Intelligent Scheduling in Health Care Domain

Srividya Bhat , Nandini S. Sidnal , Ravi S. Malashetty , Sunilkumar. S. Manvi

1 Dept. of PG Studies, VTU, Belgaum, Karnataka, India

2 Dept. of CSE, K.L.E.S.C.E.T, Belgaum, Karnataka, India

3 Dept. of PG Studies, VTU, Belgaum, Karnataka, India

4 Dept. of ECE, REVA ITM, Bangalore, Karnataka, India

Abstract

Healthcare organizations are facing the challenge of delivering high-quality services through effective process management at all levels-locally, regionally, nationally, and internationally. Patient scheduling becomes an integral part of daily work for healthcare professionals. The presented work is to build an agent based information services for mobile users. An agent is characterized by the concepts of situatedness, autonomy and flexibility. Multi-Agent systems (MAS) are appropriate in many medical domains, due to the characteristics of the problems in this area and are the basis of an emerging technology that promises to make it much easier to design and implement. The paper work integrates accessing distributed health care services in multi-agent environment to achieve better Quality of service by using java platform. This develops a framework to schedule the meeting between the patients and the relevant doctors meeting in an efficient way for routine and emergency services.

Keywords: *Mobile users, Software agents, Multiagent systems, medical ontology, FIPA-ACL.*

1. Introduction

Most of the Health care professionals use computer systems to access patient's medical record or information about hospital resources and to fix an appointment for multiple patients with potentially conflicting schedules. Meeting Scheduler in health care domain is considered as a part which will grow most rapidly and lead to economical and popular methodology with autonomous agents, which can schedule meetings and manage calendars on behalf of their users by saving the patients and physicians time. Also this system is generally designed to guide remote patients to fix the appointment with doctor through online facilities and help them to reach an appropriate hospital in an unknown city. A brief introduction of the concepts and methods are used to carry out the paper work are given.

At present most of the hospitals follow a Simple GUI based applications to maintain their information regarding the patients, Doctor and scheduling information. Effective and timely communication between patients, physicians, and other healthcare professionals is vital to good healthcare. Current communication mechanisms are based largely on paper records and prescriptions, which are old-fashioned, inefficient, and unreliable. In an age of electronic record keeping and communication, the healthcare industry is still tied to paper documents that are easily misled, often illegible, and easy to forge.

Healthcare professionals working in highly dynamic hospital environments typically have correspondingly dynamic schedules that are difficult to manage. Emergent tasks and shifting priorities result in existing schedules becoming obsolete. Managing patient appointments is an area that typically consumes a great deal of administrative overhead and cost. Clinic and office administrators are typically juggling multiple phone calls, physician requests, and patient demands. It is also a source of frustration for many patients due to the delays and inefficiencies in speaking with the clinic or office administrator. This leads to no-shows, lost revenue, and operational inefficiencies.

An increase in specialization and technology, especially in the health care department requires efficient management of the resources and timely treatment for the patients. Agents are used to solve the patient scheduling problem in the hospitals because they work well in a distributed, decentralized and dynamic environment. An agent is a software program that acts on behalf of a user, typically used to retrieve and process information. An agent is used to represent each patient and resource in the hospitals. Interaction protocols are used to reduce the search space of possible responses to an agent messages.

Multi Agent System (MAS) based Health Care Domain will address some of these issues:

- ❖ MAS contains agents that allow the user to search for medical centers satisfying a given set of requirements, to access his/her medical record or to make a booking to be visited by a particular kind of doctor.

- Some of the agents in the system can provide information about the medical centers that are available in a given city.
- The MAS also contains an agent for each medical center in town; these agents may be asked about the doctors working in that hospital, or may be requested to perform a booking in the schedule of a specific doctor.
- Providing a decomposition of the problem that matched agents to entities which could be realistic players in such a domain and to take care in who had access to which information.

Healthcare professionals working in a hospital environment typically have many responsibilities contending for their time. With tasks ranging from providing medical care and monitoring patients to undertaking administrative responsibilities, it is often the case that healthcare professionals have a seemingly endless set of changing tasks to carry out. Consequently, they must manage their time by composing their activities into prioritized to-do lists. However, hospitals are inherently highly dynamic environments in which task interruptions and delays are commonplace. Additionally, previously unforeseen tasks can emerge that may require attention alongside the already scheduled tasks. In the face of such change, static paper or whiteboard-based to-do lists can become difficult to manage and, in the worst case, obsolete.

Recent advances in embedded sensor and mobile computing technology have given rise to a range of possibilities in pervasive healthcare. Among these is the opportunity to aid healthcare professionals by automatically managing their schedules in the face of significant contextual events that can negatively impact their schedule. When a patient wants to arrange an appointment with a doctor, or a doctor must arrange a visit of a patient with a service, it is required to schedule a meeting according to different constraints such as timetable of services or doctors, and agenda of the patient.

