Role of JADE in Emergency Healthcare System

Sharmila S. Gaikwad and P.M. Dafalapurkar

Abstract-- **This paper applies agents to the highly dynamic and variable context of healthcare emergency decision-support domain. More specifically the use of mobile agents to support the deployment of an ambulance service in real-time. An implementation of the proposed agent based architecture, which was based on the specific functional and non-functional application requirements set out based on thorough analysis of literature. In this project an illustrative emergency scenario in order to demonstrate the validity and feasibility of our proposed model is also created. From the evaluation of the implementation we were able to identify some of the major technical advantages it has to offer as well as challenges one needs to address in similar attempts.**

***Index Terms*-- Mobile Agents, mobile data access system, Java agent development environment, Agent communication language.**

I. Introduction

MULTI-DATABASE is one prevalent solution that integrates distributed heterogeneous databases and provides uniform global access methods to the users while preserving local database autonomy. It provides users somewhere, sometime data access. As mobile technology advances and the wireless communication environment improves, more and more database users demand anytime, anywhere access to the globally shared information space over wireless networks. In this new mobile data access system (MDAS) environment, multi-databases must face the challenges imposed by mobile devices and the wireless medium limited computational resources and power supply, low bandwidth, frequent disconnections, etc.

The agent platform can be distributed across machines which not even need to share the same Database or OS and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by moving agents from one machine to another one, as and when required. Mobile agent technology provides an attractive and important technique for building large-scale distributed applications in heterogeneous computing environments. A mobile agent can be viewed as an autonomous program that has the ability to transport itself between the nodes of a network entirely under its own control, carrying with it the data and execution state required to resume execution at the destination host from the point it ceased on the original host. Therefore, it is the agent that decides 'when to move', 'where to move', 'what to execute' and 'how to execute it'.

We illustrate the agent-based application model employing a sample scenario. Consider a situation in which a middle-age man suffers from a cardiac arrest while on his way to work. The on-lookers immediately call 000 providing details such as the location of the patient, the nature of the accident, along with some form of identification (ID) and if possible a name (we assume that the patient will have some form of ID such as a driving license or a medical insurance card on him). The dispatch centre then relays this information to the most appropriate ambulatory service, which promptly dispatches an ambulance to the scene. The paramedics use the dashboard-mounted terminal on the ambulance to filter out the group of hospitals that are in close proximity to the patient's location. They also key in the available details such as ID/name, gender and type of treatment required. Once this is done, the paramedics use the travel time to the accident scene to launch two mobile agent assistants. These agents use the data entered by the paramedics to autonomously and asynchronously traverse the various nodes of the network to carry out specific information retrieval tasks on behalf of the paramedics.

Ms. Sharmila S. Gaikwad was born in Mumbai, on September 17, 1978. She graduated from the K.G.C.OE, at the University of Mumbai. She is doing her ME Computers from D.Y. Patil College of Engg. Pune (email :sharmila_gaikwad@yahoo.com).

Mrs. P.M. Dafalapurkar is an assistant professor at the D.Y. Patil College of Engg. Pimpri, Pune (email: *pradnya_dafalapurkar@yahoo.com*).

II. Introduction To Mobile Agent

A. Mobile Agent

"An Intelligent Agent is a self-contained software element that has a degree of intelligence, mobility and autonomy and is capable of effectively communicating with other Intelligent Agents, users and software systems". Agents can handle dedicated tasks on behalf of the user without repeatedly consulting the user. Agents are able to effectively communicate with other agents, users and software platforms. This definition excludes for example conventional software objects and rule-based expert systems. A good example is a World Wide Web off-line search agent that is capable of retrieving desired information on behalf of the user according to his profile. The agent is told what to look for, and then sent away to carry out the search off-line. After a while, the agent will come back with the desired information.

B. Characteristics of Mobile Agents

The characteristics of Mobile Agents are elaborated below

Intelligence: Intelligent agents have a certain degree of specific domain knowledge with which they are capable of reasoning and learning. This capability requires the agent to have access to a knowledge base, as well as an inference engine for reasoning. The capability of learning determines the adaptive behavior of an agent in the sense that it can effectively handle new situations/contexts.

