

Web-Based Business-to-Business Negotiation Support

Wojciech Cellary, Willy Picard and Waldemar Wiczerzycki

Department of Information Technology
University of Economics at Poznan
60-854 Poznan, Poland
email: {cellary, picard, wiecz}@kti.ae.poznan.pl.
WWW: <http://WWW.kti.ae.poznan.pl>.

Abstract. The *Agora* system presented in this paper is a tool devoted to business-to-business electronic commerce. *Agora* is composed of two main parts. The first one, called the *virtual table*, is a conferencing tool which allows negotiations between business partners. The second one is a collaborative document writing tool which supports editing of business contracts. *Agora* is independent of hardware, operating systems, browsers and database management systems. *Agora* is accessible to any *Internet* user, provided the user has been properly registered.

1 Introduction

The rapid development of the *Internet* and *World-Wide Web* (WWW) provides the possibility to apply this technology to the field of business. We can distinguish three consecutive steps of evolution of WWW applications. First, the WWW was used to distribute information only. A customer of a business unit could read some information, e.g., list of products available, their characteristics and prices, and then purchase them in a conventional manner - by a telephone, fax, or letter. Second, WWW was used to both distributing and collecting information through electronic forms. A customer could purchase some products by completing electronic forms available on web pages. Note, that in the both above steps the character of communication is human-machine. This type of communication is sufficient for customer-to-business negotiations, where the only decision is whether to buy or not to buy a given product. These forms, however, are too simplistic to cover the needs of business-to-business negotiations, where many decisions have to be taken, concerning different factors of a contract, e.g., amount, packing, transport, payment. These decisions require human-to-human communication supported by specialized

systems. We believe that the third step of web application evolution will aim at interactive and collaborative systems.

In this paper we present the system called *Agora* which supports business-to-business negotiations through the web, aiming at concluding a business contract. The *Agora* roots are related to collaborative applications developed in the domain of so called *Computer Supported Collaborative Work* [1,2,3,4,5,6,7]. *Agora* is composed of two strictly interacting functional components. The first component is the virtual negotiation table, and the second component is the generator of collaboratively written documents. *Agora* provides negotiators with an arbitrary number of virtual tables. All negotiators around the table discuss and present their positions by exchanging electronic messages. Each negotiator at the table sees all the messages exchanged. A negotiator can be involved in several negotiations simultaneously, i.e. a negotiator can virtually sit at different tables. Negotiations at different tables may concern different topics, different aspects of the same topic, or the same topic discussed by different partners. For example, a negotiator can share a table with some business partners, and in parallel the negotiator can share another table with supervisors to report the progress of negotiations and to receive guiding instructions (cf. Fig. 1).

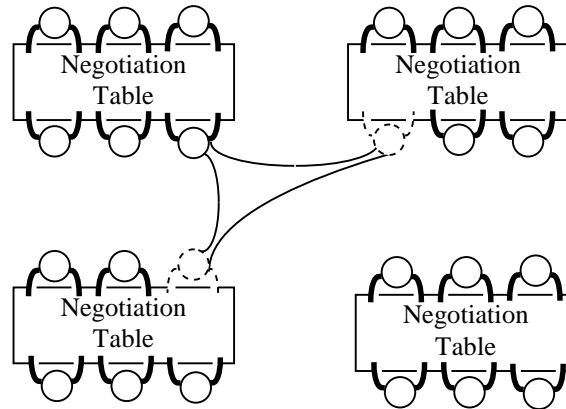


Fig. 1. Negotiations at different virtual tables

The part of *Agora* devoted to support collaborative document writing is required to prepare a final contract, which is a result of negotiations. The contract is a common document seen and accessible to all the negotiators. When a negotiator writes or modifies a paragraph of the contract and commits the changes, then it becomes instantaneously visible to other negotiators. Next, any negotiator can modify this paragraph. *Agora* provides versioning mechanisms which additionally facilitate collaborative writing [8, 9]. Historical versions of the document can be

stored in the system, which can be useful in the case of unsatisfactory results of document evolution, when a roll-back operation is required. Moreover, it is possible to conduct the document evolution simultaneously in different directions, thus preparing alternative document versions.

Agora has been implemented in the Java language and connected to the Oracle database management system through a Java Database Connectivity interface (JDBC) to provide persistency of both documents and negotiation history. The use of Java and JDBC provides *Agora* with platform independence, concerning hardware, operating systems, and database management system.

