**ISBN** 978-93-84468-20-0
**Proceedings of 2015 International Conference on Future Computational Technologies (ICFCT'2015)**
*Singapore, March 29-30, 2015, pp. 211-217*

# Mobile Agent-based Information Retrieval for Shopping Assistant

Maw Min[1], and NyeinNyein Oo[2]
Department of Information Technology Engineering, Yangon Technological University, Yangon, Myanmar
[1] aaaassss56@gmail.com, [2]nno2005@gmail.com

***Abstract*****:** *The information overload is an obstacle to the practical use of potentially useful information on the Web. The use of mobile agents in this kind of applications represents a novel approach and potentially solves most of the problems that exist in centralized client-server solutions. A mobile agent is an executing program that can migrate during execution from machine to machine in a heterogeneous network. On each machine, the agent interacts with stationary service agents and other resources to accomplish its task. Mobile agents are particularly attractive in distributed information retrieval applications. By moving to the location of an information resource, the agent can search the resource locally, eliminating the transfer of intermediate result across the network and reducing end-to-end latency. This paper presents a mobile agent application for the retrieval of distributed information in a scenario of several on-line shops. This system will be developed for Web-based distributed access to database systems.*

***Keywords:*** *Mobile Agent, Distributed Information Retrieval, Shopping Assistant*

## 1. Introduction

In traditional Client-Server way, most of the information retrieval systems do not offer enough flexibility for distributed data repositories. Generally, there are many specifications in traditional way; setting up a connection between the client and server, sending a request to the database server and receiving the result from the server. If there are N servers in the network, the user has to start N network connections and send out N database queries. The following information is needed to transfer on the network:

- Database query request and
- The result data.

The World Wide Web is rapidly being accepted a universal access mechanism for network information. The popularity of Web suggests that web browsers may offer a compelling end-user interface for a large class of applications including Image Retrieval. Mobile agent technology is used as a useful and efficient tool for searching and retrieving data in distributed environment where the data is stored at a various nodes of the system. The advantage of mobile agent is that it searches for information instead of users. Mobile agents carry code of execution or query to be fired on database.

The Distributed Information Retrieval task deals with the collection of information from multiple and usually heterogeneous information sources that exist in a distributed environment [8]. So Mobile Agent based Distributed Information Retrieval System provides solutions to the problems not to be solved by any of them [3] and [4] . Mobile agents follow the steps specified in a predefined process and network traffic, low network latency, disconnect operation, etc. It has the unique ability to transport itself from one system in a network to another. By exploiting the advantages of the agent's mobility, they are especially useful in distributed

information retrieval system. The Search Process gets the information about the papers from the remote distributed databases.

Mobile agents have several strengths. First by migrating to the location of a needed resource an agent can interact with the resource without transmitting intermediate data across the network conserving bandwidth and reducing latencies. Similarly, by migrating to the location of a user an agent can respond to user actions rapidly. In either case the agent can continue its interaction with the resource or user even if network connections go down temporarily. These features make mobile agents particularly attractive in mobile computing applications, which often must deal with low-bandwidth, high-latency and unreliable network links.

Second, mobile agents allow traditional clients and servers to offload work to each other, and to change who offloads to whom according to the capabilities and current loads of the client, server and network. Similarly, mobile agents allow an application to dynamically deploy its components to arbitrary network sites, and to re-deploy those components in response to changing network conditions. The Distributed Information Retrieval task deals with the collection of information from multiple and usually heterogeneous information sources that exist in a distributed environment. One way to address these issues is to use information agents.

Finally, most distributed applications fit naturally into the mobile-agent model, since a mobile agent can migrate sequentially through a set of machines, send out a wave of child agents to visit machines in parallel, remain stationary and interact with resources remotely, or any combination of these three extremes.

This paper presents a mobile agent application for the retrieval of distributed information in a scenario of several on-line shops. Multiple Attribute Utility (MAUT) function is used to find the best match with user preferences.

