IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 7 Issue 6, June 2020
ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72
www.ijiset.com

Mobile Agents Model For Online Booking In A Distributed Environment.

Afolabi Oluranti [1*], Ibochi Andrew Abah [1*] & Abah Sunday Ochepo [2*]

[1] Compusoft Institute of Information Technology, Bauchi, Bauchi State. Nigeria.

[2 & 3] Department of Surveying & Geoinformatics, Federal Polytechnic, Bauchi, Nigeria

ABSTRACT

For decades now, Mobile Agent systems have been discovered as a field that has future for the design and implementation of distributed applications. Before now, many important usefulness of Mobile agents have been identified such as information retrieval, automating looping tasks and workflow. To contribute to knowledge, Mobile Agents Model for Online Booking in a Distributed Environment(MAMOBIDE) has been developed. Java Agent DEvelopment Framework (JADE) was used as the platform, MySQL was used as database and Java Development Kit 8 was used as a programming language. This application is not like common software, the model is personalized, and independent, it navigates round the network looking for an item across different locations as specified by the owner(user). Performance evaluation of the Mobile Agents model and Client Server in[1] were carried out using turnaround time as performance metric. The first test shows that Mobile Agents model took turnaround time of 0.53secs to visit 13 shops and Client Server took 1.42secs. Other three test cases were also considered and results for Mobile Agents Model and Client Server were obtained as: (0.53 and 1.39)secs, (0.52 and 1.38)secs, (0.53 and 1.39)secs respectively. It was discovered that the turnaround times taken by Mobile Agent were less than Client Server because no communication of result to the ower's server until the whole 13 locations(shop) have been visited for online booking. Where as Client Server communicates each results of the visit to the owner's server and return to visit another location (shop). Mobile Agent model is therefore the best technology for implementation of e-commerce applications, since the execution is been moved closer to the users.

1. INTRODUCTION

Booking of items in an open market is dealing with confirmation of the market value of items in monetary terms and make an online reservation for it. There is a trend towards increasingly heterogeneous networks in today's communications environment. Managing these diverse networks requires the collection of large quantities of data from the network, which must be analysed before management activity can be initiated [6].

There are many ways to this approach starting from early method of physically moving from one shop to another to confirm the price of items. Most at times people move from one town to another, one state to another, one country to another etc in making enquiries on items of interest in order to make reservation for it. This type of method makes it impossible to have access to varieties of such items and to get best market price; it also encourages human influence for self interest reasons. Many organizations that are saddled with the responsibility of procuring items for people have been adopting the above method and later change to a conventional method by physically surfing the internet to search for items of their interest and make booking for them. This method involves the users sending request to the server whenever items are to be booked. The server in turn sends a response to the request and the

"handshaking" occurs again and again. Each request/response of this conventional approach requires a complete round trip across the network which consumes a lot of bandwidth. Therefore, there is need for a better method of booking and search has led us to Mobile Agent.

Mobile agents are already prevalent on the Internet, and are used for performing a variety of tasks such as collecting information, negotiating a business deal, or for online shopping and booking.

A mobile agent is a software or program that is able to migrate to some remote machine, [6], execute some functions or collect some relevant data and then migrate to other machines in order to accomplish another task. The basic idea of this paradigm is to distribute the processing throughout the networks, which is, sending the code to the data instead of bringing the data to the code. When the data needed for a computation is physically dispersed, it can be sometimes beneficial to move the computation to the data, instead of moving all the data to the code performing the computation. The use of mobile agents has been advocated for various reasons such robustness against disruptions in network connectivity, improving the network latency and reducing network load, providing more autonomy, and so on[9].

Having considered the Conventional method and Mobile Agent model it was discovered that Conventional method has a lot of limitations and constraints. Some of the challenges is as follows:

- The conventional way of booking items on the networks is the clients sending request to the server whenever the tasks are to be performed which consumes a lot of system resources such as bandwidth.
- Unrealistic Network Connection: Most of the time, the networks are not reliable, making it difficult to perform intended tasks. That is, the task has to be repeated several times because of connection problem.

