Towards An Automated Multiagent Negotiation System Based On FIPA Specifications

JAVED AHMED SHAHANI, GHULAM ALI, DR. ZUBAIR A. SHAIKH
National University of Computer & Emerging Science, Karachi

Abstract:- Revolution of e-commerce provides the buying and selling of goods or services online. It is a well established area where business is taking place electronically but it lacks some features like intelligence. On the other hand Software Agents are the autonomous programs that live and roam in the distributed environment, sense and act in the environment to achieve the goal. Agents lack the established infrastructure and the security system. There is need to overcome the deficiencies of both technologies and integrate them to get true and full benefits.

This paper surveys some established Multiagent System toolkits to select appropriate negotiating protocol. The selected protocol is implemented through a well established FIPA-complaint Agent Toolkit, JADE. The communication between buyer agent and seller agent is recorded to avoid the hacking of the negotiation pattern or policy.

Keywords:- E-commerce, Software Agents, Agent toolkits, Negotiation protocols, FIPA, JADE

1. Introduction

Software agents are computational systems that live in inhabit some complex dynamic environment; sense and act autonomously in that environment to achieve the task for which they are designed [1] [2]. Agent platforms are development environment that provide agent builders with a sufficient level of abstraction to allow them to implement intelligent agents with desired attributes, features and rules. When agents socially interact with each other, they need some environment. Such an environment is called Multiagent System (MAS) that involves coordination among agents to solve complex problem. The justification of the Multiagent System is that (a) each agent has incomplete information or capabilities for solving the problem; (b) there is no system global control; (c) data are decentralized; and (d) computation is asynchronous. The architecture that provides implementation environment to Multiagent Systems for management of the agents, coordination, communication, negotiation, etc is called Agent Platform or toolkit. The agent toolkits provide agent builders with a sufficient level of abstraction to allow them to implement intelligent agents with desired attributes, features and rules.

In last few years there has been rapid development and deployment of Multiagent Systems implementation environment such as JADE, FIPA-OS, Zeus, etc. With the increasing number of these frameworks, two parallel agent development standards have evolved: FIPA and MASIF. FIPA is standard to promote interoperable agent applications and agent systems. FIPA specifications only talk about the interfaces through which agents may communicate each other. It does not describe the implementation details. FIPA specifications are divided into five categories: Applications, Abstract architecture, Agent Communication, Agent Management and Agent Message Transport, are the five categories in which FIPA specifications are divided. The FIPA Reference Model considers an Agent Platform as a set of four components: Agents, Directory Facilitator (DF), Agent Management System (AMS), and Message Transport System (MTS). The DF and AMS support the management of the agents, while the MTS provides a message delivery service. MASIF intends to support interoperability among heterogeneous agent systems. It normally focuses on the migrations of the agents on different hosts [3] [4].

Ecommerce promises to change the way businesses interact with each other and with their customers. It is some times claimed that ecommerce has potential to create “an environment where companies will be at their most agile and market place will approach perfect efficiency” [5]. The job of ecommerce is to increase the degree and sophistication of automation in Business. Today, ecommerce encompasses very wide range of business activities and processes, from e-banking to offshore manufacturing and e-logistic. The ever growing dependence of modern industries on electronically enabled business processes gave impetus to the growth and development of supporting systems. To achieve the degree of automation in which human intervention is reduced to minimum require intelligent ecommerce applications. Agents are prefect candidate to introduce the intelligence in ecommerce applications. In this paper we make use of autonomous bidding agents, computer program that bids in electronic market without direct human intervention. We focus on the state of art agent-mediated negotiation (bargaining) mechanism, and address the issue of pattern recognition of the buyer agents to make successful deal at low price. Some work is also done in resolving this issue.

2. Negotiations and Agents

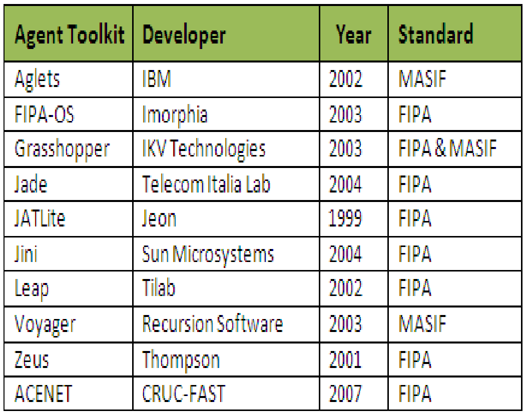
In e-commerce systems negotiation capabilities are very essential because one of the major changes in e-commerce is that dynamic pricing and personalization of offers will become the norm for many goods and customers. In human negotiations, two or more parties bargain with one another to determine the price. Software agents provide automation in negotiation process to achieve similar goals. Automated negotiation is based on some strategy. Such strategies are determined by the negotiations protocol. This protocol defines the “rules of encounter between the agents”. There is no universally best approach or technique for automated negotiations, rather protocols and strategies need to be set according to the prevailing situation. Negotiation models are divided into two categories: auctions and bilateral negotiations.

