Designing a Generic Marketplace Architecture Using Multi-Agent Based Technology

Youcef Aklouf[1] and Habiba Drias[2]
[1]Research Laboratory in Artificial Intelligence, USTHB, Algeria
[2]National Institute of Computer Science, INI, Algeria

Abstract: *Several architectures were proposed in the literature for modeling the interactions between agents. This paper deals with the integration of the multi-agent system in the electronic commerce architecture. Within the framework of this work, we describe three architectures based on the Multi-Agents design. In these systems, buyer and seller agents interact in an environment, which is similar to an electronic market in order to sell and buy goods. We propose then, an electronic architecture based on intelligent and mobile agents named Virtual Electronic Marketplace based on Mobile Agents (VEMMA). We present the generic model of this architecture as well as an implementation using the Java language and the distributed object philosophy based on Remote Method Invocation (RMI) technology. The originality of our developed architecture is that it allow users to create several kinds of agents within a set of the most required parameters for product description on one hand, and supports a complete set of methods and functions used in communication between agents on the other hand. In addition, the system is scalable and allows any updating task without major modification of the initial system.*

Keywords: *Marketplace architecture, mobile agents, multi-agents systems, Java, RMI, electronic business.*

Received February 9, 2004; accepted May 272005

1. Introduction

The distributed character, dynamic about Internet speaks in favor of scalable, opened and flexible solutions, particularly for the development of applications such as the electronic commerce, the virtual companies and the co-operative information systems. The multi-agents systems meet these aims. These systems, characterized by various agents with different capacities, which interacts in order to resolve a problem, manage the evolution of the system, allow scalability, re-use of software and open systems [11, 15]. Our interest for the electronic Marketplaces is also presented in the form of direct applications of its mechanisms to the electronic procedures of selling and buying. Our focus in this paper deals mainly with a virtual Marketplace based on intelligent agents and particularly the agents that have the characteristic of mobility i.e. the mobile agents. Our work is to present a generic model of this type of market, this model try on one hand, to define and gather the most functionalities and services proposed by a virtual market and, in the other hand, to define the most products characteristics exchanged by the involved actors. An automatic negotiation between agents is supposed to be used [7].

This paper starts by a short description of three electronic commerce infrastructures named Kasbah, Minnesota AGent Marketplace Architecture (MAGMA) and MarketSpace. This part is finished by a discussion and a comparison of these three models. Then, section 3 provides an overview of the proposed architecture by giving its different parts, followed by a detailed description of its various components and its offers strategies between buyer and seller agents.

The implementation of the described architecture using Java and Remote Method Invocation (RMI) as communication protocol is presented. The several developed modules of the implemented platform are also outlined in section 4. Finally, a conclusion and some future works are also presented in the last section of this contribution.

2. Electronic Commerce Infrastructures

Three electronic commerce prototypes are presented in this section. They are structured around several kinds of agents. *Kasbah* is a Web site based on a multi-agent system where users create agents to negotiate in order to purchase and/or to sell goods. The second one, *MarketSpace* architecture, describes two models, one for the information exchanged and the second for the interaction between agents in the market. Finally, the *MAGMA* prototype proposes a general system structure of a virtual market based on various kinds of agents. This prototype contains the necessary elements required to setup an actual electronic commerce system. In the following, each architecture is described in more detail.

2.1. Kasbah

Kasbah provides a platform where users communicate, interact, buy and sell goods. This is carried out by creating a buyer and a seller agents who access to the market and interact. The marketplace which represents a commercial space is created to handle any type of agents that supports the appropriate protocol. The agents are not sufficiently intelligent although they are completely autonomous, they do not use any mmachine learning techniques.

Once, an agent is created and starts communication with other agents in the commercial space, it negotiates and makes decisions according to its initial information without resorting to user intervention. This last, has a high level control on the agent behavior. When the user creates a new selling agent for example, they give it a description of the item they want it to sell and set several parameters to guide it as it tries to sell the specified item. These parameters are:

- Desired date to sell the item.
- The desired price.
- Lowest acceptable price.

The user has some control over the agent's negotiation "strategy". The user can specify the "Decay" function which is used by the agent to lower the asking price after expiration of the fixed time. The user has three choices for this function: *Linear*, *quadratic* and *cubic*.

