A Conceptual Architecture for a Combined Negotiation Support System

Morad Benyoucef and Rudolf K. Keller
Département d'informatique et de recherche opérationnelle
Université de Montréal
C.P. 6128 Succursale Centre-ville
Montréal, Québec, H3C 3J7, Canada
{benyouce,keller}@IRO.UMontreal.ca

Abstract

In a combined negotiation in e-commerce, the user is interested in many goods or services and consequently engages in many negotiations at the same time. The negotiations are independent of each other, whereas the goods or services are typically interdependent. Using currently available negotiation technology, the user conducts each negotiation separately, and has the burden of coordinating and reconciling them. In our research, we aim to provide a Combined Negotiation Support System (CNSS) to help the user conduct all the negotiations at the same time. The CNSS will enable the user to control and monitor the progress of the negotiations, and will make sure that the specified dependencies are respected. The architecture of the CNSS relies on workflow technology and on negotiating software agents.
The paper starts by presenting the problem and giving an overview of the envisioned CNSS. Then, underlying concepts and technology are discussed, and the proposed architecture is detailed. The paper concludes with a discussion of related work.

1. Introduction

Electronic negotiations (e-negotiations) are becoming an important research subject in the area of electronic commerce (e-commerce). The AMEC laboratory of MIT [1], for instance, puts e-negotiations at the center of its Consumer Buying Behavior (CBB) model for e-commerce. The model identifies six steps in an e-commerce transaction: (1) the identification of the need, (2) the product brokering, (3) the merchant brokering, (4) the negotiation, (5) the purchase and delivery and (6) the service evaluation.
As described by Kumar et al. [2], the most basic form of e-negotiation is no negotiation at all (also called fixed-price sale) where the seller offers her goods or services through a catalogue at take-it-or-leave-it prices. Auctions are at present the most visible type of e-negotiations on the internet as conducted by eBay [3]. E-negotiations can take a more complex form called bargaining. It involves making proposals and counter-proposals until an agreement is reached [4]. The Object Management Group (OMG) sees bargaining as bilateral and multi-lateral negotiation depending on whether there are two parties (one-to-one bargaining) or many parties (many-to-many bargaining) involved in the negotiation [5]. Negotiations are further classified as distributive or integrative [6]. In distributive negotiations only one attribute is negotiable (usually the price). The parties have opposing interests. One party tries to minimize (e.g., the buyer) and the other party tries to maximize (e.g., the seller) the price. In integrative negotiations multiple attributes of the item are negotiable (the price, the quality, the delivery date, etc.). If all the attributes are negotiable then a consumer can hope to get a small price if she can live with a poorer quality and/or can stand a long delivery time. In this case, the parties do not necessarily have opposing interests since they try to optimize different attributes. Combinatory auctions [7] are another form of negotiations that involve making bids on combinations of goods or services, and one of the main challenges is for the auctioneer to determine the winning bid.
We talk of a *combined negotiation* when a consumer is interested in many goods or services and consequently engages in many negotiations at the same time. The negotiations can be of any style (fixed-price-sale, Dutch-type-auction, bilateral bargaining, combinatory auction, etc.). The negotiations are in general totally independent of each other. The goods and services of the combined negotiation, however, are typically interdependent.
Let us look at an example of a combined negotiation, involving a company specialized in translation between several natural languages, two other companies offering printing services and a fourth one specialized in distribution. A consumer is interested in translating, printing and eventually distributing some advertising material. Let us suppose that the four companies offer their services over the web through their dedicated negotiation servers. Suppose also that each server uses a different style of negotiation. Some dependencies among the negotiated services are the number of copies, the technologies used, etc. Other constraints may involve the

preferences and priorities of the consumer (e.g., what deal to make first, the total price she is willing to pay, etc.).

Using currently available negotiation mechanisms, the consumer conducts each negotiation separately, and she has the burden of coordinating and reconciling the various negotiations. It can happen that the user makes a deal on printing ten thousand copies and later finds out (through bargaining for example) that the price of distributing fifteen thousand copies is not very different from distributing ten thousand copies.

We therefore see clearly the need for a system to support the user in conducting all the negotiations at the same time, i.e., in carrying out combined negotiations. We call such a system a Combined Negotiation Support System (CNSS). A CNSS is a tool that enables the user to track and monitor the progress of many negotiations efficiently and to respect all the constraints, dependencies and preferences of the given context. Moreover, a CNSS will support the user in taking decisions.

