CONCEPTUAL ARCHITECTURE AND SAMPLE IMPLEMENTATION OF A MULTI-AGENT E-COMMERCE SYSTEM

Amalia Pîrvănescu[1], Costin Bădică[2], Maria Ganzha[3], Marcin Paprzycki[4]

[1]*SoftExpert SRL, Str.Vasile Conta, bl.U25, Craiova, 200404, Romania*
amaliap@soft-expert.com
[2]*University of Craiova, Software Engineering Dept, Bvd.Decebal 107, Craiova, 200440, Romania*
badica_costin@software.ucv.ro
[3] *Department of Informatics, Giżycko Private Higher Educational Institute, Giżycko, Poland*
ganzha@pwsz.net
[4] *Oklahoma State University, Computer Science Department, Tulsa, OK, 74106, USA*
and SWPS, Computer Science, ul. Chodakowska 19/31, 03-815 Warszawa, Poland
marcin@cs.okstate.edu

Abstract: Recent advances in software engineering, business process management and computational intelligence resulted in methods and techniques for developing advanced e-commerce applications as well as supporting automating e-commerce business processes. Despite this fact, up to now, the most successful e-commerce systems are still based on human input to make the most important decisions in various activities along the lifecycle of an e-business transaction. Our work aims at bridging the gap between these two scenarios (the possible and the real), by proposing a conceptual architecture of a multi-agent e-commerce system. The feasibility of our approach is demonstrated with a sample implementation, using a state-of-the-art agent platform – JADE.

Keywords: multi-agent systems, e-commerce, automated negotiations, distributed systems.

1. INTRODUCTION

Already long time ago the six main stages of consumer buying behavior have been conceptualized: need identification, product brokering, merchant brokering, negotiation, payment and delivery, service and evaluation (Howard and Sheth, 1969). Current e-commerce applications operate following the principle "to select and to accept choices." Thus users browse through catalogues of needed goods (tickets, films, books CD's and so on) and utilize them to make decisions. There exist systems, which support users during the product or/and merchant brokering stages of the buying process (for instance www.shopping.com). However, the most interesting part of product buying that is not properly supported within B2C environments, are automated negotiations (however, limited use of automated negotiations within B2B is known to exist for some time now, and apparently was partially responsible for some of the more spectacular crashes on the New York stock market in the late 1990th).

As described in the recent survey (Kowalczyk, *et al.*, 2002), multi-agent technology (involving intelligent mobile agents) should help facilitate e-commerce processes. Furthermore, it can be expected to have an important economical impact, by bringing efficiency to businesses (and thus improving their profitability), as well as benefiting individual users (e.g. by assuring "price-optimality" of purchases). In this context, multi-agent systems could be very useful in the negotiation stage of business transactions by assuring price fairness and reducing the negotiation time. Here, negotiation is a process by which group of agents communicate with another to try and come to a mutually acceptable agreement on some matter (Lomuscio, *et. al.*, 2001). Two components are very important for designing automatic negotiation systems – negotiation protocols and negotiation strategies. While a large amount of work has been devoted to study agent-negotiations, the current state of the art and practice is still rather unsatisfactory. It is thus obvious that realization of fully automatic negotiation **system** will demand a lot more work. Without pursuing this goal it will be impossible to achieve the vision of a globally distributed e-commerce environment supported by intelligent software agents.

This claim is further supported by the fact that it is almost impossible to point out to an existing large-scale implementation of an e-commerce agent system. While a number of possible reasons for this situation have been suggested (see, for instance, (Paprzycki and Abraham, 2003)), some of them have been recently dispelled. First, it was shown that modern agent frameworks (e.g. JADE) can easily scale up to 1500 agents and 300000 ACL messages (Chmiel, *et al.*, 2004b). Since these results have been obtained on a set of 8 antiquated Sun workstations with 198 Mbytes of RAM, it is easy to extrapolate the true scalability of JADE on modern computers and thus **it is possible to build and experiment with large-scale agent systems**. Second, recently new methods and techniques of software engineering, business process management and computational intelligence in support of the development of advanced e-commerce applications have been proposed. For example, we now have generic software frameworks for automated negotiation (Bartolini, *et al.*, 2002), Semantic Web support (ontologies and semantic Web services), at least in theory, for the full B2B e-commerce lifecycle (Trastour, *et al.*, 2002), and finally, multi-agent solutions for business process management (Jennings, *et al.*, 1996).