## 2. Background Theory

An agent is software that becomes an extension of the user, performing task on the user's behalf. The word 'agent' is currently in vogue in the popular computing press and within the artificial intelligence and computer science communities. There is no universally accepted definition of the notion of agent. However, the following four properties are widely accepted to characterize agents: autonomy, social ability, reactivity and proactiveness. Agents are autonomous computational entities (autonomy), which interact with their environment (reactivity) and other agents (social ability) in order to achieve their own goals (proactiveness). Agents typically represent different users, and there are thus several of them in a given environment. A multi-agent system is considered as a collection of collaborating autonomous agents, each representing an independent locus of control. Agent has following properties:

- Autonomy,
- Temporal continuity,
- Reactivity,
- Goal driven,
- Mobility (in mobile agents)

### 2.1. Mobile Agent

The term software agents refer to programs that perform certain tasks on behalf of the user. Software agents can be classified as static agents and mobile agents. Static agents achieve the goal by executing on a single machine. On the other hand, mobile agents migrate from one computer to another and executes on several machines. Mobility increases the functionality of the mobile agent.

A mobile agent consists of the program code and the program execution state. Initially a mobile agent resides on a computer called the home machine. The agent is then dispatched to execute on a remote computer called a mobile agent host. When a mobile agent is dispatched the entire code of the mobile agent and the execution state of the mobile agent is transferred to the host. The host provides a suitable execution environment

for the mobile agent to execute. Another feature of mobile agent is that it can be cloned to execute on several hosts. Upon completion, the mobile agent delivers the results to the sending client or to another server.

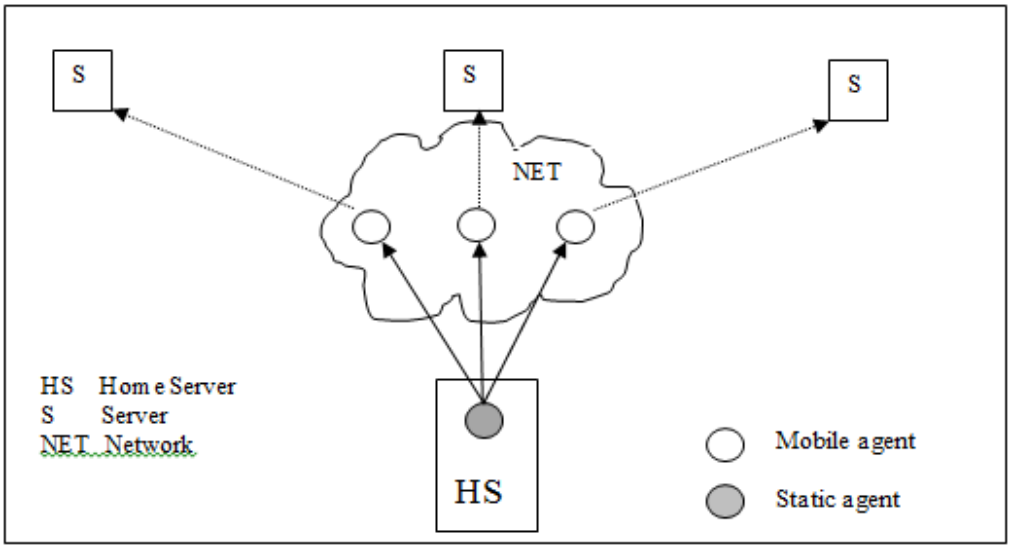


Fig. 1: Mobile Agent Design

## 2.2.Java Mobile Agent Technology

Aglet Technology is a framework for programming mobile network agents in Java developed by the IBM Japan research group. The IBM's mobile agent is called 'Aglet', is a lightweight Java object. One of the main differences between an aglet and the simple mobile code of Java applets is the itinerary or travel plan that is carried along with the aglet. By having a travel plan, aglets are capable of roaming the Internet collecting information from many places.

An aglet can be dispatched to any remote host that supports the Java Virtual Machine. This requires from the remote host to pre-install Tahiti, a tiny aglet server program implemented in Java and provided by the Aglet Framework. An aglet can be dispatched to any remote host that supports the Java Virtual Machine. This requires from the remote host to have preinstalled a tiny aglet server program implemented in Java and provided by the Aglet Framework. A running aglet server listens to the host's ports for incoming aglets, captures them, and provides them with an aglet context in which they can run their code from the state that it was halted before they were dispatched. Within its context, an aglet can communicate with other aglets, collect local information and when convenient halt its execution and be dispatched to another host. An aglet can also be cloned or disposed. An aglet world has three components:

- an aglet viewer,
- an aglets server, and
- the aglets themselves.