2. Related Literature

Mobile agents are suite of programs that have the ability to migrate across the network participating in tasks dynamically assigned by a designer, a human or agent supervisor. They can start their execution at one location (initiation), suspend execution, resume execution, and roam wide area network, interacting with foreign hosts, gathering information on behalf of their owner and come 'back home' having performed the duties set by their owner [5]. This posed some benefits such as reduction of communication costs.

According to [1], mobile agents provide a new programming paradigm for building agile distributed systems. The ability to travel allows a mobile agent system to move computation to data source systems. This decentralized approach improves network efficiency since the processing is performed locally. For example, in a market survey and reporting which relates to the major focus of this research work is e-commerce scenario and is shown in figure 1 below; where searching for the product is a client server operation but a mobile agent is used to purchase products. Once the Buyer Server receives a buyer's purchase request, it sends it to the Information Server to search wholesalers and retailers who sell the product. The Buyer Server dispatches a mobile agent visiting these wholesalers and retailers; the mobile agent negotiates with seller's agents and reports the offers to the Buyer Server. The Buyer Server evaluates all the offers, and sends a purchase mobile agent to the best offer (seller) to make the final purchase.

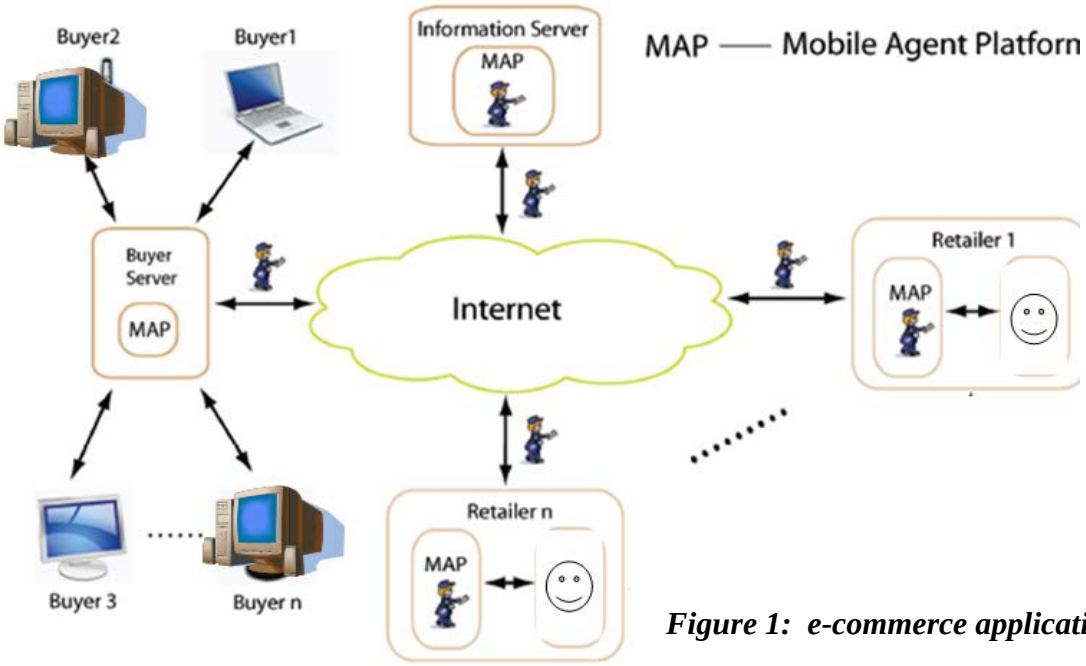

Figure 1: e-commerce application

Mobile agents were identified as a suitable tool for e-commerce [12]. A commercial transaction may require real-time access to remote resources such as stock quotes and perhaps even agent-to-agent negotiation. Different agents will have different goals, and will implement and exercise different strategies to accomplish these goals. Mobile agent technology is a very appealing solution to this kind of problem. An electronic commerce transaction may be viewed in terms of four different phases, *namely:* ***product brokering, merchant brokering, negotiation, payment and delivery*** [5].