Therefore, we have set up a goal of developing, implementing and experimenting with a large-scale agent-based e-commerce system. Since this is a long-term undertaking, at this stage our focus is on creating a system with a multitude of agents that play variety of roles and interact with each other (system skeleton). Currently, we have combined a set of lightweight agents that are capable of adaptive behavior in context of price negotiations (by dynamically loading appropriate software modules (Paprzycki, *et al.*, 2004)) with a simplistic skeleton for an e-commerce simulation (Chmiel, *et al.*, 2004a), to create a unified e-commerce environment witch supports automatic negotiations in the case of one-to-many negotiation process and experimented with the system running on two networked computers (Pîrvănescu, *et al.*, 2005).

In the paper we, first, present the top-level description of the system. We follow by a summary of implementation-specific information as well as an example illustrating its work. We conclude with the research agenda of our team.

2. SYSTEM DESCRIPTION

Our e-commerce model extends and builds on the e-commerce structures presented in (Galant, *et al.*, 2002), (Chmiel, *et al.*, 2004a) and (Paprzycki, *et al.*, 2004). Basically, our environment acts as a distributed marketplace that hosts e-shops and allows e-clients to visit them and purchase products. Clients have the option to negotiate with the shops, to bid for products and to choose the shop from which to make a purchase. Furthermore, shops may be approached "instantly" by multiple clients and consequently, through auction-type mechanisms, have an option to choose the buyer. At this stage we assume that the price is the **only** factor determining the purchase and, furthermore, only shops are allowed to advertise their products (instead of clients being also able to specify their needs). Finally, only auction-type pricing mechanisms are implemented. These are serious restrictions and we plan to address them in the near future.

Shops and clients are created through a GUI interface that links users (buyers and sellers) with their *Personal* agents, though obviously, all interactions involving multiple *Personal* agents are simulated from a central interface (in the future it will be possible to utilize our system as a "game-style" environment involving actual human users). Our environment supports dynamic agent creation and destruction and agent migration utilized in order for them to engage in negotiations. The top-level conceptual architecture of the system illustrating proposed types of agents and their interactions in a particular configuration is shown in Figure 1 (we have omitted *Personal* agents assuming that their role is obvious, as they were responsible for creating *Shop* and *Client* agents to initialize the system). Let us now describe functionalities of each agent appearing in that figure (as well as *Personal* agents).

Personal agents facilitate communication between the system and the "real-world" users (*shoppers* and *merchants*). A shopper employs its *Personal* agent to communicate to the system his sought after product(s) and possibly buying policies (currently price minimization is the only available decision making strategy). The *Personal* agent creates *Client* agents to act within the marketplace on his behalf, one *Client* agent for each product. A merchant utilizes her *Personal* agent to create a *Shop* agent, responsible for advertising and selling her products within the marketplace. After being created, both *Shop* and *Client* agents register with the *CIC* agent (a gateway to the marketplace). Agents representing "returning" users receive their existing IDs, thus supporting the future goal of agent behavior adaptability (agents in the system will be able to recognize and adjust their behavior depending on if they interact with agents representing "returning" or "new" users).

There is only one *Client Information Center* (*CIC*) agent in the system (in the future we may need to address this potential bottleneck (Chmiel, *et al.*, 2004b)) that is responsible for storing, managing and providing information about all "participants." To be able to participate in the marketplace all *Shop* and *Client* agents must register with the *CIC* agent, which stores information in the *Client Information Database* (*CICDB*). The *CICDB* combines the function of *white pages*, by storing information (including unique IDs) about all *Client* and *Shop* agents, and of *yellow pages*, by storing information about the advertised products. Thus *Client* agents communicate with the *CIC* agent to discover available stores at any given time.

Every shopper request triggers creation of a single *Client* agent for each requested product. These agents are responsible for managing stages 2-5 of the "Howard and Sheth buying process." To achieve this goal, every *Client* agent communicates with the *CIC* to find which stores sell the needed product and creates an appropriate number of "slave" *Negotiation* agents with the "buyer role" (*Buyer* agents hereafter). One *Buyer* agent is created for each store identified by the *CIC*. On the supply side, a single *Shop* agent is created for each merchant existing in the system. *Shop* agents create slave *Negotiation* agents with the "seller role" (*Seller* agents hereafter) for each product available in the e-store.

