A Software Platform for Multiprotocol E-Negotiations[1]

Gregory E. Kersten[1, 3], Ka Pong Law[2], Stefan E. Strecker[3]

School of Management, University of Ottawa
Faculty of Engineering and Computer Science, Concordia University, Montreal
John Molson School of Business, Concordia University, Montreal

Abstract

A flexible and highly customizable software platform to conduct research on e-negotiation is presented. 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. The system design is discussed and selected implementation details are presented to give an overview of the approach taken.

[1] This research was funded by grants from the Social Sciences and Humanities Research Council Canada and Natural Sciences and Engineering Research Council Canada. The authors are grateful to Nematollaah Shiri for his comments and suggestions.

1. Introduction

The use of software to support negotiation processes was put forward in the late 1970s. Keen (1978), Sprague and Carlson (1982), and others proposed to extend decision support system (DSS) capabilities to aid the negotiators. This led in the early eighties to the design of negotiation support systems (NSSs) and group support systems (GDSs) (Jarke et al. 1987; Jelassi et al. 1989; Korhonen et al. 1986).

NSSs are based on the modelling approaches formulated in decision sciences and negotiation analysis (Raiffa et al. 2003). Other approaches stemming from computer science, especially artificial intelligence, have also been used in the design of software aiding one or more negotiators (Kersten 1993; Matwin et al. 1987; Rangaswamy et al. 1989).

Until recently, NSSs and DSSs designed to support one or more negotiators were used mostly in education and research (Foroughi 1995; Jones et al. 1990; Sainfort et al. 1990). Two notable exceptions are the simulation system used in the United Nations negotiations on the Convention on the Law of the Sea (Nyhart et al. 1987) and the RAINS system designed to support negotiations on air pollution reduction in Europe (Hordijk 1991).

Internet and new software development technologies created new opportunities for the design and deployment of systems capable of supporting negotiators. Internet-based systems differ from other information systems in several key aspects. Internet provides close to ubiquitous connectivity and allows a large number of people to access systems from virtually any place. Internet-based systems also allow for tight integration of intra-enterprise business processes (e.g., supply chain management). Their user interface is provided by a client application, e.g. a web browser, which is easy to understand and common to many systems.

Decision and negotiation support systems deployed on Internet are unlike earlier systems deployed on stand-alone computers or local- and even wide-area networks in terms of the implemented solutions and employed technologies. Some systems facilitate negotiation of the documents' content and their joint preparation, e.g., contract negotiations (Schoop et al. 2001), others use email, chat and streaming video software (Lempereur 2004; Moore et al. 1999). There are also systems that allow the negotiators to enter offers, which subsequently are sent to human experts who suggest agreements (e.g., http://www.electroniccourthouse.com). The term e-negotiation systems (ENS) has been used to describe software that employs Internet technologies, is deployed on the web, and capable of supporting, aiding or replacing one or more negotiators, mediators or facilitators (Ehtamo et al. 2004).

Most of the existing ENSs implement only one negotiation protocol (Ströbel 2001), with a notable exception, the SilkRoad platform designed to support various auction protocols (Ströbel 2003). Except for SilkRoad, we know of 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 ENS's use and adoption, 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. The Invite system discussed in this paper provides a flexible and highly customizable environment to conduct e-negotiation experiments and field studies.

The purpose of this paper is to introduce an approach for the design of web-based systems that allow for customizable researcher- or user-definable protocols. While we concentrate on ENSs, the proposed approach can be extended to other systems, including decision and group support systems. The remainder of this contribution is organized as follows. Section 2 briefly reviews the existing ENSs. Section 3 introduces the negotiation theory underlying Invite and Section 4 gives an overview of the Invite system. Section 5 illustrates the system architecture and discusses the design considerations. The final section concludes the paper.

2. Literature and systems review

A number of ENSs have been designed, implemented and applied to various negotiation problems. Several of these systems are currently investigated within the SSHRC project on e-negotiation (http://interneg.org/enegotiation). In the following, we briefly review selected ENSs. For lack of space, we do not discuss ENS, which require the installation of a special client, such as SmartSettle (http://smartsettle.com), hybrid systems, which combine auction and negotiation elements, such as NegotiAuction (Teich et al. 2001) or, negotiation and workflow elements, such as GNP (Benyoucef et al. 2000). However, a thorough overview of ENSs can be found in Shim (1999) and Neumann et al. (2003).

Inspire is a bilateral ENS developed by the InterNeg group based on decision and negotiation analysis (http://interneg.org/inspire). Its main purpose is to investigate cross-cultural negotiations and to teach negotiation courses (Kersten et al. 1998). Inspire views a negotiation as a process occurring in a particular context. The system uses a 3-stage process model: pre-negotiation analysis, conduct of negotiation, and post-settlement analysis. The pre-negotiation phase involves an analysis of the situation, problem and opponent, specification of preferences, reservation levels, and strategy. The negotiation phase involves exchanges of messages and offers, evaluation of offers and the assessment of the progress of the negotiation. The post-settlement phase involves the evaluation of the negotiation outcomes generated by, and after, the negotiation activity (InterNeg 1996).

