IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 3, March 2015.
www.ijiset.com
ISSN 2348 – 7968

A Research Paper on "Design of a Prototype of E-Commerce Application using Mobile Agent"

[1]Deepak Shukla, [2]Prof. Dr.Yash Pal Singh,

[1]Research Scholar, Sunrise University, Alwar. Rajasthan India.
[2]Professor, Sunrise University, Alwar. Rajasthan India.

Abstract:

[Mobile agent paradigm provides a cleaner design for several real life applications as compared to the traditional client server model. The mapping of implementation components to real life objects is direct while using mobile agent technology. In an ecommerce scenario, a mobile agent is launched from a buyer's site with a list of market to visit, the products to look for, the desired product attributes and some other user preferences. A real life problem of getting a product or a set of product can be directly mapped onto the mobile agent design paradigm, where in, real life you send an assistant with all these parSunRiseers to get the job done, here an agent act as an assistant. Compared to the client server design model where a real life scenario has to be forced to fit in a request-response model, a mobile agent is a better and more clear design technology.]

Keywords: [Mobile agent Technology, product attributes, Prototype,

1.0 Introduction: Present paper discuss about the implementation aspects of the prototype of E-Commerce application using mobile agent. It discussed about the implementation aspects and agent framework prototype , enetities involved in prototype, architecture of the proposed prototype model . It also include Voyagers named servers and shops. The activity of buying and selling among end-consumers is a particularly time consuming and inefficient form of shopping and often includes additional steps such as negotiating on price and other features. We believe that the effective use of mobile agents can dramatically reduce transaction cost involved in electronic commerce, in general, and in business-to-consumer transaction, in particular.

2.0 Implementation aspects:

We have implemented a complete business-to-customer e-commerce based application using mobile agent technology. The goal of our prototype was to gain practical experience with mobile agent in e-commerce as well as the software engineering aspect of mobile agent technology.

ISSN 2348 – 7968

In our prototype model we have created an e-market place for buying and selling of goods using both the traditional client server paradigm and the mobile agent paradigm. Ten e-shops were hosted which handle large database of products and understand both of the design paradigms. A buyer could use any of the implementation mechanisms to get a deal . The e-commerce application that we have choose is that of a user searching for a book with desired attributes across shops of interest. The product are represented in XML in a separate file and the product information are dealt using XML tags. By changing the XML file and tags the application could easily be used for any other product discovery.

2.1 Agent framework

We have used Voyager ORB as the mobile agent framework for our applications implementation. In Voyager an agent is a special kind of object that can move independently, can continue to execute as it moves, and otherwise behaves exactly like any other object. Voyager enables objects and other agents to send standard Java messages to an agent even as the agent is moving. In addition, Voyager allows to remote enable any Java class, even a third-party library class, without modifying the class source in anyway. Voyager also includes a rich set of services for transparent distributed persistence, scalable group communication, and basic directory services.

Voyager's API for agent mobility provides us with the common basic feature of agent mobility and hence allows a more customizable application to be developed using pure Java code.

2.2 Prototype Architecture

We have implemented a customer driven market place where, the system is based on pull model of marketing i.e., the buyer moves around several shops searching for a product. The entities involved in our prototype model are buyer, mobile agent, shops and Voyager's naming server. The buyer is an entity searching for a product of its interest at various shops. The shops maintain huge products catalog and support some basic services to access and operate on product catalogs which are stored as XML files.

A buyer interested in a product launches a mobile agent and provides it with a list of shops to visit, the product specification and product evaluation logic. The buyer's mobile agent visits each of the shops in it itinerary in the specified order. On arrival at a shop, the buyer's mobile

ISSN 2348 – 7968

agent contacts a stationary local agent to get the required product. The shop's local agent hands over the buyer's mobile agent to a local salesman agent, which deals with a particular category of products. The salesman agent uses local services to search the product catalog according to buyer's criteria and returns the result to the buyer's agent. The buyer's agent then uses its evaluation logic to evaluate the product from the filtered list which match best to his taste. The buyer agent rates each of its entries then carries this information along with it and hops on to the next shop in its itinerary. The buyers mobile agent also has a list of sites to visit in-order to support mobile devices which may be disconnected for small interval of time. On completion of its itinerary the buyer's MA returns back to the buyer's site and contacts the stationary agent and handover the information. The stationary agent then displays the result and the best deal to the user. It is then to the user to make a choice and go ahead with the purchase.

