

Open Distributed Architectures for E-Market Design¹

Rudolf K. Keller

*Département d'Informatique et de Recherche Opérationnelle, Université de Montréal,
C.P. 6128 Succursale Centre-ville, Montréal, Québec, H3C 3J7, Canada*

keller@iro.umontreal.ca

Abstract. In the TEM project, we investigate novel architectures for the design of electronic marketplaces. At the core of our research is the study of various types of electronic negotiations, and the development of architectures and software tools to support those negotiations. One major focus of the TEM project are combined negotiations and combined negotiation support systems. This paper provides an overview of the TEM project and our research in combined negotiations.

Keywords. electronic commerce, electronic negotiation, electronic marketplace, combined negotiation, combined negotiation support system.

1. Introduction

Electronic marketplaces are at the center of our vision for the future of e-commerce. We believe that the next major step in the reorganization of economic structures and activities is the emergence of market- or sector-wide integrative applications such as open virtual marketplaces. The idea is to offer to the largest possible network of businesses a nexus of business services, and ultimately to allow companies to lower their costs of doing business. At the center of the marketplaces lie negotiation servers where deals are struck and prices determined; around it are complementary services, including matching and advising services, and standard e-commerce infrastructure components.

The creation of electronic marketplaces raises many challenges. Our focus is mainly on multi-product, business-to-business negotiations. We believe that this is the direction in which the least has been done, in which potentially high added-value solutions may be provided and in which inter-disciplinary research is the most required. Multi-product negotiations can take various forms such as combined negotiations (where negotiations on unrelated servers are combined by a market participant), synchronized open auctions, and combinatorial auctions. We do not take the negotiation rules as a given nor simply concentrate on how participants can make the best use of them; rather, we are exploring alternative market designs and examine their impacts on all participants. We wish to provide advice on how to set up these marketplaces.

In the remainder of this paper, we first give an overview of the TEM project which provides the framework of our research. Then, we will address combined negotiations (CN) and their support by software tools, a key aspect of the TEM project. We will wrap up the paper with a conclusion and an outlook into future work.

2. Overview of TEM Project

Our work is being conducted in the framework of the TEM (Towards Electronic Marketplaces) project, a joint industry-university project started in 1999 that involves researchers from computer science, operational research, and economic science. We are focusing around three main questions: (i) how should multi-product negotiation rules and servers be designed? (ii) what kind of tools can best help participants combine and optimize their negotiations on unrelated markets? and finally, (iii) can we replicate what can be done with a centralized multi-product market through decentralized negotiation rules and servers? and if so, how? and if not, what is gained by relating explicitly these multiple-items negotiations? In order to

¹ This research is supported by Bell Canada, BCE Emergis, NSERC (National Sciences and Engineering Research Council of Canada), and CIRANO (Centre Interuniversitaire de Recherche en ANalyse des Organisations).

make advances towards answering these questions, we are conducting research in the fields of market design and open distributed architectures.

The objective of a particular market design is to induce efficient allocation of resources among multiple (self-interested) agents. The proper design of a market must ensure an efficient and honest allocation mechanism, as well as appropriate control and reward mechanisms. Hence markets must pursue two objectives: (i) induce participating agents to reveal the necessary information, and (ii) given this information, optimize the allocation of resources. For certain cases, we aim to embed the rules required by the market into the corresponding market servers as high-level optimization tools. The design of market and auction rules has become a major field of research in economics and past experiments have shown that market rules matter a lot. We are using the methodologies of experimental economics, of multi-agent simulations and of operational research in order to investigate various possible market rules.

A further challenge is the design of open protocols and architectures for electronic negotiations. The technological challenge is to allow participants to interact dynamically through many different distributed markets and to allow them to identify and interact with as many business partners as possible. If we want participants to be able to combine negotiations on multiple market servers, we need (i) to design each individual server accordingly, and (ii) provide tools to keep track of the flow of information and control in the negotiation process. In a combined negotiation, the users connect to many servers which may each use different negotiation rules. A system that combines these negotiations must be able to understand and adapt to the various rules and process the information accordingly. Hence, there is a need to formally describe the rules governing the negotiations, to serialize them and to transfer them over the network. Both the rule descriptions and the negotiation-related information may take the form of XML-documents with the relevant versioned DTDs.

