IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 2 (Sep-Oct. 2012), PP 26-34
www.iosrjournals.org

Mobile Agents: An Intelligent Multi-Agent System for Mobile Phones

Hussain Saleem[1], M. Zamin Ali Khan[2] and Shiraz Afzal[3]

[1] *Department of Computer Science, University of Karachi, Karachi, Pakistan*
[2] *Main Communication Network Department, University of Karachi, Karachi, Pakistan*
[3] *Electronic Engineering Department, Sir Syed University of Engineering & Technology, Karachi, Pakistan*

Abstract : *The mobile applications are one of the most important utilities of modern communication world. Currently this technology is getting compact to provide compatibility and accessibility to the users with a better option of handling. This research is based on the mobile and m-commerce technology, a fastest growing technology of today. The m-commerce technology provides various services like Mobile ticketing, Mobile banking, Mobile location based services, Mobile auctions, Mobile purchasing and so on. This represents an incredible chance to enable mobile devices as universal device for mobile commerce applications. We have explored an important application view of mobile technology i.e. intelligent multi-agent system with an example application of hotel reservation through mobile phones that will not only enable the mobile phone users to access the services through internet but also provide ultimate mobility without any hassle of communication. We also explained the structure of the system with the incorporation of mobile software agent technologies.*

Keywords *- Software Agents, Mobile Computing, Communication Systems, m-Commerce Technology, Intelligent Systems.*

I. INTRODUCTION

Mobile Agents have fetched around a novel way to perform computations and helped the development of distributed applications. This has opened the learning and creative implementation with struggle for a visible position in the area of distributed and wireless computing. According to a broad definition of mobile agents: “Mobile agents are processes transmitted from a source computer to undertake a specified task [5][14]. Each mobile agent is a computation unit along with its private data and execution state encapsulated. When it is applied, the mobile agent proceeds autonomously and independently to the pointed client. When the agent appears to a server, it is delivered to an agent execution environment. If the agent possesses necessary authentication credentials, its executable parts go underway. To accomplish its task, the mobile agent can transport itself to another server, it can seed new agents, or interact with other agents. Upon completion, the mobile agent delivers the results to the source client or to another server.”

Mobile Agents appeared in the passage of the last decade of the previous millennium. They have been embraced by researchers and practitioners as a potential technology that could revolutionize the way we perform computations, develop applications and systems. They were, and still are, viewed as a unique way to approach mobile and wireless computing [22].

Certainly, this is not an over emphasis; mobile agents have been used in a variety of applications and computing areas. The driving force inspiring mobile agent-based computation is multifold: (1) Mobile agents provide a proficient, asynchronous method for searching information or services in rapidly growing network; (2) Mobile agents could be launched into the unstructured network and roam around to gather information. (3) Mobile agents support intermittent connectivity for slow networks and light-weight devices. Thus, mobile agents provide many benefits in internet system programming [7][11][14] in which there is a need for diverse varieties of integrated information, monitoring and notification, encapsulating artificial intelligence techniques, security and robustness [1][6][7][17][25]. Also the mobile agent paradigm has proved satisfactory performance when deployed for distributed access to Internet databases, distributed retrieving and filtering of information and minimizing network workload [1][6][21][23][24][30]. Hence, mobile agents proved being very effective in supporting asynchronous execution of client requests, weak connectivity & disconnected operations and the dynamic adaptation to the various types of user connectivity that are mostly common in wireless environments [4][6][22][23].

M-commerce (also called Mobile Commerce) in context to the business view is the structure of buying and selling of goods and services through wireless handheld devices such as cellular telephone and personal digital assistants i.e. PDAs. M-commerce provide lot of services like Mobile ticketing, Mobile banking, Mobile location based services, Mobile auctions, Mobile purchasing and so on. This signifies an incredible opportunity to enable mobile devices as universal device for mobile commerce applications. Here we have focused on Hotel reservation system based on Mobile Agent technology as an example application and proposed the development architecture accordingly.

With the introduction in section-1, section-2 discusses the use of mobile agent technology for hotel reservation system. Section-3 explains the software agents, while mobile agent paradigm is discussed in section-4. Section-5 explains the model design, and agent mobility is discussed in section-6, while a detail of development platforms is explored in section-7. Model implementation and the Architecture is explained in detail in section-8 and section-9. The conclusion and future work is explained in section-10.