The remainder of the paper is organized as follows. In Section 2 *Agora* negotiation support is described. Section 3 is devoted to collaborative document writing. Implementation issues are discussed in Section 4. Finally, Section 5 is a summary of the paper.

2 Negotiation Support

Negotiations are supported by *Agora* through the concept of *virtual tables*. A virtual table is a place where distributed negotiators meet together through WWW and conduct negotiations by exchanging electronic messages. The aim of negotiations is to prepare a *contract* which is an electronic document. The manner of context preparation is described in the next section. A virtual table is created by the *Agora administrator* on demand by a *negotiation leader*. A negotiation leader is responsible for allocation of negotiators to a virtual table. A negotiator may be any *Agora* user, i.e., a person who is registered in the *Agora* repository of users by the *Agora* administrator. In Figure 2 a page of the *Agora* administration tool is presented, while Figure 3 provides a page of a virtual table leader.

As mentioned before, a message exchanged during negotiations appears at the virtual table and is readable to all the negotiators. The virtual table contains:

- the name of the negotiator who sent it,
- the date of message issue,
- a subject which abstracts the message content,
- the text of the message, and optionally
- a list of attachments that may be read by the use of tools such as text editors, spreadsheets, and browsers.

If in a message a part of another message is cited, each of its lines is preceded by ">". A page of a virtual table is presented in Figure 4.

Requests		Update	Info	Help	Exit
Requests					
picard cellary wiecz	Name:	cellary			
	Email:	cellary@kti.ae.poznan.pl			
	Date:	Mon Nov 15 12:25:0 1997			
	Type:	Open Conference			
		Remove			
Virtual Tables		Table Info			
Welcome IBM Negotiations Contract1 Contract2	Address:	IBM Negotiations			
	Subject:	Internal negotiations			
	Open Date:	Mon Nov 15 12:25:21 1997			
	Leader:	cellary			
		Add table	Add OK	Cancel Add	Remove table
Users		User Profile			
admin Tom Bugs picard cellary wiecz scott	Login:	wiecz			
	Name:	Waldemar Wieczerzycki			
	Email:	wiecz@kti.ae.poznan.pl			
	Password:				
		Add User	Add OK	Cancel Add	Remove User

Fig. 2. Agora Administration page

Negotiators	Tools	Requests	Close Table	Refresh	Info	Help	Exit
Requests							
picard wiecz scott	Name:	Scott Tiger					
	Email:	scott@nowhere					
	Date:	Mon Nov 15 12:29:39 1997					
	Type:	Subscribe					
		Remove					
Negotiators		Negotiator Profile					
scott wiecz picard	Login:	picard					
	Name:	Willy Picard					
	Email:	picard@kti.ae.poznan.pl					
		Add Negotiator	Add OK	Cancel Add	Remove Negotiator		
Tools							
<input type="checkbox"/> Document Generator							