3. Software Support for Combined Negotiations

We talk of a Combined Negotiation (CN) when a consumer is interested in many items, and consequently engages in many negotiations at the same time. The negotiations can be of any type (fixed-price sale, Dutch auction, bilateral bargaining, combinatorial auction, etc.). Each negotiation is for a separate bundle of copies, items, and attributes and thus corresponds to one point in the 3D negotiation space (see Figure 1, solid lines). The negotiations are in general totally independent of each other. The goods and services of the CN, however, are typically interdependent.

As an example of a CN, let us take a vacation package consisting of three items: a transportation ticket, a hotel room, and a ski trip. The three items are obviously interrelated since the consumer would have to travel to the location where the ski trip journey initiates (or at least near it) on the date of the trip (or before it). In addition to the places and dates, there can be other constraints and dependencies between the three individual items, as we will see later on. Let us suppose that the three items are negotiable (keeping in mind that a fixed-price sale is a special case of a negotiation) and that they can be negotiated on different negotiation servers. Let us suppose also that the negotiations practiced on each single server can be of different types. Clearly, the individual negotiations are independent of each other. Using currently available negotiation technology, the consumer would have to conduct each negotiation separately, and she would have the burden of coordinating and reconciling the various negotiations. It can happen for instance that the consumer makes a deal on a plane ticket and a hotel room, and then, while negotiating the ski trip, finds out (through bargaining for example) that she is missing out on a very interesting deal only because she arrives a few hours late.

We therefore see a pressing need for a software system that supports the user in conducting all the negotiations at the same time, i.e., in carrying out CNs. 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.

In our work, we have designed the architecture and built a prototype for a CNSS that we call CONSENSUS. The architecture of CONSENSUS is based on workflow technology, negotiating software agents and rule engine technology. The originality of CONSENSUS stems from the fact that the user models her CN at build time using a workflow. The workflow itself captures the sequencing of the individual negotiations and the dependencies between them. At runtime, software agents are assigned to

individual negotiations and participate in the CN as actors in the workflow. The user of CONSENSUS can track and monitor the progress of the CN via the Workflow Monitoring and Control Tool of a Workflow Management System (WfMS). She can also monitor the work of the agents via an Agent Monitoring and Control Tool. The four-component model proposed by the Workflow Management Coalition (WfMC) [6] serves as the core of our CNSS architecture. We use Rule Engine technology to enable the agents to use negotiation strategy rules. These rules can be edited before the negotiations start and/or during the process of the negotiations. The formal description of negotiation rules is also investigated [2], as it is necessary for CONSENSUS to transfer the negotiation rules from the corresponding negotiation servers to its Negotiation Rules Repository.

