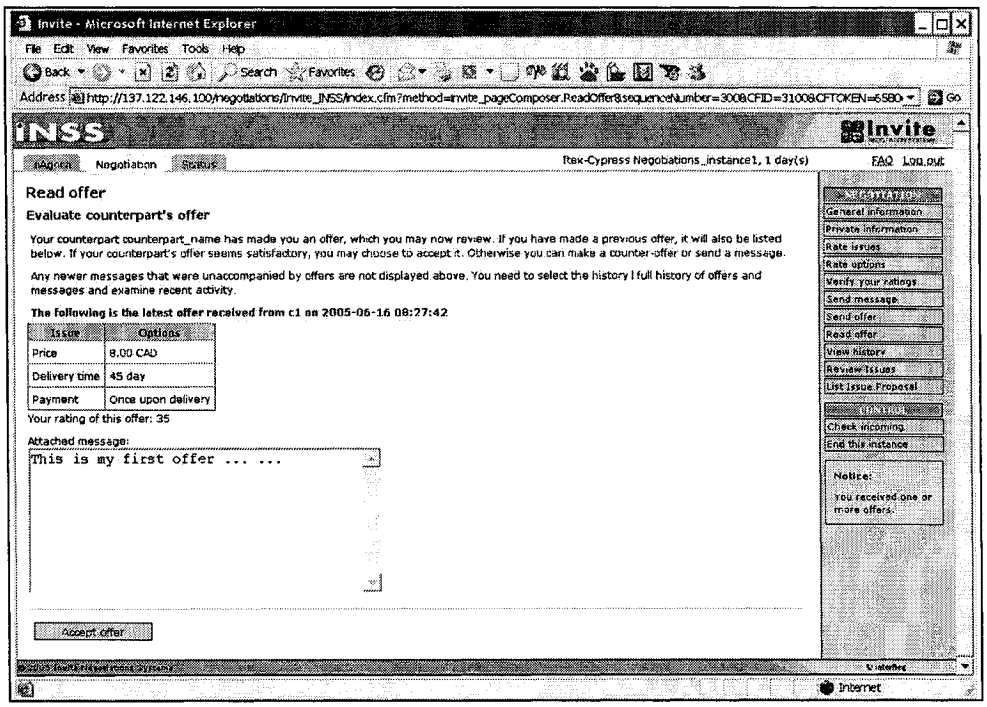

Figure D.19 Evaluating counterpart's offer (Reading offer)

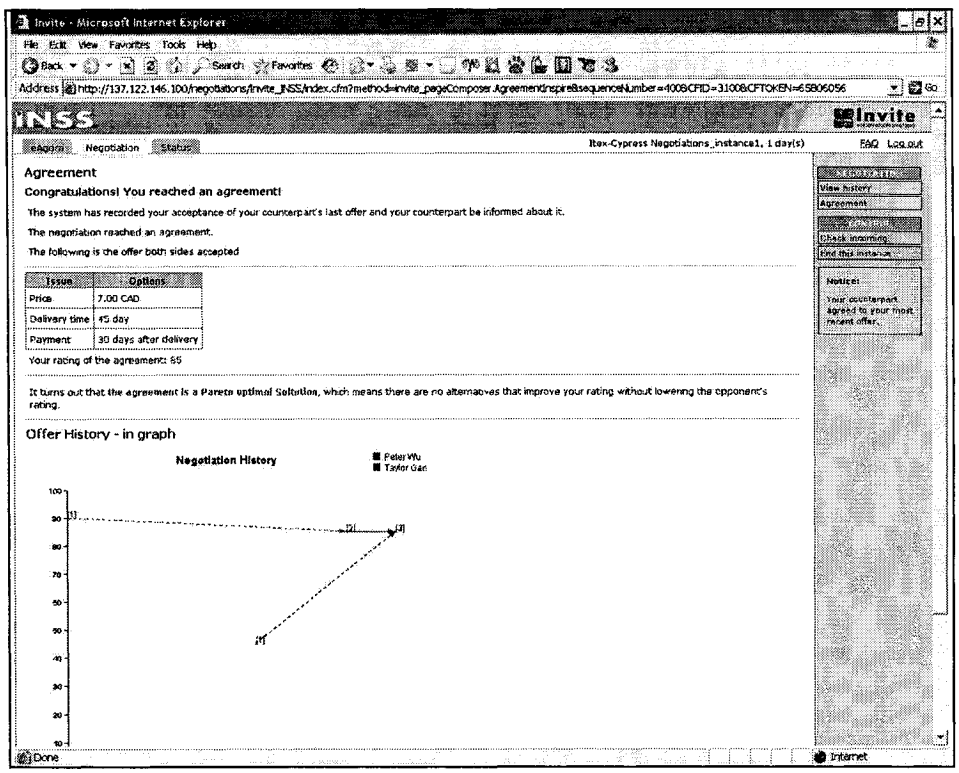

Figure D.20 Reaching an agreement (Accepting offer)