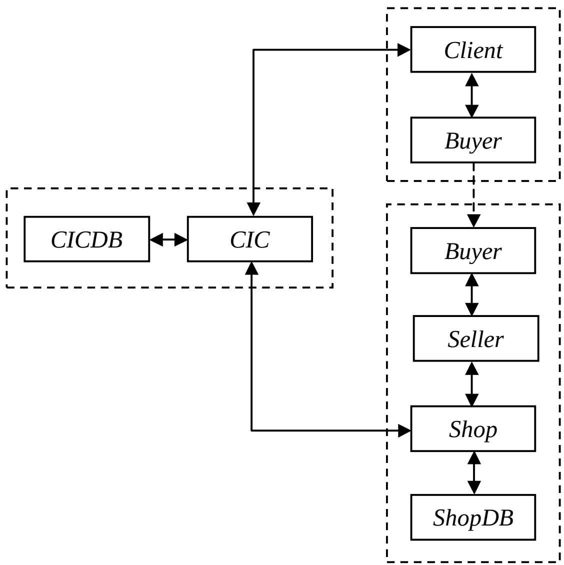

Fig. 1. The conceptual architecture of the system

Finally, *Database* agents are responsible with database updates and queries. There is one database agent per each database in the system. Thus, acting in what is a typical agent usage scenario, we use agents to decouple database management activities from the rest of the system. Currently, there are two databases in the system: a single *CICDB* (operated by the *CICDB* agent) containing the information about clients, shops and product they sell, and a single *Shop Database* (*ShopDB*, operated by the *ShopDB* agent) storing information about sales and available supplies (in the future we will replace this database with separate DBs for each *Shop* in the system).

At this stage of the development, the central part of the system operation is comprised by price negotiations. *Buyer* agents negotiate price with *Seller* agents. For this purpose, *Buyer* agents migrate to the e-stores known by the *CIC* agent to carry sought after commodity. In case of multiple *Buyer* agents attempting at purchasing the same item, they compete in an auction. Results of price negotiations are send to the *Client* agent that decides where to make a purchase. Note that the system is fully asynchronous and thus an attempt at making a purchase does not have to result in a success as by the time the offer is made other *Buyer* agents may have already purchased the last available item.

2. SYSTEM DESCRIPTION

The current implementation of the environment has been made utilizing JADE 3.3 agent platform (JADE). The main reasons for this selection are: JADE is one of the best modern agent environments. it is open-source, it is FIPA compliant, it runs on a variety of operating, and in (Chmiel, *et al.*, 2004b) we have observed its very good scalability.

3.1. System Implementation Using JADE

JADE architecture, consisting of a platform that specifies the platform within which agents "live" and containers, where agents "reside," matches well with our requirements. Negotiations between *Seller* and *Buyer* agents take place inside of JADE containers. There is one *Main* container that hosts the *CIC* agent. Users (customers and merchants) can create as many containers they need to hold their *Client* and *Shop* agents (e.g. one container for each e-store). *Buyer* agents created by the *Client* agents use JADE mobile agent technology to migrate to the *Shop* agent containers to engage in negotiations.

Figure 2 presents a mapping of our conceptual architecture from Figure 1 onto JADE. In particular, this diagram shows two machines running *Personal*, *Shop*, *Client*, *Buyer* and *Seller* agents, highlighting also JADE containers involved (*Main* container in the upper half of the figure, and *Container-1* container in the lower half, separated by the horizontal dotted line). Here, continuous lines denote agent creation. *Personal* agent P_1 represents user u_1

(shopper) and creates two *Client* agents: C_{11} and C_{12} to purchase items p_1 and p_2 respectively. Single-arrow dashed lines denote agent migration. *Buyer* agent B_{111} migrates from the *Main* container to the *Container-1* container. Double-arrow continuous lines denote negotiations. *Seller* agent S_{11} negotiates with *Buyer* agents B_{111} and B_{211} for product p_1. The sample scenario from Figure 2 is discussed in more detail in section 3.2. Further information about our implementation (i.e. information about Java classes utilized) can be found in (Pîrvănescu, *et al.*, 2005).