This work is being conducted as part of the TEM (Towards Electronic Marketplaces) project, a joint industry-university project started in early 1999. The project addresses market design issues in respect to resource allocation and control and reward mechanisms, investigates open protocols for electronic marketplaces, and explores concepts and tools for e-negotiations. We are in the process of developing a Generic Negotiation Platform (GNP) [8] which will support many types of negotiations. We will use instantiations of the GNP to implement the servers involved in a combined negotiation.

The main contribution of this paper is a conceptual CNSS architecture which is based on software agents controlled and monitored via the Process Monitoring Tool of a Workflow Management System (WfMS). The agents start as dumb assistants to the user and can subsequently be promoted to be more entrepreneuring. The promotion can be done by using a negotiation strategy that can be general or tailored to a specific situation.

Negotiating agents are an active area of research, and they have not been applied yet to combined negotiations. The use of workflow technology in our context is also novel in that we will use it to track and monitor the work of software agents instead of humans as it was intended. The four-component model proposed by the Workflow Management Coalition (WfMC) [9] will serve as the core of our CNSS architecture. Furthermore, the formal description of negotiation rules will be investigated.

This paper is organized as follows. In Section 2, we briefly describe a usage scenario and our vision of the CNSS. The underlying concepts and technology are presented in Section 3. Section 4 is dedicated to the proposed architecture of the CNSS. Section 5 discusses related work, and, finally, Section 6 concludes the paper.

2. Usage scenario and vision of the CNSS

Let us go back to our example. The consumer registers in the four negotiations. She then enters her constraints and preferences and, eventually, strategy rules to be followed during the negotiation process. One constraint could be "not to make any commitment on distribution until a deal is finalized on translation and printing". Note that the negotiations with the two printing companies are related to that of the distributor company. If you get a good deal on distribution you might as well increase the number of copies. Next, a workflow representing the combined negotiation is constructed (automatically or semi-automatically). After that, four software agents are created and assigned to each negotiation. The agents are created according to the style of negotiation practiced in the corresponding server. The negotiations are launched and the agents start by acting as simple assistants to the user. An agent might report for example that the highest bid on translation services is 10 units per page. The user can then instruct it to bid 11 units. The progress in the work of the agents is visible to the user through the Process Monitoring Tool of the WfMS. Whenever the user chooses to, she can promote all or some of her agents to "agents with a strategy".

3. Underlying concepts and technology

In this section, we discuss the underlying technology of the CNSS. We will address the formalization of negotiation rules, agent technology, and finally, workflow technology.

3.1. Formalization of negotiation rules

Cass et al. [10] suggest that an automated negotiation must have four necessary properties. It must be: *(1) Correct*: an automated negotiation must not contain errors such as deadlocks or incorrect handling of exceptions; (2) *Reasonable*: it must allow adequate time for bids to be made (or for decisions to be made in general); (3) *Robust*: it should continue to function properly after an improper action by the user; and (4) *Fast*: it has to execute and respond quickly. We believe that an automated negotiation must also be *traceable*. In order to be trusted, the software must be able to justify its actions to the user if necessary.

An important issue is the need to separate the process of negotiation from the other parts of the software. Separation permits efficient implementation, easy testing and, last but not least, encourages reuse.

Another issue is that the rules describing the negotiation should be known to the user (human or software). We believe that the rules are as important as, or perhaps even more, than the other information describing the good or

service that is the object of the negotiation. We agree with Wurman et al. [11] that “implementing auction mechanisms in a larger context will require tools to aid agents in finding appropriate auctions, inform them about auction rules, and perform many other market facilitation functions”.
We therefore need a mechanism to formally describe the rules governing a negotiation, visualize this description when necessary and serialize it in order to transfer it over the network. A review and an evaluation of the methods used to describe negotiations can be found in [12].
Our solution uses the UML’s statechart diagrams [13]. Statecharts are well established and widely used, and they are semantically rich enough to formally describe and visualize processes. An important feature of statechart diagrams is the possibility to be serialized in XML (eXtensible Markup Language) [14] and XMI (XML Metadata Interchange) [15]. Finally, off-the-shelf simulation and analysis tools are available for statecharts, such as Statemate [16], which will help validate and render the descriptions being investigated.