II. DISCUSSION

For the services in case of hotel, we generally want to choose the best hotel in prime sites, with modern facilities, clean environment and affordable rates. This can be time consuming and sometimes costly when doing this on our own physically or using human agents. The advent of the World Wide Web or internet (as we know) changed the landscape of communication and the way we conduct business dramatically, allowing us to reach beyond boundaries once separated by miles of oceans. Information that one took days and in some cases weeks, can now be received in a matter of seconds. This technology has grown significantly over the years and has been interfaced with other technologies such as smart phones, PDAs and other forms of wired and wireless networking devices, propagating the ability to receive information and conduct business on the go at the finger tip. With this background in mind we have proposed an intelligent agent based hotel reservation system that will undertake the bulk of the search and booking of hotels for various users. The system will provide users with the ability to enter some criteria for the search and then perform the search based on those criteria and book the most appropriate hotel after receiving user's confirmation. These would be carried out using JADE-LEAP on user's mobile device. There indeed is a need for such system that facilitates the humans without hassles. This system here would use an intelligent agent (instead of the human agent) to perform similar search and booking activities that can improve the speed of the search and reduce cost significantly. Hence in summary we propose developing a an agent that will move from hotel to hotel from the mobile devices like smart phones by collecting details on the list of available facilities, price, customer experience, transportation etc.

The hotel industry is one of the largest and most profitable industries in most countries. It helps to boost the economy of both developed and developing countries alike and stands as one of the major supplier for tourism in these countries. People are always 'shopping around' when vacationing for the cheapest rates for hotels and arrange bookings in advance to guarantee their stay for the period of their vacations. Several sites and booking systems [27] have developed over the years to assist in the search and booking of hotels for individuals. However these systems generally require individuals to do the bulk of the search and booking. This is not only time consuming exercise, but can be frustrating and costly as well [2].

The primary objective of this project is to define the multi-agent software technology to fulfill the requirements of hotel reservation and hotel recommendations by mobile phones. The principal and the most important task, which has to be carried out in order to meet the requirements of this effort, is to create a communication channel between the agents and the limited memory portable device like mobile phones. This area of research is still under investigation and it will be considered a significant achievement if this task could be met appropriately. The establishment of a communication channel between the intelligent software (like an agent) and the information provider server is still under progress. Agents are smart enough to fetch the data, which is relevant to the user of the system. Multi-tasking with intelligence is the basic aim of the agent-based software. To understand the system, it is important to have a clear understanding of the standards and operations of the agent technology. This manuscript covers all the key m-commerce technologies to implement the goals.

The purpose of working in the area of agent technology is to get a clear understanding of how to enable computing devices to work efficiently and intelligently. This will lead to create products which will make life easier in the future for people. Our main concern is to promote the agent based systems. In the modern computing industry, a multi-agent system is getting focal view and greater consideration and we are encouraged to explore this research project. The layout of the project follows some certain rules and procedures. The approach defined in later sections is a step by step process for the implementation of this system.

III. SOFTWARE AGENTS

The most important question arises is what is an agent? "An agent is a program that assists people and acts on their behalf. Agents perform function by allowing people to delegate work to them" [6]. According to the definition, an agent serves as a middleman between the end-user and the service provider. The software developers and the hardware engineers both are quite familiar with the meaning of middleware. The agent works in a similar manner in this latest technology. The basic idea is the same but the implementation of the application is different. The basic difference lies in the execution part. Normally, in operating systems, the agents are defined in a particular environment and they can act in an environment asynchronously and autonomously. But for the mobile agents, it is not necessary that the agents are bound in a particular environment. The agent is not bound to the system where it begins execution but it has the ability to transfer itself from one environment to another or in other words from one network to another. It can take all the characteristics of the object present on another network and this unique quality makes life more flexible for the user. It is difficult to implement the mobile agent technology as it has the ability to vary from one environment to another. So the environmental operating compatibility is also a crucial area of research now a days.

What is a software agent? This is still a big question mark as the debate on the correct definition is still in process and still not concluded. The most precise way of defining it is “Software agents are loosely defined as programs that assist people and act on their behalf” [6]. According to the definition, it is clear that the software agents are the types of program that can interact with the user and the service providers. In this project, the software agent is a mobile agent, which has the capability to communicate between the traveler and the service providers. This is a very smart agent which has the ability to change itself into many languages according to the user demands and provides the options to the customer to get appropriate help. It also provides the location and services of the hotels in that particular area and other communication medium information like telephone numbers, e-mail and fax services etc.