Autonomy: Autonomous behavior means that agents are not only passively driven by external events created by the user or other systems. In addition to these external events, an agent has internal events that cause the agent to perform a particular behavior that is in line with the objectives it 'wants' to pursue.

Mobility: Intelligent agents may have a certain degree of mobility. This means that the agent is not restricted to its home platform. It has the capability to migrate to host platforms where it can carry out tasks locally, thus reducing processing load on its home platform, and reducing communication overhead. In the context of distributed processing and load balancing, this is an attractive capability. Another advantage is the possibility of having agents that keep functioning after the location it originated from has gone off-line. Agents that can travel are called mobile agents.

Communicative / social behaviour: Intelligent agents are capable of communicating effectively with other agents, users and systems. Inter-agent communication is carried out using an Agent Communication Language (ACL) for example KQML (Knowledge Query Manipulation Language) that assumes specific ontology's for specific agent applications. For example, a personal travel agent will have to 'understand' the concept of a hotel and an aero plane in order to be able to negotiate deals with selling travel agents. It is envisaged that intelligent agents will have some basic general ontology as a minimum.

C. Model of Mobile Agent:

Each mobile agent is an intelligent agent as well. The agent model defines the internal structure of the intelligent agent it defines the autonomy, learning and co-operative characteristics of an agent.

1) A Life-Cycle model

2) Computational Model

3) A Security Model

4) A Communication Model

5) A Navigation Model

D. Advantages of Mobile Agents Technology

1) Efficiency - mobile agents consume fewer network resources since they move the computation to the data rather than the data to the computation.

2) Reduction of network traffic

3) Asynchronous autonomous interaction - tasks can be encoded into mobile agents and then dispatched. The mobile agent can operate asynchronously and independent of the sending program.

4) Interaction with real-time entities

5) Dealing with vast volumes of data - the processing of this data should be performed local to the data, instead of transmitting it over a network.

6) Robustness and fault tolerance - the ability of mobile agents to react dynamically to adverse situations makes it easier to build fault tolerant behaviour, especially in a highly distributed system.

7) Support for heterogeneous environments.

8) Support for e-commerce - mobile agents can be used to build electronic markets.

III. Introduction To JADE

A. The jade architecture

JADE (Java Agent Development Framework) is a software Framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases.

JADE is completely implemented in Java language and the minimal system requirement is the version 1.4 of JAVA (the run time environment or the JDK).

B. JADE overview

JADE is a middleware that facilitates the development of multi-agent systems. It includes

1) A runtime environment where JADE agents can "live" and that must be active on a given host before one or more agents can be executed on that host.

2) A library of classes that programmers have to/can use (directly or by specializing them) to develop their agents.

3) A suite of graphical tools that allows administrating and monitoring the activity of running agents.

Containers and Platforms

Each running instance of the JADE runtime environment is called a *Container* as it can contain several agents. The set of active containers is called a *Platform*. A single special *Main container* must always be active in a platform and all other containers register with it as soon as they start. It follows that the first container to start in a platform must be a main container while all other containers must be "normal" (i.e. non-main) containers and must "be told" where to find (host and port) their main container (i.e. the main container to register with). If another main container is started somewhere in the network it constitutes a different platform to which new normal containers can possibly register.

AMS and DF

Besides the ability of accepting registrations from other containers, a main container differs from normal containers as it holds two special agents (automatically started when the main container is launched).

The AMS (Agent Management System) that provides the naming service (i.e. ensures that each agent in the platform has a unique name) and represents the authority in the platform (for instance it is possible to create/kill agents on remote containers by requesting that to the AMS).

The DF (Directory Facilitator) provides a Yellow Pages service.

C. JADE Features

The following is the list of features that JADE offers to the agent programmer:

1) Distributed agent platform. The agent platform can be split among several hosts (provided they can be connected via RMI4). Only one Java application, and therefore only one Java Virtual Machine, is executed on each host. Agents are implemented as Java threads and live within Agent Containers that provide the runtime support to the agent execution.

2) Graphical user interface to manage several agents and agent containers from a remote host.

3) Debugging tools to help in developing multi agents applications based on JADE.