In the proposed scheme, any number of patients can access the scheduling system through patient-agent by filling all the details such as nature of disease, preferable time and date provided in the meeting request form to fix the appointments with appropriate doctor by searching the nearby hospital in the city. Upon receiving the patient request doctor agent will accordingly schedule, reschedule, or postpone the appointment meetings by viewing the available date in the doctor appointment calendar. If the patient arrival occurs at emergency case, the doctor agent will give first preference to emergency case and reschedule the appointment of already scheduled meetings and convey the same to the concerned patient. This scheduling system will reduces the conflicts between patients by negotiating best available date for meeting.

1.1 Why Agents?

Before answering the question of why agents might be useful, a few words should be said about what an agent is. Although there is no universal agreement, a popular definition, from [Wool95], describes an agent as a software entity that has the characteristics of, autonomy (acts independently), proactivity (goal-based), reactivity (responds in a timely fashion to events) and social ability (communicates with other agents to achieve goals collaboratively). Other characteristics frequently quoted include mobility (the ability to move from one host to another) and learning (the ability to improve performance overtime based on previous experiences).
Software with the above characteristics offers the possibility of systems which can lower the cost and improve the performance of businesses operations by

- Automating mundane tasks,
- Enabling users/customers to complete tasks that would otherwise be very difficult, time consuming, costly or just impossible, and
- Adapting to unexpected events or changes in the environment automatically.

Of course it may be possible to achieve cost saving and performance boosting solutions without agents, but agent technology provides a more natural model of the real world (i.e. a community of entities each with their own goals, communicating and often working together to achieve mutual benefit) compared to existing software paradigms, such as object-orientation.
Furthermore agent technology consolidates and builds upon a number of important computing technologies (object-orientation, distributed computing, parallel processing, mobile code, symbolic processing) and research results from other disciplines (artificial intelligence, biology, mathematics). In this way, agent technology offers a way to unify and simplify the use of the wide range of software technologies available today.

1.2 Problem Statement

Multi-agent systems are widely used to address large-scale distributed combinatorial real world problems. One such problem is meeting scheduling (MS) in health care domain that is characterized essentially by two features defined from both its inherently distributed and dynamic nature i.e. the presence of patient's preferences that turn it into a search for an optimal rather than a feasible solution. In this connection at least the following questions arise:

- When should the meeting take place?
- How to reach an appropriate hospital?
- What are the services available within hospital?
- How fast the doctor is available?
- Which Doctor is free to fix an appointment?

- How many patients will meet the doctor in a day, and who are they?

To solve it, today heuristics are used, because there is no optimal algorithm that fits for all possible solutions. The techniques of artificial intelligence are also used. An intelligent agent means that the agent has the knowledge about the interest and priorities of persons. Routine activities of physicians with regard to the meeting scheduling are practiced by agents in that way, that it filters and administrates information and answers questions. Supposing that every patient has got his own calendar, which is administrated by an agent, the reliability of his/her calendar will be very well. Also, a certain security of the private data is guaranteed.

"The problem is to develop a framework for distributed health care services using multi agent systems and to develop and implement an algorithm for the application of intelligent scheduling in health care domain using JAVA technology".

1.3 Scope of the study

The paper envisages development of framework and demonstration of the feasibility of distributed health care services using cooperating multi agents. Therefore a complex application of scheduling meeting for 'n' number of patients has been used. The development of parallel algorithms or task graphs for computations does not lie within scope of this paper.

1.4 Related Works

Here we present accessing of health-care related services by deploying intelligent agents. The software-agent paradigm [3] [4] was adopted due to its autonomous, reactive and/or proactive nature, which comprises of important features in real-time application deployment for dynamic systems like the one under consideration. Furthermore, software agents can incorporate coordination strategies, thus enabling them to operate in distributed environments and perform complex tasks. Software-agent technology is considered an ideal platform for providing data sharing, personalized services, and pooled knowledge. The work in [7] presents the Foundation for Intelligent Physical Agents (FIPA) that defines standards for agent interoperation. The aim in the Agent Cities is the construction of a worldwide publicly accessible network of FIPA based agent platforms. Each platform will support agents that offer services similar to those that can be found in a real city. Once the initial services have been deployed, it will be possible to implement intelligent complex compound services.

In the research literature, there are several agent-based applications reported in the healthcare domain. In particular, one of the earliest examples of work examining the role of multi-agent systems in healthcare is offered by [6]. The focus of the work presented there, and of the broader context, in which it was conducted, is upon appropriate theorem proving in decision support systems that have to deal with complex, incomplete, inconsistent and potentially conflicting data. The agent component is designed to support of tasks amongst players in the system. Heine et al [8] simulate an agent oriented environment for German hospitals with the objective to improve or optimize the appointment scheduling system, resource allocation and cost benefit of clinical trials. Nealon and Moreno [10] have discussed the potential and application of agents to assist in a wide range of activities in health care environments. Mabry et al [9] employ the Multi agent system for providing diagnosis and advice to health care personnel dealing with traumatized patients. Nealon and Moreno [2] have discussed various applications of MAS in health care e.g., coordination of organ transplants among Spanish hospitals, patient scheduling, senior citizen care etc. A research paper, called PalliaSys is offered by [15]. It incorporates information technology and multi-agent systems to improve the care given to palliative patients. An Intelligent Healthcare Knowledge Assistant [12] was developed which uses multi agent system for dynamic knowledge gathering, filtering, adaptation and acquisition from Health care Enterprise Memory unit.