It is vital for both the travel agencies and the traveler to get an agent operational because it makes life convenient for both of them. The agent will give a brief description of the hotel location, rooms, room services, charges, other services and the ranking of the hotel. This is all useful information and will be important for the customer. It gives more choices to the traveler and it reduces the interaction between the traveler and the service provider as well. The time factor is also important as the traveler gets the desired information in lesser time. The entire software agents defined in this project work as an efficient and intelligent way to facilitate the user. Figure-1 depicts the conceptual framework for mobile agent computing with layered approach.

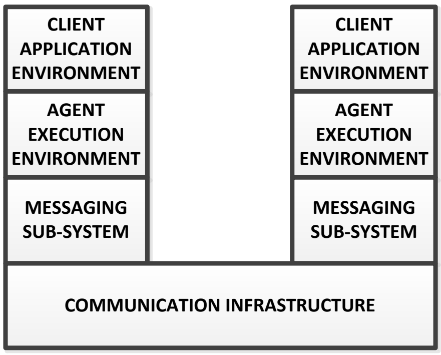
CLIENT APPLICATION ENVIRONMENT
AGENT EXECUTION ENVIRONMENT
MESSAGING SUB-SYSTEM
CLIENT APPLICATION ENVIRONMENT
AGENT EXECUTION ENVIRONMENT
MESSAGING SUB-SYSTEM
COMMUNICATION INFRASTRUCTURE

Figure-1: Conceptual model for mobile agent computing [11]

IV. MOBILE-AGENT PARADIGM

In past few years, the software growth based on mobile agents has established important attention because it has introduced a new development standard for widely distributed and heterogeneous systems. The most essential concepts of this paradigm are (mobile) agents and places [28]. In a broad sense, an agent is a software component that executes specific tasks on behalf of someone (a person) or something (an organization or another agent) with some autonomy – i.e., its actions could be not only determined by external events or interactions, but also by its own motivation (or purpose). Agents can be classified along several primary attributes that agents could exhibit, such as reactivity, pro-activity, social ability and mobility [18]. With respect to mobility – i.e., the agent’s ability to move around some computer network, and as known as migration or navigation – there are two kinds of agents: mobile and stationary (static or non-mobile) agents. Additionally, an agent can play several roles and conversely multiple agents can play the same role; a role describes external features of an agent in a particular context [8].

Usually, an agent-based application contains several agents that interact and communicate among themselves, playing different roles. A mobile-agent application furthermore involves a number of computational environments – as known as mobile-agent systems or servers – where agent execution can take place and also different services and resources are provided there. In this context, mobile agents are active entities that can move from a mobile-agent system to another one, in order to meet other agents or to access to services and resources provided there, while they are acting on behalf of someone or something. On the other hand, resources represent non-autonomous entities, such as files, objects, databases, programs or external applications, which can be used and shared by several mobile agents [8].

Additionally a mobile-agent system is a software component, which acts as a docking station accepting external agents and furnishing native resources to them. It is liable for executing agent code through protected execution environments – as known as execution places. A mobile-agent system is also responsible for providing other support features, such as agent persistence, security, and primitive operations to agent programmers, e.g., those that allow agents to migrate, communicate among them, access local resources, etc.

Furthermore, modern mobile-agent systems provide standard interfaces for interoperability with other mobile-agent systems, accepting agents implemented in different programming languages. In this perspective, execution places (or simply places) denote logical locations provided by mobile-agent systems where agents execute, meet and communicate with other agents, and operate some local resources. In general, agents, mobile-agent systems, places and resources are uniquely identified by a name or an electronic address [8]. One or more mobile-agent systems can exist in a node. A node represents a hardware infrastructure on which both agents and mobile-agents are actually executed. Typically, a node is a computer, but in the near future, this term will probably include other computing devices, such as personal digital assistants or mobile phones. A logical network of places materializes a region. The places in the region are associated to the same authority, e.g., an organization or administrative domain. The region registers each agent that is currently hosted by a place provided by a mobile-agent system of the region. If an agent moves to another place, this information is updated [8].

V. MODEL DESIGN

Up to now, the typical approach for mobile-agent applications development has been focused on the implementation phase, mainly because of the huge lack of appropriate modeling techniques or methodologies in the field. In this view, the development of powerful abstractions for the modeling of mobile agents becomes increasingly important. Agent mobility [10][15][28] – i.e., the software agent's capability of moving cross-ways the Internet while they are acting on behalf of a user – is being recognized as one of the most auspicious features for internet applications development. The mobility of an agent involves the transfer of its code, data, and possibly its execution state, towards the environments or sites where the resources it needs to access, or where other agents it needs to meet, are allocated. Such capability is particularly interesting when an agent makes random use of a valuable shared resource. Effectiveness, for instance, can be improved by moving agents performing queries over a large database to the host of the database itself. Additionally, response time and availability would improve when performing interactions over network links subject to long delays or interruptions of service [8].