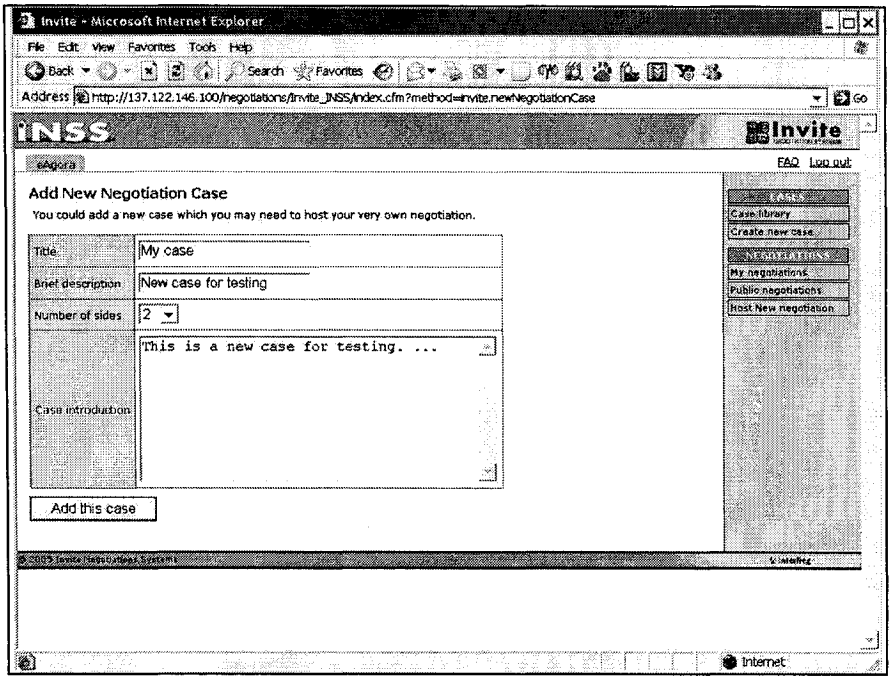

Figure D.21 Creating a new negotiation case

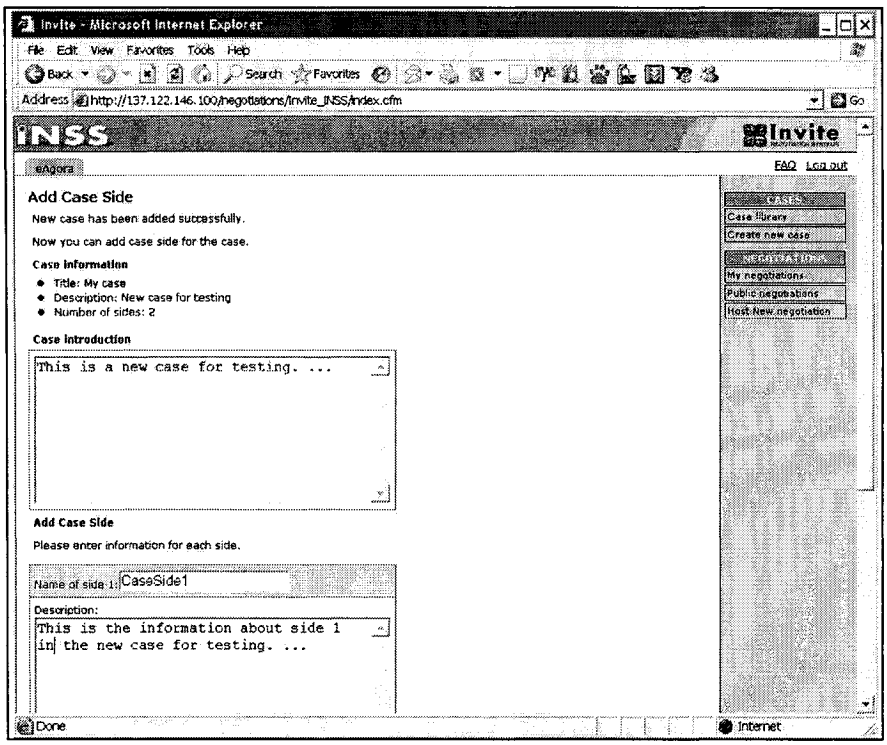

Figure D.22 Adding case sides