However, it is observed from literature survey that when the Agent Cities initiative was made public, the potential development of agents that could offer not the usual leisure-oriented services but health-care related services. The work here describes automation of a multi-agent system that caters to special types of patients or providing assistance to patients for appointments. So, the concept of intelligent agent and mobile technology is used to achieve automation, efficiency, reliability and scalability in devising Health care domain for distributed, decentralized and dynamic environment to treat the patients efficiently by cutting down the time and cost.

2. Agent Technologies

Agents are considered one of the most important paradigms that on the one hand may improve on current methods for conceptualizing, designing, and implementing software systems and on the other hand may be the solution to the legacy software integration problem.

2.1 What is an Agent?

The term 'agent' or software agent has found its way into a number of technologies and has been widely used, for example, in artificial intelligence, database, operating system and computer networks literature. Even within the Agent Research Community, there are at least the following variants on the term agent: Mobile Agents, Learning Agents, Autonomous Agents, Planning Agents, Simulation Agents, and Distributed Agents. Although there is no single definition of an agent, all definitions

agree that an agent is essentially a special software component that has autonomy that provides an interoperable interface to an arbitrary system and/or behaves like a human agent, working for some clients in pursuit of its own agenda. Even if an agent system can be based on a solitary agent working within an environment and if necessary interacting with its users, usually they consist of multiple agents. Theses multi agent systems (MAS) can model complex systems and introduce the possibility of agents having common or conflicting goals. These agents may interact with each other both indirectly (by acting on the environment) or directly (via communication and negotiation). Agents may decide to cooperate for mutual benefit or may compete to serve their own interests.

In the data processing technology, an agent is software that supports a person, by executing autonomous several processes. Persons can delegate work to agents, instead of doing them on their own. Agents represent human users. The main difference to traditional software is their relative autonomy, which can be explained as a goal-directed, proactive and self-starting behavior. Software agents run continuous and autonomous in a defined environment, together with other agents and processes. Also, agents need:

- **Social ability:** agents communicate with their users, but also with other agents, using special agent-languages.
- **Reactivity:** agents perceive their environment, which can be their owner, other agents, the internet…and they react on different influences.
- **Pro-activity:** agents not only react on signals, but they also do independent actions, to reach a goal.

2.2 Software Agents

The software agents deal with how to do something, hiding details and work from the user who describes to the agent what to do. Agents act as much as possible without human intervention by learning from users' desires and making decisions for the user. Software agents are also dynamic and responsive to a variable environment. Agents ease and quicken the use of complicated systems. The Definition of a software agent is ambiguous but several key concepts are important. Software agents are usually goal-directed processes, which perform tasks autonomously delegated to them. It is situated in, aware of and reacts to its environment. An agent is also capable of cooperating with other agents, human or software, to accomplish tasks or to get new ones. Software agents are desired to be intelligent and mobile. These capabilities offer a new way to build very large heterogeneous applications.

The agent model working at a high level can be described as skills talented in different areas. Task level skills describe what capabilities the agent has for resolving tasks that user has given and how the agent can observe the environment and ways to handle information, for example database queries. Knowledge has the rules that agent follows as it goes on with a task. This is based on the awareness of the environment. This awareness is received by an agent in different ways: the developer has specified it by programming it in the application platform; the user can specify it by answering questions that the agent needs in its task or the agent can learn it from the environment or from the other agents. Communication skills are the agent's capabilities to communicate with other agents and with the user. The most natural way to communicate with humans would be by speech and facial expressions.

2.3 Taxonomy of agents

Agents can be classified by their capabilities and method of implementation as given bellow:

- **Collaborative agents** are autonomous and they communicate with each other. They can learn, but this is not essential. Collaborative agents have their power in the group.
- **Interface agents** intend to work for the user by helping autonomously, observing users habits and imitating them. The user can also instruct interface agents or they can ask for advice from other agents.
- **Mobile agents** are agents on the move. The ideal situation would be that mobile agents are sent to the Internet to do tasks for the user and when they are ready, to return home. Mobile agents also are capable of communicating and in an ideal situation they don't all collect the same information, they ask for it from other agents.
- **Reactive agents** are impulse driven. They react by producing a response to an impulse. These agents can trigger events and are suitable for handling sensor data. A reactive agent does not actually exchange data but more like knowledge.
- **Hybrid agents** are combinations of the above agents. They can be like GOSSIP, is a combination of an information agent and a mobile agent, which goes to the Internet and collects data for the user.