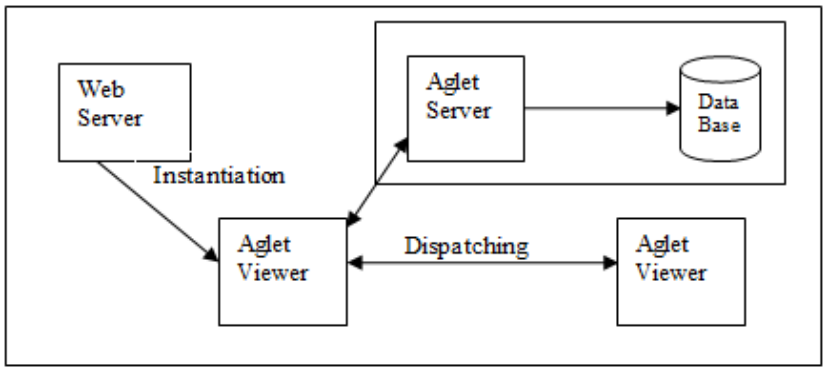


Fig. 2: Java Aglet Architecture

## 3. Related Works

Software agents have become very popular in recent years. They have been used successfully to filter information, match people with similar interests and automate repetitive behaviour. More recently, the capabilities of agents have been applied to electronic commerce, promising a revolution in the way people conduct transactions. One of these examples is the Andersen Consulting's Bargain Finder [7] and [1]. This is an automatic search tool that helps customers to find the best prices on CDs. Customers may specify what they are looking for and Bargain Finder searches nine different retailers to find the best price. A complete summary is returned where the customer can see the prices for all retailers and simply pick an option to be sent directly to the corresponding retailer. This system simply tries to ease the burden for customers to find the best price without manually going around to a number of retailers and compare prices.

Like Bargain Finder and others systems based on collaborative filtering technology [2], Firefly [6] helps consumers find products. However, instead of filtering products based on features, Firefly recommends products via an automated "word of mouth" recommendation mechanism called collaborative filtering. The system first compares a shopper's product ratings with those of other shoppers. After identifying the shopper's "nearest neighbours" (i.e., users with similar taste), the system recommends products that neighbours had rated highly but which the shopper may not yet have rated, potentially resulting in serendipitous finds. Essentially, Firefly uses the opinions of like-minded people to offer recommendations. The system is used to recommend commodity products such as music and books, as well as harder to characterize products such as restaurants' web pages. In [5], mobile agents are used for distributed database access.

## 4. Proposed System

This section presents the design of the proposed system. The main objective of the proposed system is the design of searching laptops, computers and handsets based on user preferences. Mobile agent technology is used to get distributed information at the travel radio station over the network. Mobile agents can migrate from a host node to various destinations, perform data processing there and send the relevant information back to the host. Java mobile aglet, carrying the user preferences, travels the radio station, compute similarity and get back with required information to base client. In this system; distributed information retrieval agents have following processes:

- accept a request from user,
- translate this request into a language computable by agents,
- identify the distributed information servers that contain information relevant to the request,
- surf around to those sources and search the required match item,
- collect the corresponding the results
- process the returned results and
- present the results to the client user

