Proceedings 2nd CBRC, Lahore, Pakistan
November 14, 2009

ROLE OF MULTI AGENT SYSTEM FOR ADOPTIVE E-MARKET

Khubaib Ahmed Qureshi
HIMS-IT Dept, HIMS, Hamdard University
khubaib_ahmed@yahoo.com

ABSTRACT

Agents are software robots and vitally used in e-Business applications. This paper discusses electronic markets and the role which agents can play in information transformation for automating electronic market transactions. A Multi Agent System is designed and developed which pertains to merchant brokering stage of Consumer Buying Behavior Model with the intent of appropriate framework.

Keywords: e-Markets, Buyer/Seller Agents, Java, Multi Agent Systems

I. INTRODUCTION

Software agents are programs to which one can delegate (aspects of) a task. They differ from traditional software in that they are personalized, continuously running and semi-autonomous. These qualities make agents useful for a wide variety of information and process management tasks [1]. It should come as no surprise that these same qualities are particularly useful for the information rich and process rich environment of electronic commerce.

Electronic commerce encompasses a broad range of issues including security, trust, reputation, law, payment mechanisms, advertising, ontologies, on-line catalogs, intermediaries, multimedia shopping experiences, and back-office management. Agent technologies can be applied to any of these areas where a personalized, continuously running, semi-autonomous behavior is desirable. However, certain characteristics will determine to what extent agent technologies are appropriate. For example, how much time or money could be saved if a certain process was partially automated (e.g., comparing products from multiple merchants)? How easy is it to express your preferences for the task (e.g., shopping for a gift)? What are the risks of an agent making a sub-optimal transaction decision (e.g., making stock market buying and selling decisions or buying a car)? What are the consequences for missed opportunities (e.g., not being able to effectively monitor new job postings)? Generally, the more time and money that can be saved through automation, the easier it is to express preferences, the lesser the risks of making sub-optimal transaction decisions, and the greater the loss for missed opportunities, the more appropriate it is to employ agent technologies in electronic commerce.
Software agents will play an increasing variety of roles as mediators in electronic commerce [2]. This project explores these roles, their supporting technologies, and how they relate to electronic commerce with the intent of appropriate framework.

II. LITERATURE REVIEW

Consumer Buying Behavior (CBB) Model

There are several descriptive theories and models that attempt to capture consumer buying behavior. Although different, these models all share a similar list of six fundamental stages guiding consumer buying behavior. These six stages also elucidate where agent technologies apply to the consumer shopping experience and allow us to more formally categorize existing agent-mediated electronic commerce systems:

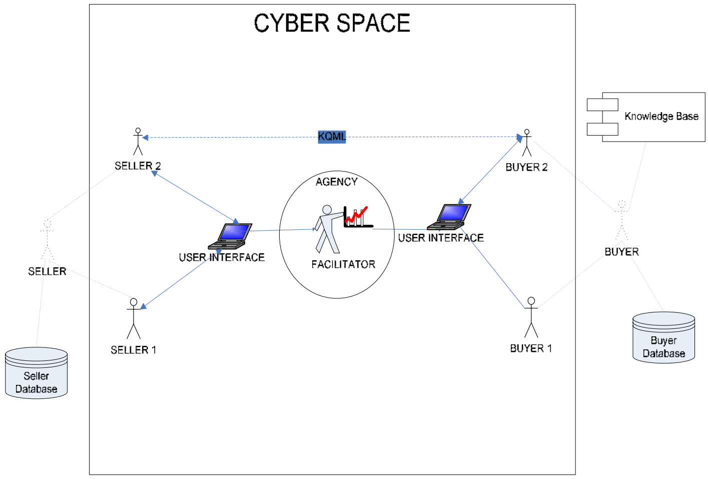


Figure 1

1. Need Identification
This stage characterizes the consumer becoming aware of some unmet need. Within this stage, the consumer can be stimulated through product information.