2.4 Intelligent agents

Agents in common are assumed to be intelligent, serving the user autonomously. The intelligence of agents comes from AI research, which has introduced different kinds of techniques like neural networks and genetic algorithms for problem solving and learning.

2.5 Multi agent cooperation

Coordination of the agents in a system is important to get the agents to reach the overall goal. Because of the distributed expertise, there is a need to coordinate everyone to prevent chaos and to make the system more efficient. Usually the different agents work toward a common goal, and therefore there is no conflict between them. The individual agent's objective does not matter, only the overall system. This is what Wooldridge mean about "benevolence assumption". In contrast some agents are self-interested. These types of agents have goals that will be in conflict with other agents. However, they still need to cooperate and it is important to find the best way to cooperate.

Coherence and coordination are two issues that need to be considered to decide how successful a multi agent system is. Coherence is the ability of a system to behave as a unit. Coherence is measured in terms of solution quality, efficiency of resource usage, conceptual clarity of operation, or how well system performance degrades in the presence of uncertainty or failure. Coordination is a process in which agents engage in order to ensure their community acts in a coherent manner. In a perfectly coordinated system agent do not need to bother about others sub-goal while achieving a common goal.

There are several ways different agents can work together to solve problems. Contracting is one solution to coordinate agents to work together. By using the contract net protocol standardized by FIPA (Foundation for Intelligent Physical Agents), the agents can cooperate by sharing tasks. A manager announces the problem to the other agents. As the agents listen to announcements and evaluate them with respect to their own resources, they place a bid if they find a suitable task. Several agents can bid for the same task, and then the manager has to decide from the information of the bid which agent should win the bidding round and then will be awarding the contract.

Another way to let agents cooperate to solve a problem, is result sharing. This will typically be that each agent solve small problems which later on will be become larger solution. Result sharing is when agents may share information relevant to their sub-problems. Durfee has suggested 4 ways to improve group performance:

- **Confidence:** When independently derived solutions can be crosschecked, the confidence in the overall solution is increased.
- **Completeness:** Agents that share their local views to achieve a better overall global view.
- **Precision:** The precision of the overall solution is increased when agents share results.
- **Timeliness:** As several agents work on the solution, the result could be derived more quickly.

2.6 Multi agent Systems

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Examples of problems which are appropriate to multi-agent systems research include online trading, disaster response, and modeling social structures. The agents in a multi-agent system have several important characteristics:

- **Autonomy:** the agents are at least partially autonomous i.e. agents operate without direct human intervention and have control over their own actions.
- **Local views:** no agent has a full global view of the system, or the system is too complex for an agent to make practical use of such knowledge
- **Decentralization:** there is no one controlling agent.

3. Proposed Scheme

This section describes the proposed model in terms of the network environment, hospital environment, patient environment and agencies involved in building and maintaining the medical data center, the agent interactions in discovering and building an automated meeting scheduler in health care domain to access distributed health care services, also the advantages and limitations of this proposed system.

3.1 Network Environment

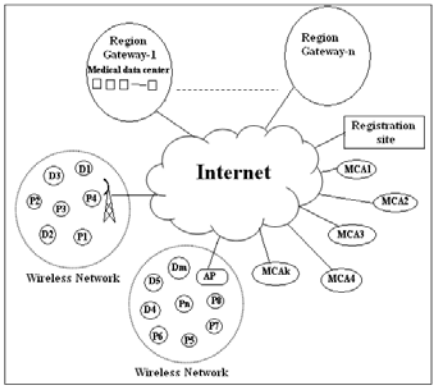

Fig. 1 Network Environment

Network environment for the proposed work is depicted in Fig. 1. The network environment consists of clusters of medical center agents (MCA1…MCAk) in a fixed network, regional gateways, a registration site, mobile patients (P1…Pn) and doctors (D1…Dm) in the wireless environment. Clusters are categorized based on their physical geographical locations where each cluster consists of medical center agents hosting several medical centers. The gateways are connected to the network based on the regions.

Mobile users or patients are in the vicinity of a wireless local area or in a cellular network. The mobile users or patients in a particular region request its regional gateway to fix / schedule an appointment with the doctor. The gateway comprises of medical center data, case base and the patient preferences to identify the relevant medical centers and doctors to coordinate the meeting scheduling process. An agent platform exists in all the components of network environment to facilitate agent based activities, since the information that must be dealt with is geographically distributed. The servers hosting medical centers are reliable and have sufficient bandwidth with good connectivity to accept requests from large number of mobile patients.