WebNS is a Java-based ENS, which focuses purely on the conduct of negotiation and does not offer any analytical negotiation support. It is based on a negotiation process model derived from Gulliver (1979) and divides the negotiation process into two main phases: preparation and offer exchange. Preparation is supported by tools such as a session description and private notes. The main support of WebNS is in the conduct of negotiations. The system uses real-time chat and video conferencing to exchange offers and counter-offers as well as short messages. The protocol underlying WebNS treats every issue separately and, hence, does not explicitly support the discussion of tradeoffs among issues (Yuan et al. 1999).

Negoisst is an ENS developed by the Electronic Negotiation Group at RWTH Aachen. Its primary purpose is to enable complex dynamic negotiations between human negotiators, to guide them through the complex negotiation process, and to provide support rather than automation of negotiations. Negoisst takes a document-oriented as opposed to a communication-oriented approach towards negotiation support and uses speech act theory to model communication between negotiators. The system architecture is a Java-based 3-tier client/server architecture based on the DOC.COM framework (Schoop et al. 2001). The negotiation protocol used in Negoisst distinguishes between different types of messages, e.g. a question or clarification. The protocol is based on a deterministic finite automaton with symmetric states and constraints on some states (Schoop et al. 2003).

SilkRoad is the platform for designing and implementing electronic negotiations (Ströbel 2003). Its design methodology is based on reusable negotiation support components, e.g. for matching offers and mediating compromises. To set up a specific ENS, a negotiation designer uses SilkRoad to define the negotiation (case, issues and options etc.), and to define the protocol. On the basis of a complete negotiation design, run-time specifications are generated. These specifications are then deployed to a server running the SilkRoad run-time environment. Through the activation of these run-time specifications, the server is now customised to support an electronic negotiation among agents in exactly the way specified by the negotiation designer.

SimpleNS has been developed for teaching and comparative studies on the use and effectiveness of different ENSs (http://mis.concordia.ca/simplens). It provides a virtual negotiation table allowing its users to exchange offers and messages. This system displays the negotiation case and other information required to conduct the negotiation, presents a form in which users write messages and offers, and shows the negotiation history in which all messages and offers are displayed in one table together with the time when they were made. SimpleNS is based on a three-stage process model similar to Inspire. However, it does not offer analytical support to the negotiators. SimpleNS has been used in teaching at the University of Ottawa, Concordia University, Vienna University, Austria and National Sun-yat Sen University, Taiwan.

This brief overview illustrates the breadth of approaches towards ENSs design. Some systems are grounded in negotiation theory, some draw on one or more models from negotiation research, and some do not use any methodological foundation at all. However, the most ENSs make very limited use of the existing theory and are restricted to a single negotiation protocol. Hence, the user can neither adapt the ENS to her needs nor follow a consistent methodology.

3. Negotiation activities

3.1 Negotiation process and activities

The use of software in negotiations requires that a *process model* and a *protocol* be constructed (Kersten et al. 2003; Kim et al. 2003). The process model describes negotiation phases and assigns different activities to them. Its significance is in that it allows the negotiators to follow a methodologically sound approach (Lewicki et al. 1999). The protocol is a formal model, often represented by a set of rules, which govern software processing and communication tasks, and imposes restrictions on activities through the specification of permissible inputs (Jennings et al. 2001).

There have been no behavioural studies on e-negotiations and, therefore, no process models specific to e-negotiation have been developed. For the purpose of Invite we use a five-phase model based on Gulliver's eight-phase model (1979). This model, presented in Figure 1, allows for the consideration of a wide range of negotiated decisions, including those which use ENSs.

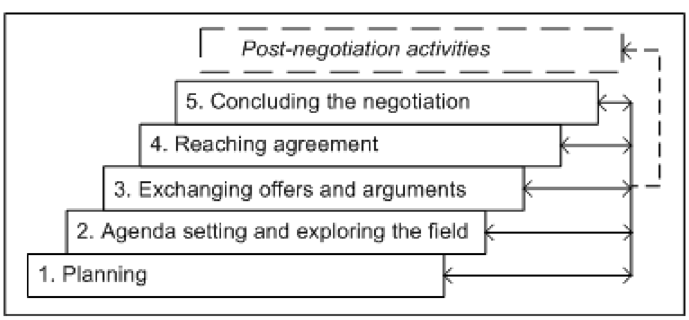

Figure 1. Negotiation process model

Negotiation phase model provides a structure to the negotiation process. This is not to say that the negotiator who conducts activities belonging to one phase cannot return to another. This possibility of revisiting previous phases and then returning to the current phase is indicated in Figure 1. The possibility of ignoring some phases is also shown in Figure 1. This practice, while it occurs in real-life negotiation, should not be allowed in ENS-supported negotiations. Instead, the process model and protocol underlying an ENS should be based on a sound methodology and carried out, among others, by the sequencing imposed by the phase model.