2. Product Brokering
This stage comprises the retrieval of information to help determine *what* to buy. This encompasses the evaluation of product alternatives based on consumer-provided criteria. The result of this stage is the “consideration set” of products.

3. Merchant Brokering
This stage combines the “consideration set” from the previous stage with merchant-specific information to help determine *who* to buy from. This includes the evaluation of merchant alternatives based on consumer-provided criteria (e.g., price, warranty, availability, delivery time, reputation, etc.).

4. Negotiation
This stage is about *how* to determine the terms of the transaction. Negotiation varies in duration and complexity depending on the market. In traditional retail markets, prices and other aspects of the trans-action are often fixed leaving no room for negotiation. In other markets (e.g., stocks, automobile, fine art, local markets, etc.), the negotiation of price or other aspects of the deal are integral to product and merchant brokering. Traditional CBB models do not identify this stage explicitly, but the conclusion of the Negotiation stage is comparable to the Choice or Decision stage found in other models.

5. Purchase and Delivery
The purchase and delivery of a product can either signal the termination of the negotiation stage or occur sometime afterwards (in either order). In some cases, the available payment (e.g., cash only) or delivery options can influence product and merchant brokering.

6. Product Service and Evaluation
This post-purchase stage involves product service, customer service, and an evaluation of the satisfaction of the overall buying experience and decision. The nature of this stage (and others) depends upon for whom the product was purchased. As with most models, these stages represent an approximation and simplification of complex behaviors. As noted, CBB stages often overlap and migration from one to another can be non-linear and iterative.
From this CBB perspective, we can identify the roles of agents as mediators in electronic commerce [5]. The personalized, continuously-running autonomous nature of agents makes them well-suited for mediating those consumer behaviors involving information filtering and retrieval, personalized evaluations, complex co-ordinations, and time-based interactions.

PERSONA LOGIC

Persona Logic is a tool that enables consumers to narrow down the products that best meet their needs by guiding them through a large product feature space. The system filters out unwanted products within a given domain by allowing shoppers to specify constraints on a product’s features. A constraint satisfaction engine then returns an ordered list of only those products that satisfy all of the hard constraints.

FIREFLY

Firefly recommends products via a “word of mouth” recommendation mechanism called automated collaborative filtering (ACF). ACF first compares a shopper’s product ratings with those of other shoppers. After identifying the shopper’s “nearest neighbors” (i.e., users with similar tastes), ACF recommends products that they rated highly but which the shopper has not yet rated, potentially resulting in serendipitous finds. Essentially, Firefly uses the opinions of like-minded people to offer recommendations. The system is currently used to recommend commodity products such as music and books.

JANGO

Jango can be viewed as an advanced Bargain Finder. The original Jango version solved the merchant blocking issue by having the product requests originate from each consumer’s Web browser instead of from a central site as in Bargain Finder. This way, requests to merchants from a Jango-augmented Web browser appeared as requests from ‘real’ customers. This kind of aggressive interoperability

Jango’s modus operandi is simple: once a shopper has identified a specific product, Jango can simultaneously query merchant sites (from a list now maintained by Excite, Inc.) for its price. These results allow a consumer to compare merchant offerings on price.

KASBAH

The MIT Media Lab’s Kasbah is an on-line, multiagent classified ad system. A user wanting to buy or sell a good creates an agent, gives it some strategic direction, and sends it off into a centralized agent marketplace. Kasbah agents proactively seek out potential buyers or sellers and negotiate with them on behalf of their owners. Each agent’s goal is to complete an acceptable deal, subject to a set of user-specified constraints such as a desired price, a highest (or lowest) acceptable price, and a date by which to complete the transaction. The latest version of Kasbah incorporates a distributed trust and reputation mechanism called the Better Business Bureau. Upon the completion of a transaction, both parties may rate how well the other party managed their half of the deal (e.g., accuracy of product condition, completion of the transaction, etc.).Agents can then use these ratings to determine if they should negotiate with agents whose owners fall below a user-specified reputation threshold.