2.2. MarketSpace

MarketSpace is an infrastructure based on *agents and interactio*n paradigm. The basic idea of this architecture is to have and architecture in which information about services and products is readable both by humans and machines and where each participant can announce their own interest to other participants. Interaction has to make sense for both agents and humans, and will be easier to automate. It will not be centralized like today's marketplaces, but distributed like the World Wide Web [3].

The *MarketSpace* server has three main components:

- *The Kernel*: Handles events which starts the activities of the system and manages the communication with the external components.
- *The Protocol Handlers*: Register protocols to and get the corresponding events from the event handlers when data arrived.
- *The Agent Environment*: Implements the runtime environment for agents.

MarketSpace can be divided in two sub-systems: The events system and the stream and protocol system. The *event system* is a subscription and distribution central for events. It allows other system components to subscribe to events, add events to the queue, and register new event types.

The *stream and protocol system* generates the stream events and takes care of stream communication with the outside world. The main component is the stream handler that sets up a port through which it receives all incoming data.

2.3. MAGMA

MAGMA is a prototype of a virtual market. Currently Magma includes a multiple commerce agents and an advertising server. These agents are responsible of all activities realized in the market, such as purchasing and selling a goods, in addition to the negotiation between participants (agents). To facilitate the communication between agents, Magma uses a relay server that maintains all sockets connections and routes messages between agents based on unique agents names [13]. All agents in the current version of Magma support SQL by using Java DataBase Connectivity (JDBC) protocol. This would enable agents to interface with existing relational inventory databases, as well as banking and virtual catalog databases.

The several electronic commerce prototypes presented herein are structured around various kinds of agents. These agents are limited in setting up of an adequate strategy for selling and buying, due to the limited parameters used by agents. More developed and intelligent agents are required, they must have a sufficient number of parameters in order to have an efficient adjusting strategy of selling and buying. The lack of such standard infrastructure for electronic commerce due to the large kinds of trade and to the specificity of each participant explains the multiple existing plate-forms today on the Internet. Actually, the choice of an infrastructure is enormously related to the kind of trade that the participant is interested in.

3. The VEMMA Model

The Virtual Electronic MaketPlace based on Mobile Agent (VEMMA) system proposed herein aims to gives an value added to agents and try to make as possible the agents to be more intelligent and to use a suitable language facilitating their access to various marketplaces. Thus, allowing users to build and to use agents who follow exactly the specificity of each user requirements, suggests the use of an advanced communication language between agents [8, 9]. VEMMA Architecture includes four types of agents: buyer agents, seller agents, broker agents and administrator agent of the MarketPlace.

The administrator agent represents the main component of the market, it is the operator of the most important realized tasks, and the broker agent play a mediator role between buyer and seller, he search a correspondence between the buyer agents requirements and the seller agents offers. However, these agents are

operators of a second level, they manage a limited set of agents gathered in groups and categories. The agents in VEMMA are able to do automatically the products search step according to the characteristics introduced by the users. In addition, the agents can negotiate the prices/availability of products and obtain as possible an agreement between agents, this is done also in an automatic way [12, 14].

3.1. Specification of the Market Entities

The entities used in the VEMMA virtual Marketplace are the products, the administrator agent, the buyers, the sellers and the brokers agents. Each entity is defined in a formal way as a set of characteristics. Moreover, the basic activities managed by the Marketplace are defined as a set of functions used by the various entities. These activities gathers: The inscription, removing of the buyer and seller agents, adding a new products published by the seller agents and updating the products prices and availability, responding to the buyer agents for its several asking requests related to products, providing a selection mechanism for the seller agents based on some criteria and a buyer requests, and managing of the various activities of the brokers agents etc [5]. Formally, each entity of the system is represented as a record with a set of parameters. For example, the seller agent information will be presented by:

- *The Seller Identifier*: Associated to each seller. This identifier is used by the broker and the administrator agent in order to identify in a unique way, each agent of the system, in this case, the seller.
- *The Address*: Represents the seller contact address.
- *The Brokers List*: Contains a list of brokers with which the seller is registered.
- *The Products List*: Represent the list of products that the seller offers. This list contains separately for each product:
  - The link to the product description.
  - The asked selling price.
  - The reserved price (the lower price).
  - The offered quantity.
  - The desired date for selling (not to exceed).

