ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 5 Issue 09 September 2016 Page No.18019-18024

Auction System for Automated E-Commerce: JADE based Multi-agent Application

Aakanksha Gupta1, Durgesh Kumar Srivastava[2],Saket Jain[3]
[1]Student,BRCM College of Engineering and Technology Bahal
[2]Assistant Professor, BRCM College of Engineering and Technology Bahal
[3]Senior Engineer, Business Support System Ericsson Global Pvt. Ltd Gurgaon

Abstract--With the advancements in World Wide Web and high speed internet the scenario of traditional shopping system has changed completely and e-shopping has gained popularity as it furnishes human with a feeling of easiness. But this e-shopping system is also being badly off for the involvement of human efforts and time to a great extent. To minimize this human intervention automation in e-commerce is like a blessing and use of agent technology is the best way to achieve this. In this context use of negotiation strategies and protocols mentioned in literature in developing systems which automates the whole e-shopping system is required. And so, here in this paper we present a multi-agent application using JADE, based on real world e-commerce; it automates an English auction by applying Kasbah negotiation strategies in order to minimize human intervention and ensures that the product is being sold at its best price.
Keywords--multi-agent system, e-commerce, jade, kasbah strategy, negotiation, english auction, agent technology, software agent.

I. INTRODUCTION

A marketplace is a term coined to specify a common place where buyers and sellers meet and negotiate to sell/purchase one or more goods/products. Traditionally, all marketplaces exist physically where sellers need to go to sell goods and buyers need to go there to purchase whatever they want to have. Sellers need to advertise their products via placards, announcements, visuals and by word of mouth. There was no other way to tell a customer about their products. Similarly buyers need to travel and visit various shops in search for required products. This approach was time consuming and requires a lot of manual effort.

Slowly and gradually with the advancement in internet technologies, this scenario has changed completely and Marketplace turned to E-Marketplace where buyers and sellers can meet each other at a single place without moving physically from their respective places. This approach makes the work of buyer as well as seller very easy, to be completed in less time and with less cost. This is a big advancement, but it is sad to say that still, at almost all stages of buyer to seller or seller to buyer processing, humans are involved, like buyer is searching the required product, comparing on different sites then finalizing and in the end confirming the order and in making transactions. This drawback has thrown light on a new direction of research that is *Automation in E-commerce*, and use of agents or agent technology is the best way for employing automation in e-commerce.

Software Agents are programs to which one can delegate (aspects of) a task [1]. For automating e-commerce usage of agent technology was recognized long years ago, experiments were done other by researchers like Maria Ganzha, Marcin Paprzycki, Chmiel, Costin Badica[2][3][4][5] and in MIT Media Labs [6][7][8][9][10]. Initiatives taken by MIT labs in the direction of using software agents in implementing large scale multi agent E-Marketplace have done a lot of work and the very first generated system was *Kasbah* [8]. Kasbah helped human user in selecting the desired product allows them to create the agents. Kasbah has proposed some price negotiation strategies using which agent negotiate with each other and finds the best deal. In series of experimentation with Kasbah MIT conducted one full day experiment [9] and also proposed a new approach PDA@Shop [10].

Among six stages of Consumer Buying Behavior model which says a lot about buying behavior of a consumer either in commerce or e-commerce, negotiation is one of the most important stage, which ensures that both buyer as well as the seller should be satisfied from the deal while buying or selling any product [11]. For automating e-commerce we have to focus on this negotiation part. In our previous paper we have specified a plan to develop a system by combining Kasbah negotiation strategies along with FIPA negotiation protocols [12] in order to automate e-commerce and to reduce human efforts and time [13]. So here in our work we are implementing English auction by combining Kasbah negotiation strategies and auction a product in e-market and for this we are using JADE platform.

Slowly, with the advancement in the agent technology, various agent development frameworks/ platforms emerged [14]. These frameworks helped a lot to developers in developing multi-agent systems. These are powerful, handy and easy to program with. Many platforms came into picture such as Concordia, Voyager, Odyssey, Aglet and so on. Among these, the most powerful platform that we found, according to a research done to test efficiency [15] is JADE (Java Agent Development Framework) [16] is open source and FIPA compliant [17].