In the past decade, many agent-oriented modeling techniques and methodologies have been developed [12][13][29][31]. Conversely, none of these techniques can yet be observed as a comprehensive methodology for the analysis and design of multi-agent systems. Likewise, very minute work has been made defining concepts and notations to model mobile agents. As a result, the usefulness of the existing Agent-Oriented Software Engineering (AOSE) methodologies for internet applications development is restricted. In spite of these efforts, there is yet little evidence of an engineering approach to the development of mobile agent-based applications [12][13][16]. Up to now, the usual approach for this development has been focusing on the implementation phase; applications are directly implemented in an ad-hoc fashion by selecting and using a particular platform supporting agent mobility, following little or no rigorous modeling technique or methodology. This approach subsequently produces inadequate specifications concerning relevant requirements, analysis or design aspects of mobile agents [8].

At present, the Unified Modeling Language (UML) [19][20] is being recognized as a satisfactory medium to support agent-oriented modeling [12]. Although UML is the standard language for documenting systems in the object-oriented paradigm, there are considerable resemblances between this paradigm and the agent-oriented one [29].

VI. AGENT MOBILITY

The primary feature of mobile agents is their ability to migrate between an arbitrary set of execution places. In general, agent mobility is reinforced by two mechanisms: Migration and Remote Cloning. The migration mechanism suspends the execution of a mobile agent, transmits it to the destination place, and then resumes it. The remote cloning mechanism creates a copy of a mobile agent at a remote place; cloning differs from migration because the original agent is not detached from its current location (place). Alternatively, agent mobility can be either Proactive or Reactive. In proactive mobility, the time and destination for migration are autonomously determined by the migrating agent. With reactive mobility, in contrast, the movement is triggered by a different entity that has some kind of relationship with the mobile agent to be migrated. As in migration, cloning can be either proactive or reactive [8].

On the other hand, from the implementation point of view, a mobile agent represents an execution unit with an internal structure including: a code segment, which provides the static sketch for its behavior; and a state composed of a data space and an execution state. The data space is the set of references to resources that can be opened by the agent. The execution state includes private data that cannot be shared, as well as control information, such as the call stack and the instruction pointer. In principle, each of these elements might move independently. The portion of a mobile agent that needs to be moved – i.e., its code, data, or execution state – is determined by composing orthogonal mechanisms supporting (1) mobility of code and execution state with mechanisms for (2) data space management, and facilitation for Communication between Agents [8].

VII. DEVELOPMENT PLATFORMS

Several mobile agent platforms have been established. They can be roughly categorized as Java and non-Java based ones, and they can be further divided into experimental and commercial ones. The Java-based mobile agent platforms include IBM's Aglets Workbench [22], Recursion Software's Voyager [22], Mitsubishi's Concordia [22], IKV++ Grasshopper [22] and General Magic's Odyssey [14]. The non-Java-based systems include, for example, TACOMA [9] and Agent Tcl [9]. There is an increasing interest in Java-based platforms due to the integral advantages of Java, namely: (1) platform independence support, (2) highly secure program execution, and (3) small size of compiled code. These flexible nature of Java joined with its simple database connectivity interface (JDBC) facilitates application access to various relational databases on the internet, which brand the Java applications highly appealing. Surely Java orientation of the Mobile Agent technology gave it boost so far and in future, this technology can do and offer much more [22].

VIII. MODEL IMPLEMENTATION

8.1 Design Standards

The standards of the design are also vital. Agents running in the system should must be thoroughly illustrated for the sake of clear understanding. The IEEE Foundation for Intelligent Physical Agents (FIPA) has provided the standards of the agent development and those standards must be followed. FIPA is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies. The comprehensive detail of the design standards is available at [9].

8.2 Design Structure and Design View

The design structure of the system is well-defined in three different ways. The core is the (1) architecture view, (2) UML (Unified Modular Language) diagram and (3) the activity diagram.