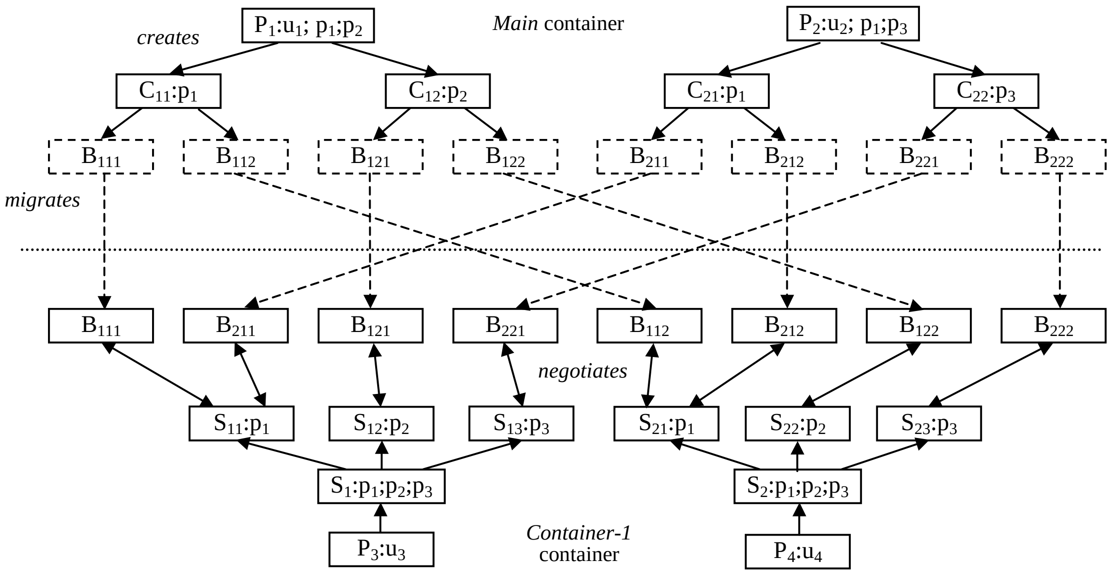

Fig. 2. Mapping the conceptual architecture of the system to JADE

3.2. Running the System

Here, we describe two simple experiments to illustrate main features of our implementation.

For the first experiment, we set up JADE on 2 computers. On the first computer, the *Main* container is initialized. On the second computer, a second container *Container-1* that is linked with the *Main* container on the first computer was started. On both computers we have set-up MySQL database. Both the *CIC* and the *CICDB* agents are created by default within the *Main* container, while the *ShopDB* and the *ShopDB* agent are instantiated in *Container-1*.

In this experiment we have chosen a simple scenario with 2 merchants – u_3 and u_4 and 2 customers – u_1 and u_2. Customer u_1 is requesting products p_1 and p_2, and customer u_2 is requesting products p_1 and p_3. We assume that both customers are seeking a common product – $p_{1,}$ in order to enable price competition. Both merchants u_3 and u_4 are advertising 3 products: p_1, p_2 and p_3, and are thus competing on selling them. Figure 2 illustrates this scenario.

Customers and merchants used *Personal* agents to create *Client* and *Shop* agents. In this experiment merchants used *Personal* agents running in *Container-1* container to create two *Shop* agents (Figure 3, upper left panel) and customers used *Personal* agents running in *Main* container to create two *Client* agents (Figure 3, upper right panel).

The process of starting *Shop* agents involved their registration with the *CIC* agent. Hereafter, for each product offered, a *Seller* agent was created in *Container-1*, finally resulting in 6 *Seller* agents being created. Similarly, starting *Client* agents involves their registration with the *CIC* agent, followed by the "search" of *Shop* agents that sell sought products and creation of a *Buyer* agent for every *Shop* agent found. So, finally, 4 *Buyer* agents were created (4 *Client* agents send 2 *Buyer* agents each to 2 e-stores).