Online auction provide millions of globally dispersed customers with more varieties of goods that can be selected within a flexible pricing mechanism. In online auction, one or more agents initiate the auction, and several other agents make bids according to imposed protocol. There are many different types of auctions. In our paper we give an overview of the four common types of single sided auction. In English auction the agent’s dominant strategy is to bid small amount more than current highest bid and stop when the user valuation is reached; In First price sealed bid auction (FPSB) the price of bid and time to stop are functions of he agent’s own valuation of the item and its beliefs about the valuation of other bidders; In Vickrey auction: the dominant strategy is to bid the user’s true valuation, while in Dutch auction prices decline until a buyer makes a bid [6] [7].

The negotiation type used in the scenario of this paper is single sided Dutch auction.

3. Selection of Agent Toolkit

It has been discussed earlier that MASIF only focuses on mobility while FIPA talks in detail about the communication and interaction protocols. So it is obvious that instead of using MASIF-compliant agent platform we must choose a FIPA-compliant platform. Under FIPA standardization JADE is the appropriate agent toolkit for our implementation. Because it is rich in interaction protocols, negotiations, etc. It is open source and being updated continuously. JADE is supported by all operating systems and distributed in nature because it is developed in JAVA [8]. In future we also plan to run our application over ACENET (Agent Collaborative Environment based on .NET).



Agent Toolkit	Developer	Year	Standard
Aglets	IBM	2002	MASIF
FIPA-OS	Imorphia	2003	FIPA
Grasshopper	IKV Technologies	2003	FIPA & MASIF
Jade	Telecom Italia Lab	2004	FIPA
JATLite	Jeon	1999	FIPA
Jini	Sun Microsystems	2004	FIPA
Leap	Tilab	2002	FIPA
Voyager	Recursion Software	2003	MASIF
Zeus	Thompson	2001	FIPA
ACENET	CRUC-FAST	2007	FIPA

Fig. 1. List of surveyed Agent toolkits

We have implemented our problem through FIPA’s basic interaction protocol, i.e. FIPA Propose Interaction Protocol (IP). It allows an agent to propose to receiving agents that the initiator will do the actions described in propose communicative act when the receiving agent accepts the proposal.

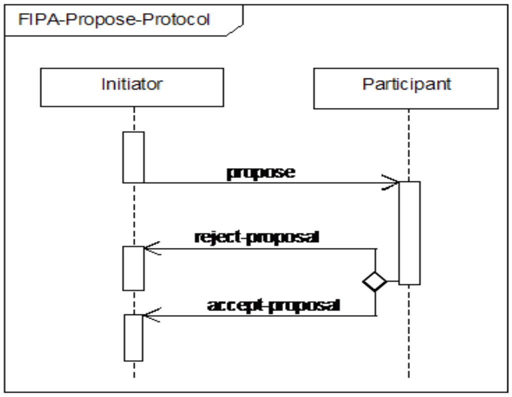


Fig. 2. FIPA Propose Interaction protocol

Fig.2 shows that Initiator sends a propose message to the Participant indicating that it will perform some action if the Participant agrees. The Participant responds by either accepting or rejecting the proposal, communicating this with the accept-proposal or reject-proposal communicative act, accordingly [9]. We have completely implemented negotiation protocol. The agents are iteratively negotiating until they reach on an agreement, either acceptance or the rejection of the proposal. We have implemented Dutch auctions.