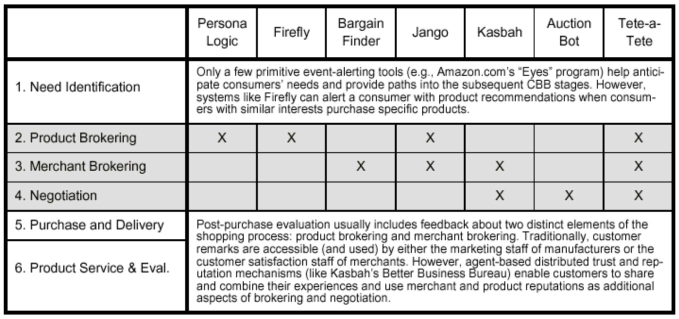
AUCTIONBOT

AuctionBot is a general purpose Internet auction server at the University of Michigan. Auction Bot users create new auctions to sell products by choosing from a selection of auction types and then specifying its parameters (e.g., clearing times, method for resolving bidding ties, the number of sellers permitted, etc.). Buyers and sellers can then bid according to the multi-lateral distributive negotiation protocols of the created auction. In a typical scenario, a seller would bid a reservation price after creating the auction and let AuctionBot manage and enforce buyer bidding according to the auction protocol and parameters.

What makes AuctionBot different from most other auction sites, however, is that it provides an application programmable interface (API) for users to create their own software agents to autonomously compete in the AuctionBot marketplace. Such an API provides a semantically sound interface to the marketplace. It is left to the users to encode their own bidding strategies. Fish market is not currently being used as a real-world system, but it has hosted tournaments to compare opponents’ hand-crafted bidding strategies along the lines of Axelrod’s prisoner’s dilemma tournaments.

TETE-A-TETE

Tete-a-Tete provides a unique negotiation approach to retail sales. Unlike most other on-line negotiation systems which competitively negotiate over price, Tete-a-Tete agents cooperatively negotiate across multiple terms of a transaction e.g., warranties, delivery times, service contracts, return policies, loan options, gift services, and other merchant value added services. Like Kasbah, this negotiation takes the form of multi-agent, bilateral bargaining but not using simple raise or decay functions as in Kasbah. Instead, Tete-a-Tete shopping agents follow an argumentative style of negotiation with sales agents and use the evaluation constraints captured during the Product Brokering and Merchant Brokering stages as dimensions of a multi-attribute utility. This utility is used by a consumer’s shopping agent to rank order merchant offerings based on how well they satisfy the consumer’s preferences. In essence, Tete-a-Tete integrates all three of the Product Brokering, Merchant Brokering, and Negotiation CBB stages.



	Persona Logic	Firefly	Bargain Finder	Jango	Kasbah	Auction Bot	Tete-a-Tete
1. Need Identification	Only a few primitive event-alerting tools (e.g., Amazon.com's "Eyes" program) help anticipate consumers' needs and provide paths into the subsequent CBB stages. However, systems like Firefly can alert a consumer with product recommendations when consumers with similar interests purchase specific products.						
2. Product Brokering	X	X		X			X
3. Merchant Brokering			X	X	X		X
4. Negotiation					X	X	X
5. Purchase and Delivery	Post-purchase evaluation usually includes feedback about two distinct elements of the shopping process: product brokering and merchant brokering. Traditionally, customer remarks are accessible (and used) by either the marketing staff of manufacturers or the customer satisfaction staff of merchants. However, agent-based distributed trust and reputation mechanisms (like Kasbah's Better Business Bureau) enable customers to share and combine their experiences and use merchant and product reputations as additional aspects of brokering and negotiation.						
6. Product Service & Eval.							

Figure 2

III. FUNCTIONAL SPECIFICATIONS

1. Each Store agent represents one retail store and knows the prices and available quantities of the products in its store. Store agents receive requests for price quotes from the Buyer agents and respond by telling the Buyer agent the prices, shipment costs and quantities that they can deliver.

2. Agents will be capable of communicating with each other over an intranet or the Internet. These agents communicate with each other using the Knowledge Query and Manipulation Language (KQML).