4) Intra-platform agent mobility, including transfer of both the state and the code (when necessary) of the agent.

5) Support to the execution of multiple, parallel and concurrent agent activities via the behavior model. JADE schedules the agent behaviors in a non-preemptive fashion.

6) Library of FIPA interaction protocols ready to be used.

7) Automatic registration and deregistration of agents with the AMS.

8) InProcess Interface to allow external applications to launch autonomous agents.

IV. System Analysis

A. Problem Definition:

Consider a situation in which a middle-age man suffers from a cardiac arrest while on his way to work. The on-lookers immediately the dispatch center providing details such as the location of the patient, the nature of the accident, along with some form of identification (ID) and if possible a name. The dispatch centre then relays this information to the most appropriate ambulatory service, which promptly dispatches an ambulance to the scene. The paramedics use the dashboard-mounted terminal on the ambulance to filter out the group of hospitals that are in close proximity to the patient's location. They also key in the available details such as ID/name, gender and type of treatment required. Once this is done, the paramedics use the travel time to the accident scene to launch two mobile agent assistants. These agents use the data entered by the paramedics to autonomously and asynchronously traverse the various nodes of the network to carry out specific information retrieval tasks on behalf of the paramedics. Finally they collaborate their results and present it to the user in a transparent manner. We call out first mobile agent as the 'hospital assistant'. This agent is tasked to retrieve the latest resource details form the list of hospitals assigned to it. As mentioned, paramedics' shortlist a group of hospitals based on the patient's location (they may not do this in which case the agent will visit all hospitals in the city irrespective of their distance form the scene of accident). Each hospital is essentially a node in our distributed medical environment. Once launched, the mobile assistant uses the list of hospitals it was assigned and migrates to each of the hospitals one by one. When it arrives at a destination hospital node it interacts locally with the stationary hospital agent to retrieve the latest resource consumptions details for the hospital.

As part of its interaction with the stationary agent the assistant acquires information about

1) Facility type offered by the hospital (24-Hr Emergency Centre, Cardiology Ward etc.)
2) Total number of doctors present
3) Number of doctors available to attend patient at present
4) Total number of nurses present
5) Number of nurses available to attend patient at present
6) Total number of beds at the hospital
7) Number of beds available.

After receiving the requested information, the assistant then visits the next hospital on the list and repeats its request.

It keeps doing this till it reaches the last hospital on its list. Upon completion the agent moves back to the ambulance to relay its results.

While the hospital assistant retrieves the latest resource utilization details from various hospitals, the second mobile agent, whom we refer to as the 'data assistant' simultaneously, retrieves the patient's medical history along with the latest route status for the selected hospitals. Based on the patient's ID details, the data assistant travels to a central medical repository (Medical Archives) interacts with the stationary archive assistant and acquires the medical details of the patient along with his past medical history. As part of the of the patients details the assistant acquires

1) Name of the patient
2) Health insurance number (if available)
3) Driving license number (if available)
4) Address details
5) Known drug allergies.

As part of the patients' medical history the assistant acquires information about

1) Hospital/Clinic visited by the patient
2) Date of visit
3) Details of visit (medical procedures performed etc.)
4) Name of physician attended.

It is important to note that for our scenario we assume that every individual has a unique identifier (ID) that is used to tag all his medical related details. If no ID details were entered by the paramedic (or were not available in the first place) the data assistant automatically skips going to the archive node and migrates directly to the road authority website for the latest situation on roads leading to each of the selected hospitals. After retrieving the relevant patient details and the necessary road updates, the agent moves back to the ambulance to relay this information. However, before both the hospital assistant and the data assistant publish their results, they coordinate their results with each other.

The UI then uses this set of results to rank the list of hospitals in order of suitability. In order to rank the hospitals the following are taken into consideration

1) Current resource availability at each hospital
2) Special medical facility offered by each hospital
3) Patient's medical history with each hospital and
4) Current road status information.