DOI: 10.18535/ijecs/v5i9.39

II. SYSTEM DESCRIPTION

Here in our work we are implementing a multi-agent application shows how sellers and buyers communicate, negotiate and buys and sells product in e-market and how agents are simplifying their tasks by automating the whole shopping process.

A. System Architecture

In our work we have covered a very specific area of e-commerce where there is an e-marketplace having buyers or clients buying some products and sellers selling some product/goods/items. Seller here is an auctioneer who is the host of an auction. Buyers who want to buy the specific product auctioned by the seller become participants of auction. Auctioneer or seller starts auction, bidding is done and finally one of the buyer among all the buyers buys the product by offering the highest bid and becomes the winner of the auction.

Keeping this picture in mind we have created two types of agents, these are: the *Buying Agent* responsible for buying product on buyer's behalf and the *Selling Agent* responsible for selling product on behalf of seller. Since in an auction there is a single seller and a number of buyers so we have to create one *Selling agent* and two or more *Buying agent* which are created through a GUI. The conceptual model of the proposed scenario is shown in Figure 1 which shows the interaction among the agents.

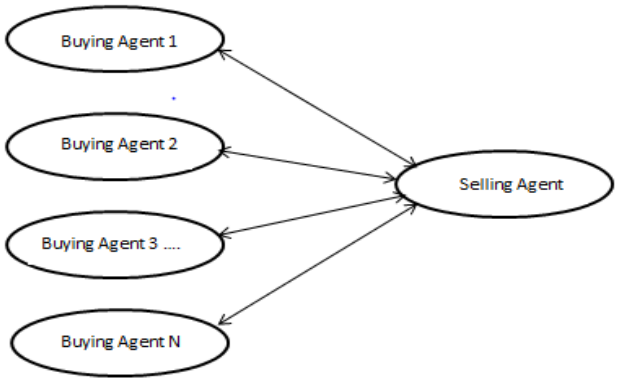

Fig. 1 Conceptual Model of our System

Let's have a look on each agent's description and functionalities respectively.

A *Buying agent* is created by buyer using GUI with desired details. Once created, on behalf of buyers these agents roam in the market place and try to find appropriate seller, here in our work it will find an auctioneer, and participates in the auction. Bids are offered, *Buying agent* either accepts or rejects it by checking buyer's pre-decided price value which the product worth, accordingly agent decides to quit or continue further in the auction and informs the real buyer and also if this agent accepts the highest bid it becomes the winner of the auction and buys the product.

On the other side, the seller is responsible for creating *Selling agent* using GUI by specifying desired details. The *Selling agent* acts as an auctioneer here and starts auction of the product it wants to sell. In marketplace it search for the buyers who are willing to buy the product the seller is selling and invites them to participate in the auction. Auction starts and this agent offer bids and wait for the acceptance of participants, if any of the participants has accepted the bid *Selling agent* raises its bid value by a particular amount which is calculated by negotiation strategy the seller has chosen while creating *Selling agent*. Again *Selling agent* offers bid and whole process repeats and finally auction ends either with a success or failure. Auction fails or terminate if none of the participant is ready to accept the minimum price bid, the price which seller has decided below which it will not sell its product else auction successfully ends by selling product to the buyer who has accepted the maximum bid where others quit.

B. System Operation and Interaction

The system starts with the creation of agents, the *Buying Agent* and the *Selling Agent*. Once created their GUI comes up where desired details are specified like for *Selling agent* it obtains name of the product to be auctioned, a reserve price which is the minimum price to sell product, a maximum price which seller expects that highest bid should cross, a price for the first bid to start auction, negotiation strategy through which value of next bid is begin calculated and selling time in which *Selling agent* has to sell the product. Similarly for *Buying agent* name of the product which it has to buy and a reserve price, here it is the maximum price which buyer can pay to buy the product. On getting all these details, the *Buying agent* and the *Selling agent* start searching each other eagerly in the market place. Figure 2 and 3 illustrates the whole process.