3.2 Hospital Environment

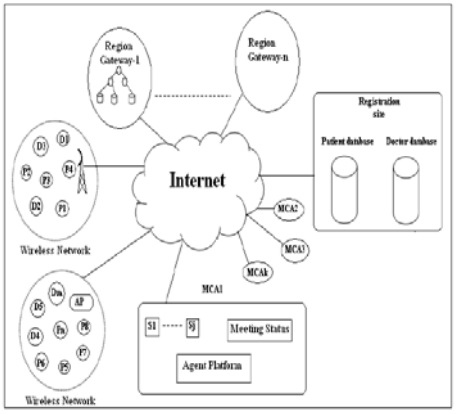

Fig. 2 Hospital Environment

The Fig. 2 depicts the Hospital Environment. The hospital environment consists of a number of Medical center agents that run concurrently on various servers that are connected to the WWW. Each medical center agent will keep its own data, and each doctor will have his/her personal information i.e. an up-to-date daily schedule in a personal computer comprising an agent platform that hosts an agency to carry out the meetings and communication with the patients. Meeting status contains each patients scheduling information such as confirmation of the appointments, postpone or rescheduling of the meetings. The registration site maintains the databases of patients and doctors also the medical records of the potential patients of the system. The medical data center on the regional gateway provides the information about all the medical centers and doctors working in that medical center. The agent platform supports persistence, security, communication and computing services.

3.3 Patient Environment

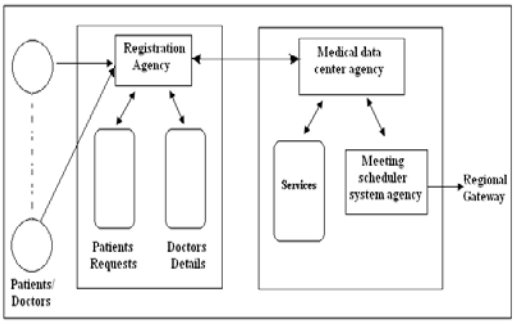

Fig. 3 Patient Environment

The Fig. 3 depicts the Patient Environment. The mobile patients register themselves in the registration desk of the regional gateway to fix an appointment with appropriate doctor by searching the nearby medical centers. The agencies involved in patient environment are registration agency, medical data center agency and meeting scheduler system agency. These agencies employ static and mobile agents to perform the dedicated tasks and focuses on scheduling the meeting for patients and building repository of services for mobile patients based on the patient agenda. Also these agencies will automate the process by enabling the mobile patients to complete the meeting scheduling process successfully without continuous online presence.

3.4 Advantages

The advantages of the proposed systems are as follows:

- Autonomy of fixing the appointments with doctors as per the patient's requests is achieved.
- Secure user access to medical records at any time.
- Support for user queries about the medical centers, and availability of doctors in the medical centers.
- Online booking for appointments with specialist doctors, whose offices in turn, automatically receive the appropriate medical records for reference and updating.
- High level accuracy and system reliability.
- Better time efficiency and flexibility due to quick and efficient retrieval of information any time.

4. Requirement Analysis and Specification

It is widely believed that the next generation of computer desktop applications will be significantly more proactive in helping users to achieve their goals than those which currently exist. Rather than the user having to specify each and every step of a given task, the desktop of the future will be composed of a series of intelligent agents to which a number of high level tasks can be delegated. These agents will be responsible for autonomously deciding how the task is to be achieved and actually

performing the necessary set of actions, including handling possible interactions with other intelligent agents. This chapter reports on the requirement analysis and specification of a particular agent-based application which arranges meetings for patients and doctors.

4.1 Problem Statement

To implement an agent based meeting scheduling system, which can schedule meetings for a set of patients. The patient, who wants to schedule a meeting between doctors, just fills the meeting request form that will be provided by an interface. When the patient submits it, the negotiation and scheduling process have to be automatic initiated and its control should be taken by the agent residing in the machine. The agents have to cooperate and do the negotiation on the behalf of each patient. An agent should work for every patient, so that the negotiation is done by it and the participant patients need not be present on their machine, provided their computer machine should be on and running this scheduling system, so the patients can view his updated meeting status any time.

4.2 Functional Requirements

Input: The patient fills all the details provided in the meeting request form to fix the appointments with appropriate doctor and submits it.
Output: The meeting is scheduled on a date and time convenient for all patients taking into account all the patient preferences. And the details of the scheduled meeting are added and displayed in the meeting status of every concerned patient.

5. Design and Implementation

A detailed design using Unified Modeling Language (UML) notation with diagrams and implementation details are given.

5.1 System Design

Designing a system mainly focuses on the detailed implementation of the proposed systems. It emphasizes on translating performance specifications recorded at the time of system study into design specifications. System Design phase is a transition from a user-oriented document to on the methods adopted for developing the system. Design part is the pivotal point in the system development life cycle.

5.1.1 Design

In this phase the architecture of the proposed system is conceived and developed. The architectural diagram helps in a smooth transition between the design stage and the implementation stage. The various factors that are considered before developing the system architecture are cost, reliability, accuracy, security, control, integration, expandability, availability, and acceptability.
The elegant design achieves its objectives with minimum use or resources. The system analyst must have clear understanding of the objectives that the design is aiming to fulfill. There is usually more than one way of achieving a desired set of results.