The named parameters used in the implementation are described in as follows:

- *SellerId:* Seller identifier.
- *SbrokersList*: The list of brokers.
- *SAddr*: The seller address.
- *ProdOffer*: Represents the list of a products offered by the seller. The description of each product is similar to the product requested by the buyer.
- *ProductId*: A link to the product description.
- *SPriceD*: The desired price.
- *SpriceR*: The reserved price (the best price of the products).
- *SQty*: The offered quantity.
- *SDate*: Represents the date before which the seller wishes its products to be sold.

Figure 1 shows graphically the structure of the seller agent. The other agents are defined approximately in a similar way with some changes about parameters.

- **Identifier**
- Seller address
- A brokers list
- A products list
  - **Product Identifier**
  - Desired price
  - Reserved Price
  - Offered quantity
  - Selling Date

Figure 1. Representation of a seller agent.

3.2. The VEMMA System Activities

The basic activities handled by the VEMMA Marketplace are the registration of buyer, seller and broker agents, adding a news product offers, updating of products prices, responding to the buyer agents on the various requests related to the products and providing a selection mechanism for the sellers. The management of the various activities of the brokers' agents are also a part of the activities functions. When these activities occur, the broker must react to these events and execute the necessary processing using the available information in its local database. We defined a set of functions required to achieve such activities. The most important functions among these activities are listed in Table 1. More than 30 functions are shared between several agents involved in exchange.

3.3. A VEMMA Framework

Figure 2 presents the system entities as a diagram, each agent has some characteristics and supports several methods allowed by the system administrator, communication can be done without restriction between all agents.

3.4. General Representation of the System

The generic model described previously is implemented according to the VEMMA platform schema of Figure 3. It consists of an organizational model carried out by the system procedures. The use of several kinds of agents for structuring the platform and to get a scalable system is proposed with the goal of achieving a high level of flexibility. Actually, in the implementation point of view, VEMMA is a collection of Java packages that implements the administrator agent, several brokers agents, several seller and buyer agents, in addition to the basic libraries for messages

structure and communication.

The administrator agent is the first agent created by the system, it is a server, the administrator handle a list of brokers, each brokers is responsible of some products categories and maintain a list of products information, this list is used to inform buyer agents about the available offers and if the buyer is interested by such offer, the broker try to search using a searching algorithms [10] and contact the seller or the owner of this offer, by consulting its local database or the administrator database if the seller information contact are not available in the broker local database.

The broker gives a list of sellers which proposes this product and wait for the buyer selection decision, the buyer make a decision according to its preferences and its local parameters. Once, the seller agent is localized by the broker according to the buyer requirement, a negotiation between this last and the seller agent can be started, to do this, each agent creates a mobile agent (mobile Bagent, mobile Sagent in Figure 3) that migrate to the broker siteand participate in the negotiation in order to conclude a possible agreement. This several tasks are executed under administrator supervision.

Table 1. A subset of activities functions managed by VEMM.

Function Name	Definition
Add_buyer (BrokerId, BuyerId, Adm)	Add the buyer identified by BuyerId to the broker list who has BrokerId as identifier
Del_buyer (BrokerId, BuyerId, Adm)	Remove the buyer from the broker buyers list
Add_seller (BrokerId, SellerId, Adm)	Add a seller
Del_seller (BrokerId, BuyerId, Adm)	Remove a seller
Add_Broker (BrokerId, Adm)	Add a new broker to the system
Del_Broker (BrokerId, Adm)	Remove a broker
Ch_bAddr (BuyerId, Adr, Adm)	Update the buyer address
Add_prodInt (bId, pId, mk)	Add a product PId to the market mk
InformOffer (SellerId, Offer, Adm)	Announces a new offer (a new product) by the seller identified by SellerId.
Chg_BPrice (BuyerId, ProductId, BpriceD, Adm)	Update the price suggested by the buyer