Fig. 3. The page of a virtual table leader

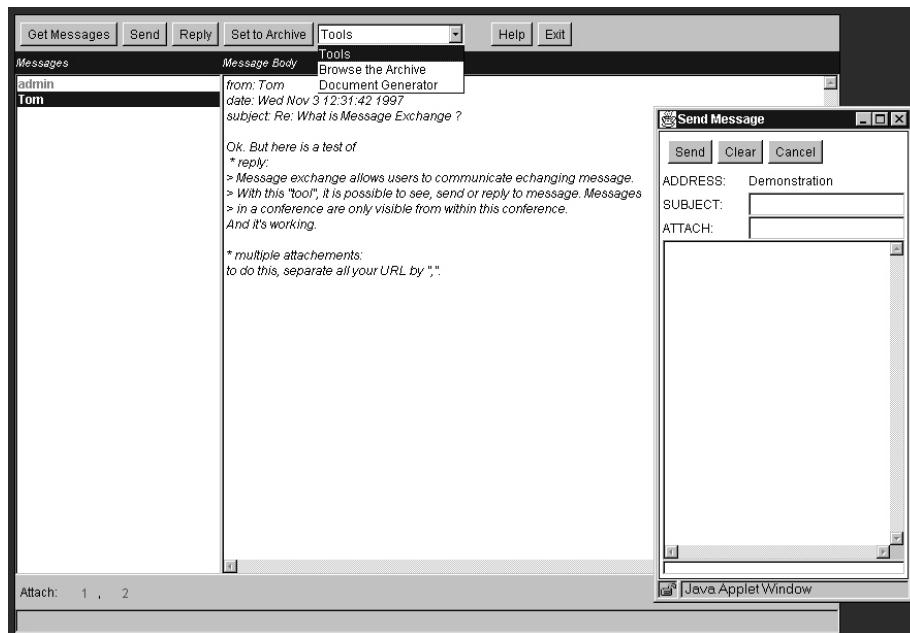


Fig. 4. The page of a virtual table

During the negotiations, the negotiation history is permanently available through scrolling up and down the table, so a negotiator may come back to any position from the past. After the negotiations, their history is stored in the *Agora* database for 30 days. This feature means that the history is still available to all the negotiators. Before this deadline, the negotiation leader may archive negotiation history, thus making it persistent in the *Agora* database.

3 Collaborative Document Writing

As mentioned in Section 1, the general aim of negotiations is to prepare a business contract which is an electronic document. *Agora* provides negotiators with an integrated tool for collaborative development of this document.

A document is defined as a linear set of *paragraphs*. A paragraph is an item terminated by a *New-line* character, e.g., a sentence, a header, a title, a mathematical formula, or a figure. Every paragraph is a terminal unit of document composition, which means that it can not be further decomposed. Thus, the layout attributes defined for the paragraph are common to all items it contains, e.g., words and characters. Though the structure of a document is flat and linear, it is visualized with respect to the hierarchy of document headers. *Agora* substantially facilitates the navigation through the document content (cf. Figure 5).

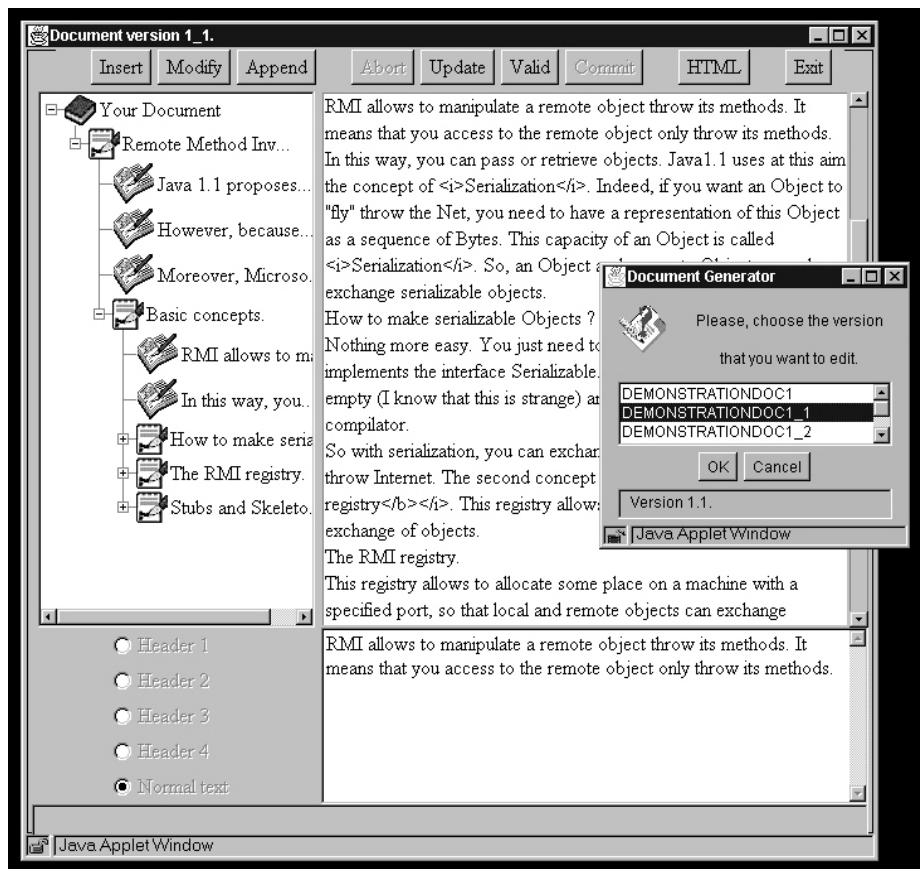


Fig. 5. Collaborative document writing

Collaborative writing is performed as follows. *Agora* keeps all the documents in the database which is accessible through the network, i.e., the database need not be installed on the same machine as the *Agora* server. Each document may exist in several versions which are uniquely identified. A copy of a document version maintained at the server is called *global*. If a negotiator intends to read or edit a version of the document, *Agora* prepares a local copy for the negotiator. Assuming a single paragraph has been selected, a negotiator may perform the following actions on it:

- modify the paragraph, in particular remove it from the document,
- append a new paragraph directly after the selected one,
- insert a new paragraph just before the selected one,
- validate changes made on the paragraph in a local copy of a document version.