By the time paramedics reach the scene of emergency their onboard terminal already reflects the list of nearby hospitals ranked form the most relevant to the current case, to the least relevant. This is information is also supplemented with the patient's medical details and current road situation for each hospital. Once at the scene the medics stabilize the patient, administer medications based on the medical history (such as allergies) and load him in. It is important to note that although the application ranks the hospitals in their order of priority, the final decision with respect to selecting the most appropriate hospital still rests with the paramedics. In essence the application provides for a decision support tool that aids the paramedic by empowering him with real-time, mission critical information at the point-of-care. The paramedics review the ranked hospitals and the ranking details associated with each one and then make their final selection. On the way to the selected hospital, a message is sent to the associated hospital agent, which then updates the relevant hospital databases with the details of the patient 'en route' to the hospital. By the time the ambulance reaches the hospital this information has already enabled the emergency room staff to prepare for the arrival of the patient.

More specifically the project demonstrates the use of mobile agents to support the deployment of an ambulance support services in real-time. We presented an implementation of the proposed agent based architecture, which was based on the specific functional and non-functional application requirements set out based on thorough analysis of literature. We describe an emergency scenario to demonstrate the validity and feasibility of our proposed model. We were able to identify some of the major technical advantages as well as challenges one needs to be aware of in similar attempts. We choose mobile agents as the key enabling technology because they offer a single, general framework in which large-scale distributed real-time decision support applications can be implemented more efficiently.

B. Scope

This architecture involves a mix of stationary (hospitals) and mobile nodes (ambulances) supported by a combination of static and wireless networks. These network of nodes communicate and share information using mobile agents. The various agents autonomously and asynchronously migrate between nodes and interact with other stationary or mobile agents in order to accomplish their assigned tasks. The application gets initiated when the paramedic selects a list of hospital based on the accident location, keys in the patient ID and clicks on the 'Launch Agent' button on the on the user interface component of the ambulance node. This deploys the two mobile assistants associated with the application – 'hospital assistant' and 'data assistant'. The hospital assistant is tasked to visit a list of hospitals identified by the paramedic. Once launched, it migrates to the various hospital nodes, interacts with the stationary 'hospital agent' and retrieves the latest resource utilization details. The data assistant on the other hand is tasked to retrieve the patient's medical details and get the latest road status information for the selected hospitals. Once launched, it migrates to the 'medical archives node' and interacts with the stationary archive agent to retrieve the patient medical history based on his ID. It then travels to the 'road authority web site' and gets the latest updates on road status information for each of the selected hospital. When their respective tasks are done, the mobile agents then move back to their home node (ambulance which initially launched them). Upon arriving back at the home base, they coordinate with each other to collaborate their results before publishing it to the user interface.

The five major components, which comprise the proposed architecture are detailed in the following section.

1) Ambulance Node
2) Hospital Node
3) Medical Archives Node
4) Mobile Hospital Assistant
5) .Data Assistant Node

C. *Object Oriented Analysis:*

OOA is the challenge of understanding the problem domain, and then the system's responsibilities in that light. I have done analysis of the project using Unified approach of OOA starting from identification of the actors and finding the activities performed by them.

1) *Activity Diagram*

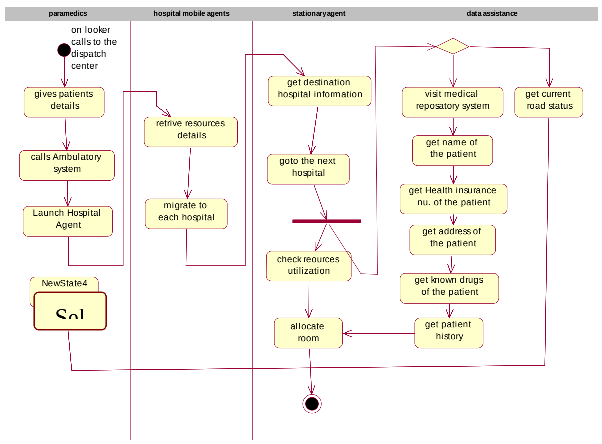

Figure 1 Activity Diagram

2) *Sequence Diagram*

An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them. A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. Graphically, a sequence diagram is a table that shows objects arranged along the X axis and messages. Sequence diagrams commonly contain:

1. Objects
2. Links
3. Messages

Sequence diagrams have two features that distinguish the from collaboration diagrams. First, there is the object lifeline. An object lifeline is the vertical dashed line that represents the existence of an object over a period of time. Second, there is the focus of control. The focus of control is tall, thin rectangle that shows the period of time during which an object is performing an action. The sequence diagram for resource utilization is shown in Figure 2.