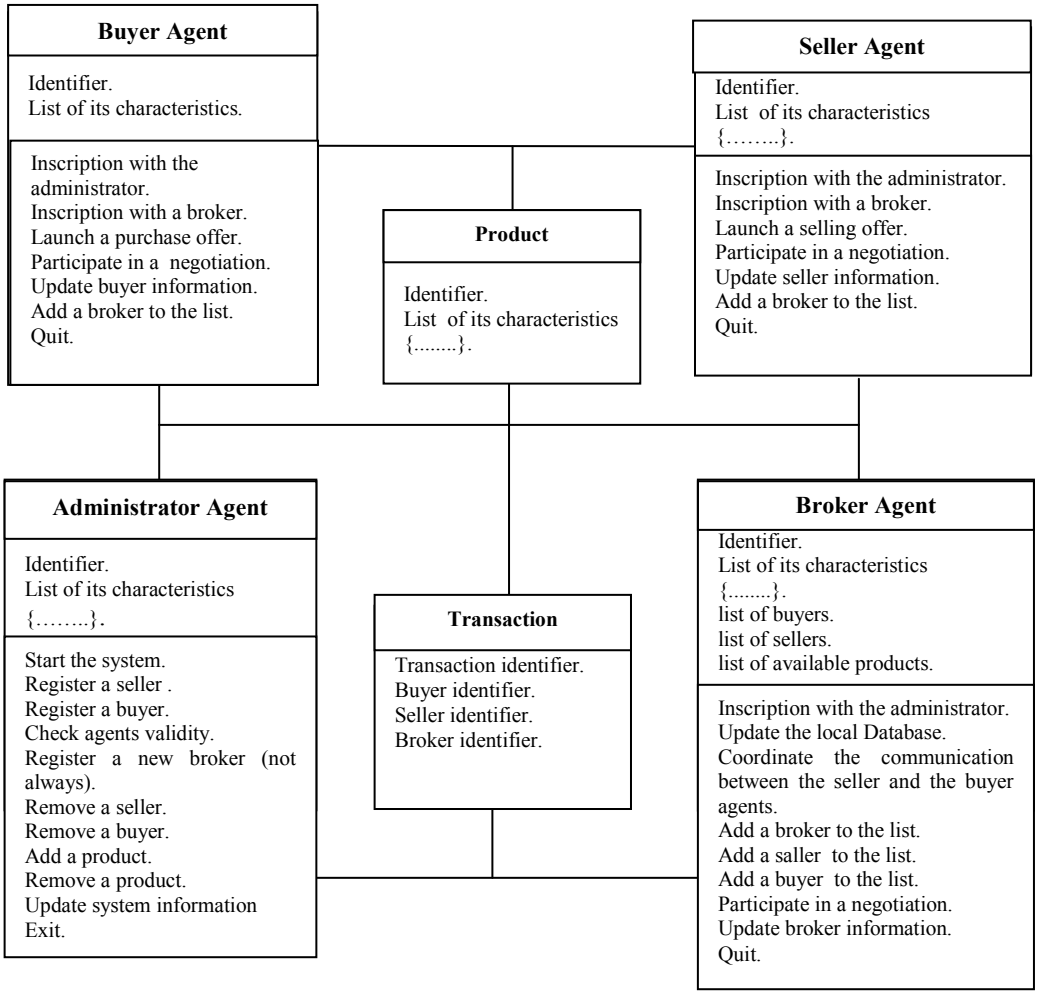

Figure 2. A VEMMA marketplace framework.

4. Implementation of the VEMMA System

The VEMMA developed architecture [4] is a distributed application implemented using Java language and RMI technology with JDBC-ODBC protocol to access databases. A visual version of Java language is used. This version is the version 9.0 of Jbuilder. This application is a platform which gathers the various entities defined in the VEMMA system, we found the administrator agent as the most important component, the seller and buyer agents created by users, a set of brokers agents according to some categories of products and finally the product component which is the unit of exchange in the communication. The system is scalable and allows evolution, due to these characteristics, a new functionalities can be easily added to the system.

4.1. Components Architecture of the Kernel

4.1.1. Administrator Agent (AdminAgent)

AdminAgent represents the application server. It includes mainly two classes, AdminAgent and AdminAgentFrame, the first classe is used for the application processing and the second, for data presentation. Both classes handles also some connections to the database via methods implemented through a common interface called AgentApi. AgentApi is a class that provides to the buyer and the seller agents syntactic methods for accessing data through administrator agent.