At this stage, *Buyer* agents move to *Container-1* and register with appropriate *Shop* agents. As a result of message exchanges (Figure 3, bottom panels) negotiation protocol is identified and negotiation modules loaded by *Buyer* agents. Next, *Buyer* agents subscribe with *Seller* agents that sell sought products. *Seller* agents react to a timer that periodically triggers start of auctions with subscribed *Buyer* agents (an English auction in this experiment). Thus we have 6 auctions – 2 for selling each product p_1, p_2 and p_3. Note that because both customers u_1 and u_2 are requesting product p_1, *Buyer* agents B_{111} and B_{211}, and respectively B_{112} and B_{212} are competing for buying p_1 from *Seller* agents S_{11} and respectively S_{21}.

When negotiations end, *Shop* agents pass their result to *Client* agents. The *Client* agent collects all results and decides where from to buy the sought product, informing the *Shop* agent accordingly.

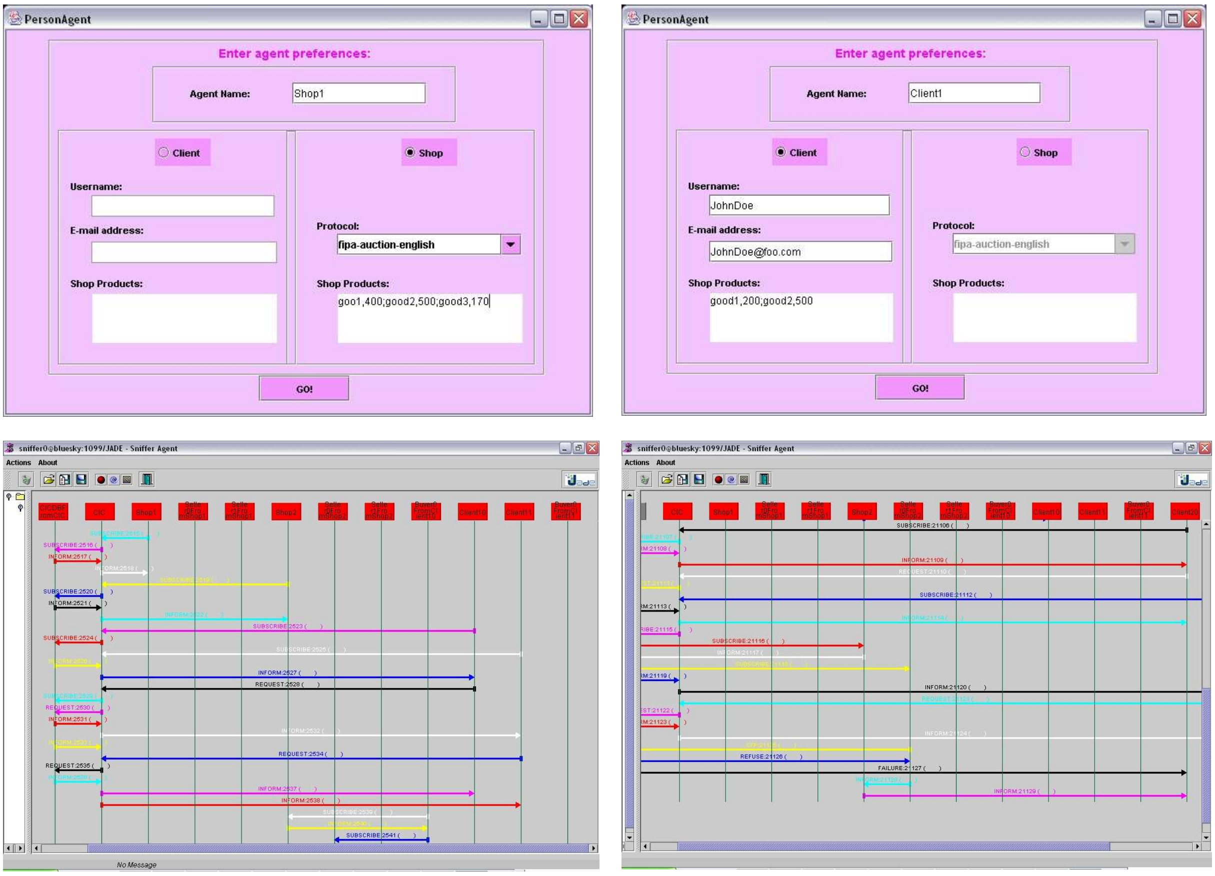

Fig. 3. Screen captures showing our system running on 2 computers

Figure 3 (bottom panels) presents message exchanges captured in the experiment using a *sniffer* agent. Figure 3 (left bottom panel) shows: i) *Shop* and *Client* agents subscribing to *CIC* agent; ii) *Client* agents asking *CIC* agent where to find out a specific product; iii) *Buyer* agents subscribing to *Seller* agents for negotiation; iv) the start of a negotiation when a *Seller* agent issues a call-for-proposal request to a *Buyer* agent. Figure 3 (right bottom panel, the last two message exchanges) shows: i) a *Seller* agent informing its “master” *Shop* agent about the result of a negotiation; ii) *Shop* agent informing *Client* about the possibility of establishing an agreement for transacting the sought product requested by *Client* agent. The process continues from here with *Client* eventually receiving other proposals and then deciding where to make the purchase.