3. Each agent has a graphical interface that allows the user to control the operation of the agent or examine the status of agent activity. The Buyer agent's interface allows the user to select a product and quantity to purchase.

4. A pull down menu lists the known products and a text field is available for specifying the quantity to be purchased.

5. Pressing the **Shop** button will cause the Buyer agent to begin shopping for the specified product and quantity.

6. Pressing the **Quit** button will cause the Buyer agent to send messages to shut down all of the Store agents and then shut down itself.

7. Each agent interface also displays the agent's inventory, account balance, and the agent's activities.

8. After the user selects a product and quantity and clicks on the **Shop** button, the Buyer agent sends requests for price quotes to the Store agents. The Store agents respond with price quotes for the specified product. The Buyer agent decides which store has the lowest price and sends a purchase request to the Store agent with the lowest price.

9. The Store agent then confirms the purchase, decreases its available inventory, adds the purchase price to its account balance, and sends a purchase confirmation message to the Buyer agent. After receiving the purchase confirmation, the Buyer agent increases its inventory and subtracts the purchase price from its account balance.

IV. KNOWLEDGE BASE

a. Knowledge Representation
There are different methods of knowledge representation like:

- Rules
- Objects
- Frames

In Buyer/Seller Agency we have followed Rule Based Knowledge Representation which is very flexible and compatible way of representing heuristics.

b. Inference Method

Two standard methods of inference are:

I) Forward Chaining – Data Driven Approach

II) Backward Chaining – Goal Driven Approach

Buyer/Seller Agency is making use of Data Driven Approach which is also called Forward Chaining for Selection of best offer according to Buyer's preferences (parameters).

BUYER'S PREFERENCES

Buyer Agent will be informed about the following parameters (price of the product, ranking of the seller, and flexibility required in shipment period) against which the best offer will be selected.

SELECTION CRITERIA

Buyer Agent will deliver Price, Preferences, and Selected Ranking to the Facilitator Agent (Agency) which will intelligently select appropriate ranked sellers. Facilitator will send Request to the selected sellers. Seller will check its inventory for sufficient quantity of the required product and its shipment option. Then Seller sends an Offer to the Buyer providing non-negotiable price with shipment period. After the Deadline offers from the Sellers will be evaluated. The evaluation will be done on the basis of price and shipment period against the preferences provided by Buyer. For example, in case of higher price preference, offer will higher price will be given priority over lower priced offers. Similarly an offer within stipulated time frame will be given higher priority.

Some of the possible options for Buyer's preferences are described below:

i) Price
   - i. High
   - ii. Medium
   - iii. Low

ii) Ranking
   - i. High
   - ii. Medium
   - iii. Low

iii) Shipment Period
   - i. On Stipulated time
   - ii. Within 20% of stipulated time
   - iii. With 40% of stipulated time

Figure 3 represents the Execution Process of an Agent and how an agent's mental model is changed in its decision making process.

V. KQML

There are different languages for multi-agent interaction [14]. Including KQML which is a high-level language intended to support interoperability among intelligent agents in distributed applications. It is both a message format and a message-handling protocol to support runtime knowledge sharing among agents. KQML can also be used for two or more intelligent systems to share knowledge in support of cooperative problem solving.

Why:

- Existing protocols, such as RPC, are insufficient.
- Nor are there standard models for programming in an environment where some of the *data is supplied by processes running on remote machines* and some of the *results are needed by other similarly distant processes*.
- Must be easy and flexible to communicate.

Layer of Communication: Agents may have different, and even conflicting, agendas. KQML message is defined in terms of constraints on the message sender rather than the message receiver who choose a

course of action that is compatible with other aspects of its function and strive for maximal cooperation. KQML language can be viewed as being divided into three layers:

- Content layer,
- Message layer,
- Communication layer.