The system has processing in two active distributions. One at server side, and another at client side. Client side is called "front-end", while Server side is called "back-end". The project is divided into front-end (client side) and back-end (server side) technology. *Client Side Technologies*: The front end or client side technologies used to develop the system are JADE-LEAP and J2ME. To build a java-based application, the tools that support java application are also used. J-builder, Java Server Pages (JSP) and eclipse are used to create the application. The front-end technologies are defined as the technologies that work on higher level of the system. The system working on any device is described as the logical implementation of the system structure. *Server Side Technologies*: Two types of databases are used to develop Server side or back-end technologies. One is the Registered-User database and other is the Hotel database. Registered-User database and the Hotel database are developed in MS-SQL Server 2012. Java also supports all types of databases. Both databases provide the higher connectivity to other high level languages like JAVA. The entity relationship model is used to represent the graphical view of the databases. This provides the clear view of the system.

IX. THE ARCHITECTURE VIEW

Figure-2 best explains the architectural view of the system that defines all the steps of implementation of the project. The architectural view provides the clear view of the System.

9.1 Agent

The three kinds of agents are shown in Figure-2. They are (1) Mobile Agent, (2) Authenticator Agent and (3) the Update Agent. All agents are performing different tasks and play their role in the system.

9.2 Mobile Agent

The mobile agent plays a vital and an important task in the project. The basic task is to provide the connectivity between the system and Mobile. The Mobile Agent provides such kind of connectivity in the system. It is difficult to write and execute program on mobile devices due to limitation of memory. The functionalities on the mobile phones require limited memory access. LEAP adds-on (Light Extensible Agent Platform) facilitates developer to create mobile phone agent and provide connectivity to the computer agent to perform the task. If this agent can interact with the agent on the computer systems, the biggest goal of the software is achieved. Also the mobile agent defined above diagram controls the data on the mobile phone.

9.3 Authenticator

The task of this agent is for the security purpose. To achieve the security the Authenticator agent will assign a User-ID and password to all users of the system. This agent will perform its task by connecting with the registered users database and checks that the user who wants to use the service is a registered user or not. This is a security measure integrated in the system. Because the System is application software and the service can be charged, security measures inclusion is important to allow the registered users.

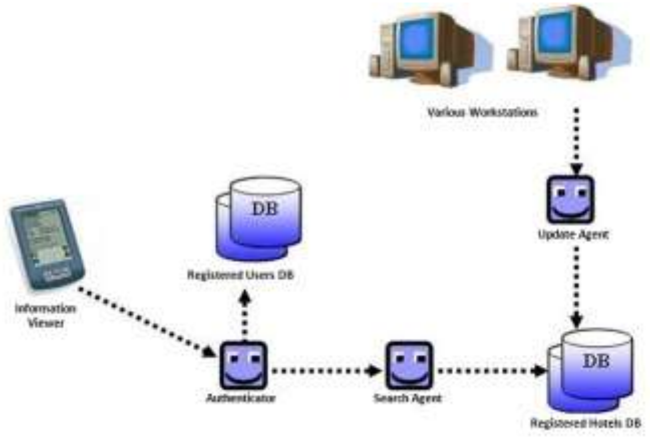

Figure-2: Architecture view of the system

9.4 Search Agent

The search agent searches for the request sent by end-user. Either the request will be fulfilled or it would not. It will find out that the exact demand of the customer has to be met. This will be achieved by connecting Search agent with the database. The user can select multiple hotels at the same time. The search agent is an intelligent agent which will be connected to the database server and acquire the result that matches the user input. The search agent works like any search engines e.g. www.google.com or www.altavista.com. The intelligence comes when it transfers data which fulfills the customer demand and provide multiple data options. This agent interacts with the registered user database after receiving the authentication. It connects to the main hotels database in order to find the relevant matched information.

9.5 Update Agent

The third agent is the update agent. This agent performs the task of updating the data in the hotels databases. For example, if a hotel room is reserved for a user and he checks-in, the update agent can reduce the room from that hotel's database. Similarly, if someone checks-out, the update agent will add that room to the particular database as it will be available to the customers if they will try to access the system. This agent is important for the purpose of correct information provided to the traveler. This agent communicates with the hotels database directly and can update changes in the database.

9.6 Registered Users Database

When authenticator agent performs its task, it accesses the registered user database. The update agent is a software agent which updates information in the hotel interface provided by the registered hotel database. The information is retrieved through this database at the time of authentication. MS-SQL Server is used in this application for the purpose of development of the database. It contains all the required data and the functionality as defined by the developer.

9.7 Registered Hotels Database

The registered hotels database consists of all the information about the hotels that are registered with the system to get the business. This database is a subset of hotel interface. It is also a client of hotel interface. When the application is developed, the hotel must register to the system to provide the information about the rooms and facilities to the travellers. This data might be local or remote. This application is based on remote data. Both Search and Update agents connect directly to the hotels registered database to manage this database. Both agents are responsible to provide the correct data from database.