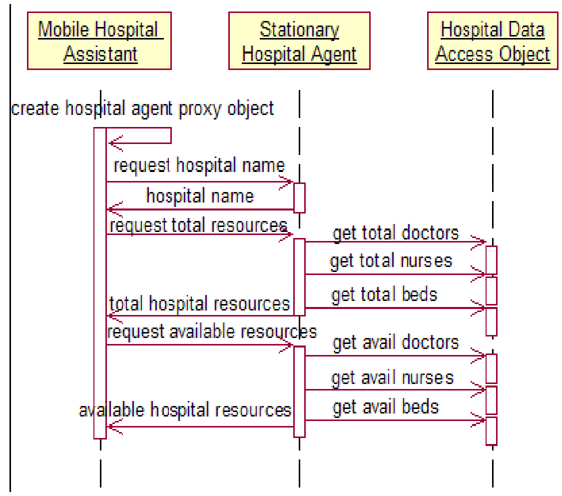

Figure 2 Obtaining Resource Utilization Details

The sequence diagram for Patient's Medical details is shown in Figure 3.

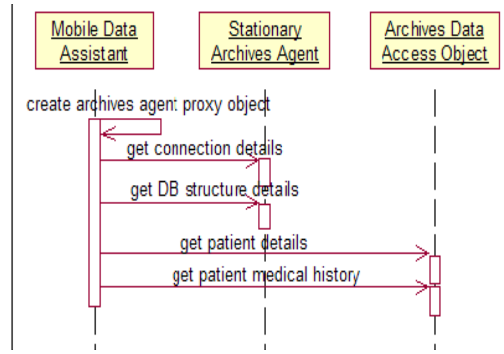

Figure 3 Obtaining Patient's Medical Details

The sequence diagram for getting Rad Current status information is shown in Figure 4.

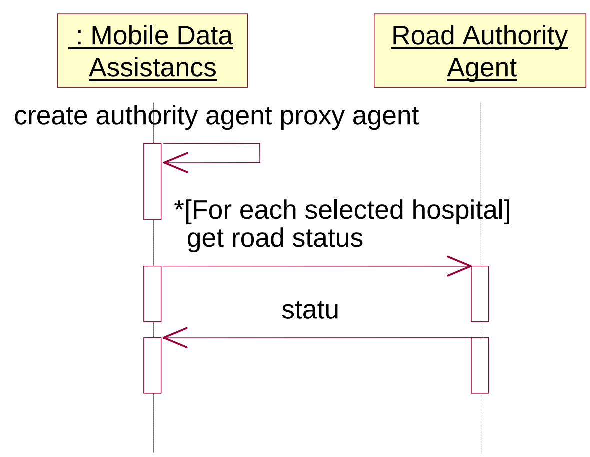

Figure 4 Obtaining Route Status Updates

D. *4.3.4 Software Requirements*

1) JDK 1.5
2) Swing
3) JADE
4) MS Access
5) MySQL Server 4.1
6) SQL Server 2000
7) UML (Rational Rose)

V. Results Output Screenshots

HDMS using MA is developed using JADE, the main container is the main repository system. And then different sites i.e. the different heterogeneous hospital node, the ambulance node, Data Assistant are created as container on the Main container.

We had identified some of the key requirements that need to be addressed by our application for it to be successfully deployed in the dynamic and variable context of healthcare.

A single interaction process is actually made up a series of requests. By being able to migrate to its destination host, the hospital assistant makes this entire process local thus bypassing the network.

The paramedics then utilize the travel time to the scene of emergency to launch a set of two mobile agents. The first mobile agent (referred to as *hospital assistant*) starts off by contacting to 'discover' the three hospital nodes that paramedics have short-listed. When these nodes have been successfully discovered, the agent migrates to the first hospital on its list. Upon arrival, it interacts with the stationary agent present at this node and retrieves the latest resource utilization details for the hospital as shown in the figure 5.