1. As stated above, a *Buying* agent search for appropriate seller in market place (see Figure 2). If it finds appropriate seller it sends request to *Selling agent* and participates in the auction.
2. The *Selling agent* too after getting details approaches one or more *Buying agents* and invites them to participate in the auction (see Figure 3).As the *Selling agent* gets one or more participants it starts the auction and offers bid to buyers.
3. The *Buying agent* either accepts or reject bids by checking the maximum price it can pay to buy product, if rejects it quits the auction and keeps on searching another seller else it accepts the proposal of the *Selling agent* (see Figure 2).

DOI: 10.18535/ijecs/v5i9.39

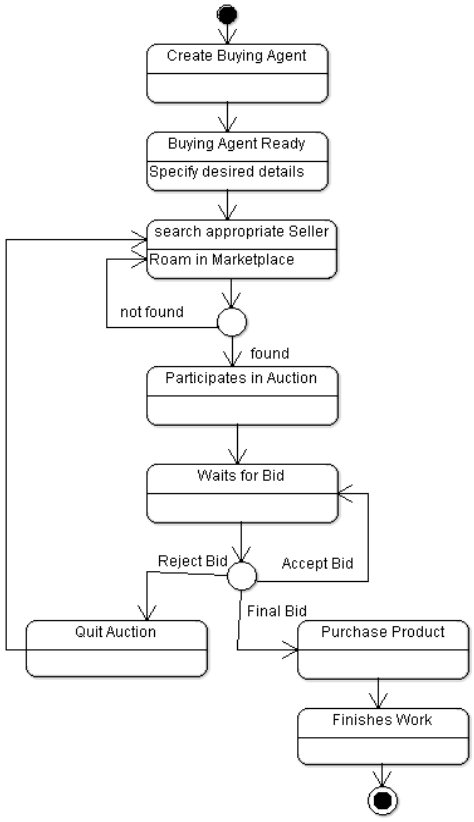

Fig. 2 UML State Chart Diagram of Buying Agent

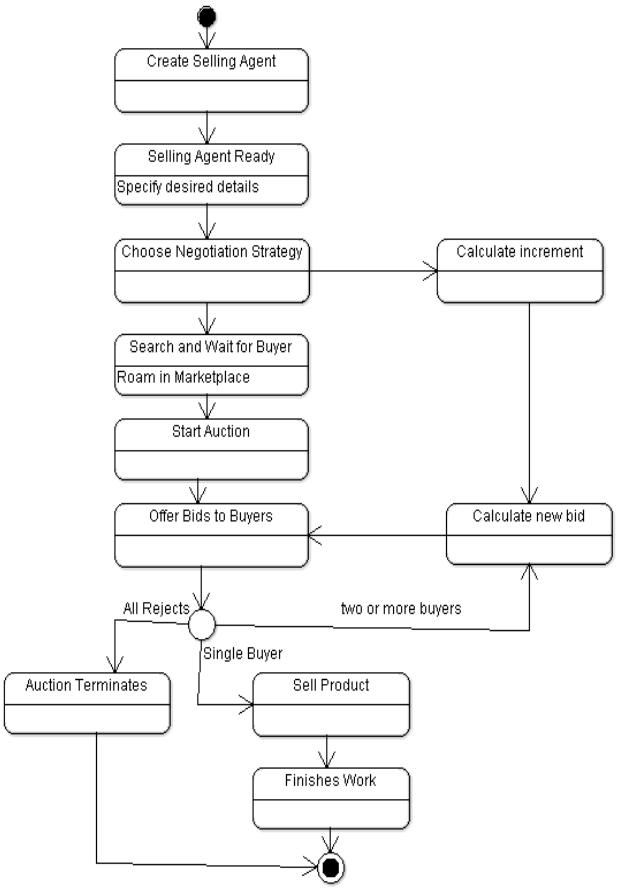

Fig. 3 UML State Chart Diagram of Selling Agent.

4. The *Selling agent* periodically checks for *Buying agents* who have accepted bid for the product. If one or more *Buying agents* accepts bid, the *Selling agent* offers a new bid having price value greater than previous bid by calculating using negotiation strategy chosen by seller and at the end of auction *Selling agent* informs the corresponding *Buying agent* that it has won the auction and can purchase the product(see Figure 2 and 3).
5. The *Buying agent* obtains auction result from the *Selling agent* and buys the product.