9.8 Hotels Interface

The hotels interface is linked with the central database where all the information about the hotels is present. It means that it is the system which is used to update the data from the database. This database is scalable and it could be extended on demand. This interface can keep the hotel's personal database up-to-date.

9.9 Connectors

The last thing to describe in the system architecture is the Connectors i.e. Arrows. The connectors give the view of communication channel and the connectivity among the parts of the system. These communication channels and the connectivity links are different for different parts of the system.

9.10 Database Models

The declaration of the entity relationship model is the second phase of the system development. There are three view of Entity relationship model: (1) The External view, (2) the Logical view, and (3) the Physical view or the descriptive view. External View: The external view defines the class diagram designed in Unified Modeling Language (UML) for the system. The UML diagram is included in the system for the purpose of defining the system. Logical View: The middle level in the three level architecture is the logical or conceptual level. It is the community view of data and includes a description of all of the data that is available to be shared. It is a complete model or view of the working of the organization in its environment. The logical view defines the entity and their relations along with attributes. It represents the logical diagrams of the attributes. Physical View: The database that exists on the physical location represents the physical view. The internal or physical level covers the physical implementation of the database. It includes the data structure and file organization used to store data on physical storage devices [3].

9.11 Activity Diagram

Activity diagrams, from the behavioral or dynamic model view, are variations of state diagrams that depict the dynamic behavior of a system using action states and flows. Activity diagrams are also known as activity graphs [26]. All the behavior and changes of the system is defined in the Activity diagram. Figure-3 explains the concept view very well.

9.12 User Login Request

The first step in the system is to user login request. The authentication is provided at the start of the use of the system. The function that it performs at this stage is to check either the authentication is provided or not.

9.13 Search Agent Performance

After getting the request from the authentic user, following phase searches for the desired hotels. The hotels are either one or more. The selections of the hotel by the search agent can be {0, 1, …, n}. It means that the system can search for any possible selections and the list is provided to the user.

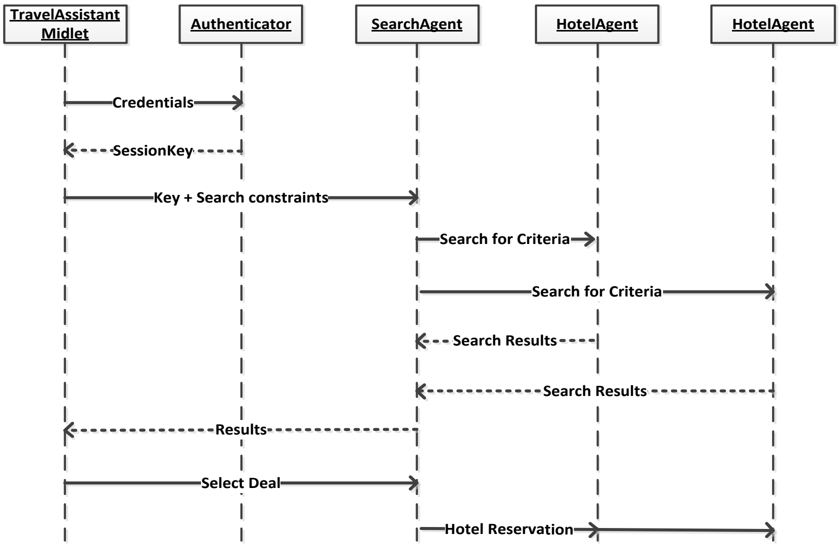

Figure-3: Activity Diagram of the System Process

9.14 Update Agent Performance

The next phase is the update agent performance. To show the update agent performance, it is essential to understand what will be the consequences for the update agent. What tasks an update agent performs? All the addition and subtraction of the rooms in case of check-in and check-out is done by Update Agent Performance.

9.15 Message Passing

The conversation between the agents is important. How agents communicate with each other and with other parts of the system is the key task of the project. The ACL messaging is the standard way agents can communicate with each other. All the vocabulary of communication language and standards among the agents is defined in FIPA's Agent [9]. The request is sent from one agent to another in the form of ACL and the reply from the agent will define the perfect communication between the software agents.