2.1 Classification of Mobile Agents in E-Commerce

The application of Mobile Agents to electronic-commerce gives another dimension of conducting business-to-business, business-to-consumer, and consumer-to-consumer transactions. The existing Mobile Agents applications in electronic-commerce are categorised into three, namely; shopping agents, salesman agents, and auction agents[8].

Shopping agents: These are Mobile Agents that buys in e-marketplaces on behalf of their owner as specified by the user. A shopping agent visits several online stores, compare features of different products and report the best choice to its owner. The Mobile Agent's owner specified set of features to be considered and their ideal values and carries it along as It may be given one or more sites to visit and may dynamically visit other sites based on subsequent information. Mobile Agent goes to the source of information; therefore overhead repeatedly transferring potentially large amounts of information over a network is eliminated. One example of a system that implements shopping agents is Mobile Agents for Networked Electronic Trading (MAgNET), where agents deal with procurement of the many components needed to manufacture a complex product [9].

2. Salesman agents

These Mobile Agents behave like a travelling salesman who visits customers to sell his wares. This model of e-commerce uses a supplier driven marketplace and is particularly attractive for products with a short shelf-life. A supplier creates and dispatches a Mobile Agent to potential buyers by giving it a list of sites to visit. The Mobile Agents carries with it information about available stock and price of the product. [5].

3. **Auction agents:** These categories of Mobile Agents bids for and sell items in an online auction on behalf of their owners. Each of the Mobile Agent carries along its bidding information as specified by the owner; for example bidding range, time within which the item is to be procured, bidding pattern, and other relevant attributes.[10].

2.3 Mobility Patterns

Mobility in Mobile Agents can be characterized by the set of destinations that Mobile Agent visits, and the order in which it visits them. Therefore, the following parameters to characterize the mobility of Mobile Agent are identified: [7].

(1) **Itinerary:** the set of sites that an Mobile Agent has to visit. This could either be statically fixed at the time of agent initialization, or dynamically determined by the Mobile Agent.

(2) **Order:** the order in which a Mobile Agent visits sites in its itinerary. This may also be determined statically or dynamically.

2.4 The Lifecycle of a Mobile Agent

Agents have a well-defined six(6)lifecycles . The figure 2 below illustrates the states that made up these lifecycles.

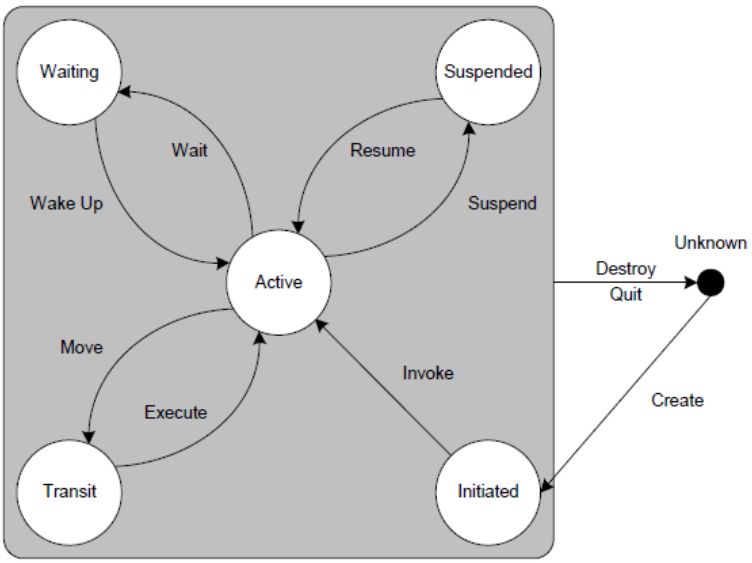

Figure 2 - Agent life-cycle (Source: **[4]**)

- **Initiated:** the Agent object is built, but hasn't registered itself yet with the AMS, has neither a name nor an address and cannot communicate with other agents.
- **Active:** the Agent object is registered with the AMS, has a regular name and address and can access all the various JADE features.
- **Suspended:** the Agent object is currently stopped.
- **Waiting:** the Agent object is blocked, waiting for something. (waiting till some conditions are met.
- **Deleted:** the Agent is dead and the is no more registered with the AMS.
- **Transit:** a mobile agent enters this state while it is migrating to the new location.