1. Planning comprises activities that each negotiator undertakes both individually and jointly. They formulate their representation of the negotiation problem including the specification of issues and options. In this phase the negotiators specify their objectives and preferences, and such negotiation-specific constructs as BATNA and reservation levels (Fisher et al. 1991). If the negotiators know or can learn about their opponents, they decide on strategies to be used. This phase's joint activity is the selection of the negotiation location and time, and the communication modes the negotiators will use.

2. Agenda setting and exploring the field includes the negotiators' discussion about the negotiated issues and their meaning. The discussion's result may be that new issues and options are added or some are deleted. The negotiators may also discuss the protocol they will follow, the timing of the exchanges, the deadline and—in some negotiations—their objectives, priorities and constraints. The result of these discussions is that the negotiators may have to revise the problem, objectives and preferences, and also their strategies and initial tactics.

3. Exchanging offers and arguments allows the parties to learn of the others' limitations, and to identify the key issues and critical areas of disagreement. During this phase, the parties realize the potential of a compromise and can assess its main features. The analysis of a negotiation may focus on the modification of strategies, the determination of concessions and revision of aspiration levels, and on the restriction of efficient solutions to those which may be acceptable to the parties.

4. Reaching an agreement means that the parties realize that the negotiation will be successful. Having identified the critical issues, they may develop joint proposals or soften their individual limitations. The parties may also identify a limited number of possible compromises.

5. Concluding the negotiation takes place when the negotiators reached an agreement. They evaluate this compromise and consider its possible improvements. They also may discuss additional issues which, however, have no impact on the negotiations (e.g., the agreement implementation).

The post-negotiation activities phase is oriented on future negotiations. Its focus is on assessing the negotiation and its outcomes, and the negotiator's performance in order to learn what worked well and what did not. This phase is of particular importance for inter-organizational negotiations; it is used to record best practices and for knowledge transfer (Ertel 1999).

3.2 Negotiation protocol

The negotiation process model allows to group activities undertaken by the negotiators and/or software. The purpose of the different negotiation phases is to provide the participants with a framework and rationale for activities conducted in each phase. The consideration of phases helps to specify negotiation activities which are undertaken and the relationships among them.

In order to be able to implement an ENS, it is necessary to precisely define the activities and their sequence. This is done with the use of the negotiation protocol, which defines the activities that are permissible in every state of the negotiation, their input and output requirements. The key concepts used to define the activities and to specify their sequencing are presented in Figure 2. Based on a behavioural negotiation theory, approaches, and models, the process model, strategies, tactics and activities are formulated. Behavioural research does not, however, provide sufficiently precise insights regarding ordering of activities within each negotiation phase. This has to be done with a protocol, which—as illustrated in Figure 1—depends on the process model and the selected strategy and tactic.

Negotiation strategies and tactics often require the use of different protocols. Also, different types of negotiations, e.g., single or multiple issues, and different roles of the negotiators, e.g., buyer or seller, require different protocols. To meet the requirements of the negotiators and researchers who design negotiation experiments we need to equip an ENS with several protocols and be able to add new ones.

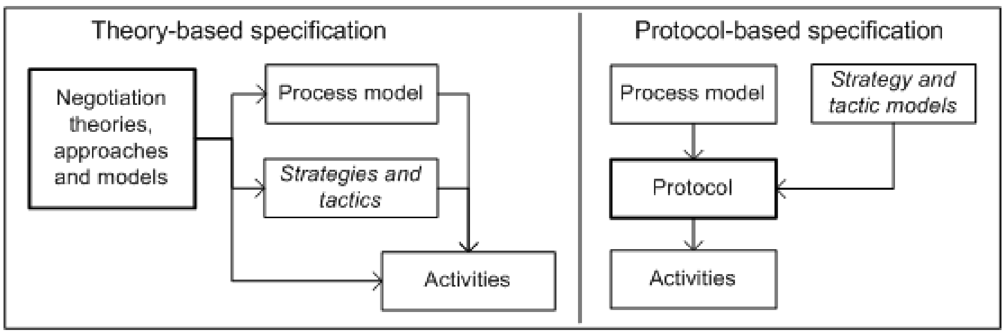

Figure 2. Theory- and protocol-based activity specification

We illustrate the construction of a negotiation protocol using a simple example. A seller *S* who wants to sell a product provides its short description, and specifies the issues (e.g., price, quantity and delivery time) and the initial issue values. Then the seller waits for a potential buyer *B*. The buyer reviews the information and asks about the warranty. *S* adds the warranty issue and its value and informs *B* who, in turn, suggest a different price and quantity. After exchanging several offers *S* and *B* arrive at an agreement.

The Seller protocol used in this simple scenario is illustrated in Figure 3. Note that there may be several different protocols that can be used to sequence the same set of activities for one scenario.