III. SYSTEM IMPLEMENTATION AND EXPERIMENT

A. System Implementation

Our system is implemented in two parts: (i) Front-end, A GUI which handles all user interactions is implemented in Java and FXML. All human interaction is confined to this end. (ii) Back-end, A marketplace where actually all agents live and communicate with each other, is implemented using a power agent development platform JADE 4.4.0.

As of now, in initial implementation back-end and front-end are residing on the same machine.

JADE is a software framework written in java. It is open source and under continuous development [14]. JADE facilitates creation of agents under FIPA standards and thus this framework supports development of multi-agent systems [13]. JADE contains predefined classes for creating agents and for their behaviors. It is a superb platform that allows programmers to create and play with agents easily and efficiently.

JADE platform is composed of containers, and container contains agents. Everything done by agent takes place inside the container. According to the need user can create containers. Here in our system we are using the main container only, to hold all the *Buying agents* and the *Selling agents.*

- Classes used to implement our system are

JADE Agent Classes: In our system we have implemented two classes, class *BuyerAgent* and class *SellerAgent* to create agents the *Buying agent* and the *Selling agent* respectively. Being a developer or programmer of JADE we only needs to identify the correct class and need to extend our own classes from the predefined ones, so here we have extended the *Agent* class of JADE and have overridden the *setup()* and *takedown()* methods to implement our classes.

JADE Agent Behaviour Classes: These are also known as agent activity classes as agent behaviours define the actual task an agent will carry out after creation. The actual work of an agent is defined by implementing an object of class which extends the class behaviour. By using *addbehaviour()* method behavior is added to the agent. : In our system we have implemented two classes, class *BuyerAgent* and class *SellerAgent* which extends the *CyclicBehaviour*, the *OneShotBehaviour*, the *TickerBehaviour* classes of the class behaviour.

DOI: 10.18535/ijecs/v5i9.39

JavaFX Application classes: In our system we have implemented the class *SellerGUI* and the *BuyerGUI* which extends the *Application* class of JavaFX. The *Application* class manages framework for JavaFX applications. It creates *JavaFX Application Thread* for executing the *start()* and *stop()* methods of the class.

Other Classes of the System: Here including the above mentioned classes' rest we have two types of classes, classes having .java extension and that of having .fxml extension, which are used to implement variours other functions of the system, like for gui the classes *Buyer.fxml* and *Seller.fxml* are used. The classes *BuyerController.java* and the *SellerController* are intermediary between other java classes and fxml classes. And the other java claases are the class *Book* and the *BuyerBook* used to add details about book and implementing system's functionalities.

- FIPA Messages for Agent Communication

The most exciting feature of agent is their ability of message communication. Agent communicates with other agents by sending and receiving messages. These messages follow a format specified by FIPA under Agent Communication Language (ACL). Here in our system we have used following messages for letting agents to communicate with each other: REQUEST, INFORM, CALL FOR PROPOSAL (CFP), PROPOSE, ACCEPT-PROPOSAL. REQUEST message is used by the *Selling agent* to request to environment for the bid price. Bid price is informed to the *Selling agent* by using INFORM informative as a reply of REQUEST. CALL FOR PROPOSAL (CFP) and ACCEPT-PROPOSAL are again used by *Selling agent.* The *Selling agent* sends a CFP to all the *Buying agents* and so invites buyers to participate in the auction. The *Buying agents* in response send PROPOSE informative through which it sends the *Selling agent* its proposal. If the *Selling agent* likes the proposal it sends ACCEPT-PROPOSAL to *Buying agent* as a reply of proposal. Figure 4 shows the usage of all these informative in our system.