Many authors have been making use of internet to improve e-commerce([2]; [11]; [8]; [3])

3. METHODOLOGY

3.1 System Analysis and Proposed System modelling

The traditional method of procuring items is to move from one vendor to another in making enquiries on items of interest in order to make reservation for it**.** This is a very slow process and time consuming. The next method of procurement is physically surfing the internet to search for items of interest which involves the users sending request to the server whenever items are to be procured which consumes a lot of system resources such as bandwidth. Therefore, there is need for more efficient and effective method of procuring items which is Mobile Agent.

Buyer who is interested in acquiring 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, mobile agent contacts a stationary local agent to get the required product. The shop's local agent hands over the 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 a given criteria and returns the result to the its agent.

Moreover, the agent then uses its evaluation logic to evaluate the product from the filtered list which match best to his taste and make booking for it for a particular period as it may be allowed by each online shops. The agent rates each of its entries then carries this information along with it and move on to the next shop in its itinerary.

On completion of sites in its itinerary, it returns back to the it's site and contacts the stationary agent and handover the information. The stationary agent then displays the results to the user who will decide when the purchase of the booked items will be made.

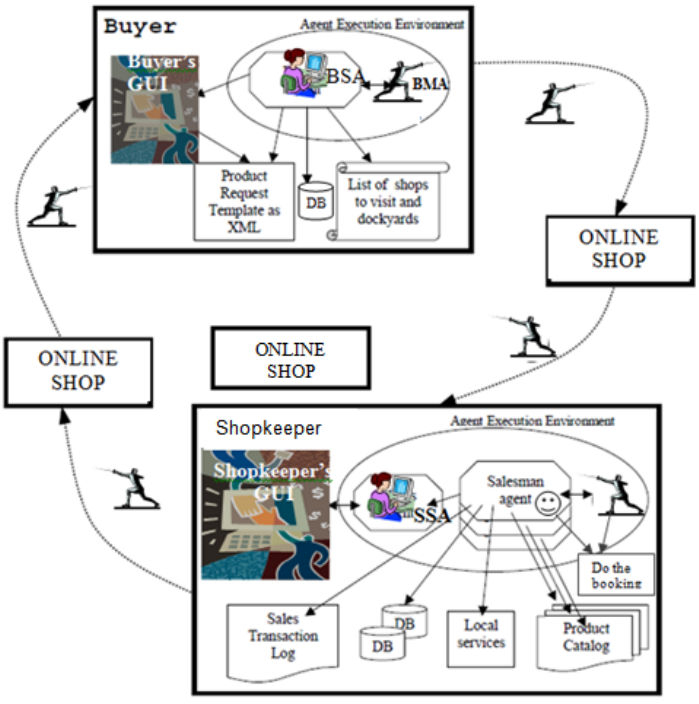

Figure 3: Proposed System Architecture

3.3 System Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.

1. Database Design

In this paper, the buyer maintains a single database at the buyer's end. Once a buyer supply his name or company name, the product to book, minimum rating, maximum price per product, maximum budget and maximum product count then agent can go ahead for search.

At the seller's end, because they are at different locations, they have different databases. Product category, product name, rating, price and product sold are parameters in each of the seller's database.

Figure 4 below shows the Entity Relationship diagram for the system.