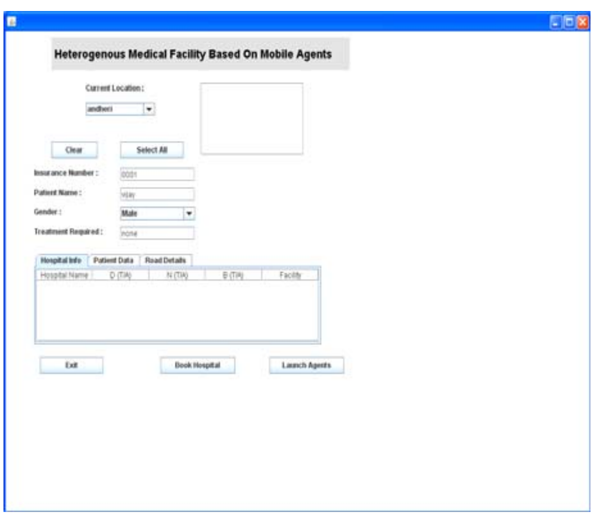

Figure 5 GUI at Ambulance node

When we press the button launch the agent, two mobile agent are getting launched one is Hospital agent which visit several near by hospitals and search for the facilities and availability of doctor and other resources. And give the result in the tab provided for hospital details. As shown in figure 6. and other is Data assistant agent which collect the patients info from the main repository system database like patient personal info., medical history etc as shown in figure 7.

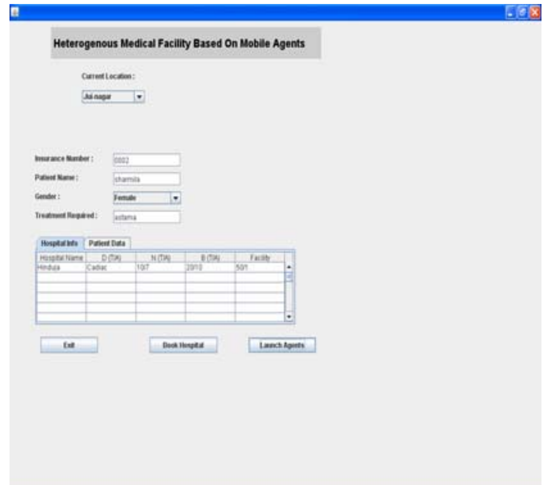

Figure 6 GUI Hospital Agent Launched

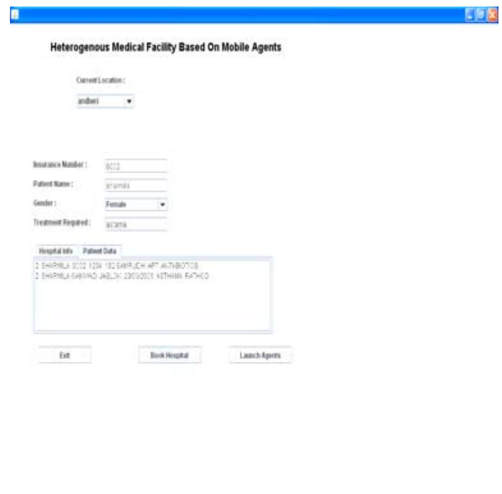

Figure 7 Data Assistant agent Launched

VI. Conclusions

In this paper we have demonstrated how the agent's technology can be used to provide decision support in a dynamic, context sensitive environment. Healthcare example was selected as an appropriate field due to the fact that it lends itself very well for such implementation from the point of view of business requirements as well as recently increased interest in using technologies by medical practitioners.

This is despite the fact that there are many potential application areas where mobile agents can be ideally deployed. While numerous agent system have been developed and agents have been applied to a wide variety of applications ranging from purely commercial to military to educational , none of these has been able to 'jump-start' the use of agents .

Unlike traditional paradigms, agents exhibit the property of being autonomous and interactive. Coupled with mobility, they are capable of performing dynamic and intelligent inference tasks during their execution. Utilizing a framework based on the mobile agent paradigm provided for a higher degree of flexibility by allowing applications to dynamically adapt to the changing demands of their execution environments.