3.2. Agent technology

In the area of e-commerce, software agents are well suited for information filtering and retrieval, personalized evaluation, complex coordination and time-based interaction. This last feature is vital to e-negotiations. The research conducted at the AMEC laboratory, for instance, uses negotiating mobile agents that are able to go from site to site and negotiate deals on behalf of their creator [17]. The research focuses on the negotiation phase (Phase 4 in their CBB model [1]), whereas several commercial products already use agents in the other phases. The system designed at AMEC is an on-line multi-agent classified ad system called KASBAH. The user (seller or buyer) of this system creates an agent, gives it instructions and sends it to a centralized agent marketplace. The user has high-level control over the behavior of the agent. At the time of its creation, the user (a seller in this case) sets the desired price, the desired time to sell, the lowest acceptable price and the negotiation strategy. The user can check on her agent to see what it is doing and, she can also modify the parameters and give approbation before a deal is struck. If the user goes off-line, she can be notified of an eventual deal by e-mail.
The AuctionBot project at the University of Michigan takes a different approach by combining human and software agents [11]. It is a general purpose internet auction server. The user can create new auctions to sell products by choosing from a selection of auction types and by setting a set of parameters (clearing time, method for solving bidding ties, number of sellers permitted in the auction, etc.).

3.3. Workflow technology

A workflow describes the flow of information and control in a business process. Some examples of business processes are: the processing of an insurance claim, the processing of a purchase order, the processing of customer support, etc. A WfMS is a software that manages the workflow efficiently by tracking and controlling its execution. The WfMC came up with a four components reference architecture [9]: (1) The Process Definition Tool which is used to enter the workflow in the computer, (2) the Workflow Engine which reads the workflow definition and executes and tracks it, (3) the Workflow Client Application through which the participants interact with the workflow, and (4) the Administration and Monitoring Tool which allows the administrator to track the status of the workflow.
WfMSs are an active area of research, and their application to e-negotiations is novel. One of the aspects that will be investigated in TEM is the representation of workflows for our particular application domain. We are currently studying various approaches to such representation, including the MENTOR approach, which is based on activity diagrams and statecharts [18]. As we plan to use existing WfMS technology, another challenge will be to map our conceptual CNSS architecture onto the WfMS reference architecture, and from there onto concrete WfMS tools.

4. Proposed architecture

The description of the architecture is divided in two parts. We first present the start-up time architecture during which the workflow is constructed, then we look at the run-time architecture during which the combined negotiation is conducted.

4.1. Start-up time architecture

We suppose that the negotiation servers have already been identified which means that the product and merchant brokering phases of the CBB [1] model have been successfully completed either manually or automatically and that the user has registered on the corresponding servers. After that, the formal descriptions of the negotiation rules on each server are downloaded into the Negotiation Repository (NR), for use in the instantiation of the agents. The constraints entered by the user via the GUI Tool go to the Constraint Repository (CR) and will be reflected in the workflow. The negotiation strategy rules are also entered via the GUI tool and go to the Strategy Repository (SR). We intend to use a declarative language to express these rules. The Process Definition Tool of a WfMS is used to enter the workflow in the CNSS. We will investigate ways to build the workflow either automatically or semi-automatically. Figure 1 represents the architecture at start-up time just after registration.

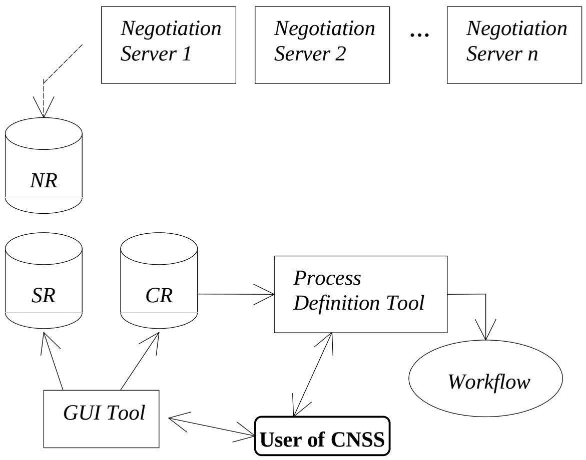

Figure 1. Start-up time architecture

Going back to our scenario, suppose that the negotiation taking place on the first server is an English auction, and that one negotiation on printing services is a fixed price sale and the other one a Dutch auction. Suppose also that the fourth negotiation is a bilateral bargaining. The descriptions of the four negotiations are downloaded from the corresponding servers in the form of UML statecharts serialized in XMI. Eventually, the constraints of the user and the negotiation strategy rules are entered via the GUI Tool. The workflow is then constructed based on the contents of the CR repository.