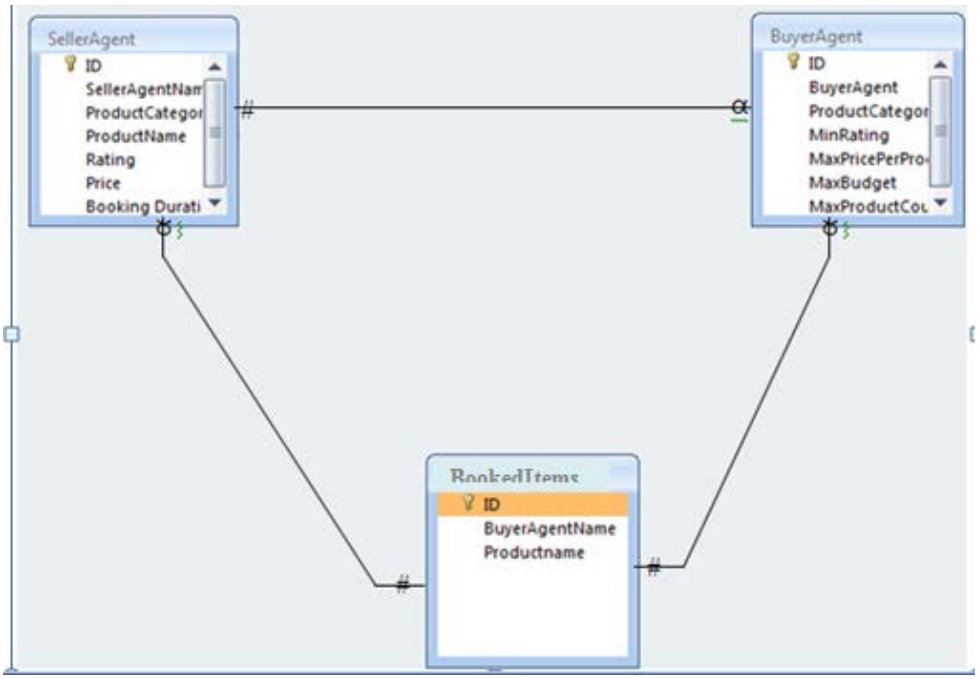

Figure 4: Entity Relationship diagram for MAMOBIDE

2. Program Design

Algorithm for agent's behavior in the market

```
1       /* Agent creation */
        private BuyerAgent agent = this;
          private BuyerGui ui;
          Logger.info(agent, "Creating Buyer Agent UI...");
        /* Buyer provides criterion for booking to be made */
2.       this.category = category;
        this.maxprice=maxprice
       this.minRating = minRating;
        this.prodcount=prodcount
```

this.maxBudgetPerProduct = maxBudgetPerProduct;
this.mail=mail
/*Test if buyer's category name e.g Laptop, Desktop, Tablets etc is available in seller's shop */
3. if (selectedcategoryname = sellercategoryname) then goto 9
4. if (buyerproductname=sellerproductname) then goto 9
5. if(buyeragentrating>=selleragentrating) then goto 9
6. if(buyerprice<=sellerprice) then goto 9
7. totalamout=maxprice*prodcount
8. if(totalamount>maxBudgetPerProduct) then
Logger.info("Insufificient fund") goto 10
9. call booking subprogram and book the buyer agent
10. if(ShopsInBuyerAgentItenerary=0)then goto 11
Else goto 2
11. Display Results
12. End

4. RESULTS AND DISCUSSIONS

4.1 Results

Getting Started

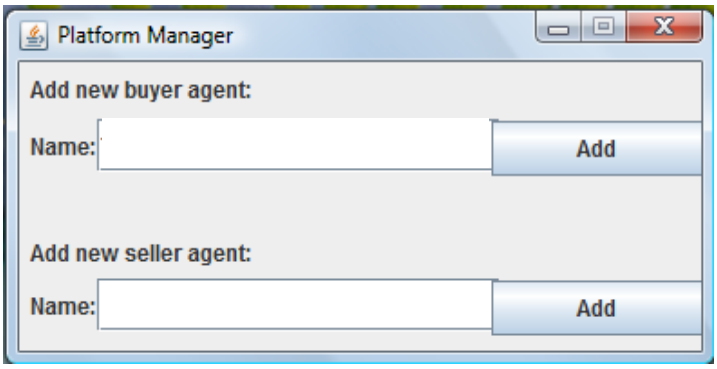

Figure 5: How Buyer Agent is being created

The name of the buyer agent will be typed as shown in the figure 5 above, then "Add" will be clicked for that agent to be created.
Once agent is created, the next thing is to supply the specifications of the product to search and book. i.e input data

Figure 6: input data for the buyer agent

From the above; “category” is Laptop. In the category combo box, there are other types of computer like Desktop, Tablet etc.