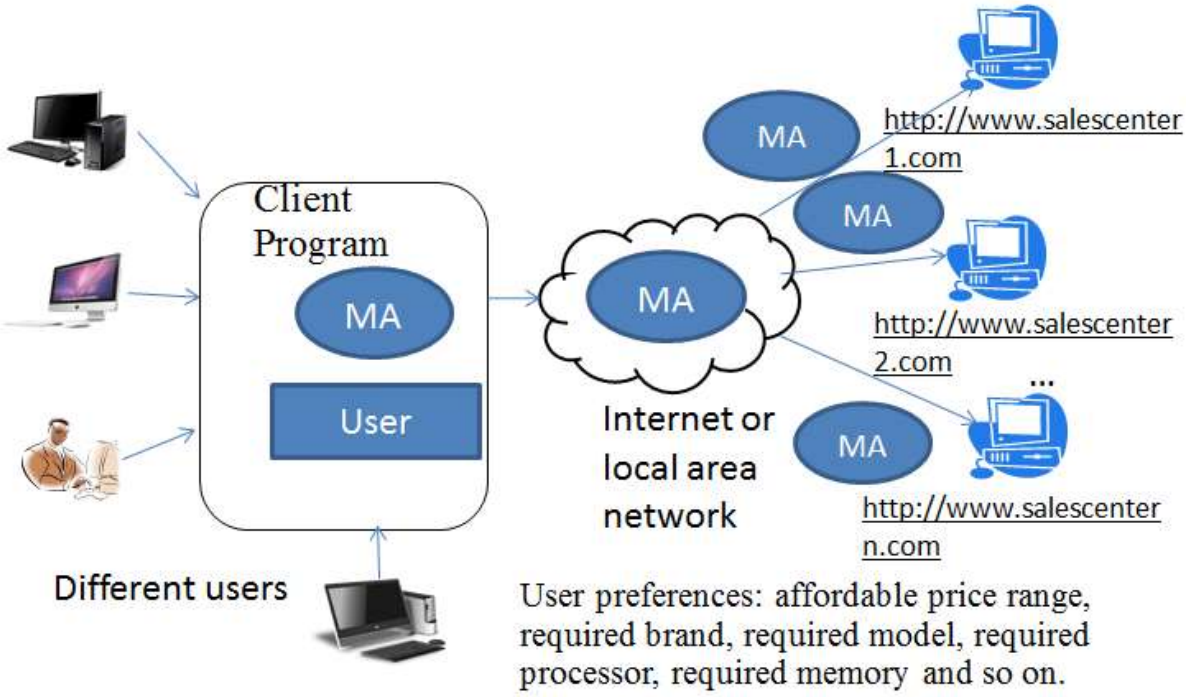


Fig. 3: Proposed System Design

Multiple Attribute Utility function (MAUT) is used as mobile code carried by aglets. Multi-Attribute Utility Theory (MAUT) is a decision-making method used when the decision maker has to take multiple objectives into account. MAUT [9] is widely used for the measurement of subjective judgments in various fields of a decision making. A structured methodology designed to handle the tradeoffs among incomparable and conflicting multiple objectives, captured by multiple attributes (e.g: better performance and lower price of computer). The basic goal of MAUT is to substitute information with an arbitrary measure called utilities. The method is recommended when prospective alternatives must be evaluated to determine which alternative perform best. The process of MAUT involves these stages:

- Identify the attributes, which collectively describe the overall utility of all relevant decision options.
- Identify the alternative.
- Weight the attributes in terms if their importance. (for example, if user sets price in most important; price attribute is set highest).
- Transform the attribute scores, measured in different unit, into similar measurable standard units.

Additive Utility function is as follows:

- A Simplified Utility Model
  - Ignores interactions among attributes
- For a consequence set that has values $x_1$, $x_2$, …, $x_m$ on the attributes of $m$ objectives, its overall utility is computed as

$$\mathrm{U}(x_1, x_2, \cdots, x_m) = k_1\mathrm{U}_1(x_1) + k_2\mathrm{U}_2(x_2) + \cdots + k_m\mathrm{U}_m(x_m)$$
$$= \sum_{i=1}^{m} k_i\mathrm{U}_i(x_i)$$

$\mathrm{U}_i(x_i)$ – the utility function of the ith attribute

$k_i$– the weight of the ith attribute $(k_1 + k_2 + \cdots + k_m = 1)$

$0 \le \mathrm{U}(x_1, x_2, \cdots, x_m) \le 1$

## 5. System Implementation

This section presents the implementation results. The applications used in the study are the followings: Tomcat web server, java web development and aglet implementation by Tahiti agent server. It has following processes:

- User enters his / her required keywords or preferences into the system.
- Then mobile agent is created and it moves along the network carrying user preferences.
- Mobile agent travels the web according to the URL stored in the URL Queue database.
- URL is fetched and mobile agent moves to that address URL, matched information with user preference is stored in mobile agent.
- After surfing all URLs in the database, it returns to client user and results are displayed.