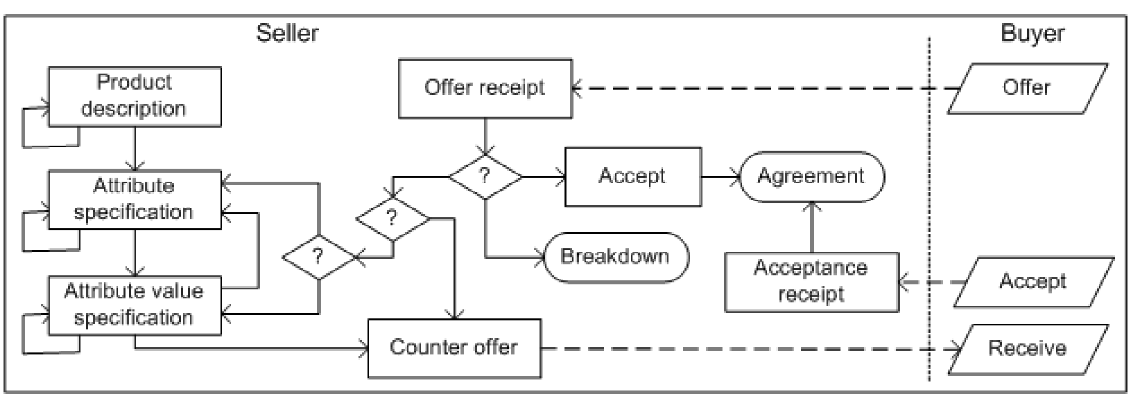

Figure 3. A simple Seller protocol

The Seller protocol begins with the product description—an activity which can be repeated several times. This activity and the following two activities comprise the planning phase. Then, S waits for an offer from B. If B makes an offer, S enters the activity offer receipt. S decides to accept, reject and/or terminate the negotiation or make a counter offer. If S decides to make a counter offer, she may first enter either the issue specification or the issue value specification activities.

It can be seen from this example, that the same set of activities may be used in the construction of many different protocols. We could construct a buyer protocol, a protocol for price-only negotiation, and a protocol where the seller makes the first offer.

The difficulty in constructing a protocol can be illustrated with this simple example. The issue value specification can precede the counter offer activity only if S earlier undertook the offer receipt activity. This shows that the set of preceding and following activities may change depending on other activities that are not elements of these two sets.

Finally, this example illustrates the requirement for constructing protocols for each negotiator separately. Protocol design involves the specification of the negotiator's roles (e.g., buyer, seller) and the interaction of activities undertaken by the negotiators. The activities are analyzed from the point of view of the input they require and output they produce. This analysis leads to: (1) specification of detailed actions undertaken by software and low-level processes which comprise an activity; and (2) assignment of these actions and processes to the negotiators and to software.

4. Overview of the Invite system

4.1 Activities, components, and pages

Activities are—from the perspective of the negotiators—the most concrete components of a negotiation. The activities, as shown in Figure 2, are formulated based on theories, approaches and models. In order to describe the Invite system and its use in e-negotiations, we take a bottom-up approach and begin with the representation of activities.

ENSs, like other web-based systems, interact with their users through pages. A page provides information to the user (e.g., about the negotiation problem or an offer) and/or may require that the users enter information (e.g., determine problem issues or construct an offer). This indicates that one or more activities may be undertaken on one page (e.g., a page may both display an offer and ask the user to construct a counter-offer).

We represent the negotiation activities with components; every activity is associated with at least one component. A component is a program that generates output and/or obtains input from the user, e.g., two activity components are used to generate the page shown in Figure 8, the components used to enter a message and to construct an offer. Component 1 produces C1 output, which corresponds to Activity 1 (see Figure 4). A component may use data stored in databases and one or more external applications, e.g., for creating a graphic or calculating utility values.

It is important to note that not every component corresponds to an activity. In the example in Figure 4, component 2 produces an output, which is not associated with any activity. Examples of such components are headers, footers or the information about the status of the negotiation. These components facilitate customization of a negotiation to a particular organization or research experiment.

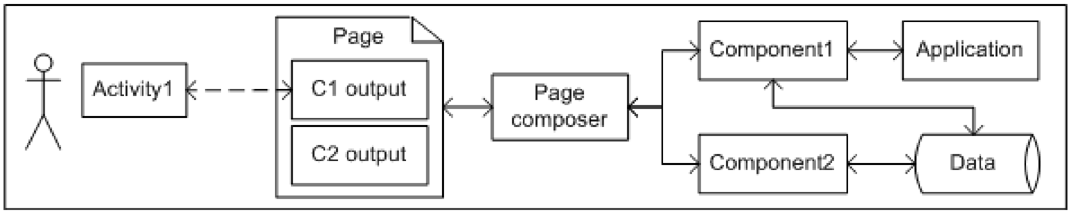

Figure 4. Activities, components and pages

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 of the positioning of components on a page on the negotiators' efficiency), but also for the internationalization of 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 (e.g., offer receipt, counter offer construction and message activities may be placed on one or on separate pages).

4.2 Page composers, process models, and protocols

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 relationship between page composers; this relationship depends on the context. Examples of this dependency are given in Section 3.2. After S introduced a new issue, she has to add the issue values as well. Also, S may add a new issue value during the offer exchange phase, however only if she obtained an offer from the buyer.

The negotiation protocol may be implemented in several ways, for instance, by a rule-based expert system, or by a database. We chose a database-driven approach illustrated in Section 5. This approach requires that a distinction between a protocol and its protocol be made. 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. The protocol instance is the implementation of the protocol; it is also a table that is being filled in during the negotiation.