From the figure 6 above, after category might have been selected the next thing is “Mini rating” that is minimum rating which is the user’s assessment of a particular item in term of quality. An item is been rated by assigning number that correspond to the qualitative performance of the item for instance “0” is bad, “1”is fair, “2” is Better, “3” is Good, “4” is Very .good and “5” is Best. For the above input data(figure 6), Minimum rating is 2. Furthermore, “Maxi price per product” is highest amount that the buyer can buy the product. Also, “Max budget” is total amount of money a buyer has reserved for the transaction. “Max product count” is the total quantity of the product a buyer intends to buy. Lastly, e-mail is the buyer’s personal contact through which seller agent can to communicate to the buyer at the expiration of the booking.

Once criteria are entered and “Find” is clicked, agent will go round visiting one online shop to another with the specified criterions and come back with the booking reports. Four test cases were carried out using 13 shops and results obtained were compared with [1].

Performance Evaluation Results of Mobile Agent model and Bo (2010)’s Client Server)

The performance metric considered is the user turnaround time, which is the time elapsed between a user initiating a request and receiving the results and this includes the time taken for agent creation, time taken to visit, time for booking and the processing time to extract the required information.

Also, the following parameters for comparing the performance of Mobile Agent model and Client Server were considered:

- number of stores (varies from 1 to 13);
- processing time for servicing each request;

System was timed to record time taken for agent creation(t_c), time taken to visit the shop(t_v), time for booking(t_b), processing time to extract the required information(t_e) and time taken to return back to buyer’s server(t_r).

Turnaround time $(t_t) = t_{c+} t_{v+} t_{b+} t_{e+} t_r$

The results obtained from the simulated model were presented in the tables 1 to 4 below. Summary of the results also presented in table 5 below.

			No Communication	**Communication**
Results of Test case 1	Booking Duration Allowed	No. of Shops Visited	MA	CS
HP 4.0 GHZ PROCESSOR, 8.0GB RAM[LAPTOP] [R:4.0] [P:175000]@EMMAPET NIG. LTD. SONY 3.0 GHZ PROCESSOR, 4.0GB RAM [LAPTOP] [R:4.0] [P:100000]@DANBAUCHI STORE DELL 4.0 GHZ PROCESSOR, 4.0 GB RAM [LAPTOP] [R:4.0] [P:60000]@LUKA NIG. LTD. LENOVO 2.0 GHZ PROCESSOR, 4.0GB RAM [LAPTOP] [R:5.0] [P:122000] @ A. A. WAZIRI NIG. LTD. Turnaround time	48 hours 12 hours 24 hours 72 hours	0 1 2 3 4 5 6 7 8 9 10 11 12 13	0 0.26 0.28 0.29 0.31 0.32 0.34 0.36 0.38 0.40 0.42 0.44 0.51 **0.53**	0 0.16 0.41 0.45 0.50 0.57 0.66 0.70 0.76 0.90 1.01 1.21 1.30 **1.42**

Table 1. Effect of communication on turnaround time for Mobile Agents model and Client Server for Test Case 1

MA-------->Mobile Agent
CS---------->Client Server

			No Communication	**Communication**
Results of Test case 2	Booking Duration Allowed	No. of Shops Visited	MA	CS
TOSHIBA 2.0 GHZ PROCESSOR, 4.0GB RAM [LAPTOP] [R:4.0] [90000]@A. A. WAZIRI NIG. VENTURE. IBM 2.0 GHZ PROCESSOR, 4.0 GB RAM [LAPTOP] [R:4.0] [P:60000]@ADAMU NIG. LTD.	48 hours 24 hours	0	0	0
		1	0.26	0.15
		2	0.28	0.40
		3	0.29	0.44
		4	0.31	0.49
		5	0.32	0.56
		6	0.34	0.65
		7	0.36	0.69
		8	0.38	0.75
		9	0.40	0.90
		10	0.42	0.99
		11	0.44	1.20
		12	0.51	1.29
Turnaround time		13	**0.53**	**1.39**