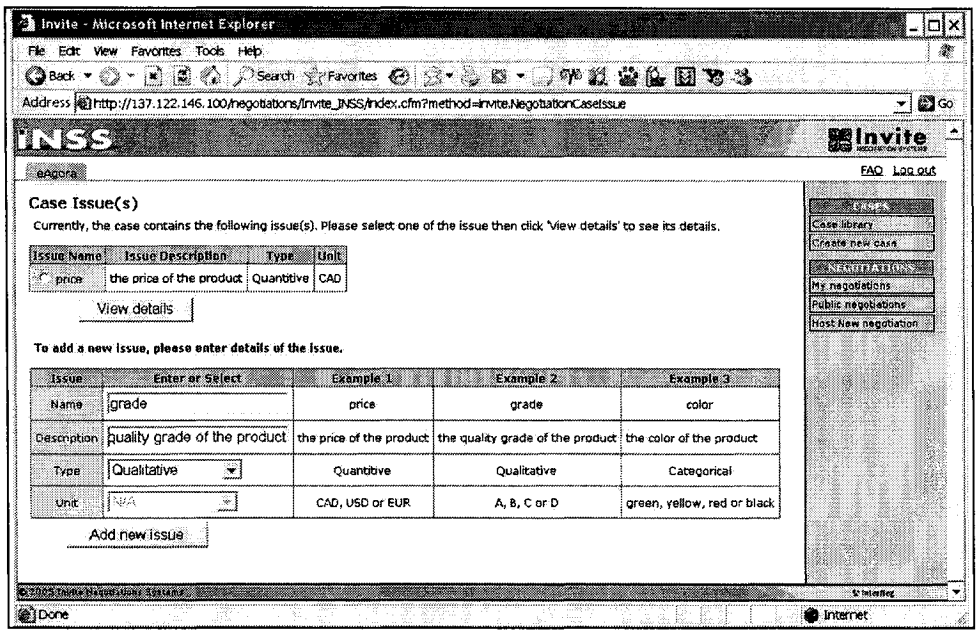

Figure D.23 Adding a new issue to the negotiation case

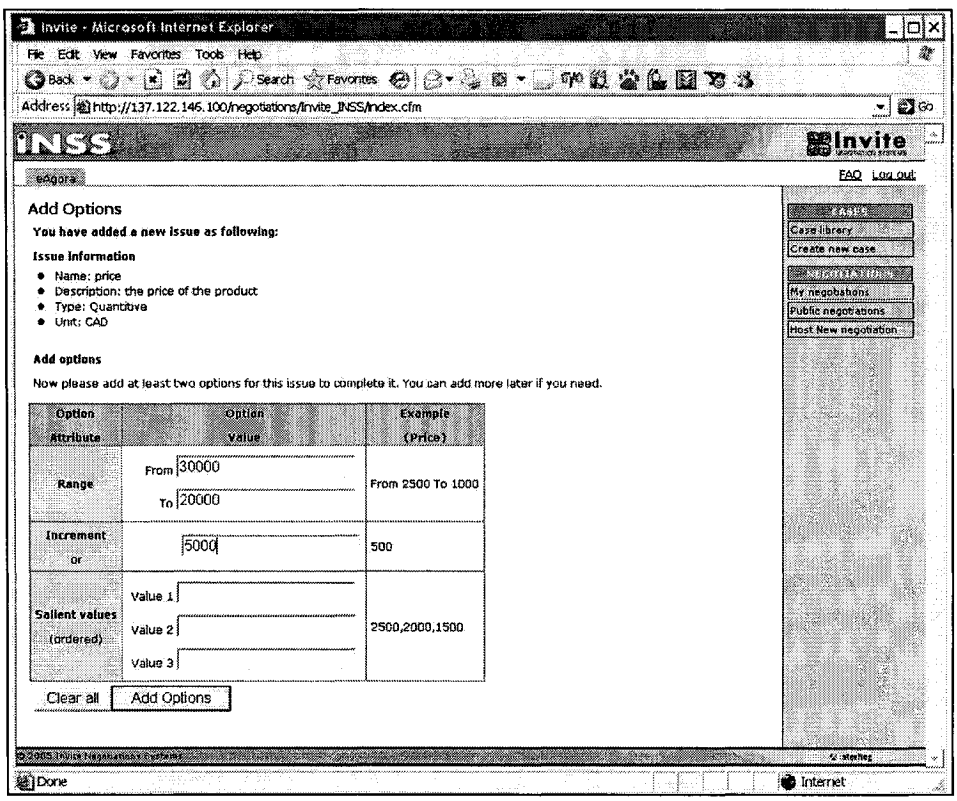