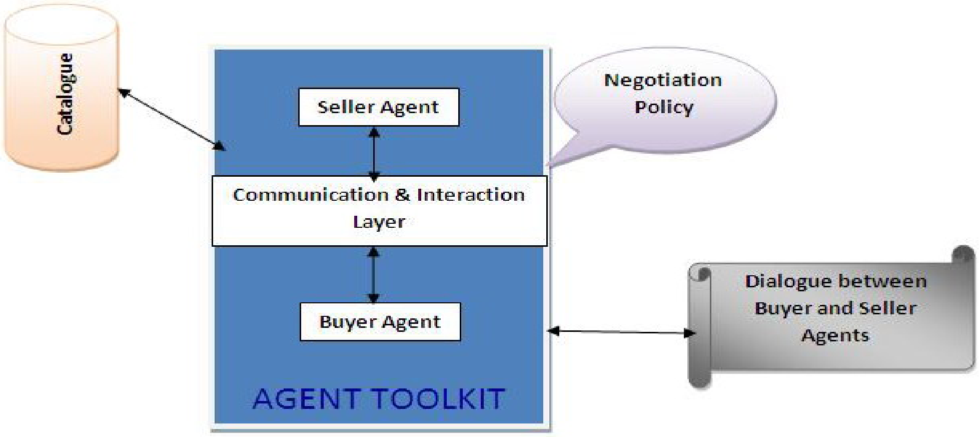

Catalogue
Seller Agent
Negotiation Policy
Communication & Interaction Layer
Buyer Agent
AGENT TOOLKIT
Dialogue between Buyer and Seller Agents

Fig. 3. The conceptual Architecture of the proposed Model

4. Architecture and Design

The proposed automated multi-agent negotiation system is based on FIPA Specifications. This specification has layered architecture, which consists of four layers. The proposed system is based on five major components:

a) Database System
b) Seller Agents
c) Buyer Agents
d) Web service - to dump the recorded conversation between seller and buyer agent so that an owner of the company may check and ensure that the fair deal is taking place
e) Negotiation policies

Catalog using database management system provides information about the availability of product in stock, and price related information. Currently we are considering only price and the availability of the stock. We do not consider any other feature in our system. But in future we will work on multi-attribute contracts using bilateral negotiation which will cover price, quality, delivery date, and so on. Seller Agent varies its offering depending on the customer it is deal with. Buyer Agent crawls all the available outlets to find the most suitable for purchasing the chosen good. Negotiation Policy defines the rules of encounter between the agents for the certain deal. Seller agent varies its offering based on the policy. Web service is used to dump the recorded conversation between seller and buyer agent so that an owner of the company may check and ensure that the fair deal is taking place.

Buyer agent confirms its range from the user and start bargaining with seller agent. Seller Agent reject the proposal if the price offered is even less than the actual price of the product, but if seller agent finds the proposed price less than offered price and more actual price than continues to giving offers until buyer reaches to certain profit margin. All the transactions are being recorded, which can be accessed by the market owner. Market owner uses these transaction records for assessment fairness of the deal, if any prove of pattern recognition is found ultimate the policy is changed by the market owner.

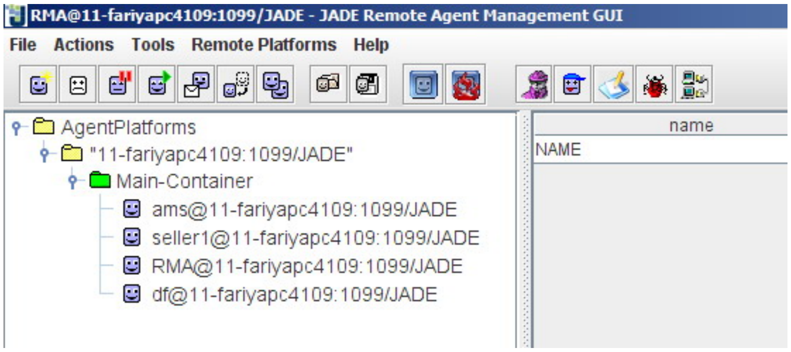

RMA@11-fariyapc4109:1099/JADE - JADE Remote Agent Management GUI
File
Actions
Tools
Remote Platforms
Help
AgentPlatforms
"11-fariyapc4109:1099/JADE"
Main-Container
ams@11-fariyapc4109:1099/JADE
seller1@11-fariyapc4109:1099/JADE
RMA@11-fariyapc4109:1099/JADE
df@11-fariyapc4109:1099/JADE
name
NAME

Fig.4. JADE's main container where Seller Agent resides

5. Implementation and Results

The model has been implemented in JADE [10]. JADE architecture, consisting of a platform within which agents "live" and containers, where agents "reside" matches well with our requirements. Negotiations between *Seller* and *Buyer* agents take place inside JADE containers. There is one Main container that hosts the Seller agent. Users can create as many containers they need to hold their Buyer agents. Agent Communication Language (ACL) provided by JADE has been used for message passing among agents. Seller agent and buyer agent talk to each other through ACL message passing. *Agent class has been used* for describing the agent types. Tow main methods are automatically invoked by the platform during the agent lifecycle, i.e. *setup()* and *takedown()*. Behaviours class is used for describing the activities performed by agents in the system. A behavior is an abstraction that represents an atomic activity performed by an agent. The class *Behaviour* is the root of a class hierarchy abstracting various agent behavior types. The suitable performatives of FIPA-Propose protocol are implemented to create real scenario of negotiation.