Table 2. Effect of communication on turnaround time for Mobile Agents model and Client Server for Test Case 2
MA-------->Mobile Agent
CS---------->Client Server

			No Communication	**Communication**
Results of Test case 3	Booking Duration Allowed	No. of Shops Visited	MA	CS
No Result	No Booking	0	0	0
		1	0.25	0.13
		2	0.27	0.38
		3	0.28	0.42
		4	0.30	0.47
		5	0.31	0.54
		6	0.33	0.63
		7	0.35	0.67
		8	0.37	0.74
		9	0.39	0.86
		10	0.41	0.98
		11	0.43	1.17
		12	0.50	1.26
Total Turnaround time		13 →	**0.52**	**1.38**

Table 3. Effect of communication on turnaround time for Mobile Agents model and Client Server for Test Case 3

MA-------->Mobile Agent
CS---------->Client Server

			No Communication	**Communication**
Results of Test case 2	Booking Duration Allowed	No. of Shops Visited	MA	CS
HP 4.0 GHZ PROCESSOR, 8.0GB RAM [LAPTOP] [R:4.0] [70000]@EMMAPET NIG. LTD.	48 hours	0	0	0
		1	0.26	0.15
		2	0.28	0.40
		3	0.29	0.44
		4	0.31	0.49
IBM 2.0 GHZ PROCESSOR, 4.0 GB RAM [LAPTOP] [R:4.0] [P:60000]@ADAMU NIG. LTD.	24 hours	5	0.32	0.56
		6	0.34	0.65
		7	0.36	0.69
		8	0.38	0.75
		9	0.40	0.90
		10	0.42	0.99
		11	0.44	1.20
		12	0.51	1.29
Turnaround time ⟶		13	**0.53**	**1.39**

Table 4. Effect of communication on turnaround time for Mobile Agent model and Client Server for Test Case 4

MA--------->Mobile Agent
CS---------->Client Server

The table 5 below shows summary of tables 1- 4 above for turnaround times of visiting 13 shops for Mobile Agents model and Client Server.

Table 5. Turnaround times of visiting 13 shops for Mobile Agent model and Client Server for Test Cases 1 to 4

	When there is No Communication	**When there is Communication**
Test Cases	MA	CS
Test Cases 1	0.53	1.42
Test Cases 2	0.53	1.39
Test Cases 3	0.52	1.38
Test Cases 4	0.53	1.39

Discussion on Performance Evaluation

From results shown in the tables 1 to 4, some of our observations are:

- The performance of Mobile Agents model with turnaround time remains the same (except the scenario where there is no booking at all) while performance of Client Server varied. From table 5, test case 1 shows that Mobile Agent model took turnaround time of 0.53secs to visit 13 online shops, while Client Server took turnaround time of 1.42secs to visit the same number of shops.
- Test case 2 shows that Mobile Agent model took turnaround time of 0.53secs to visit 13 shops while Client Server took turnaround time of 1.39secs to visit the same shops.
- Test case 3 shows that Mobile Agent model took turnaround time of 0.52secs to visit 13 shops, while Client Server took turnaround time of 1.38secs to visit the same number of shops.
- Test case 4 shows that Mobile Agent model took turnaround time of 0.53secs to visit 13 shops while Client Server took turnaround time of 1.39secs to visit the same shops.

Looking at work done by [1] in his research, the Buyer Server receives a buyer's purchase request, and sends it to the Information Server to search wholesalers and retailers who sells the product. The Buyer Server generates and dispatches a mobile agent visiting these wholesalers and retailers, the mobile agent then visits and reports the offers to the Buyer Server. The Buyer Server then evaluates all the offers.