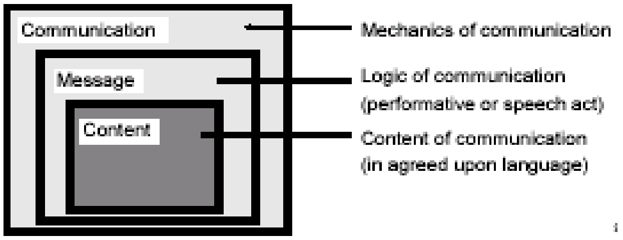

Communication
Message
Content
Mechanics of communication
Logic of communication
(performative or speech act)
Content of communication
(in agreed upon language)

Figure 3

Content Layer: The actual content of the message in the program's own representation language. KQML can carry any representation language, including languages expressed as ASCII strings & those expressed using a binary notation.

Communication Layer: Encodes a set of features to the message which describe the lower level communication parameters, such as the identity of the sender and recipient, and a unique identifier associated with the communication.

Message Layer: The message layer forms the core of the language. The primary function of the message layer is to identify the protocol to be used to deliver the message and to supply a speech act, or performative, which the sender attaches to the content. The performative signifies that the content is an assertion, a query, a command, or any of a set of known performatives.

Message Parts: A KQML message consists of a *performative*, its associated *arguments* which include the real *content of the message*, and a set of *optional arguments*. The main focus of KQML is on its extensible set of performatives, which defines the permissible operations that agents may attempt on each other's knowledge and goal stores at run time.

Example:

```
(ask :content (prodoffer purchase(?prod, ?price, ?days) )
        :sender buyer1                                Performative
        :receiver seller 1
                                                      Parameters
        :in-reply-to <label of the reply>
        :reply-with <label to be replied>                               Value
        :language KQML
        :ontology  buyer-seller                   )
```

Ontologies:

Describe relevant objects and relations in a domain [4]:

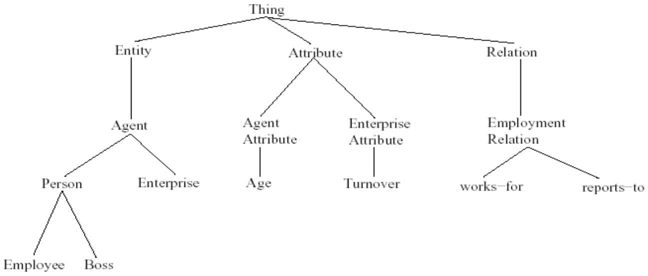


Figure 4

VI. DESIGN

Agent Roles and Description

There will be three types of agents in the marketplace:

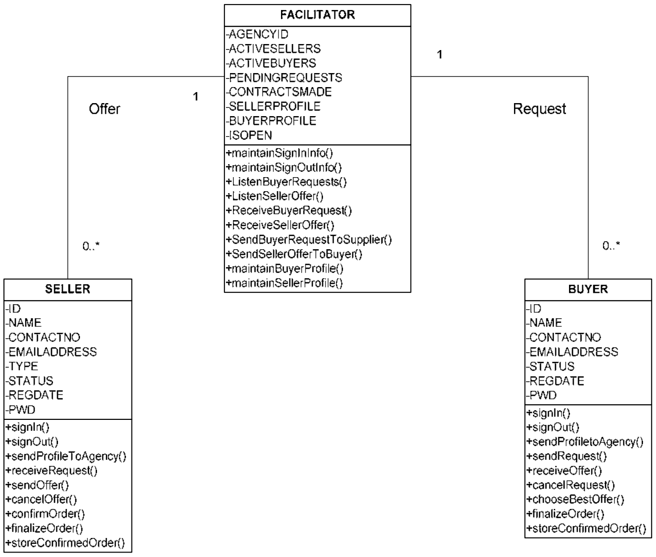


Figure 5

i) Facilitator

Activities performed by Facilitator are:

- maintainSignInInfo

Facilitator maintains information of Buyer and Seller Agents who have signed in and are active. So Facilitator will send and receive messages from active agents only.