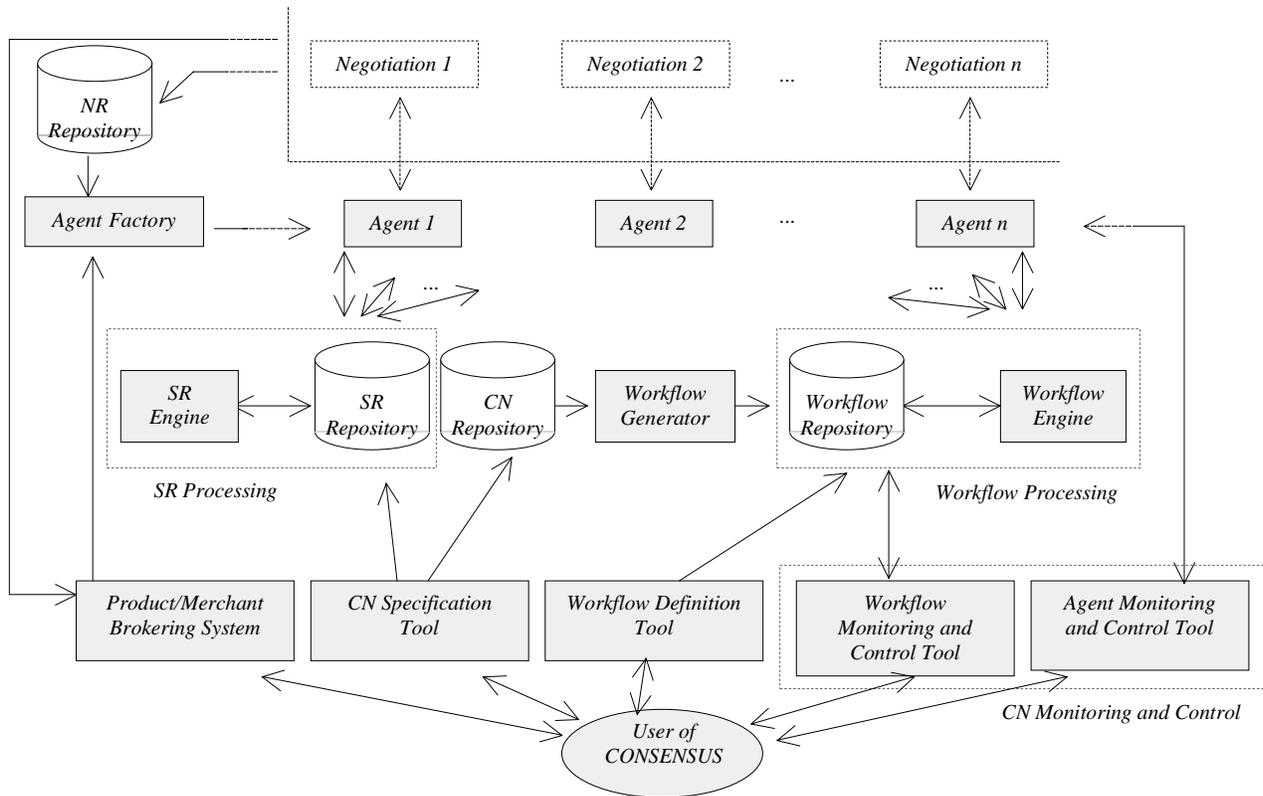


Figure 1: Architecture of CONSENSUS.

Figure 1 shows the architecture of CONSENSUS. The *Product/Merchant Brokering System* (responsible for phases 2 and 3 of the Consumer Buying Behavior model for e-commerce [5]) is used by the consumer to select the products and the providers she is willing to negotiate with. The system will return a list of candidate negotiations, and the user selects the ones she is interested in. This list is passed on to the *Agent Factory*. The Negotiation Rules (NR) are then downloaded from the corresponding negotiation servers. We are using XMI [7] as a vehicle for such a transfer. The rules are stored in the *NR Repository* and are used by the *Agent Factory* to instantiate the *Agents* that will be responsible for the individual negotiations. Each agent will be instantiated according to the negotiation type practiced by the negotiation server assigned to it. After they are instantiated, the agents connect to and register on the corresponding servers. The consumer uses the *CN Specification Tool* (a GUI tool) to specify the CN. A constraint language will be used to represent, manipulate and store the constraints of the CN. The constraints are stored in the *CN Repository*. Another part of the specification deals with negotiation Strategy Rules (SR). These go into the *SR Repository*. When the CN specification task is completed, the workflow modeling the CN is generated and stored in the *Workflow Repository*. The consumer can use the *Workflow Definition Tool* to see and

debug the resulting workflow. This ends the modeling part. At runtime, an instance of the workflow is started. The workflow can be monitored via the *Workflow Monitoring and Control Tool*. The *Workflow Engine* executes the workflow by dispatching work to the agents (which are actors in the workflow) and enforces the sequencing of the activities in the workflow. The user can monitor the work of the agents (what is specific to individual negotiations) using the *Agent Monitoring and Control Tool*. Strategy rules can be entered and/or edited at run-time. Each *agent* instantiates an *SR Engine* to make inferences based on the rules found in the *SR Repository*. Note that *Negotiation 1*, *Negotiation 2*, ... *Negotiation n* are the negotiations in which the agents participate. They can be conducted on one single negotiation server or on different ones as long as they are independent of each other.