Comparing our Mobile Agents Model with his work, In our own work, once the mobile agent is created, it is the agent that will do the searching without depending on the outcome of information server. This simply means that our agent is autonomous and it makes process to be faster since agent doesn't need to wait until information server searches the shops that sells the product. Further more, in the model developed, Mobile agent has Evaluation Logic. Mobile Agent Evaluation Logic

evaluates rates products that match user's specification and carries the offer that is within its specification from the list of items in the shop. Therefore, there is no need to carry obtained result to the buyer server for evaluation any more as agent has taken care of that at different shops. Moreover, the model is tolerant to network failures and support disconnected operation. It can operate without an active connection between the destination and the home host, therefore the problem of unrealistic network connection is solved. Lastly, as the model developed is moving to each shop the number of information exchange is not over the network, it is local; therefore saving network latencies and load; this solves the problem of bandwidth consumption.

Observation from Comparism.

From the above comparison, it is clear that our agent was able to carry out all the tasks from the beginning to the end without intervention. This simply means that our mobile agent is more intelligent. The turnaround time taken by Mobile Agent was less than Client Server because no communication of result to the buyer server after each visit, communication of result was done once, that is when whole 13 shops has been visited. Where as Client server communicates each results to the buyer server and return to visit another shop. Furthermore, Client Server implementations are recommended for applications where a little amount of information has to be brought out from few remote information sources, and most importantly when the level of processing required is not high (above tables 1 to 4 shows clear indication of these parameters). Nevertheless, these conditions may not be effective for most real world electronic commerce applications. Mobile Agents performs more brilliantly across the above parameters; therefore, it was discovered that Mobile Agents model is an exact technology for building efficient electronic-commerce applications.

5. CONCLUSION

This model was able to search, book items and communicate back the expiration time of booking to the buyer via e-mail. The results of the search were obtained and compared with [1]'s Client Server. It was discovered that Mobile Agent model's turnaround time is less than that of Client Server, which makes Mobile Agent model better than Client Server.

From the result obtained through different scenarios in the test cases, it is very clear that the model developed is very intelligent and efficient.

References

[1] **Bo C. (2010),** Mobile Agents in E-Commerce. *Michigan Technological University, USA. pp. 846-847.*

[2] **Chavez A. and Maes P.(1996),** Kasbah: An agent marketplace for buying and selling goods. *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK (April 1996), pp. 75-90.*

[3] **Doorenbos R., Etzioni O. and Weld D.(1997)**, A scalable comparison/shopping agent for the World Wide Web. *Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA.*

World Wide Web Journal.

[4] **Fabio B, Giovanni C., Tiziana T. and Giovanni R.(2010)**, JADE programmer's Guide, *University of Parma.*

.

[5] **Maes, P., Guttman R.H. and Moukas A.G(1999).**: Agents that Buy and Sell: Transforming Commerce as we Know It. *In Communications of the ACM, Vol.42, No.3, pp.81- 91.*

[6] **Pagurek B., Yanrong L., Andrzej B. and Gatot S.(1998),** Network Configuration Management In Heterogeneous ATM Environments. *Proc. of the Int'l Workshop on Agents in Telecommunications Applications (IATA '98), AgentWorld'98, Paris, France, July 4-7.*

[7] **Rahul J**., *Mobile Agents for e-commerce,* Roll No:99329011. www.it.iitb.ac.in/~sri/students/rahul-thesis.pdf *retrieved on 20th Nov., 2014.*

[8] **Rodriguez J, Noriega P., Sierra C. and Padget J. (1997),** FM96.5: A Java-based electronic auction house. *Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, London, UK , pp. 207-224.*

[9] **Todd P.(2000),** Mobile Information Agents for CyberSpace - State of the Art and Visions.*To appear in Proc. of Cooperative Information Agents (CIA- 2000).*

[10] **Thomas S. and Qianbo H.(2000)**, "Nomad: Mobile Agent System for an Internet-Based Auction House ", IEEE Internet Computing , vol. 4, no.2.

[11] **Tsvetovatyy M., Mobasher B., Gini M. and Wieckowski Z. (1997)**, MAGMA: An agent based virtual market for electronic commerce. *Applied Artificial Intelligence, vol. 11, no. 6, pp. 501-523.*

[12] **Wenna L. and Deli Y.(2004),** Research on Mobile Agent-based E-Commerce System Framework. *The Fourth 440 International Conference on Electronic Business (ICEB2004) / Beijing. PP 440-444.*