2.3. Entities involved:

The entities involved in our prototype are:

1.Voyager's naming server 2. Buyer 3. Mobile agent and 4. Shops

These entities have several components which constitute the whole system. Figure 20 show the components of our prototype model and their interaction

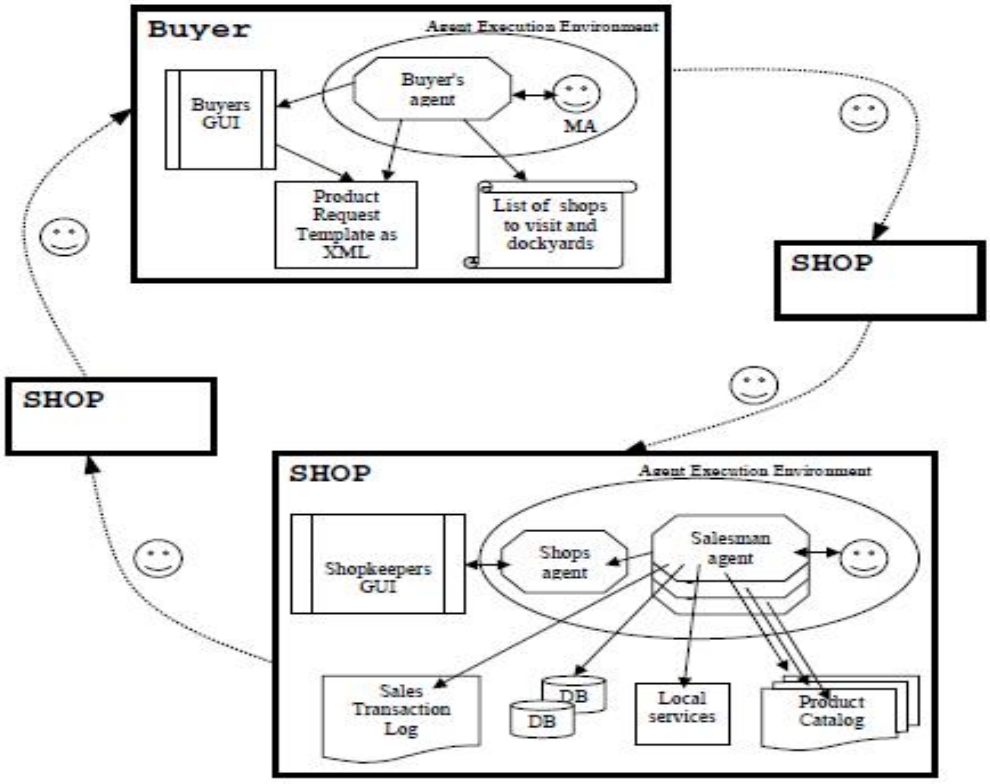

Figure : Architecture of Our Prototype Model

resources and services. A mobile agent visiting the buyer's site passes on his request to access some of the local resource or services to the stationary agent which in turn carries out the operation for the mobile agent. Buyer's stationary agent can thus also be used for security and authentication purposes. On user's submission of product parSunRiseers using the buyer's graphical user interface the information is collected by the buyer's stationary agent and stored in a data structure. The buyer's stationary agent then creates a buyer's mobile agent to get the product. The buyer's mobile agent is provided with a list of shops to visit, the product specification, and product evaluation logic and sent off to get the product.

2.3.1 Voyagers naming server:

Voyager contains an integrated directory service that allows associating an alias with any object and connecting to the object via its alias at a later time, even if it has moved. Voyager's directory service allows creating and connecting network directories together to form a large federated directory service. The shops and host in our prototype designed are registered with the Voyager's naming service and are identified by unique names.