Figure D.24 Adding options to an issue

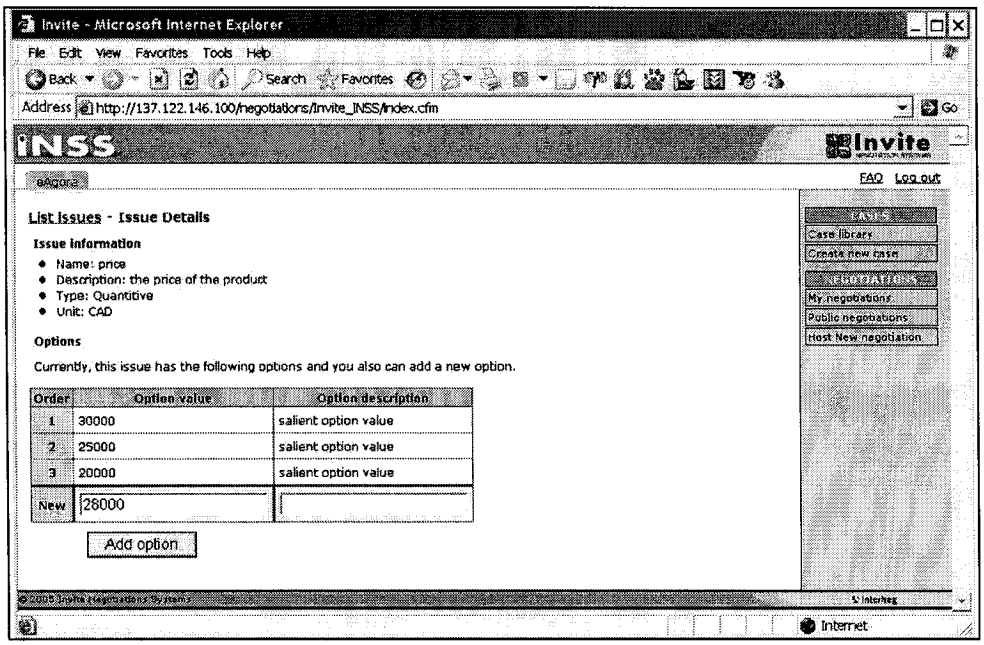

Figure D.25 Adding a new option to current issue (quantitative issue)