X. CONCLUSION & FUTURE WORK

There are several implementation issues that are needed to be addressed. The technology is growing and getting advanced. Some of the problems are as under:

i. The multi-agent technology is still immature and it is difficult to find the resources and documentation which will provide a clear understanding of how agents can communicate with each other.
ii. Because of the above problem, it is also difficult to find out how agents can communicate and perform on limited memory and memory oriented devices.
iii. An intelligent system is also needed to communicate with the database server on back-end to fetch the right data at the right time and also the addition and deletion of records in the database.
iv. Because of immature technology, the code which will be executed to get the desired results will not be easy to find.
v. One of the resources to get a clear understanding of how the technology will work is from the past work done by using that technology. As usually developers work on re-engineering paradigm, rather than engineering approach. These are the loop holes.
vi. The available environments made to produce agent based software's are also very limited. There is not a big competition in this regard amongst the market leaders and that's why, quality products are not produced.
vii. It is a challenging task to re-engineer an information system for hand-held devices like PDA's and mobile phones to work intelligently just like they work on memory oriented machines.

There is a necessity to address above issues for a successful implementation of the technology in order to get maximum benefit. However the idea of mobile agents with self-intelligence is novel and unique.

REFERENCES

[1] A. Pitsillides, G. Samaras, M. Dikaiakos, E. Christodoulou, "*DITIS: Collaborative Virtual Medical team for home healthcare of cancer patients*", Conf. on the Information Society and Telematics Applications, Catania, Italy, (16-18 April 1999).

[2] C. McTavish, S. Sankaranarayanan, "*Intelligent Agent based Hotel Search & Booking System*", IEEE Intl. Conf. on Electro/Information Technology (EIT), (2010).

[3] C. Ricardo, "*Database Systems*", (Maxwell McMillan International editions, 1990), pp.129-130.

[4] C. Spyrou, G. Samaras, E. Pitoura, P. Evripidou, "*Mobile Agents for Wireless Computing: The Convergence of Wireless Computational Models with Mobile-Agent Technologies*", Journal of ACM/Baltzer Mobile Networking and Applications (MONET), special issue on "Mobility in Databases & Distributed Systems", (2004).

[5] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris and G. Tsudik, "*Itinerant Agents for Mobile Computing*", Journal IEEE Personal Communications, (October, 1995), Vol.2, No. 5.

[6] D.B. Lange, M. Oshima, "Programming and Deploying Java Mobile Agents with Aglets", (Addison-Wesley, 1998).

[7] D.B. Lange, M. Oshima, "*Seven Good Reasons for Mobile Agents*", Comm. of the ACM, 42(3):88-91, (March 1999).

[8] Edgardo A.B. & Claudia A.M., "*Modeling of Mobile-Agent Applications with UML*", In Proc. of the Fourth Argentine Symposium on Software Engineering ASSE'2003, (2003).

[9] FIPA: The Foundation of Intelligent Mobile Agents, Web: http://www.fipa.org/ (Accessed on 20 September, 2012)

[10] Fuggetta A., Picco G. & Vigna G., "*Understanding Code Mobility*", IEEE Transactions on Software Engineering, (1998), Vol. 24, No. 5, pp. 342-361.

[11] Harrison C.G., Chess D.M., Kershenbaum A., "*Mobile Agents: Are they a good idea?*", IBM Research Division, T.J. Watson Research Center, NY, U.S.A. (1995).

[12] Huget M.P., Odell J. & Bauer B., "*UML and Agents: Current Trends and Future Directions*", Workshop on Agent-oriented methodologies, OOPSLA2002, Seattle, USA, (2002).

[13] Iglesias C.A. & González J.C., "*A Survey of Agent-Oriented Methodologies*", In Proc. of ATAL'98, LNAI, No.1555, pp.317-330, Springer-Verlag, (1998).

[14] J. E. White, "*Mobile Agents*", General Magic White Paper, (1996).

[15] Karnik N. & Tripathi A. R., "*Design Issues in Mobile Agent Programming Systems*", IEEE Concurrency, Vol.6, No.3, (1998).

[16] Klein C., Rausch A., Sihling M. & Wen Z., "*Extension of the Unified Modeling Language for Mobile Agents*". In Siau K. and Halpin T. (Eds.): Unified Modeling Language. Systems Analysis, Design and Development Issues, chapter VIII. (Idea Group Publishing, 2001).

[17] M. Dikaiakos, D. Gunopoulos, "*FIGI: The Architecture of an Internet-based Financial Information Gathering Infrastructure*", In Proc. of the 1st Intl. Workshop on Advanced Issues of E-Commerce and Web-based Information Systems. IEEE-Computer Society, pp.91-94, (1999).

[18] Nwana H., "*Software Agents: An Overview*", Knowledge Engineering Review, Vol.11, No.2, pp.205-244, (1996).