Additionally, a negotiator may perform the following actions concerning the entire document version:

- commit changes to the global copy of a document version,
- abort all the changes done (i.e., all non-committed changes),
- refresh the local copy by re-reading the global copy of the document version concerned, thus visualizing the recent changes committed by other negotiators at the same virtual table.

When a negotiator starts to modify a paragraph of a document version, this fact is annotated in its global copy. Another negotiator who reads or refreshes the same version of this document finds this paragraph distinguished by color. The second negotiator is free to modify any other paragraph in parallel. However, if a negotiator needs to modify the same one a conflict arises. There are two possible resolutions of such a conflict which are provided by *Agora*. The first resolution consists of negotiations through the virtual table which aim to decide who will wait for whom with the modifications intended. The second resolution consists of deriving a new version of the document. Then, both negotiators can continue their work in parallel, creating two different versions of the same paragraph. Afterwards the negotiators can compare these two versions of a paragraph and either choose one of them or merge them into one consistent version.

4 Implementation issues

The *Agora* architecture is client-server based, as presented in Figure 6. Both the clients and the server are implemented as *Java* objects that communicate by the *Remote Method Invocation (RMI)*. The *Agora* server is connected to a database management system by a *Java-Database Connectivity (JDBC)* interface.

The structure of an *Agora* client and server is presented in Figure 7. The main part of a client is an *Agora Client Applet*, which operates on a *Java Virtual Machine (JVM)*. The applet is accessed through *Internet* by the use of a standard *WWW* browser. The main part of the server is an *Agora Server Kernel* that also operates on a *Java Virtual Machine*. The server is directly accessible through *Internet*. The server uses the *Java-Database Connectivity (JDBC)* interface to access a *Database Management System (DBMS)*, currently the *Oracle DBMS*.

The structure of an *Agora Client Applet* and *Agora Server Kernel* is presented in Figure 8. The *Agora Client Applet* is composed of the *Graphical User Interface* supported by the *Graphical Widget Library* and the *Remote Method Invocation* interface. The first two components allow for communication with a user, while the third component communicates with the *Agora Server Kernel*.