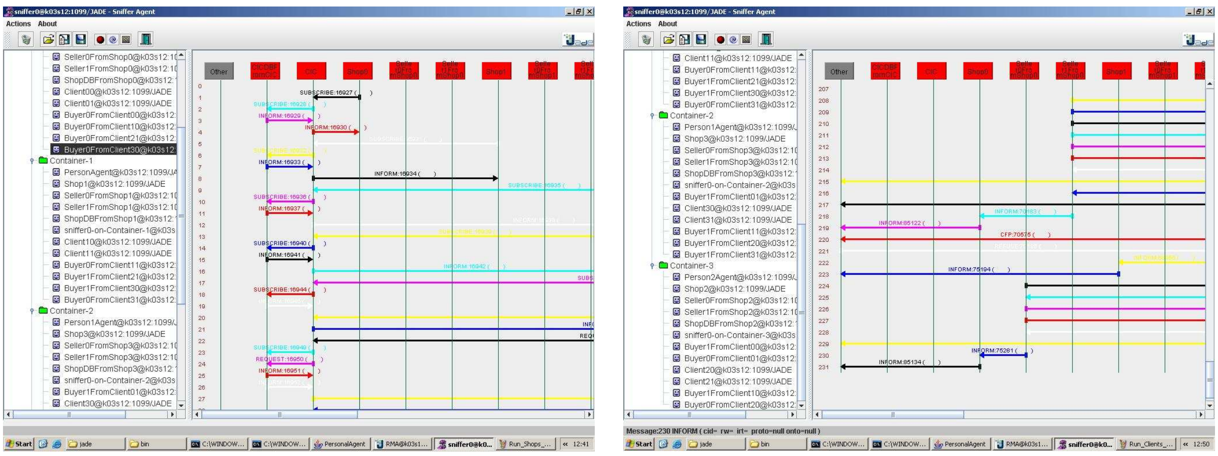

Fig. 4. Screen captures showing our system running on 4 computers

In the second experiment we ran our system on four computers. Here, the “user oriented activities” and the *CIC* were located on one computer, while the shops have been created on all 4 machines. Finally, 4 containers have been instantiated: *Main*, *Container-1*, *Container-2*, *Container-3*. So, we have increased the number of e-shops to 4 and the total number of *Buyer* agents to 16 (4 *Buyer* agents visiting each e-store, one e-store per container, one container per machine). Running of the system is illustrated in Figure 4, where in the left panel we can see the initial state of the system – the first lines of agent interactions as reported by the *sniffer* agent. Furthermore, we can see list of all agents residing at this moment in the *Container-1* container as well as partial lists of agents residing at this moment in the *Main* container and in the *Container-2* container. In the right panel we can see the list of all agents residing in the *Container-2* container and partial lists of agents populating the *Container-1* and *Container-3* containers as well as the final stages of system work (lines 207-231) of the report generated by the *sniffer* agent. Here, for instance, line 230 indicates that an *inform*-type message has been sent from the *Seller* agent in shop labeled *Shop0* to the *Shop* agent residing in this shop (when clicking on that arrow one would be able to find the message content).