2.3.2 Buyer.

1) Buyer: Buyer is the user who is willing to get a product from several online stores and isnot interested in searching for information all over the net. The buyer interacts with a graphical user interface and provides it with his preferences for the product. The submission in turn leads to the creation of a buyer's mobile agent to get the job done. A buyer site is registered with the Voyager's naming service and is identified by a unique name.

2) Buyer's Graphical User Interface (BGUI): The buyer's graphical user interface is theinterface between the buyer (user) and the stationary agent sitting at that host. A buyer interface allows the user to input parSunRiseers for a specified product. A buyer's-id is required to uniquely identify the buyer in the marketplace, this could be the buyer's email-id or some agreed upon protocol of identity assignment. For our prototype we use the buyer's email-id as buyer's-id.

3) Buyer's Stationary Agent (BSA): Buyer's stationary agent is the local agent the sit at thebuyer's site and is responsible for managing resources and service at the buyer's end. The buyer's stationary agent executes in the agent execution environment and is responsible for

ISSN 2348 – 7968

management of mobile agent i.e. agent creation, receiving and dispatching agent. The buyer's stationary agent is the interface for the mobile agent to access.

4) Agent Execution Environment (AEE): Voyager's ORB is our agent executionenvironment which provides support for agent creation, arrival, dispatch and agent management. Voyager is an ORB (Object Request Broker) implemented in Java and provides support for remote messaging, mobile objects and autonomous mobile agents.

5) Buyer's Mobile Agent (BMA): The Buyer's mobile agent is a Voyager's mobile agentwhich moves around the network visiting shops in its itinerary to get its job done. Buyer's mobile agent is created and launched from the buyer's site to search a product. The buyer's mobile agent is provided with a list of shops to visit, the product specification and product evaluation logic. The BMA visits all the shops in its itinerary and returns back to the buyer with the result. At each shop the BMA interacts with the shop's stationary agent and further with the salesman agent. The list of products that matches the user's parSunRiseers are stored in vector of product data structure and carried along with the mobile agent. On completion of its itinerary the mobile agent returns back to the buyer's site and handover the information to the buyer's local agent which in turns displays the result to the user.

6) Mobile Agent's Evaluation Logic (MAEL) : MAEL is the buyer's product evaluationcriteria which is carried along with the mobile agent and executed at each shop before adding a product offer to the mobile agent's product list. MAEL evaluates and rates products that match user's specification and carries the top three offers from the list. The MAEL provides user specific operation on the product catalogs, so a buyer is not restricted to just the operation provided at the shops. The MAEL is created at the buyer's end and attached to the mobile agent hence it helps in selecting products according to user choice.

7) Product Request Template (PRT): The PRT is the product parSunRiseers specified by theusers which are stored in XML data format structure and is attached to the mobile agent by buyer's local agent. At a shop the mobile agent passes on its product request XML tree to the salesman agent which in turn does the filtering of products form the product catalog which is stored as DOM tree.

ISSN 2348 – 7968

8) List of shops to visit: On agent creation the agent is provided with a set of shops to visit inits itinerary. Shop's name are registered at the Voyager's naming service and the mobile agent gets the network addresses of these shops using the Voyager's naming service.

9) List of host supporting disconnected operations: A buyers mobile agent also has a listof site to sit in case the host is a mobile device and faces small interval of disconnection. This provides support for disconnected operations for mobile devices. The mobile agents pings for the mobile host at regular intervals to see if it is connected.

2.3.3 Mobile agent

The mobile agent in our prototype is the buyer's mobile agent (BMA). The mobile agent is responsible for automating the whole task of shopping by acting on behalf of the user. The mobile agent move around different nodes in the network & accesses local resources at the site by communicating with the local agents. The mobile agent uses. Voyager's naming service to move across shops in our system. The mobile agent carries along with it, list of shops to visit, the product specification and product evaluation logic. In case of a mobile host the mobile agent has a list of hosts to sit when disconnected.