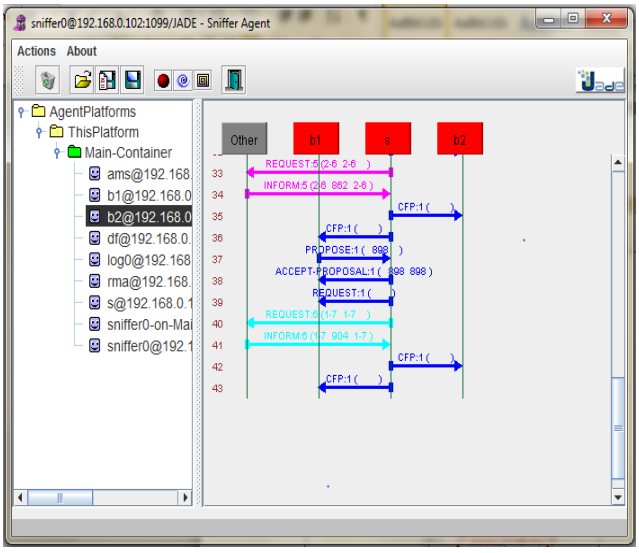

Fig. 4 Sniffer Agent: Usage of Informative

B. System Execution

We are going to run an example on our system to see the working of these agents in our Auction system. When we run the system two windows appears, the JADE platform along with the seller GUI or we can say the *Selling agent* window (see Figure 5). Next seller need to enter details of item(s) need to be auctioned in the seller/auctioneer form as per the availability of different product(s), say the seller added a book of Java (see Figure 6).Now the *Selling agent* is ready to auction the product and is searching for buyers to invite them in the auction. Let's say we have three *Buying agent* created by buyer to buy books from the market, two of them are searching for Java and one is searching book of C++. So the *Selling agent* will invite all of the three buying agent but only two of them who wish to buy Java book will participate in the auction and the third one will roam in the market place in search of appropriate seller (see Figure 7) and the auction starts. As per the chosen negotiation strategy the seller will increase the bid price and offers bids to buyers and buyer have to give their acceptance or rejection on the bid, the auction continues and sooner or later buyers start leaving the auction as the bid price exceeds their maximum price, when only one buyer remains in the auction and rejected the next offered bid of the auctioneer then the auction stops and this buyer becomes the winner of the auction and buys the book. If more than one buyer continues till end, a situation of *tie* comes up, then auctioneer resolves this tie by checking that who has participated in the auction first, becomes the winner (see Figure 8) shows the message the winner will get at the end.