4. CONCLUSIONS

In this paper we have presented basic features of an e-commerce modeling agent system that we are in the process of developing. We have also illustrated its work on 2 and 4 computers with a number of agents migrating to participate in auctions aimed at purchasing sought after products. At this stage capabilities of our system are rather limited, but we have already considered some future research directions: adding other factors that determine the purchase (speed of delivery, trust, history of involvement with a given merchant), letting customers advertise their needs, adding additional pricing mechanisms (fixed price, fixed-price with discount, iterative bargaining), studying system scalability and identifying performance bottlenecks, extending the ontological support beyond that very simple ontology of customer preferences, and adding more realistic strategy modules.

REFERENCES

Bartolini, C., Preist, C., Jennings, N.R. (2002). Architecting for Reuse: A Software Framework for Automated Negotiation. In: *Proc. 3rd Int. Workshop on Agent-Oriented Software Engineering*, Bologna, Italy, LNCS 2585, pp. 88-100, Springer Verlag.

Chmiel, K., Czech, D., Paprzycki, M. (2004a). Agent Technology in Modelling E-Commerce Processes; Sample Implementation. In: C.Danilowicz (ed.), *Multimedia and Network Information Systems*, vol.2, pp. 13-22, Wrocław University of Technology Press, Poland.

Chmiel, K., Tomiak, D., Gawinecki, M., Karczmarek, P., Szymczak, Paprzycki, M. (2004b). Testing the Efficiency of JADE Agent Platform. In: *Proc.3rd International Symposium on Parallel and Distributed Computing*, Cork, Ireland, pp. 49-57, IEEE Computer Society Press, Los Alamitos, CA, USA.

FIPA. The foundation for intelligent physical agents. See `http://www.fipa.org.`

Galant, V., Jakubczyc, J., Paprzycki, M. (2002). Infrastructure for E-Commerce. In: *Proc.10th Conference on Knowledge Extraction from Databases*, pp. 32-47, Wrocław University of Economics Press, Poland.

Howard, J., Sheth, J. (1969). *The theory of buyer behavior*, Wiley.

JADE. Java Agent Development Framework. See `http://jade.cselt.it.`

Jennings, N.R., Faratin, P., Johnson, M.J., Norman, T.J., O'Brien, P., Wiegand, M.E. (1996). Agent-Based Business Process management. In: *International Journal of Cooperative Information Systems*, 5(2&3), pp. 105-130.

Kowalczyk, R., Ulieru, M., Unland, R. (2002). Integrating Mobile and Intelligent Agents in Advanced E-commerce: A Survey. In: *Agent Technologies, Infrastructures, Tools, and Applications for E-Services, Proc.NODe'2002 Agent-Related Workshops*, Erfurt, Germany, LNAI 2592, pp. 295-313, Springer-Verlag.

Lomuscio, A.R., Wooldridge, M., Jennings, N.R. (2001). A classification scheme for negotiation in electronic commerce. In: F. Dignum, C. Sierra (Eds.): *Agent Mediated Electronic Commerce: The European AgentLink Perspective*, LNCS 1991, pp.19-33.

Paprzycki, M., Abraham, A. (2003). Agent Systems Today; Methodological Considerations. In: *Proceedings of 2003 International Conference on Management of e-Commerce and e-Government*, Nanchang, China, pp. 416-421, Jangxi Science and Technology Press, China.

Paprzycki, M., Abraham, A.. Pîrvănescu, A., Bădică, C. (2004). Implementing Agents Capable of Dynamic Negotiations. In: D. Petcu, V. Negru (eds.) *Proc.SYNASC04: Symbolic and Numeric Algorithms for Scientific Computing*, Timişoara, Romania, Mirton Press, pp. 369-380, Romania.

Pîrvănescu, A., Bădică, C., Paprzycki, M. (2005). Developing a JADE-based Multi-Agent E-Commerce Environment. In: N.Guimarães and P.Isaías (eds.) *Proc.IADIS AC'05: International Conference on Applied Computing*, Algarve, Portugal, pp.425-432, IADIS Press, Lisbon.

Trastour, D., Bartolini, C., Preist, C. (2002). Semantic Web Support for the Business-to-Business E-Commerce Lifecycle. In: *Proc.WWW'02: International World Wide Web Conference*, Hawaii, USA, pp. 89-98, ACM Press, New York, USA.