2.3.4 Shops:

1) Shopkeeper: Shopkeeper manages his shop at a given site. The shopkeeper is responsiblefor adding new products, catalogs. The Shopkeeper interacts with the system using the shopkeeper's graphical user interface. The Shopkeeper uses shopkeeper's graphical user interface to see the list of products at his shop and the transactions made at his shop. The shopkeeper keeps track of sales transaction with the help of sales transaction log, which is also displayed at the shopkeeper's graphical user interface.

2) Shopkeeper's graphical user interface (SGUI): The shopkeeper's graphical userinterface provides an interface to the shopkeeper to surf through its product catalogs. SGUI also provides information of transactions made at the shop. SGUI could also be used to add and modify product catalogs; in our prototype model we have not implemented functionalities of shopkeeper being able to add and modify its product catalogs.

3) Shop's stationary agent (SSA): Shop's stationary agent is the local agent the sits at theshopkeeper's site and in responsible for:

ISSN 2348 – 7968

- ✓ Mobile agent management i.e., receiving and dispatching mobile agents to / from its site. A buyer's mobile agent looking for a shop, first contacts the SSA. The SSA handles all request from the mobile agent and depending on requirement spawns salesman agent. Further interaction is between the salesman agent and the mobile agent.
- ✓ Interacting with the shopkeeper for product and catalog management.

The SSA also maintains a list of mobile agent visiting and currently active at its shop. All access to local resource and services by the mobile agent are passed on to either SSA or the salesman agent; which then execute it, for the mobile agent thus imposing security and authority restrictions.

4) Agent execution environment (AEE): Voyager's ORB is our agent executionenvironment which provides support for agent creation, arrival, dispatch and agent management. Voyager is an ORB (Object Request Broker) implemented in Java and provides support for remote messaging, mobile objects and autonomous mobile agents.

5) Shop's Salesman agent (SSmA): A SSmA is spawned on a mobile agents request to theSSA for accessing a specific product catalog. Each product catalog is maintained by a specific type of SSmA which has all the information about the product. The SSmA uses local service such as filtering, searching, etc. to serve mobile agents requests. The buyer's mobile agent operate on product catalog through SSmA which returns the buyer's mobile agent the resultant product sub-tree.

6) Product catalogs in XML: With regard to interoperability and platform independencerequired for the Internet application for information interchange. We have used XML as our data representation format for product catalogs. Catalogs of different products are maintained by different stationary salesman agents.

7) Local services: Each shops host some local service to provide access and operation tolocal resources. Local services could be availed by mobile agent by request to local agents. For our product catalog we have provided two local service of filtering and searching. Local services provides support and functionalities for salesman agents.

8) Sales transaction log (STL): Each shop maintains a STL which records user request andtransaction made by the user. This could help the shopkeeper study user's behavior.

ISSN 2348 – 7968

3.0 Features and suitability of our design

In the domain of e-commerce where, in the current scenario a lot of time is being spent on product discover, finding best deal and negotiating at different stages. These operations not only involve larger user interaction time but leads to extra network load and larger number of message interchange over the network. e-commerce applications require :

- e-commerce application demands faster response and real time decision making, for example in an auction house or a stock market the user would want to have the price changes as soon as possible, make his decision on the changed price and convey to the other party. In traditional computingmodels network delay and disconnected operation provide a majorhindrance to real time interaction.

- Also marketing is always customer driven i.e. a user buys goods to his requirements and specifications. Hence a need of customer specific queries and operation are must for e-commerce.

In current scenario the queries are limited to as that provided by the shops. Also with mobile devices being pervasive support for disconnected operation is must.

3.0.1 Data representation

The data representation format for our application is XML. XML represents data in a familiar (HTML-like) tagged textual form, and explicitly separates the treatment of data from its representation. This achieves the platform-independence required for the Internet and the well-appreciated feature of human-readability. In addition, XML can be made capable of representing whatever kind of data and entity one is likely to find in the Internet: complex documents, service interfaces and objects, communication protocols, and agents. These characteristics let us think that interoperability in the Internet will be information-oriented and based on XML, rather than service oriented and based on CORBA. With regard for inter-operability and as a communication mode for mobile agent the product catalogs, product requests and filtered product results are all represented in XML.