This database-driven approach requires a program which consults the protocol, based on the context, updates the protocol instance, and controls the execution of the page composers. This program, the *negotiation controller*, and its function are illustrated in Figure 5.

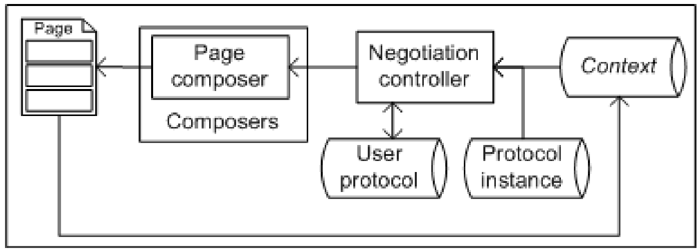

Figure 5. Pages, protocols and process models

The negotiation controller runs a protocol instance assigned to users' negotiations. The context, in which each instance of a negotiation runs, is the stage in which the negotiators are at a given moment and is defined by the activities undertaken so far. It selects and activates the appropriate page composer program according to a protocol instance (see Figure 5).

The negotiation controller is independent of the sets of components, page composers and rules, so that new, additional components, page composers and rules can be introduced into the system without changing the implementation of the negotiation controller providing essential flexibility. In fact, when the negotiation controller executes a negotiation instance, it is not "aware" of any internals specific to a component, page composer or rule. The available sets of components, page composers and the rules constrain the execution of a protocol instance, but do not modify the controller itself.

5. Invite architecture and design specifications

5.1 System architecture

Invite is designed as a typical three tier web application 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 (see Figure 6).

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 (e.g., offers and messages), because the user data is kept separate from system data (e.g., user accounts and negotiation case information).

The Invite 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, the protocol instance as well as external applications.

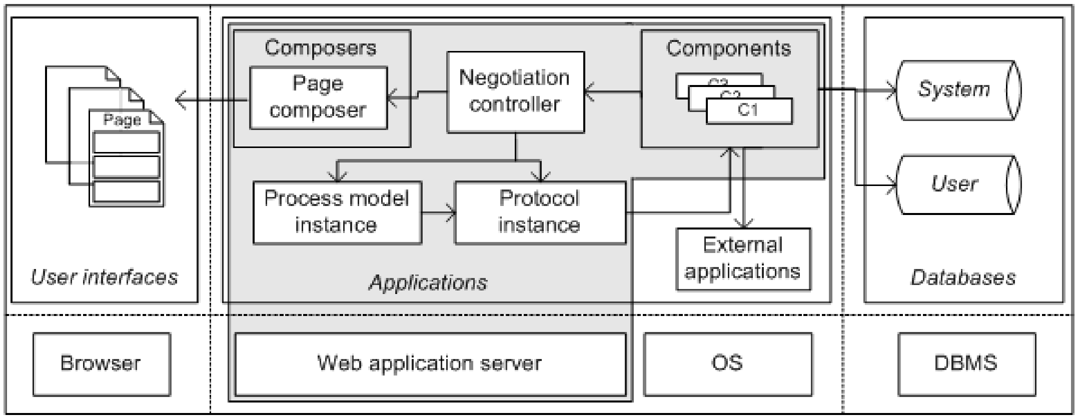

Figure 6. The Invite system architecture

5.2 Design methodology and implementation

The Invite system is designed and implemented using the Fusebox Lifecycle Process (FLiP) as software engineering methodology. FLiP is essentially an evolutionary prototyping approach to building web applications, and is based on best practices, constant user feedback and early changes in the process (http://fusebox.org). It is especially useful in connection with the Fusebox framework for implementing web applications. The Fusebox framework is an architectural framework for building web applications that establishes a standard for how to organize a web application and how to process requests from a web browser. It is implemented by a set of core files which are collectively known as "Fusebox" (Quarto-vonTivadar et al. 2003).

The Invite system uses the Fusebox framework in connection with the ColdFusion Markup Language (Brooks-Bilson 2003). Except for external applications, all Invite applications run inside the Macromedia ColdFusion MX application server. The Invite databases reside in a MySQL database. The development and production environment is Microsoft Windows 2000.

5.3 Invite Database Overview

As mentioned in Section 5.1, Invite separates negotiation and user data from system data. The advantages of using two different databases are enhanced security and user data portability. It also allows users or their organizations storing the negotiation data on their own computers.

The core relations of the Invite data model are presented in Figure 7. According to the system architecture (see Section 5.1), entities in the system and in the user database are separated. The diagram is divided into four sections. User account information is stored in the USER table. Information for a predefined negotiation case is kept in the CASE, SIDE, ISSUES and OPTIONS tables. The tables in the user database are stored information about the state and history of a negotiation instance and associate it with a protocol instance. A specific protocol instance itself is represented by the tables in the upper area and maps a state to a page composer.