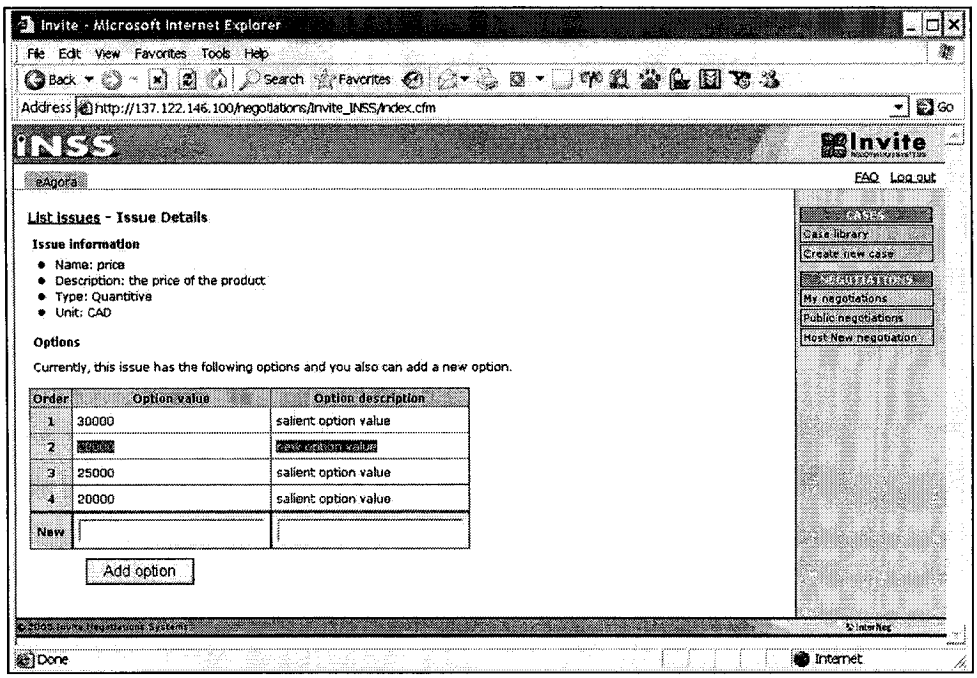

Figure D.26 New option added to current issue (quantitative issue)

Appendix E: Questionnaire for usability test of INSS

Part I: Pre-Test

1. How often do you access the Internet?
   - ❑ Every day
   - ❑ Several times a week
   - ❑ Few times a month
   - ❑ Almost never
2. Have you tried to buy/sell any thing via the Internet?
   - ❑ Yes
   - ❑ No
3. How would you rate your experience at negotiating?
   - ❑ Very experienced
   - ❑ Experienced
   - ❑ Beginner
   - ❑ No experience
4. Have you used a negotiation support system ever before?
   - ❑ Yes
   - ❑ No

Part II: Post-Test (for adding new issues and options)

1. Would you use INSS for (or to prepare) your negotiations?
   - ❑ Yes
   - ❑ No
2. Do you think that INSS is easy to use?
   - ❑ Yes
   - ❑ No
3. Have you proposed/added new issue or options during your negotiation?
   - ❑ Yes
   - ❑ No

   If yes, how many issues? _____ and how many options?_____
4. Did you reach an agreement with your counterpart?
   - ❑ Yes
   - ❑ No

   If not, why? _____

   _____
5. Do you think it is helpful to allow adding new issue and options during negotiation?
   - ❑ Yes
   - ❑ No
6. How would you rate your INSS experience overall?
   - ❑ Excellent
   - ❑ Great
   - ❑ Very good
   - ❑ Good
   - ❑ Acceptable

- ❑ Poor

7. What did you like about INSS? _____
   _____
8. What did you dislike about INSS? _____
   _____

Part III: Post-Test (for hosting negotiations and instances)

1. Would you use INSS for (or to prepare) your negotiations?
   - ❑ Yes
   - ❑ No
2. Do you think that INSS is easy to use?
   - ❑ Yes
   - ❑ No
3. How did you host your negotiation?
   - ❑ By using existing case
   - ❑ By creating new case
4. How many negotiation instances did you conduct?
   - ❑ 1
   - ❑ 2
   - ❑ 3 or more
5. Did you reach an agreement with any negotiator?
   - ❑ Yes
   - ❑ No

   If not, why? _____
   _____
6. Do you think it is helpful to improve your negotiation outcome by conducting more instances?
   - ❑ Yes
   - ❑ No
7. How would you rate your INSS experience overall?
   - ❑ Excellent
   - ❑ Great
   - ❑ Very good
   - ❑ Good
   - ❑ Acceptable
   - ❑ Poor
8. What did you like about INSS? _____
   _____
9. What did you dislike about INSS? _____
   _____

Appendix F: Results for usability Test

Table F.1 Results of Pre-test Questions

Part I: Pre-Test Questions	
1. Percentage of participants that access the Internet	
▪ Every day ▪ Several times a week	86 % 14 %
2. Percentage of participants that have tried to buy/sell anything via the Internet	86 %
3. Percentage of participants that have experience at negotiating	73 %
4. Percentage of participants that have used a negotiation support system	21 %

Table F.2 Results of Post-test Questions (Scenario 1)

Part II: Post-Test Questions (for adding new issues and options)	
1. Percentage of participants that would use INSS to conduct or prepare negotiations	93 %
2. Percentage of participants that claimed INSS is easy to use	89 %
3. Percentage of participants that proposed/added new issue/options during negotiation	100 %
▪ Average number of issues that participants added during negotiation ▪ Average number of options that participants added during negotiation	1.8 6.1
4. Percentage of participants that have reached an agreement using INSS (Participants that did not reach an agreement cited the reason to be: hard to compromise between issues or time limitation)	72 %
5. Percentage of participants that claimed it is helpful to allow adding new issue/options during negotiation	93 %
6. Average rating for participants' satisfaction with INSS	Very good
7. INSS's most cited advantages:	
▪ Easy to add new issue/options during negotiation ▪ Can calculate new option's rating value for quantitative issue	
8. INSS's most cited disadvantages:	
▪ Takes more time to negotiate with more issues ▪ Does not have enough instructions for some pages ▪ E-mail notification even for online negotiators	