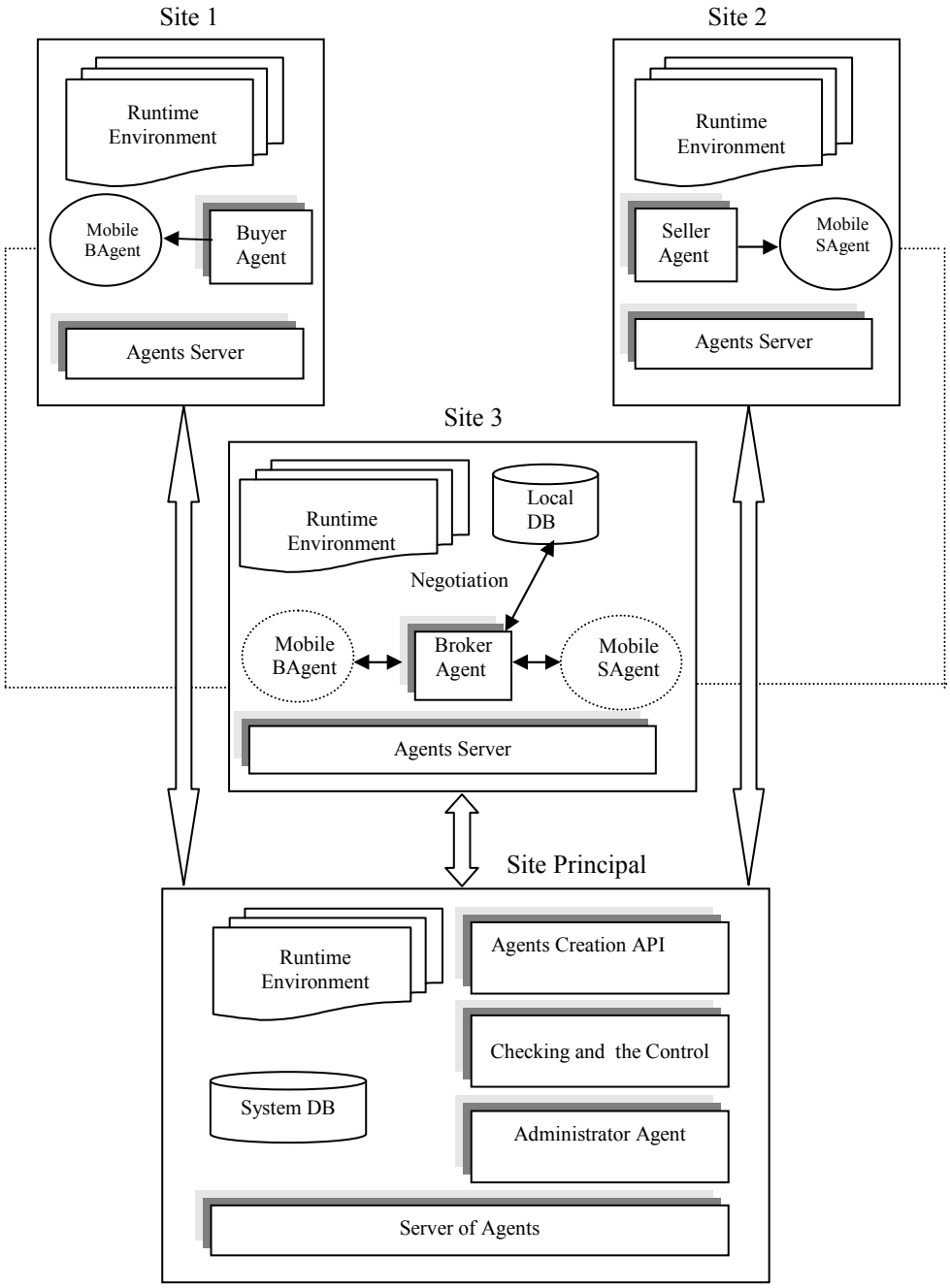

Figure 3. General architecture of VEMMA system.