[19] OMG, "*Specification of the Unified Modeling Language (UML)*", v.1.4, http://www.omg.org, (2001).

[20] Rumbaugh J., Jacobson I. & Booch G., "*The Unified Modeling Language*", Ref. Manual, Addison-Wesley, (1999).

[21] S. Papastavrou, G. Samaras, E. Pitoura, "*Mobile Agents for WWW Distributed Database Access*", IEEE Transactions on Knowledge and Data Engineering Journal (TKDE), Vol. 12, No. 5, pp. 802-820, (2000).

[22] Samaras G., "*Mobile Agents: What about them? Did they deliver what they promised? Are they here to stay? (Panel)*", IEEE Intl. Conf. Proc. Mobile Data Management, pp.294-5, (2004).

[23] Samaras G., A. Pitsillides, "*Client/Intercept: A Computational Model for Wireless Environments*", Proc. 4th Intl. Conf. on Telecommunications (ICT'97), Melbourne, Australia, (1997).

[24] Samaras G., Dikaiakos M., Spyrou C., Liberdos A., "*Mobile Agent Platforms for Web-Databases: A Qualitative and Quantitative Assessment*", The Joint Symposium ASA/MA'99. First International Symposium on Agent Systems and Applications (ASA'99). Third International Symposium on Mobile Agents (MA'99), pp.50-64, USA, (1999).

[25] Samaras G., P. Evripidou, & P. Evangelia, "*Mobile-Agents based Infrastructure for eWork and eBussiness Applications*", The eBusiness and eWork Conference, eWork2000, pp.1092-1098, Madrid, Spain, (2000).

[26] Sinan Si Alhir, "*Guide to applying the UML*", Springer-Verlag New York, pp.108, (2002).

[27] Tait, N., "Online Booking Application", Napier University, (April 2001).

[28] White J., "Mobile Agents. In Software Agents", J. M. Bradshaw (Ed.), pp.437-472, (AAAI/MIT Press, 1997).

[29] Wooldridge M. & Ciancarini P., "*Agent-Oriented Software Engineering: The State of the Art*", In Handbook of Software Engineering and Knowledge Engineering, World Scientific Publishing, (2001).

[30] Y. Villate, A. Illarramendi, and E. Pitoura, "*Data Lockers: Mobile-Agent Based Middleware for the Security and Availability of Roaming Users Data*", Proc. of the Fifth Intl. Conf. on Cooperative Information Systems (CoopIS´2000), Israel, (2000).

[31] Zambonelli F., Jennings N. R., Omicini A. & Wooldridge M., "*Agent-Oriented Engineering for Internet Applications*", In Coordination of Internet Agents: Models, Technologies and Applications, Chapter XIII. Omicini A., Zambonelli F., Klusch M. and Tolksdorf (eds.), Springer-Verlag, (2000).

Hussain Saleem is currently Assistant Professor and Ph.D. Research Scholar at Department of Computer Science, University of Karachi, Pakistan. He received B.S. in Electronics Engineering from Sir Syed University of Engineering & Technology, Karachi in 1997 and has done Masters in Computer Science from University of Karachi in 2001. He bears vast experience of about 15 years of University Teaching, Administration and Research in various dimensions of Computer Science. Hussain is the Author of several International Journal publications. His field of interest is Software Science, System Automation, Hardware design and engineering, and Simulation & Modeling. He is senior member of Pakistan Engineering Council (PEC).

Dr. M. Zamin Ali Khan is a Head of Main Communication network department at University of Karachi. He has received B.E (Electrical Engineering) from NED University, Karachi, Pakistan and MS (Electrical and Computer Engineering) from Concordia University, Montreal, Canada and PhD in Computer Science from UoK. He has more than 18 years of experience of teaching and industry. He has worked in Victhom Human Bionics, Canada as an Engineer Scientist. Currently, he is a senior member of Pakistan Engineering Council, Canadian Engineering Council and IEEE. His research interest includes VLSI, Digital design, Digital signal processing and Analog front end of wireless devices.

Shiraz Afzal is a full time faculty member at Sir Syed University of Engineering & Technology, Karachi, Pakistan. He received his B.S. in Electronics from Sir Syed University of Engineering and Technology Karachi, Pakistan and M.E degree in Electronics with specialization in Micro-System design from NED University of Engineering and Technology Karachi, Pakistan in 2006 and 2012 respectively. His research interest includes Microelectronic circuit design. He is also a member of PEC.