5.2 Architecture

The architectural diagram for the MAS is shown in Fig. 4. The aim of the MAS is to provide access to the basic health care services in a given city to the patient and to schedule the meeting between patient and doctor. The architecture shows interactions among agents, and also the interactions between humans/resources and agents.

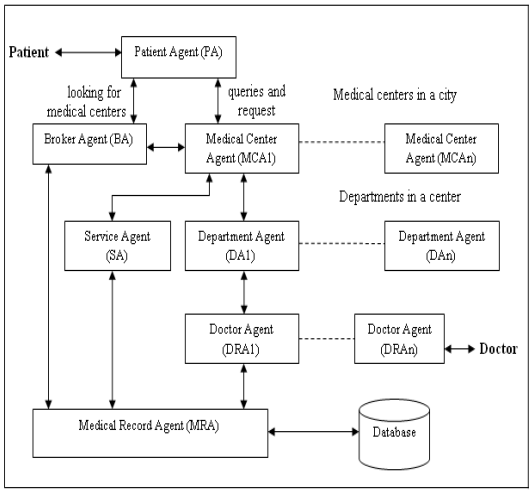

Fig. 4 Architectural diagram for the MAS

A patient interacts with the system through a Patient-agent (PA), provided a GUI through which patients could make queries and receives answers. This agent stores static data related to the patient such as the national healthcare number, name, address, phone number, and information for allowing a secure access to the system (login, password, keys). It also stores dynamic data such as the agenda of the patient. The static data will be used to identify the patient in the system (authentication and ciphering). The agents of the system will exchange required data automatically in each step, *e.g.* a doctor needs to know personal details of a patient before the medical visit, in order to retrieve his/her medical record from a database. The dynamic data is very useful to guide negotiations between any PA and other agents, because a PA can avoid coincidences in those negotiations, *e.g.* if the patient works from 9:00AM to 14:00PM, his agent would arrange meetings during the afternoon and night.

All PAs can talk with a Broker Agent (BA) provided an interface between all the agents internal to the system and the patient-agents. The BA is the bridge between

patients and the medical centers, and it is used to discover information about the system. All PAs can ask this agent in order to find medical centers satisfying certain criteria. The BA covers all the medical centers located in a city or nearby area.

Patient can access the system through the Medical Centre Agent (MCA) that centralizes and monitors the outsider's accesses. Each medical center is represented by Medical center Agent which contains all the information related to the medical center such as address, phone number, opening times, location, and so on. A MCA monitors all of its departments, represented by Department Agents (DAs), and a set of general services represented by Service Agents (SAs), such as a blood test service, ambulance etc. Each department is formed by several doctors represented by Doctor Agents (DRA) with specialization, free time and day of doctors.

Database is used to store all patients' medical records which can be accessed through the Medical Record Agent (MRA). This agent provides a secure access to the data using authentication. When a patient wants to arrange an appointment with a doctor, or a doctor must arrange a visit of a patient with a service, it is required to schedule a meeting according to different constraints such as timetable of services or doctors, and agenda of the patient. Here the patient-agent will search nearby hospitals by selecting city or area and category of hospitals also the available services in the hospitals. The patient-agent will then request for the appointment dates with the doctor through the doctor agent. The doctor agent will view the list of appointment request and accordingly it confirms the request or reschedules the appointment date and time as per the free time and day of doctor and confirm the same to the respective patient. If the patient arrival occurs at emergency case, then doctor agent will give first preference to emergency case and reschedule the appointment of already scheduled meetings and convey the same to the concerned patient. This scheduling system will helps in reducing the conflicts between patients by negotiating best available date for meeting.

The goal is to create an automated meeting scheduling agent in health care domain that is:

- It allows the patients to input his/her meeting request.
- Negotiates with the agents of the other requested patients.
- Finds out its best fitting and free time slots.
- Compares them with the sent fitting slots of the patient-agents and find out the best ones.
- Reacts to the incoming patient request by sending back its best fitting free time slots.
- Shows all fixed meeting in a time table.
- Allows the patient to input his/her preferences.

5.3 Phases

This paper implements the above mentioned goals by using Java programming tools. The phases include the complete life cycle of the multiagent system to schedule the meeting between the patients and doctors are given bellow in steps.

Step 1: Register the members to the centralized database server. This centralized database server maintains a list of hospitals in a city, with each hospital containing different departments with associated services and list of doctors with different specialization and free time and day of doctors.

Step 2: When the application on the centralized server is executed, any number of patients can access the system through patient-agent and send the meeting request form by filling all the details such as nature of disease, preferable time and date to fix the appointments with appropriate doctor by searching the nearby hospital in the city.

Step 3: The patient will first open the login page. If he/she is a new patient then he/she will click new patient and Register. The window will be the registration page of the patient. Once the patient registers he will be activated by the broker agent and can easily login.