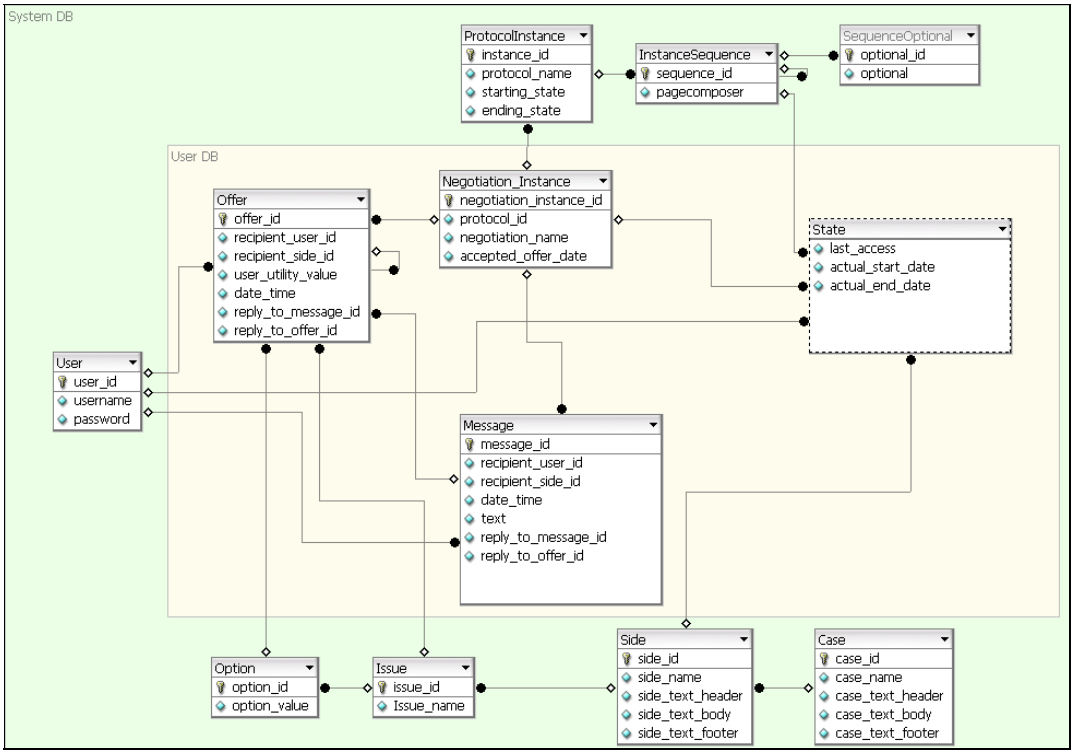

Figure 7. Invite data model

5.4 Implementation examples

Following the buyer-seller example in Section 3.2, this section describes selected implementation details in order to illustrate the inner workings of the Invite system. In our example, we take on the role of seller *S*. Buyer *B* has asked about the warranty. Before seller *S* logs out, *S* adds the warranty issue. Next time, when *S* logs into the system, the negotiation controller reads the current state of the negotiation by querying the STATE table given the data in the context (see Table 1), e.g. the primary keys of the USER (user_id of seller *S*), NEGOTIATION_INSTANCE and PROTOCOL_INSTANCE tables (see Figure 7).

Table 1. Negotiation controller

```
…
<CFQUERY name="Readstate" datasource="someDSN">
SELECT last_accesssequence_id
FROM STATE and NEGOTIATION_INSTANCE
 WHERE side_id='seller'
AND  negotiation_instance_id=#client.user_instance_number#
AND user_id=#client.user_id#
</CFQUERY>
…
<CFQUERY name="UpdateState" datasource="someDSN">
UPDATE last_access
FROM STATE
WHERE user_id=#client.user_id#
</CFQUERY>
…
```

The controller then calls the page composer associated with that state, effectively forwarding the user to the current negotiation state. The invoked page composer, called SendMessageOffer in our example, in turn creates the layout of the user interface and calls two components, SendMessage and SendOffer (see Table 2).

The page composer invokes two components, comp_sendmessage and comp_sendoffer. Basically, the components correspond to HTML FORM tags with dynamically added form actions. The advantage of having the page composer set the variables is that it allows us to combine two or more components in a page composer and create “combined” HTML forms, as e.g. in the SendMessageOffer page composer in Table 2.

Table 2. SendMessageOffer page composer

```
…
<TABLE>
<TR>
  <TD><CFINCLUDE TEMPLATE="comp_sendmessage.cfm"></TD>
  <TD><CFINCLUDE TEMPLATE="comp_sendoffer.cfm"></TD>
</TR>
<TR>
 <TD COLSPAN=”2”><INPUT NAME="reset" TYPE=reset VALUE='Clear'>
 <INPUT NAME="button" TYPE=button VALUE='Send'></TD>
</TR>
</TABLE>
…
```

The form action page referred to by the page composer stores the user input, i.e. message and/or the offer in the MESSAGE and OFFER table in the database by calling the respective component. Note that both tables reference to earlier messages and offers, respectively. Next, it calls another component, which displays a success or failure notification to the user.