Fig. 4-8 show the results our implemented model.

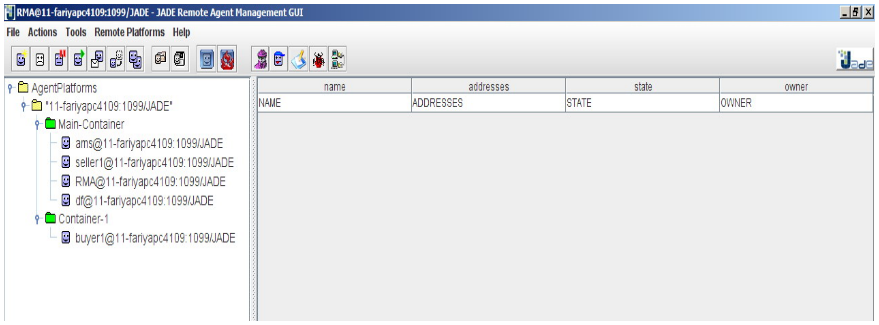
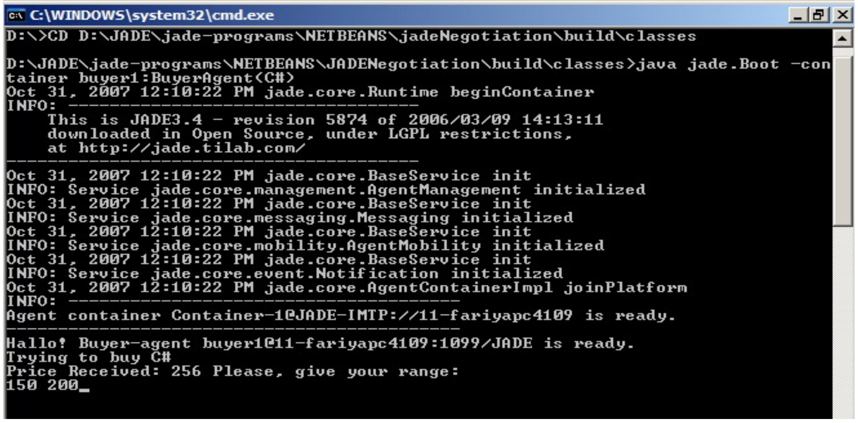
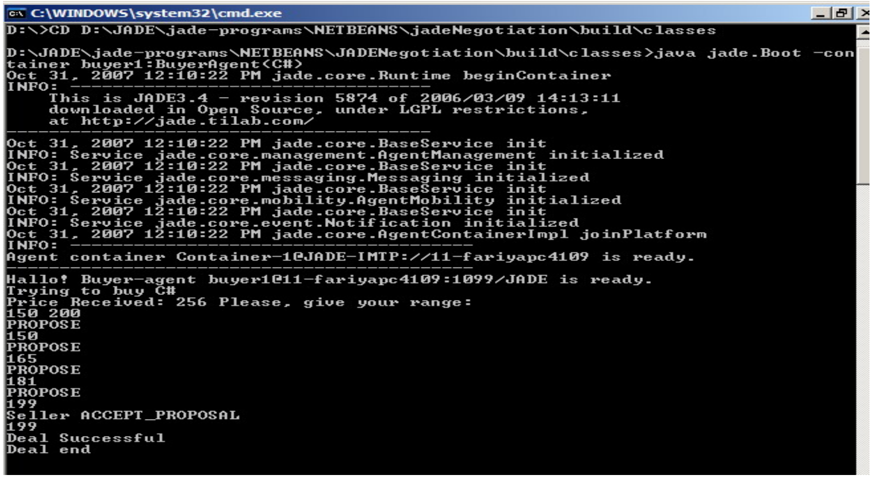