Table F.3 Results of Post-test Questions (Scenario 2)

Part III: Post-Test Questions (for hosting negotiations and instances)	
1. Percentage of participants that would use INSS to conduct or prepare negotiations	93 %
2. Percentage of participants that claimed INSS is easy to use	89 %
3. The way to host negotiation:	
▪ Percentage of participants that hosted negotiation by using existing case ▪ Percentage of participants that hosted negotiation by creating new case	83 % 31 %
4. Average number of negotiation instances that participants conducted	1.5
5. Percentage of participants that have reached an agreement using INSS (Participants that did not reach an agreement cited the reason to be: hard to compromise between issues or time limitation)	87 %
6. Percentage of participants that claimed conducting more instances is helpful to improve their negotiation outcome	93 %
7. Average rating for participants' satisfaction with INSS	Good
8. INSS's most cited advantages:	
▪ Easy to host negotiations to negotiate with many people ▪ Can create their very own case for negotiation	
9. INSS's most cited disadvantages:	
▪ Does not have enough instructions for some pages ▪ Need to change negotiation instance to view its status	

Appendix G: List of hardwired components for INSS

Table G.1 List of hardwired components for INSS

Component	Note
qry_getCurrentSequenceState.cfm	For proposing issue (outgoing intervening) as well as updating additional session variables
qry_updateNegotiationStatus.cfm	
act_processExitPoint.cfm	
act_validateUser.cfm	
qry_getNegotiationInstance.cfm	Need to check negotiation_id too
dsp_selectNegotiationInstance.cfm	Within eAgora part for selecting one instance of the same negotiation
qry_getOpponentNegoInfo.cfm	Get negotiation (instance) info by instance_id but not the instance_name (not only for INSS, but also more unique)
qry_getUserNegoInfo.cfm	
act_setNegotiationInstance.cfm	With more session variables, such as previousSequenceNumber, previousStateNumber, etc
dsp_linkConstructionTop0.cfm	For pages related to eAgora part, right01 for negotiation cases and right02 for negotiations (Currently have be done independently, but need to be specified in layoutdefault.cfm)
dsp_linkConstructionRight01.cfm	
dsp_linkConstructionRight02.cfm	

Appendix H: Research plan

Table H.1 Research plan

Activities	Start Date	End Date	Notes
Basic theory and concepts	May 1st, 2003	Aug 31st, 2003	Negotiation, auctions, DSS, E-commerce, E-negotiation, ENS, E-marketplace, etc.
Research on development tools (technologies)	June 1st, 2003	Dec 31st, 2003	CF, MySQL, DecisionScript, Fusebox
Literature review	Aug 7th, 2003	Nov 28th, 2003	Thesis proposal
Review and migration of eAgora system	Nov 15th, 2003	Dec 31st, 2003	On sever: 137.122.146.100
General system requirements	Jan 1st, 2004	Mar 31st, 2004	Old version of INSS; Invite platform; Negotiation protocol
Design INSS protocol	Mar 1st, 2004	July 31st, 2004	Flowchart
Design INSS database	Apr 1st, 2004	July 31st, 2004	
Design and implementation of components for new features	Apr 15th, 2004	July 31st, 2004	Based on eAgora
Implementation	May 1st, 2004	July 31st, 2004	Invite-INSS demo (without protocol)
Design and implementation of components for Invite	June 1st, 2004	Sep 15th, 2004	Page composers for Inspire
Refine page composers for Invite	Sep 1st, 2004	Dec 15th, 2004	
Design and refine INSS protocol	Sep 1st, 2004	Dec 31st, 2004	Initial sequence/states tables
Deployment of components in Invite	Sep 1st, 2004	Nov 30th, 2004	Invite-INSS demo (with protocol)
Implementation	Sep 1st, 2004	Dec 31st, 2004	
Design and refine INSS protocol	Jan 1st, 2005	Feb 15th, 2005	Intervening rules table
Implementation	Feb 1st, 2005	Mar 31st, 2005	
Usability test	Apr 1st, 2005	Jun 15th, 2005	
Writing	Oct 1st, 2004	Jun 30th, 2005	Thesis