Step 4: The patient may request information about all the medical centers available in a particular city. If the patient is aware of a specific medical centre in the area, he/she may request information about the medical services and doctors in that centre. Also it is possible to book a visit to a doctor. In this kind of request the patient-agent has to select the Broker Agent as the recipient of the message. As BA is aware of all Medical Center Agents in town, it will find out which of them satisfy the patient's constraints.

Step 5: Broker Agent will have a predefined user name and password through which he will do various operations and he will insert, update any data from the database based on complaints received from patient-agent. Broker agent will deactivate any member at any time. Also Broker agent will add new area, category of hospital, specialty of hospital and new hospital to the database.

Step 6: The Patient-agents sends a request (REQ) to the MCA through BA. This REQ is forwarded to the department selected by the patient-agent. The Department Agents (DA) will send the REQ to the Doctor Agents. The Doctor Agents (DRA) replies to the request, in which it displays the earliest time in which the doctor has a free slot for making a visit. The patient will view his Meeting status any time such as confirmed, postponed, or rescheduling of the meetings.

Step 7: The Doctor will login any time and view the List of recent appointment request, List of forthcoming appointments and the calendar showing the available

dates and time for meeting and scheduled meeting time table.

Step 8: The Doctor Agent will assign the time slots of each day, week with their respective priorities for doctors and fix the meeting using date, priority and time into consideration.

Step 9: If the patient arrival occurs at emergency case, the doctor agent will give first preference to emergency case and find a best fitting time slot for the meeting. Finally, the doctor agent confirms that the schedule of the doctor has been modified, and this confirmation is sent to the patient-agent through department agent and medical center agent.

Step 10: The medical records of the patients are stored in a database called Medical Record Agent (MRA), the access to which is controlled by Database Wrapper (DW).

6. Results

In this section, the simulated results obtained with the proposed work are discussed.

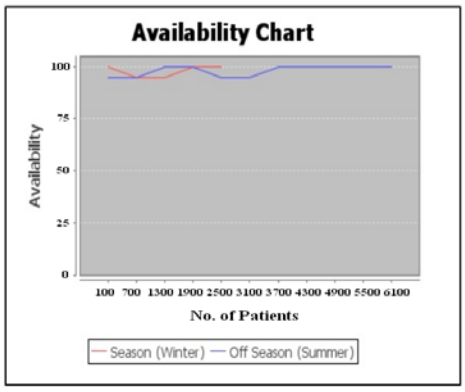

Fig. 5 Success Rate of Getting an Appointment

The Fig. 5 depicts the Success Rate of Getting an Appointment. The X-axis represents the number of patients and Y-axis represents the availability of doctor in terms of percentage during season and off season. During off season, the number of disease will be less (summer) and the availability of doctor will be more when compared to season (winter). During season, the number of diseases will be more, hence the requests will be more and also the availability of doctor will be less.

Season here means - when there are more patients i.e. when there is transition in climate.

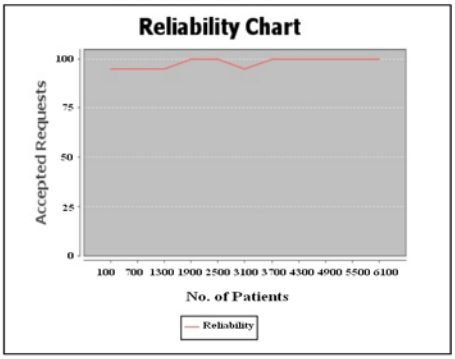

Fig. 6 Reliability of the System

The Fig. 6 depicts the Reliability of the System. The x-axis represents the number of patients and y-axis represents the number of accepted requests in terms of percentage. As the number of patient's increases i.e. the number of requests will be high, so higher the system reliability.

The Figure depicts the Response Time of the System. The X axis represents the number of patients and Y axis represents the time in terms of mili seconds. Response time is the time required to process the request. i.e. after sending the request, how fast it will get confirmed was calculated.

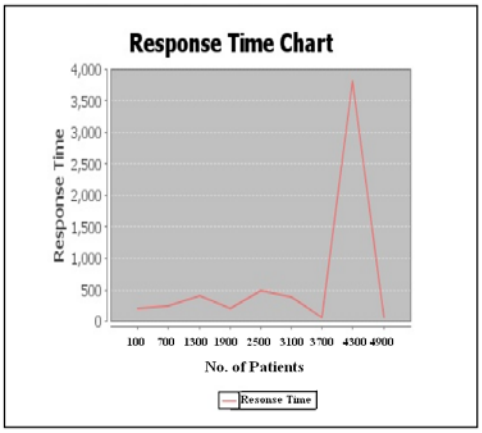

Fig. 7 Response Time of the System

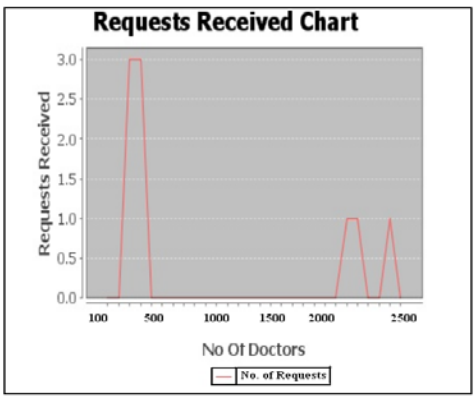

Fig. 8 Success Rate of Getting the Patients

The Fig. 8 depicts the Availability of Patients. The X-axis represents the doctors in the hospitals and Y-axis represents the Number of request received from the patients. More the number of patients, as the number of request received were more.