4.2. Run-time architecture

At this stage, the agents will be instantiated using the NR repository. They will act as participants in the workflow. Each one will be specialized in the style of negotiation practiced on the server assigned to it. At first, the agents will use no strategy. The Process Monitoring Tool will be used to track the agents, cancel (if allowed) their work, instruct them, etc. The user can see what each agent is doing at any given time and can intervene whenever she sees the need for it. The agents depend on the user in taking decisions (my bid was beaten, do I make another bid? How much do I bid?, etc.). The Workflow Engine executes the workflow by dispatching work to the participants. The user can modify/add constraints at run-time via the GUI Tool which will change the content of the CR repository. This adds flexibility to the combined negotiation by making it possible for the user to adjust to new situations in order to get what she wants. The same thing can be said about the negotiation strategy rules which can be modified/augmented at run-time, too.

At any time, the user might instruct her agents to use the strategy rules stored in the SR repository. Figure 2 shows the run-time architecture of the CNSS.

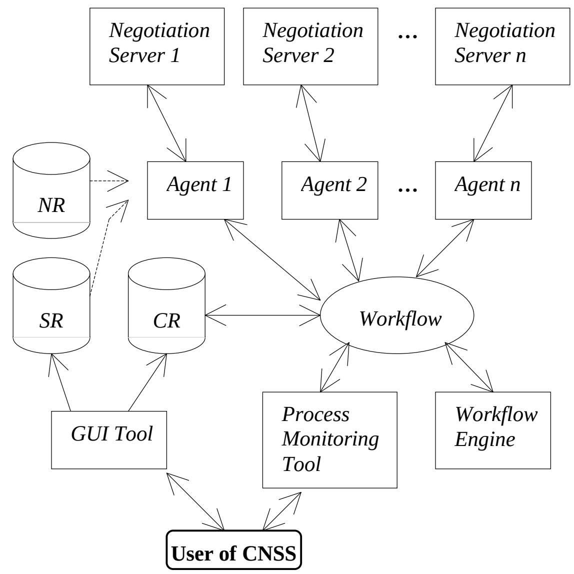

Figure 2. Run-time architecture

Now, let us go back to our scenario. The agents are instantiated and the bidding starts at the first negotiation server. The user decides to promote the corresponding agent by giving it the following strategy: "make bids starting at the current highest bid by adding two units. Wait 5 units of time before bidding, and when the highest bid reaches 100 units inform the user". The bargaining on distribution needs the involvement of the user so the corresponding agent will play only the minor role of acknowledging the proposals and counter-proposals. The two agents assigned to the second and third negotiations (printing) will have to wait for the outcome of the first negotiation. The bargaining on distribution can influence or be influenced by the outcome of the other negotiations. This process goes on until a deal that satisfies the user is reached or until the user pulls out of the negotiations.

5. Related work

Below, we present two approaches that we think are most closely related to the CNSS approach.

The approach by Su et al. [4] is based on the idea that a consumer registers on a proxy negotiation server by giving a description of the goods or services she wants, her preferences and a negotiation strategy. Then the server takes over and looks for a supplier that matches the consumer. When it finds one it starts a bargaining type negotiation until eventually a deal is reached. The negotiation strategy supplied by the consumer is used by the server. Notice that in this case, the merchant brokering phase is the responsibility of the server, and that the user has no control over the negotiation. Many negotiations can be started on the same proxy server and, they will be stored as persistent objects in a database.

With this solution, the user will have no control over the negotiations once they are started. She can only see their progress, but cannot intervene in them. We prefer the user to be in control and intervene whenever she wants in the combined negotiation. The workflow representing the combined negotiation would have to be constructed, stored, and executed at the negotiation server (a feature that is not supported by the proxy server as it is described in [4]). We prefer the workflow representing the combined negotiation to be constructed, stored and executed on the client side rather than on the negotiation server side. In contrast to our solution, the proxy negotiation server approach does not explicitly support combined negotiations. Furthermore, our solution is decentralized and supports any type of negotiations, whereas the server solution supports only bilateral bargaining.