4. Conclusion

Prior to the inception of the TEM project, we have built a negotiation server which is being commercialized by Bell Canada and BCE Emergis within the wood and paper industry. The implemented market rules were relatively simple, and thus we set out to devise a more functional platform for our TEM research. This resulted in a generic negotiation server infrastructure called GENESIS [1] which is available as a proof-of-concept prototype. We use instances of GENESIS as prototypical negotiation servers with which the CONSENSUS system interacts. The CONSENSUS prototype is available now in its version 0.2. This version is functional enough to conduct initial experiments of CNs. CONSENSUS is further detailed in [3, 4].

In this paper, we described the TEM project and then focused on CNs and ways to support CNs by a CNSS. A CN involves negotiating many interdependent items by engaging in different and independent negotiations. CONSENSUS is a CNSS which helps the user model, track and monitor a CN. Its architecture is based on agent technology, as well as workflow management and rule engine technology. CONSENSUS can be used by a consumer from her home computer to negotiate, for instance, a vacation package (B2C), or it can be used in B2B, for example, by a travel agency to negotiate travel packages for its customers.

We claim that a CN can be specified using a workflow. We are certain that a workflow can capture the sequencing of the negotiations and some dependencies between them (we call this the CN know-how), and currently, we are experimenting with more complex examples to further substantiate our claim. The software agents know the rules of the individual negotiations in which they are involved (we call this individual negotiation know-how). But knowing the rules of the negotiation does not make you a good negotiator. What is needed are negotiation strategies to be used by the agents in order to make them better negotiators (we call this negotiation strategy know-how).

As future work, we aim to further investigate several research issues that are central to our solution. One such issue is how to instantiate software agents based on a description of the negotiation rules. Another issue is the high-level specification language we would like to define and implement in order to free the user from the task of modeling a workflow. The question of breaking commitments in a negotiation will also be investigated along with its effect on CNs. Negotiation strategies are an important research area for us. We will have to come up with strategies that apply to individual negotiations and strategies that apply to a CN. We will also have to see to what extent the negotiation strategies depend on the negotiation type. Our conceptual work will be completed by proof-of-concept prototypes, as incepted with CONSENSUS versions 0.1 and 0.2

In conclusion, we believe that a CNSS such as CONSENSUS is a much-needed tool for coping with the ever-increasing scope and complexity of e-negotiations.

Acknowledgment: CONSENSUS is being designed and implemented by Morad Benyoucef, Mathieu Vézeau, and Hakim Alj. I also gratefully acknowledge the contributions of my colleagues and fellow investigators in the TEM project, Gilbert Babin, Teodor Crainic, Michel Gendreau, Robert Gérin-Lajoie, Peter Kropf, and Jacques Robert.

References

- [1] M. Benyoucef, R. K. Keller, S. Lamouroux, J. Robert, and V. Trussart. Towards a Generic E-Negotiation Platform. In Proceedings of the Sixth Conf. on Re-Technologies for Information Systems, pp. 95-109, Zurich, Switzerland, February 2000.
- [2] M. Benyoucef and R. K. Keller. An Evaluation of Formalisms for Combined Negotiations in E-Commerce. In Proceedings of the Workshop on DCW, pages 45-54, Quebec City, QC, Canada, June 2000. Springer. LNCS 1830.
- [3] M. Benyoucef and R. K. Keller. A Conceptual Architecture for a Combined Negotiation Support System. In Proceedings of the Eleventh International Workshop on Database and Expert Systems Applications (DEXA 2000), pages 1015-1019, Greenwich, London, Britain, September 2000. IEEE. Presented in W17: Workshop on Negotiations in Electronic Markets - Beyond Price Discovery - E - Negotiations.
- [4] M. Benyoucef, H. Alj, M. Vézeau, and R. K. Keller. Combined negotiations in e-commerce: Concepts and architecture. Electronic Commerce Research Journal, 2, 2001. Special issue on Theory and Application of Electronic Market Design. Baltzer Science Publishers. To appear.
- [5] P. Maes, R. H. Guttman, and A. G. Moukas. Agents that Buy and Sell: Transforming commerce as we know it. Communications of the ACM, March 1999.
- [6] The WfMC. <http://www.aiim.org/wfmc/mainframe.htm>.
- [7] XMI. <ftp://ftp.omg.org/pub/docs/ad/98-10-05.pdf>.