4.1.2. Seller and Buyer Agents

Agent contains also two Java classes, the SellerAgent (BuyerAgent) class and the SellerFrame (BuyerFrame) class. The same technical method is used, the first class is used for data processing and executing all allowed methods such as searching, negotiating, etc. The second method is used for all graphical presentation of the selected data.

4.1.3. Database

The administrator agent manages a database which maintains all system information about entities created by the market in addition to the broker's information, products catalogs, etc. To access to this Database, a JDBC-ODBC connection is used, allowing an easy exploitation of a relational databases.

The main VEMMA interfaces are illustrated in Figure 4.

5. Conclusion

In this work, we proposed a generic model of a virtual Marketplace for selling and buying goods between consumers represented by software agents. This model can be used as a prototype. We have focused in modeling the behavior of the buyer and seller agents participating in the market as well as the brokers. In addition, our proposed approach has been designed as a Marketplace called VEMMA based on multi-agents system. We claim that multi-agents paradigm is the most adequate technology for automating the defined steps of the commercial process. The originality of our model lies in the fact that it allows the buyer and seller agents to participate in the negotiation by decreasing the traffic network, because these agents create a mobile agents that migrates to the broker site in order to negotiate about offers for eventual agreement. To validate this proposal, we have implemented a prototype that showed the easy communication and collaboration between VEMMA entities in an operational way.

For the future, we plan to extend our implementation prototype with a Web service technology and also to use an existing multi-agents platform for integrating more intelligent functionalities to agents.

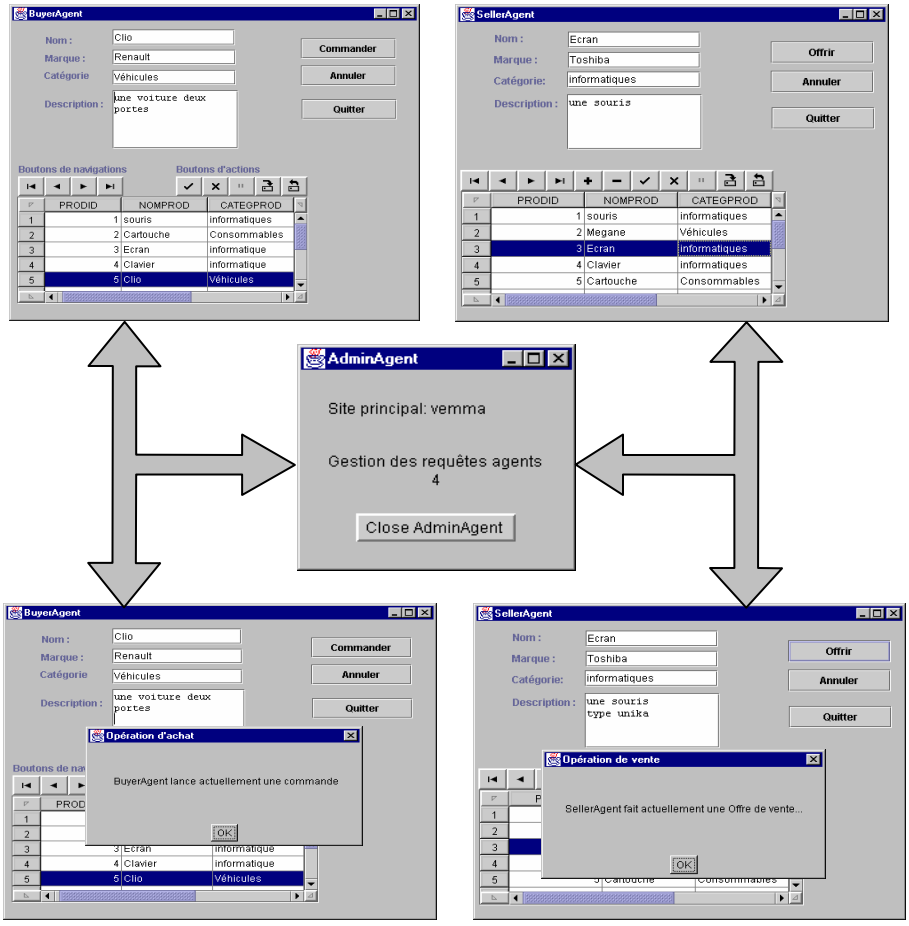

Figure 4. The kernel architecture.