3.0.2 Static and mobile agents

We have distinctly separated and identified use of static and mobile agents in our design. We have static agents at the buyer's and shoppers' end which are responsible for handling mobile agents and mobile agent's request. Our static agent presently only manage mobile agents and resource at that site but they could be extended to being AI agent which study

ISSN 2348 – 7968

the buyers behaviour and then accordingly creates a mobile agent on behalf of the buyer. Similarly the idea could be extended to the shopkeeper's stationary agent which could mine the users transaction to identify user's buying behaviour.

3.0.3 Evaluation Logic

The Mobile Agent's Evaluation Logic (MAEL) is another strong and extendable feature of our prototype. MAEL is the users evaluation and rating criteria, for a product in request. This module is dynamically aggregated with the mobile agent and hence any time a change in preference is to be implemented new modules need to be attached.

3.0.4 Salesman agent

Use of salesman agent is a cleaner and better way of implementing shops. Different salesman agents are responsible for managing different product catalogs. A user (mobile agent) interested in a product need to contact that specific salesman agent. This help in providing different administrative right to different products.

3.0.5 Support for disconnected operations

Our mobile agent also maintains a list of host to sit in case it could not migrate itself to a mobile host from where it was launched because of temporary disconnection. This feature provides support for disconnected operations. The agent moves to any of these hosts (dockyards) and ping the destination at regular intervals to determine when the host is connected.

4.0. Results: (The e- market place)

We created an e-market place for buying and selling of books using both the traditional client server paradigm and the mobile agent paradigm. Ten e-shops were hosted which handle large databases of books and understand both of the design paradigms. A user interested in some book enters parSunRiseers such as, title of the book, publisher's name, author's name etc. A mobile agent is created then, which visit all the ten shops, searching for the book. At each shop the mobile agent contacts the shop's local agents. At each shop it evaluates the result and finally returns back to the buyer with the matching results.

ISSN 2348 – 7968

5.0. References:

1.W. C. Jakes (ed.), Microwave Mobile Communications. New York, NY: John Wiley &Sons, 1974.

2.J.-S. R. Jang, "Self-learning fuzzy controller based on temporal back-propagation," IEEETrans. Neural Networks, vol. 3, no. 5, pp. 714-723, Sept. 1992.

3.J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," IEEE Trans.Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 665-685, May 1993.

4.T. Kohonen, "The self-organizing map," Proc. IEEE, vol. 78, no. 9, pp. 1464-1480, Sep. 1990.

5.W. Pedrycz, "Fuzzy sets in pattern recognition: Methodology and methods," PatternRecognition, vol. 23, no. 1, pp. 121-146, 1990.

6.R. Roy, T. Furuhashi, and P. K. Chawdhry (eds.), Advances in Computer techniques for mobile computing: Engineering

7.Design and Manufacture. London, UK: Springer, 1998.

8.D. E. Rumehart and J. L. McClelland (eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, Cambridge, MA: MIT Press, 1986.

9. B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition,"Computer, vol. 21, no. 3, pp. 25-39, Mar. 1988.

10.B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron,madaline, and backpropagation," Proc. IEEE, vol. 78, no. 9, pp. 1415-1442, Sep. 1990.138

11.A. Y. Zomaya, "Reinforcement learning for the adaptive control of nonlinear systems,"IEEE Trans. Systems, Man, and Cybernetics, vol. 24, no. 2, pp. 357-363, Feb. 1994.

6.0 About The Authors.

1. **Deepak Shukla:** Deepak Shukla, A research scholar with SunRise University, Alwar Rajasthan Pursuing his Ph.D.Degree in Computer Science. He has attended Seminar and Conference on his topic. It is his third research paper.

2. **Prof. Dr. Y.P.Singh,**Guide and supervisor , working as Director in an engineering college. Has published almost 178 papers in international conference and Journals, His area of research is wireless and wifi techniques.

ISSN 2348 – 7968