RMA@11-fariyapc4109:1099/JADE - JADE Remote Agent Management GUI
File
Actions
Tools
Remote Platforms
Help
AgentPlatforms
"11-fariyapc4109:1099/JADE"
Main-Container
ams@11-fariyapc4109:1099/JADE
seller1@11-fariyapc4109:1099/JADE
RMA@11-fariyapc4109:1099/JADE
df@11-fariyapc4109:1099/JADE
Container-1
buyer1@11-fariyapc4109:1099/JADE
name
addresses
state
owner
NAME
ADDRESSES
STATE
OWNER

Fig.5. JADE's container-1 where Buyer Agent resides



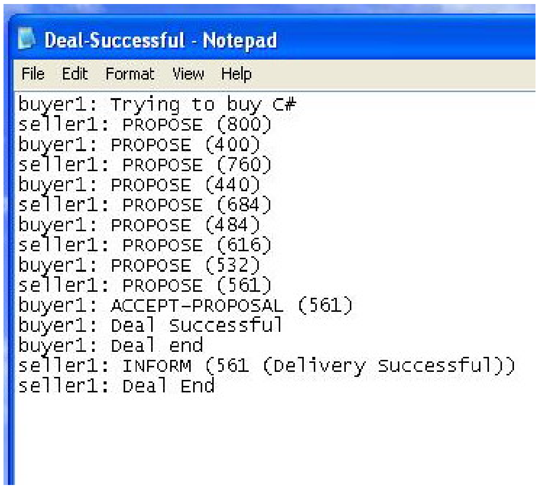
```
D:\>CD D:\JADE\jade-programs\NETBEANS\jadeNegotiation\build\classes

D:\JADE\jade-programs\NETBEANS\JADENegotiation\build\classes>java jade.Boot -con
tainer buyer1:BuyerAgent(C#)
Oct 31, 2007 12:10:22 PM jade.core.Runtime beginContainer
INFO: ----------------------------------
    This is JADE3.4 - revision 5874 of 2006/03/09 14:13:11
    downloaded in Open Source, under LGPL restrictions,
    at http://jade.tilab.com/
----------------------------------------
Oct 31, 2007 12:10:22 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Oct 31, 2007 12:10:22 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Oct 31, 2007 12:10:22 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Oct 31, 2007 12:10:22 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Oct 31, 2007 12:10:22 PM jade.core.AgentContainerImpl joinPlatform
INFO: --------------------------------------
Agent container Container-1@JADE-IMTP://11-fariyapc4109 is ready.
--------------------------------------------
Hallo! Buyer-agent buyer1@11-fariyapc4109:1099/JADE is ready.
Trying to buy C#
Price Received: 256 Please, give your range:
150 200_
```

Fig. 6. Console where a buyer agent asks the negotiating range from its owner



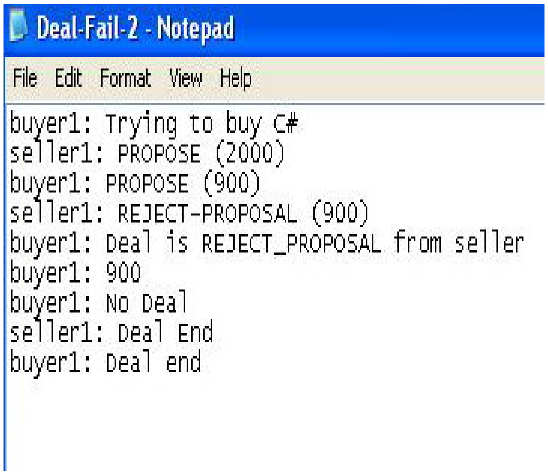
```
D:\>CD D:\JADE\jade-programs\NETBEANS\jadeNegotiation\build\classes

D:\JADE\jade-programs\NETBEANS\JADENegotiation\build\classes>java jade.Boot -con
tainer buyer1:BuyerAgent(C#)
Oct 31, 2007 12:10:22 PM jade.core.Runtime beginContainer
INFO: ----------------------------------
    This is JADE3.4 - revision 5874 of 2006/03/09 14:13:11
    downloaded in Open Source, under LGPL restrictions,
    at http://jade.tilab.com/
----------------------------------------
Oct 31, 2007 12:10:22 PM jade.core.BaseService init
INFO: Service jade.core.management.AgentManagement initialized
Oct 31, 2007 12:10:22 PM jade.core.BaseService init
INFO: Service jade.core.messaging.Messaging initialized
Oct 31, 2007 12:10:22 PM jade.core.BaseService init
INFO: Service jade.core.mobility.AgentMobility initialized
Oct 31, 2007 12:10:22 PM jade.core.BaseService init
INFO: Service jade.core.event.Notification initialized
Oct 31, 2007 12:10:22 PM jade.core.AgentContainerImpl joinPlatform
INFO: --------------------------------------
Agent container Container-1@JADE-IMTP://11-fariyapc4109 is ready.
--------------------------------------------
Hallo! Buyer-agent buyer1@11-fariyapc4109:1099/JADE is ready.
Trying to buy C#
Price Received: 256 Please, give your range:
150 200
PROPOSE
150
PROPOSE
165
PROPOSE
181
PROPOSE
199
Seller ACCEPT_PROPOSAL
199
Deal Successful
Deal end
```