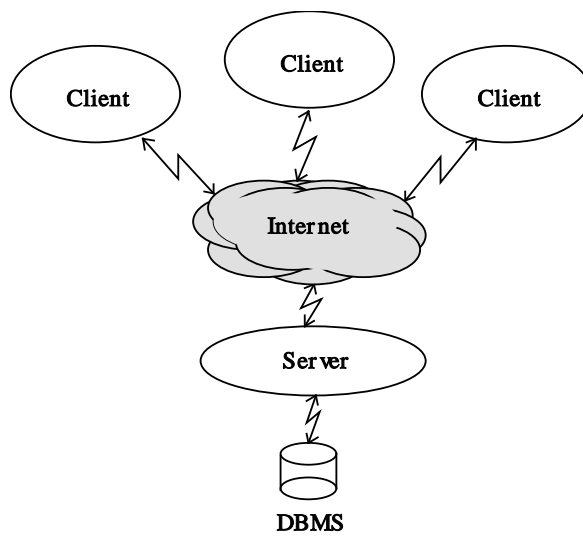


Fig. 6. *Agora architecture*

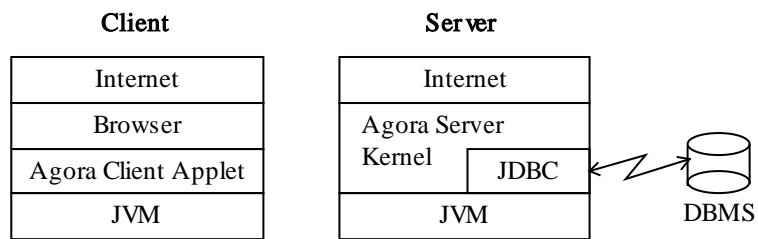


Fig. 7. *Agora client and server structure*

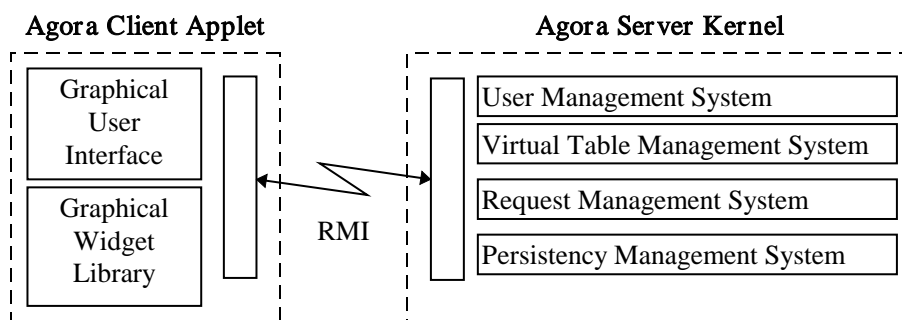


Fig. 8. *The structure of an Agora Client Applet and Agora Server Kernel*

The *Agora Server Kernel* is composed of the:

- *User Management System*, which is responsible for *Agora* user registration and removal, and maintenance of user profiles;
- *Virtual Table Management System*, which is responsible for table creation and removal, negotiator registration, message exchange, and collaboration during document writing;
- *Request Management System*, which is responsible for collecting and processing requests of user and virtual tables registration and removal;
- *Persistency Management System*, which is responsible for collaboration with the database management system to store and retrieve negotiation history and documents, as well as user and virtual table profiles;
- *Remote Method Invocation* interface, which is used to communicate with *Agora* clients.

Agora Server Kernel is built over the database management system. Currently the Oracle DBMS is used for implementation. The Oracle DBMS is used to provide persistency for all the data that are managed by *Agora*, namely, the data concerning users and negotiators, virtual tables, negotiation histories, and documents. The use of *JDBC* as an interface provides platform independence. Indeed, any database management system may be used to support *Agora* only if there is access through *JDBC*.

5 Summary

The *Agora* prototype system presented in this paper responds to the needs of business-to-business electronic commerce which will be the next step of the *Internet* and *WWW* business application evolution. *Agora* is implemented by the use of open system tools, which make it independent of hardware and operating system platform, as well as of browsers and database management systems. *Agora* is easy to use, secure, and reliable. The system is available to anybody who has access to the *Internet*. *Agora* supports negotiations and collaborative document writing, which are two basic functions required by business-to-business electronic commerce. Persistency of negotiation histories and documents provided by *Agora*, as well as versioning mechanisms provided to documents under preparation, offer interesting possibilities of post-processing, for example, to analyze negotiation strategies of different negotiators.

6 References

1. Crowley T., Milazzo P., Baker E., Forsdick H., and Tomlinson R., "MMConf: An Infrastructure for Building Shared Multimedia Applications", *Proc. of ACM Conference on Computer Supported Cooperative Work*, October 1990, pp. 329-342.
2. Ellis C.A., Gibbs S.J., and Rein G.L., "Groupware: Some Issues and Experiences", *CACM* 34:1 (January 1991), pp. 38-58.
3. Ensor J.R., Ahuja S.R., Horn D.N., and Lucco S.E., "The Rapport Multimedia Conferencing System: A Software Overview", *Proceedings of the 2nd IEEE Conference on Computer Workstations*, March 1988, pp. 52-58.
4. Garfinkel D., Welti B., and Yip T., "HP Shared X: A Tool for Real-Time Collaboration", *Hewlett-Packard Journal*, April 1994, pp. 23-24.
5. Hill R., Brinck T., Rohall S., Patterson J., and Wilner W., "The Rendezvous Architecture and Language for Constructing Multiuser Applications", *ACM Transactions on Computer Human Interaction* 1:2 (June 1994).
6. Sadowski M., Wiśniewski J., "Forum - Asynchronous Collaborative Educational Web Conferencing System", *Master Thesis*, EFP - The Franco-Polish School of New Information and Communication Technologies, 1997.
7. Stefik M., Foster G., Bobrow D.G., Kahn K., Lanning S., and Suchman L., "Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings", *CACM* 30:1 (January 1987), pp. 32-47.
8. Wiczerzycki W., "Modeling and Execution of Editing Processes in Collaborative Writing Applications Supported by a Database System", *3rd Int. Conf. on Next Generation of Information Technologies and Systems*, NGITS'97, Israel, June 1997.
9. Wiczerzycki W., "Versioning Technique for Collaborative Writing Tools", *Proc. of 7th International Conference and Workshop on Database and Expert Systems Applications DEXA'96*, IEEE Computer Society Press, Zurich, Switzerland, September 1996, pp. 463-468.