- maintainSignOutInfo

Facilitator keeps a check when the Buyer or Seller has signed out. After sign out it marks its status as inactive.

- listenBuyRequests

Facilitator listens for the buy requests from the buyer.

- receiveBuyRequest

As soon as the listener listens to a buy request, this method receives the buy request and stores it in its own memory. This method also automatically triggers the sendBuyerRequest method.

- sendBuyerRequest

This method checks the Buy Request. It looks for Product Specifications and Buyer's Preferences. It sends the Buyer's Request to only those Sellers who are active and which fulfill the buyer's preferences.

- receiveSellerOffer

Seller sends its offers against Buyer's Requests. These offers are received by the Facilitator.

- sendSellerOffertoBuyer

Facilitator sends the Offers to the respective buyer.

- maintainBuyerProfile

Facilitator maintains buyer profile. How many requests it had made and how many orders it had confirmed. Against this profile it assigns a rating to the buyer.

- maintainSellerProfile

Facilitator maintains seller profile. Facilitator assigns it a rating based on its past performance.

ii) Buyer

Activities performed by the Buyer are:

- signIn

Whenever the buyer signs in, it informs the Facilitator that it is active and it opens the communications channels.

- signOut

Whenever the buyer signs out, it informs the Facilitator that it is inactive and it will not be able to communicate.

- sendProfiletoAgency

Buyer sends its profile to Facilitator in which its basic information is kept, and this profile can be forwarded to seller if buyer agrees to it.

- sendRequest

Buyer sends request for a specific product. The parameters sent with the Request are quantity required, shipment time and relaxation in it(if any), price preference, and seller's rating.

- receiveOffer

Buyer receives offers from the sellers via Facilitator.

- cancelRequest

Buyer can cancel a request.

- chooseBestOffer

Buyer can evaluate and then choose best possible offer. Evaluation is done on the basis of preferences given by the end user.

- finalizeOrder

After choosing the best option, buyer confirms the order and then order is finalized. From this stage onwards buyer cannot cancel the order.

- storeConfirmedOrder

Buyer Agent also stores the confirmed order into its own repository for record keeping and future references.

iii) Seller

- signIn

Whenever the Seller signs in, it informs the Facilitator that it is active and it opens the communications channels.

- signOut

Whenever the Seller signs out, it informs the Facilitator that it is inactive and it will not be able to communicate.

- sendProfiletoAgency

Seller sends its profile to Facilitator in which its basic information is kept, and this profile can be forwarded to Buyer if seller agrees to it.

- receiveRequest

Seller receives request from the Buyer for a specific product via Facilitator.

- sendOffer

Seller sends the offer to the Buyer via Facilitator.

- cancelOffer

Seller can cancel an offer.

- finalizeOrder

Once buyer accepts Seller's offer, seller finalizes the order.

- storeConfirmedOrder

Seller Agent stores the confirmed order into its own repository for record keeping and future references.

AGENT INTERACTION PROTOCOL

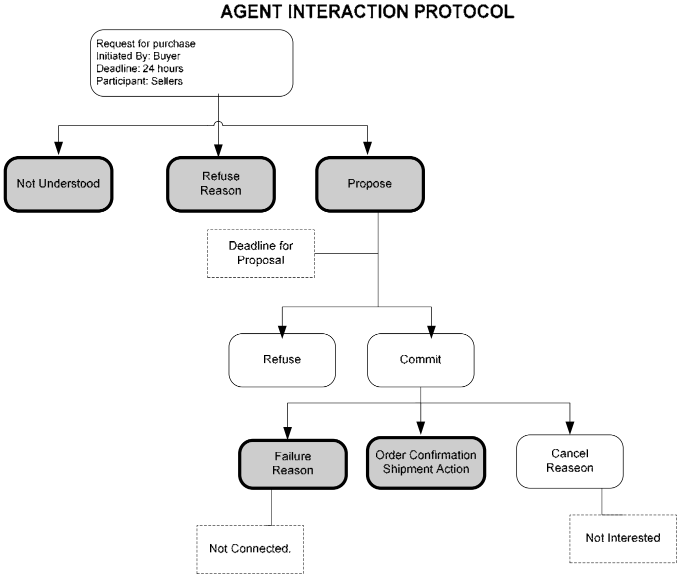


Figure 6

VII. UNDERLYING PROTOCOLS

- Sockets
- TCP/IP

VIII. DEVELOPMENT TOOLS

- JCreator (Java Editor)