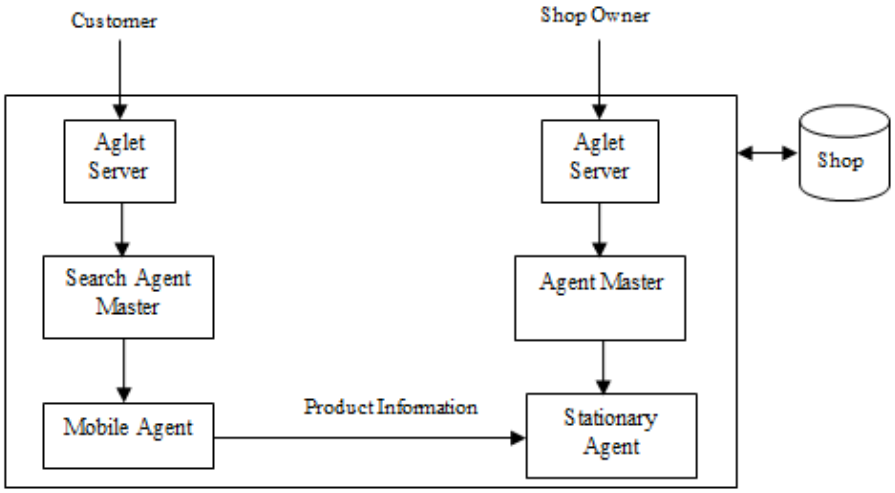


Fig. 4: Aglet Architecture of Proposed System

## 6. Conclusion

Certainly online shopping will continue to grow and the trend towards more powerful online shopping agents will continue. Mobile agent technology enhances with the emerging paradigm of intelligent human-computer interaction. The architecture of this system is multi-agent one, human and artificial agent together to achieve the shopping task based on mobile agent. This system presents an implementation of shopping assistant and suggesting the designs for the users using agent technology. Buyers can indicate their required preferences to the system. Then as buyers stroll through a mall, the system informs them of the availability of items of interest to the available in the immediately surrounding stores online. It is an agent-based model for intelligent shopping assistant and its application, presented architecture of a multi-agent model in virtual environment. It provides the buyer in valuable information and support decision to choose required items. Buyers can save of time in searching required information and able to select from the most suitable products.

## 7. Acknowledgement

I would like to thank to my teachers for their fruitful suggestions, grateful attitude and kind permissions to write this paper. I am also grateful to all teachers and friends who helped during this work.

## 8. References

[1] Barcin Kozbe and Michael N. Huhns, "The Requirements for Personal Mobile Assistants in a Mobile Telecommunications Environment", Center for Information Technology Department of Electrical and Computer Engineering University of South Carolina Columbia, SC, U.S.A.

[2] Benjamin Marlin, “Collaborative Filtering:A Machine Learning Perspective”, Department of Computer Science University of Toronto,2004.

[3] Ichiro Satoh, “ Mobile Agent”, National Institute of Informatics, Hitotsubashi, Chiyoda-ku, Tokyo.

[4] Katia P.Sycara, “Multiagent Systems”.

[5] Mohamad Eid, Hassan Artail, Ayman Kayssi, and Ali Chehab ,“Trends in Mobile Agent Applications”, Department of Electrical and Computer Engineering American University of Beirut.

[6] Robert H. Guttman, Alexandros G. Moukas, and Pattie Maes, “Agents as Mediators in Electronic Commerce”, Software Agents Group MIT Media Laboratory,Cambridge.

[7] Rosane Maria Martins, Luci Pirmez and Luiz Fernando Rust da Costa Carmo, Yan Wang and Vijay Varadharajan, “Databases and Information Retrieval:Taking advantage of Mobile Agent Technology”, NCE/UFRJ - Núcleo de Computação Eletrônica Universidade Federa l do Rio de Janeiro.YarivAridor and Mitsuru Oshima, “Infrastructure for Mobile Agents: Requirement and Design”, IBM Research, Tokyo Research Laboratory, Japan, 1998.

[8] Shivam H. Agrawal, Ankit Kumar Singh, Aliasger S. Najmi, Chetan S. Chavan, “Mobile Agent Based Data Retrieval from Distributed Database”, Department of information Technology, Sandip Institute of Technology and Research Center,Nashik.

[9] Yan Liu, “Multi-attribute Utility Theory (MAUT)”, Department of Biomedial Industrial and Human Factors Engineering Wright State University.