The separation of the components and page composers provides two types of flexibility. The first type allows to construct different pages for similar or the same type of activities. In Figure 8, the output of the SendMessageOffer page composer is presented. One may study the impact of the ordering of message and offer input on the negotiators’ efficiency by using composers that differently position the components outputs on the page. A modification of the SendMessageOffer page compose is easily achieved, e.g. by reversing the order and by creating a SendOfferMessage page composer, which is virtually identically to the SendMessageOffer page composer except for the different display of the two components on the page. More importantly, this separation is also required to implement different protocols; it allows constructing and sequencing pages containing components corresponding to different negotiation activities.

Another type of flexibility is in system customization for the same negotiation protocol. This is achieved through the use of different components that do not correspond to the negotiation activities. To illustrate this flexibility in Figure 8, a component that displays the Invite’s logo has been replaced with the MakeMoney.com logo.

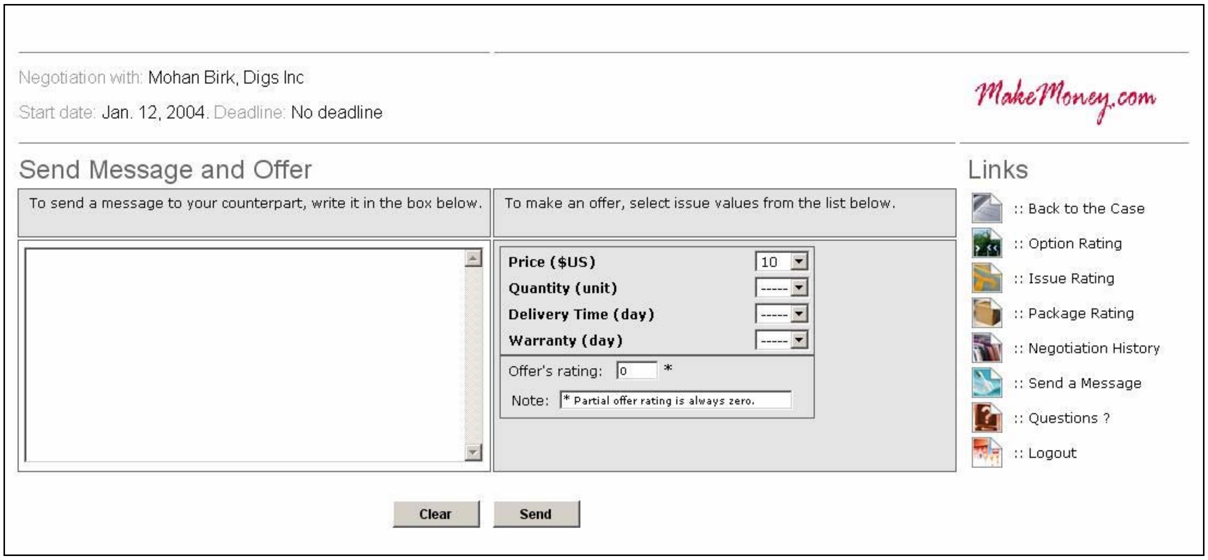

Figure 8. Screenshot of the output of the SendMessageOffer page composer

6. Discussion and conclusions

Out of necessity, experimental research in technology adoption and the relationships between the tasks, user characteristics and the outcomes achieved from the user-system interaction focuses on studying one system (Kersten et al. 1998; Starke et al. 1999), either in the same or different settings (e.g., different cases and user groups) or on the theoretical comparison of different systems features (Benyoucef et al. 2000; Jelassi et al. 1989). In the area of e-negotiation, we have been conducting both types of studies; a longitudinal study of Inspire negotiations and, more recently, an e-negotiation tournament of six ENSs (http://enegotiation.org). From the perspective of system assessment both types of research have their limitations. Firstly, the different solutions implemented in one system and its interface confounds their efficacy. Secondly, the use of several systems does not allow comparing their particular features and solutions, because they are closely interrelated with others. The comparison of the systems is also difficult because of the multiplicity of the differences and the users' inability to assess one feature in isolation from others.

For these reasons we decided to develop Invite. This system will allow us to study the protocols and solutions implemented in Inspire, INSS, Aspire and SimpleNS, as well as solutions implemented in other systems, without the noise caused by their different interfaces and components. We will also be able to study the impact of interface design on the e-negotiation process and outcomes.

Another important purpose behind Invite is the relationship between the users' characteristics and the system use, and—more broadly—the need of different solutions for different user populations. We know that users' culture influences not only system use, but users from different cultures have different requirements regarding the system features (Köszegi et al. 2004). We know little, however, about what specific solutions different users prefer and how they would apply them. For example, it appears that users from low-context cultures value analytical support tools higher than the users from high-context cultures, and the latter value communication facilities higher than the former. However, since such a study was conducted with only one system we cannot conclude, for example, that different analytical tools are equally valued by one group or that there are no analytical tools that would have a high value for users from low-context cultures. A system that allows for a ‘plug-and-play’ of various tools and solutions would provide insights that currently cannot be obtained.

Users of ENSs and other systems may not be experts in negotiations and decision making. Also, they may want to delegate some of their tasks to software agents. An important research question is to determine the suitable tasks, the specific needs of different user groups and their preferred ways of interacting with the agents. Again, this type of experimental research can be undertaken with the Invite system.