- JDK 1.4
- Swing Components
- MS Access (RDBMS)
- Concept Building Tool: C#, Jack, Agent Builder[13]

IX. CONCLUSION & FUTURE DIRECTIONS

Just like any real-life business agents can work on behalf of customer by transforming provided information into knowledge. Buyer/Seller Agency is a system where users create agents to search and filter offers for the purchase and sale of goods on their behalf. We have built a simple prototype to test the basic concepts and feasibility. Future work is focused on making smarter agents which are direct able at a more natural level for users. Though we have only just scratched the surface in terms of making a truly useful system, we are excited about this work and think it has the chance to fundamentally change the way people buy and sell goods and services in the not-too-distant future.

REFERENCES

[1]P. Maes. “Agents that Reduce Work and Information Overload.” Comm. of the ACM, Vol:37, No:7, pp 31-40, July 1994.

[2] J. Bailey and Y. Bakos. “An Exploratory Study of the Emerging Role of Electronic Intermediaries.” International Journal of Electronic Commerce, Vol:1, No:3, Spring 1997.

[3] Karl Kurbel, Iouri Loutchko “A Framework for Multi-agent Electronic Marketplaces: Analysis and Classification of Existing Systems” Procedings of ICSC, 2001

[4] Marcel Albers, Catholijn M. Jonker, Mehrzad Karami, Jan Treur “Agent Models and Different User Ontologies for an Electronic Market Place” Inter. Journal Knowl. Inf. Syst. Vol:6 No.1, 2004

[5] Minghua He, Nicholas R. Jennings, Ho-Fung Leung “On Agent-mediated Electronic Commerce”, IEEE Trans. Know. & Data Eng. Vol:15 No.4, 2003

[6] Jie Zhang, Robin Cohen, “Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings”, Proc. Of 8th inter. Conf. on e-comm. ICEC '06., Vol. 156, Aug. 2006.

[7] Shiu-li Huang, Fu-ren Lin, “Designing intelligent sales-agent for online selling”, Proce. of 7th intern. Conf. on e-comm. ICEC '05, Vol. 113, Aug. 2005

[8] P. Mariano, A. Pereira, L.Correia, R. A. Ribeiro, V. Abramov, N.Szirbik, J. Goossenaerts, T. Marwala, P. De Wilde. “Simulation of a trading multi-agent system. Proc. IEEE SMC´01, TESADI´01, USA, Oct 2001.

[9] Kwang Hyoun JOO, Tetsuo knoshita, Norio Shirtori “Design and Implemenation of an Agent-Based Grocery Shopping System” IEICE Trans. on IS, Vol:E83-D, No:11, pp. 1940-1951, Nov 2000

[10] Anthony Chavez, Pattie Maes “Kasbah: An Agent Marketplace for Buying and Selling Goods” PAAM, 1996

[11] Pattie Maes, Robert H. Guttman, and Alexandros G. Moukas “Agents That Buy and Sell” Comm of the ACM, Vol:42, Iss: 3, 1999

[12] Robert H. Guttman, Pattie Maes, Anthony Chavez, Daniel Dreilinger “Results from a Multi-Agent Electronic Marketplace Experiment” Proceedings of Modeling Autonomous Agents in a Multi-Agent 1997

[13] An integrated toolkit for constructing intelligent software agents, “Agent Builder User Guide”, Version 1.4 June 2004.

[14] Stefan Poslad, “Specifying protocols for multi-agent systems interaction” ACM Trans. on Autonomous & Adaptive Sys.,Vol. 2, No. 4, Nov. 2007.