VII. Future Work

In this research we demonstrated the efficiency and usefulness of mobile agents in distributed systems by applying them to the highly mobile and dynamic area of ambulatory services. Though our implementation and extensive experimental studies have shown the feasibility of this approach, we have however made certain assumptions in our system design so as to ease the actual implementation process. Most of these assumptions are made due to the highly controlled and sensitive nature of the healthcare domain or because of time limitations on our part. There are several extensions, we have identified that could be applied to the proposed model to further equip it for a real-world deployment

A. Location Awareness

In the current implementation we do not factor in the key element of distance while ranking the selected list of hospitals. Ideally, our model should be integrated with some form of location tracking technology (such as GPS) that will enable it to determine the exact distance of each hospital from the location of emergency. This will not only make the ranking process more intuitive but will also enable the model to cater to a wider variety of emergency scenarios. For example consider a scenario is which the patient is triaged critical. In this case the nearest hospital should be selected rather than the one, which has a higher level of available resources.

B. Enhanced Security Features

Security and privacy are of primary concern when dealing with medical data. However, this is an interesting area as providing a higher level of security generally means reducing availability and timely delivery of services. So far, we have utilized the in-built security mechanisms provided by Grasshopper platform. We note that this level of basic security will not be sufficient when dealing with actual patient data. Therefore, moving forward the model would need to incorporate a more comprehensive security policy. Future work could include arming the agents with authentication mechanisms that prevent unauthorized access to server data. Along with authentication, organizational nodes could also define and manage data access policies for different agent types. Further, a simple encryption technique could be put in place to secure rudimentary communication.

C. Incorporating Generic Web-based Nodes into the Ambulatory Environment

Instead of utilizing the Webhopper extension to enable an agent to interact with a website (as we did in the case of the road-authority node), a better solution would be to allow agents to 'talk' directly to such nodes. Webhopper was used to maintain consistency across all nodes that comprised our architecture as well as to take advantage of the communication and security mechanisms of the underlying agent platform. However, with enhanced security features in place, doing away with Webhopper will not only make the design more generic but also scalable by allowing standard web-based nodes to be included as part of the architecture.

D. Deployment in real-world settings

To realize the true potential of our proposed application, both in terms of its feasibility as well as its accuracy and efficiency, it should ideally be deployed and tested in an actual ambulatory environment with the size and complexity of real data sets. However, this would involve the collaboration of a number of healthcare organizations (hospitals, clinics etc.) and require access to their information systems. We note that due to highly sensitive and closed nature of this environment, an actual deployment could face many challenges. Moreover, our research prototype has been developed based on a few key assumptions in relation to the operating environment (such as unique patient identifier, agent host on every node), which may not hold true in the real world.

VIII. Acknowledgements

I express my sincere and profound thanks to Guide Prof. P. M. Daflapurkar, who always stood by me as the helping and guiding support.

IX. References

[1] Software Agents: A Review, 1997, Trinity College, Dublin, Intelligent Agen tGroup overview of the agent technology .http://www.cs.tcd.ie/research_groups/aig/iag/pubreview/hap5/chap5.html mobile agents.

[2] A Distributed Heterogeneous Database System based on Mobile Agents, A. Di Stefano, L. Lo Bello , C. Santoro.

[3] Mobile Agents in Mobile Heterogeneous Database Environment Performance and Power Consumption Analysis By Yu Jiao and Ali R. Hurson, Department of Computer Science and Engineering , The Pennsylvania State University ,University Park, PA 16802-6106.

[4] Mobile Agent Technology for the Management of Distributed Systems-a case study by Claudia Raibulet and Claudio Demartini.
Using Mobile Agents in Real World: A Survey and Evaluation of Agent Platforms Josef Altmann, Franz Gruber, Ludwig Klug, Wolfgang Stockner, Edgar Weippl Software Competence Center Hagenberg Hauptstr. 99 A-4232 Hagenberg, Austria +43 7236 3343 800

X. Biographies

Sharmila Gaikwad was born in Mumbai, on September 17, 1978. She graduated from the K.G.C.OE, at the University of Mumbai. Doing ME Computers from D.Y. Patil college of engg. Pune.