Finally, we expect that the ability to easily extend and customize the Invite system would allow us to test it in the field and—if the results are successful—provide e-negotiation services useful in solving real-life problems.

References

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

Brooks-Bilson, R. *Programming ColdFusion MX*, (2nd ed.) O'Reilly, Cambridge, MA et al., 2003.

Ehtamo, H., Hämäläinen, R.P., and Koskinen, V. "An e-learning module on negotiation analysis," Hawai'i International Conference on System Sciences, IEEE Computer Society Press, Hawai'i, 2004.

Ertel, D. "Turning Negotiation into a Corporate Capability," *Harvard Business Review* (77:3) 1999, pp 55-70.

Fisher, R., Ury, W., and Patton, B. *Getting to Yes: Negotiating Agreement Without Giving In*, (2nd ed.) Penguin Books, New York, 1991.

Foroughi, A. "An Empirical-Study of An Interactive, Session-Oriented Computerized Negotiation Support System (NSS)," *Group Decision and Negotiation* (4:6) 1995, pp 485 - 512.

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

Hordijk, L. "Use of the RAINS Model in Acid Rain Negotiation in Europe," *Environmental Science Technology* (25:4) 1991, pp 596-603.

InterNeg "Inspire, Demo Case," Ottawa, 1996.

Jarke, M., Jelassi, M.T., and Shakun, M.F. "MEDIATOR: Toward a Negotiation Support System," *European Journal of Operational Research* (31:3) 1987, pp 314-334.

Jelassi, M.T., and Foroughi, A. "Negotiation Support Systems: An Overview of Design Issues and Existing Software," *Decision Support Systems* (5) 1989, pp 167-181.

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

Jones, B.H., and Jelassi, M.T. "The Effect of Computer Intervention and Task Structure on Bargaining Outcomes," *Theory and Decision* (28) 1990, pp 355-377.

Keen, P.G.W., and Scott-Morton, M.S. *Decision Support Systems: An organizational Perspective* Addison-Wesley, Reading, 1978.

Kersten, G.E. "Rule-based Modeling of Negotiation Processes," *Theory and Decisions* (34:2) 1993.

Kersten, G.E., and Lo, G. "Aspire: Integration of Negotiation Support System and Software Agents for E-Business Negotiation," *International Journal of Internet and Enterprise Management (IJIEM)* (1:3) 2003, pp 293-315.

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

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

Korhonen, P., Moskowtz, H., Wallenius, J., and Zionts, S. "An Interactive Approach to Multiple Criteria Optimization with Multiple Decision-Makers," *Naval Research Logistics Quarterly* (33) 1986, pp 589-602.

Köszegi, S., Vetschera, R., and Kersten, G.E. "Cultural Influences on the Use and Perception of Internet-based NSS - An Exploratory Analysis," *International Negotiations Journal* (9:1) 2004, p (to appear).

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

Lewicki, R.J., Saunders, D.M., and Minton, J.W. *Negotiation*, (3rd ed.) McGraw-Hill, Boston, MA, 1999.

Matwin, S., S. Szpakowicz, Kersten, G.E., and Michalowski, W. "Logic-Based System for Negotiation Support," Proceedings of the 1987 Symposium on Logic Programming, IEEE Computer Society Press, San Francisco, 1987, pp. 499-506.

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

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

Nyhart, J.D., and Goeltner, C. "Computer Models as Support for Complex Negotiations," International Conference for the Society for General System Research, Budapest, 1987, pp. 40-48.

Quarto-vonTivadar, J., Kotek, B., LeRoux, B., Clark, S., and Woodin, P. *Discovering Fusebox 4 with ColdFusion* Techspedition, Las Vegas, NV, 2003.

Raiffa, H., Richardson, J., and Metcalfe, D. *Negotiation Analysis. The Science and Art of Collaborative Decision Making*, Harvard University Press, Cambridge, 2003.

Rangaswamy, A., Eliasberg, J., Burke, R.R., and Wind, J. "Developing Marketing Expert Systems: An Application to International Negotiations," *Journal of Marketing* (53) 1989, pp 24-39.

Sainfort, F.C., David, H.G., Bosworth, K., and Hawkins, R.P. "Decision Support Systems Effectiveness: Conceptual Frameworks and Empirical Investigations," *Organization Behavior and Human Decision Processes* (45:2) 1990, pp 232-252.

Schoop, M., Jertila, A., and List, T. "Negoisst: a negotiation support system for electronic business-to-business negotiations in e-commerce," *Data and Knowledge Engineering* (47:3) 2003, pp 371-401.

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

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

Sprague, R.M., and Carlson, E.D. *Building Effective Decision Support Systems* Prentice-Hall, Englewood Cliffs, 1982.

Starke, K., and Rangaswamy, A. "Computer-Mediated Negotiations: Review and Research Opportunities," 5-2000, eBusiness Research Centre, University Park, p. 37.

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

Ströbel, M. *Engineering Electronic Negotiations* Kluwer, New York, 2003.

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

Yuan, Y., Ding, H., Rose, J.B., and Archer, N. "Multi-party interaction in a web-based negotiation support system,") 1999.