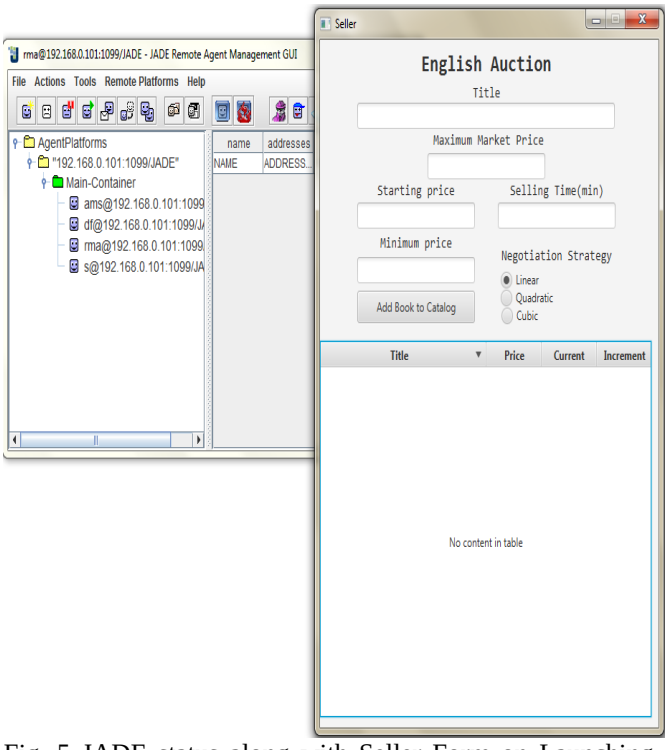

Fig. 5 JADE status along with Seller Form on Launching System

DOI: 10.18535/ijecs/v5i9.39

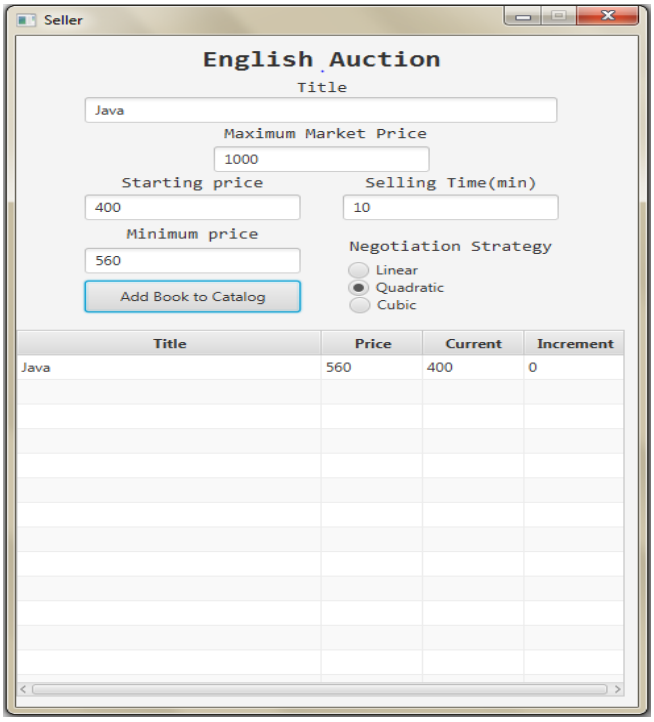

Fig. 6 Seller/Auctioneer/Selling Agent Form with Desired Details

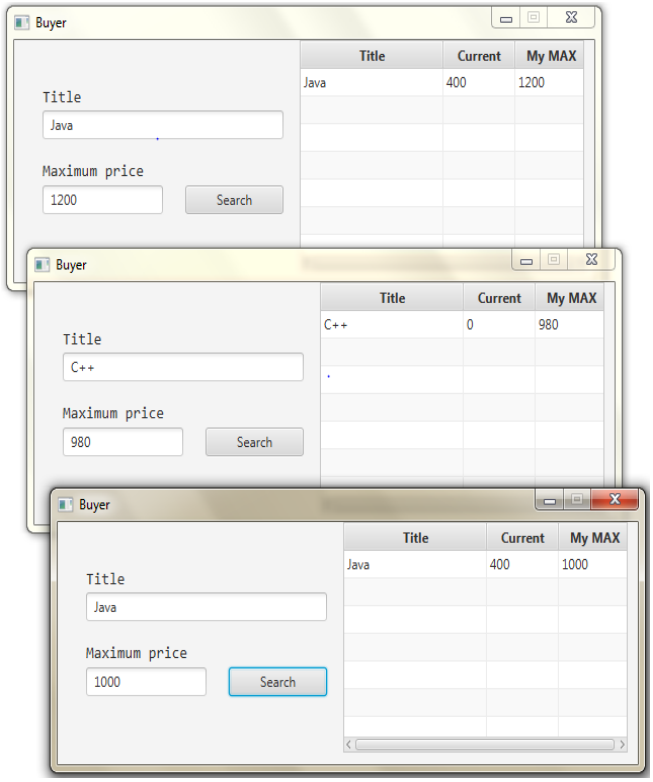

Fig. 7 Buyer/Buying Agent Form with Desired Details

Fig. 8 Message sent to the Winner Buying Agent

IV. CONCLUSION

In this paper we have presented model of an e-marketplace with multi-agents and implemented it using JADE platform. Our aim is to decrease the human efforts and time which they spend in buying or selling product and in doing negotiation. We have done so by automating E-Marketplace by employing software agents for the various time and price consuming tasks of human user, and thus by developing a system using Kasbah negotiation strategies and FIPA English auction interaction protocol which reduces seller's effort and time and ensures that a product is being sold at its best price.

V. FUTURE WORK

As future work we can do a lot of things since agent technology is much wider than we have used here and can help us to automate the whole process of e-commerce as here we have automated a very small part of it. We can change the current system by adding features like:

1. Currently the system is developed for seller's respect, as we have applied negotiation strategies only on the seller's side, the same could be implemented for buyers too.
2. In the current system FIPA English auction interaction protocol is used, we can extend it for others too like Dutch interaction protocol.