If we compare the services expected from the CNSS to those offered by the KASBAH system [17], we see some similarities. The agents can be monitored by their creator via GUI tools. They also can have a negotiation strategy. For KASBAH, the strategy is very simple and is chosen from a set of predefined strategies. In the CNSS, however, the strategy is entered rule by rule. Consequently, it can be more sophisticated and cannot easily be guessed by the server or the other participants. In the KASBAH system for instance, it is possible to guess the negotiation strategy of an agent just by observing it for a while. An agent for which the strategy is known by its opponents is at a disadvantage [19]. The agents in KASBAH are mobile, ours are not. The agents in KASBAH practice only one style of negotiation which is bargaining, ours will practice any kind of negotiation.

6. Conclusion

In this paper we stated the problem of combined negotiations and showed the need for a CNSS to support the user in conducting many negotiations in parallel. We then detailed the conceptual architecture of the CNSS. We made the case for the use of workflow technology to coordinate negotiating software agents and showed the need for a formalized description of negotiation rules. We compared our agents to those of the KASBAH system and our entire system to a proxy negotiation server and pointed out the originality of the envisioned CNSS.

Our next step will be to implement a prototype as a proof-of-concept of this conceptual architecture. We intend to use the already functional GNP [8] to instantiate negotiation servers, and we will leverage existing WfMS and agent technologies.

Acknowledgement: This research is supported by Bell Canada, BCE Emergis, NSERC (National Sciences and Engineering Research Council of Canada), and CIRANO.

References

[1] The AMEC Laboratory. http://ecommerce.media.mit.edu/.
[2] M. Kumar and S. I. Feldman. Internet auctions. Tech. report, IBM Inst. for Adv. Commerce, Yorktown Heights, NY, 11/98.
[3] EBay Auctions. http://www.ebay.com.
[4] S. Y. Su, C. Huang, and J. Hammer. A replicable web-based negotiation server for e-commerce. In 33rd International Conference on System Sciences, Hawaii, 2000.
[5] OMG Negotiation Facility. http://www.oms.net/ecdtf.html.
[6] M. Strobel. Effects of electronic markets on negotiation processes - evaluating protocol suitability. Tech. Report 93237, IBM, Zurich Research Laboratory, Switzerland, 1999.
[7] T. Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In IJCAI, pp. 542-547, Stockholm, Sweden, 1999.
[8] M. Benyoucef, R. K. Keller, S. Lamouroux, J. Robert, and V. Trussart. Towards a generic e-negotiation platform. In Proc. of the Sixth Conf. on Re-Technologies for Information Systems, pp. 95-109, Zurich, Switzerland, February 2000.
[9] The WfMC. http://www.aiim.org/wfmc/mainframe.htm.
[10] A. G. Cass, H. Lee, B. Staudt Lerner, and L. J. Osterweil. Formally defining coordination processes to support contract negotiation. Technical Report UM-CS-1999-039, University of Massachusetts, Amherst, MA, June 1999.
[11] P. R. Wurman, M. P. Wellman, and W. E. Walsh. The Michigan internet AuctionBot. In Second Intl. Conf. on Autonomous Agents, pp 301-308, Minneapolis, MN, May 1998.
[12] M. Benyoucef and R. K. Keller. An evaluation of formalisms for combined negotiations in e-commerce. In Proceedings of the Workshop on DCW, Quebec City, QC, Canada, June 2000. Springer-Verlag. LNCS. To appear.
[13] J. Rumbaugh, I. Jacobson, and G. Booch. The Unified Modeling Language Reference Manual. Addison-Wesley. 1999.
[14] XML. http://www.w3.org/TR/1998/REC-xml-19980210.
[15] XMI. ftp://ftp.omg.org/pub/docs/ad/98-10-05.pdf.
[16] D. Harel, H. Lachover, A. Naamad, A. Pnueli, M. Politi, R. Sherman, A. Shtull-Trauring, and M. Trakhtenbrot. STATEMATE: A working environment for the development of complex reactive systems. IEEE Trans. on Software Engineering. 16(4):403-414, 1990.
[17] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In 1st Intl Conf on Practical Appli. of Intelligent Agents Technology, London, UK, April 1996.
[18] P. Muth, D. Wodtke, J. Weissenfels, G. Weikum, and A. K. Dittrich. Enterprise wide workflow management based on state and activity charts. In: A. Dogac, L. Kalinichenko, T. Özsu, A. Sheth (Eds.): Workflow Management Systems and Interoperability, Springer Verlag, 1998.
[19] C. Beam and A. Segev. Automated negotiations: A survey of the state of the art. Technical Report 97-WO-1022, Haas School of Business, UC Berkeley, 1997.