References

[1] Aklouf Y. and Drias H., "Mobile Agent and Java Technology for Electronic Marketplace Architecture," *in Proceeding of International Arab Conference on Information Technology*, Mantouri University of Constatine, Algeria, December 2004.

[2] Chavez A. and Maes P., "Kasbah: An Agent Marketplace for Buying and Selling Goods," *MIT Lab Media*, 1996.

[3] Eriksson J., Finne N., and Janson S., "MarketSpace: An Open Agent Based Market Infrastructure," *UPMAIL Technical Carry-forward*, no. 147, Sweden 1997.

[4] Fu-ren L., and Kuang-yi C., "A Multiagent Framework for Automated Online Bargaining," *IEEE Intelligent Systems*, vol. 16, no. 4, 2001.

[5] Guttman R. and Maes P., "Agent Mediated Integrative Negotiation for Retail Electronic Commerce," *in Proceedings of the 2nd International Conference on Autonomous Agents*, 1998.

[6] Joanis M. and Gérin-Lajoie R., "Un Serveur de Négotiation Electronique, *Center inter Academics of Research in Analysis of Organizations (CIRANO)*, Montreal, July 1998.

[7] Kumar M. and Stuart I. F., "Negotiation on the Internet Auction," *IBM Research Division T. J. Watson Research Center Yorktown Height*, NY, 1999.

[8] Kurbel K. and Loutchko I., "Towards Multi-Agent Electronic Marketplaces: What is There and What Is Missing?" *The Knowledge Engineering Review*, vol. 18, no. 1, 2003.

[9] Labrou Y., "Semantics for an Agent Communication Language," *PhD Thesis*, University of Maryland, Baltimore, Maryland, 1996.

[10] Leung C. S., Sum J. P. F., Shen H., Wu J., and Young G., "Analysis and Design of an Agent Searching Algorithm for e-Marketplaces," *Cluster Computing,* vol. 7, no. 1, 2004.

[11] Nadri B. and Miller J., "Collaborative, Programmable Intelligent Agents," *Communication of the ACM*, vol. 41, no. 3, pp. 96-104, March 1998.

[12] Patra M. R. and Moore R., "A Formal Model of an Agent-Mediated Electronic Market," *The United Nations University (UNU/IIST)*, 2000.

[13] Tsvetovatyy M., Gini B., Mobasher Z., and Weickowski R., "MAGMA: An Agent–Based Virtual Market for Electronic Commerce," *Journal of Applied Artificial Intelligence*, vol. 11, no. 6, pp. 501-523, September 1997.

[14] White A. and Daniel E. M., "Electronic Marketplace to Marketplace Alliances: Emerging Trends and Strategic Rationales," *in Proceedings of the 5th international Conference on Electronic Commerce*, 2003.

[15] With M. and Troudi N., "NetSa: Une Architecture Multi Agent Pour la Recherche sur Internet," *Université Laval, Pavillon Pouliot, Ste-Foy*, G 1K 7 P4, Canada, March 1998.

[16] Wurman P., Wellman M., and Wash W., "The Michigan Internet AuctionBot," *in Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pp. 301-308, 1998.

Youcef Aklouf obtained his Engineering and MS degrees from the Computer Science Department of the University of Science and Technology of Algeria (USTHB) in 1998 and 2002, respectively. He is preparing his PhD at the University of Poitiers and is a member of the data engineering team of LISI/ENSMA. He is also a lecturer at Algeria USTHB University and a member of the Laboratory of Research in Artificial Intelligence. His research areas include e-commerce, web services, ontology, and multi agents systems.

Habiba Drias obtained her Master degree in computer science from Case Western Reserve University, Cleveland, Ohio, USA in 1984 and the PhD prepared at Paris6 University from USTHB, Algeria, in 1993. She has directed the computer science institute of USTHB and then the laboratory of research in artificial intelligence for many years. She has over 50 published papers in the domain of artificial intelligence, e-commerce, computational complexity, and the satisifiability problem. Currently, she is the principal of the Algerian National Institute of Computer Science (INI).