We will report the further extensions of the system in our future papers.

VI. REFERENCES

[1] Jeffrey M. Bradshaw, "An Introduction to Software Agents", ISBN 0262522349, MIT Press, 1997.

[2] Maria Ghanza, Marcin Paprzycki, Amalia Pirvanescu, Costin Badica, and Ajith Abraham "Jade based Multi agent E-Commerce Environment: Initial Implementation", Presented at 6th Int. Symposium SYNASC04, Timisoara, Romania

[3] Krzysztof Chmiel, Dariusz Czech, Marcin Paprzycki "Agent Technology in Modeling E-Commerce Processes; Sample Implementation"

[4] Costin Badica, Maria Ghanza, Marcin Paprzycki and Amalia Pirvanescu "Experimenting with a MultiAgent E-Commerce Environment"

[5] Amalia Pirvanescu, Costin Badica, Maria Ghanza, Marcin Paprzycki "Conceptual Architecture and Sample implementation of a multi agent e-commerce system"

[6] Anthony Chavez and Pattie Maes , "Kasbah: An Agent Marketplace for Buying and Selling Goods", Proceedings of the First International Conference on the Practical Applications of the Intelligent Agents and MultiAgent Technology(PAAM‟96). London, UK, April 1996.

[7] Robert H. Guttman, Pattie Maes, Anthony Chavez, Daniel Dreilinger, "Results from a Multi- Agent Electronic Marketplace Experiment", MIT Media Labs

[8] Giorgos Zacharia, Alexandros Moukas, Robert Guttman and Pattie Maes "An agent system for comparative shopping at the point of sale", Software Agent Group, E15-305,MIT Media Lab, Submitted for publication (PAAM‟98)

[9] Alexandros Moukas, Robert Guttman and Pattie Maes "Agent-mediated Electronic Commerce: An MIT Media Laboratory Perspective" Software Agents Group, MIT Media Lab, ICEC98 Submission

[10] Robert H. Guttman, Alexandros G. Moukas and Pattie Maes "Agent-mediated Electronic Commerce: A Survey", Software Agents Group, MIT Media Labs. 80

[11] Robert H. Guttman, Alexandros G. Moukas and Pattie Maes "Agent-mediated Electronic Commerce: A Survey", Software Agents Group, MIT Media Labs.

[12] FIPA. Website: http://www.fipa.org/repository/ips.php3

[13] Aakanksha Gupta, Durgesh Kumar Srivastava, Saket Jain "Evaluating Negotiation Protocols and Negotiation Strategies for Automated E-Commerce", International Journal of Engineering Research & Technology (IJERT), Vol. 5 Issue 07, July-2016

[14] AgentBuilder. Website: http://www.agentbuilder.com

[15] 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.

[16] JADE. Website: http://jade.tilab.com.

[17] Introduction to Software Agents, Wikipedia web reference Website: http://en.wikipedia.org/wiki/Software_agent

Aakanksha Gupta received B.Tech in IT from B.K.Birla Institute of Engineering and Technology, RTU and pursuing M.Tech from BRCM Bahal, MDU. She is having 3 years of working experience as Assistant Professor. Her research interest includes Artificial Intelligence and Multi-Agents. She has published a paper in journal.

Durgesh Kumar Srivastava received B.Tech in IT from MIET Meerut, M.Tech in SE from BIT Mesra and pursuing PHD from Punjab Technical University Jalandhar. He is having around 8 years of work experience. Presently he is working as an Assistant Professor in BRCM College of Engineering and Technology, Bahal. His research interest includes Soft Computing. He has published 9 papers in journals and 3 papers in conference.

Saket Jain received B.Tech in CS from JNIT Jaipur, RTU. PGDM in Systems from IMT Gaziabad. He is having around 6 years of work experience. He is presently working as Senior Engineer in Ericsson Global Pvt. Ltd, Gurgaon. His research area includes Unix/Linux, and Artificial Intelligence. He has written a book named "Unix Administration Quick Guide". He has published a paper in journal.