Fig. 7. Buyer Console where agent shows the Deal



```
buyer1: Trying to buy C#
seller1: PROPOSE (800)
buyer1: PROPOSE (400)
seller1: PROPOSE (760)
buyer1: PROPOSE (440)
seller1: PROPOSE (684)
buyer1: PROPOSE (484)
seller1: PROPOSE (616)
buyer1: PROPOSE (532)
seller1: PROPOSE (561)
buyer1: ACCEPT-PROPOSAL (561)
buyer1: Deal successful
buyer1: Deal end
seller1: INFORM (561 (Delivery successful))
seller1: Deal End
```

Fig. 8(a) The actual negotiation recorded and dumped to website or text file (Successful)



```
buyer1: Trying to buy C#
seller1: PROPOSE (2000)
buyer1: PROPOSE (900)
seller1: REJECT-PROPOSAL (900)
buyer1: Deal is REJECT_PROPOSAL from seller
buyer1: 900
buyer1: No Deal
seller1: Deal End
buyer1: Deal end
```

Fig. 8(b) The actual negotiation recorded and dumped to website or text file (Unsuccessful)

6. Conclusion

In this paper we developed and implemented an agent based negotiation model. The basic intelligence to the buyer and selling agents has been provided. Buyer agent confirms the negotiation range from its owner before starting negotiation. Currently the capabilities of the agents are limited and very few factors have been considered in the implementation. In future some advanced negotiation protocols will be considered. The conversation regarding any deal between buyer and seller is recorded so that the company owner may analyze it to avoid the hacking of the pattern of the negotiation or the policy of the organization.

References:-

[1] Franklin, S., and Graesser, A., Is It an Agent or Just a Program? A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages.* New York: Springer-Verlag.

[2] Jennings, N.R., An Agent based approach for building complex software systems. Communication of the ACM, 44(4), 35-41, 2001

[3] Mallah, G. A, Shaikh, Z. A, Nisar, M. J, A Decentralized Fault Tolerant FIPA-Complaint Agent Framework based on .NET, Accepted in *2nd All Pakistan UTECH and ACM Technical Conference 2007 to be held on 9-10 November 2007 at Usman Institute of Technology, Karachi.*

[4] Ghulam Ali, Zubair A. Shaikh, Noor A Shaikh, The Design and Implementation of an Agent Framework to Support Distributed Problem Solving, published in *Conference Proceedings, 25-27 September 2007, Athens, Greece, and accepted for Springer-Verlag publications to be released in November 2007.*

[5] Mark S. Ackerman, Lorrie Faith Cranor, Privacy in e-commerce: examining user scenarios and privacy preferences, November 1999, *Proceedings of the 1st ACM conference on Electronic commerce EC '99.*

[6] Minghua He, Nicholas R. Jennings, and Ho-Fung Leung On Agent-Mediated Electronic Commerce IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 15, NO. 4, JULY/AUGUST 2003

[7] R. Guttman and P. Maes. "Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce." *Proceedings of the Second International Workshop on Cooperative Information Agents (CIA'98).* Paris, France, July 1998.

[8] Chmiel, K. et al, Testing the Efficiency of JADE Agent Platform, *Proceedings of the 3rd International Symposium on Parallel and Distributed Computing, Cork, Ireland* 2004, IEEE Computer Society Press, Los Alamitos, CA, pp. 49-57

[9] FIPA Propose Interaction Protocol Specification, FIPA 2000, Standards, Agents, Interaction Protocols, Conversations, Propose http://www.fipa.org/specs/fipa00036/

[10]JADE, Java Agent Development Framework. See http://jade.cselt.it.