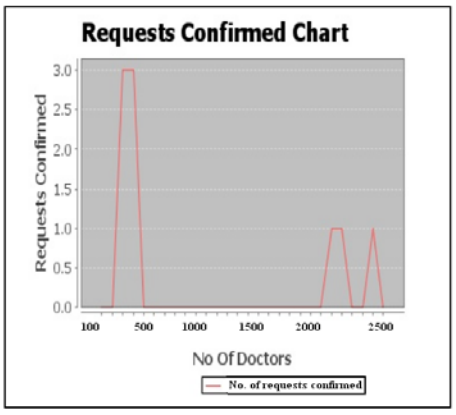

Fig. 9 Reliability of the System

The Fig. 9 depicts the Reliability of the System. The X axis represents the Doctors in the hospitals and Y axis represents the number of accepted requests (processed or confirmed request). As the number of requests increases, the number of requests accepted by the doctor will decrease.

7. Conclusion and Future Work

This paper presented a framework of Intelligent Scheduling in Health Care Domain. The use of agents in health care has experimented an important growth. One of the main benefits of this paradigm is to allow the interoperability of preexisting systems for improving its general performance. We have designed and implemented an agent-based information services for mobile users. The architecture defines the interaction between agents, also between humans and agents. The interaction human-agent is made through personal agents that could be located in computers or mobile devices.

The characteristics of the agent such as the concepts of situatedness, autonomy and flexibility will helps in solving many problems that appear in the health care domain. One such problem we discussed here is access to distributed medical information of a city to schedule the meeting for patients and relevant doctors meeting in an efficient way for routine and emergency services.

Multi agent system was developed to represent the real conditions, courses, and the human decision behavior and to present the overall design of the proposed MAS, emphasizing its architecture and the behavior of each agent of the model, as well as on the scheduling model which provide the activity scheduling process of care and the agent interaction protocol to ensure cooperation between agents that perform coordination tasks for the users. The system implements services as reusable as possible also, it could easily allow the addition of new agents or features to further improve the time efficiency.

In the current implementation the users/patients personal assistant is simulated through a web interface as discussed. In this prototype all the agents are running in the same computer; in order to be usable in a real mobile environment, WAP-accessible version of the MAS must be used. The latest version of JADE, JADE-LEAP v2.4 can be used to achieve better performance and reliability than the existing mechanism.

8. References

1. Zgaya, H. Design and distributed optimization of an information system to aid urban mobility: A multiagent approach to research and composition of services related to transportation. Doctoral Thesis, Ecole Centrale of Lille, 2007.

2. Nealon, J. and Moreno, A. The application of agent technology to health care, Proceedings of the Workshop AgentCities: Research in Large-scale Open Agent Environments, in the 1st International Joint Conference on Autonomous Agents and Multi- Agent Systems (AAMAS „02), pages 169-73, Bologna, Italy, 2002.

3. Weiss, G.: Multiagent systems. A modern approach to Distributed Artificial Intelligence. M.I.T. Press (1999).

4. M. Becker, C. Heine, R. Herrler, and K.-H. Krempels. OntHoS – an ontology for hospital scenarios. In John L. Nealon and Antonio Moreno, editors, Applications of Software Agent Technology in the Health Care Domain, Whitestein Series in Software Agent Technologies, pages 87–104. Birkh¨auser Verlag, Basel, Switzerland, 2003.

5. R. Haux, E. Ammenwerth, W. Herzog, and P. Knaup. Health care in the information society. A prognosis for the year 2013. International Journal of Medical Informatics, 66:3–21, 2002.

6. V. Shankararaman, V. Ambrosiadou, T. Panchal, and B. Robinson. Agents in health care. In V. Shankararaman, editor, Workshop on Autonomous Agents in Health Care, pages

7. FIPA Agent Communication Language: FIPA ACL Message Structure Specification

8. Heine, C., Herrler, R., and Stefan, K. Agentbased Optimisation and Management of Clinical Processes. Proceedings of the 16th European Conference on Artificial Intelligence (ECAI”4)-The 2nd Workshop on Agents Applied in Health Care. 2004.

9. Mabry, Susan L., Hug, Caleb R., Roundy, Russell C. Clinical Decision Support with IM-Agents and ERMA Multi-agents .cbms.page.242. 17th IEEE Symposium on Computer-Based Medical Systems (CBMS“04), 2004.

10. Nealon, J. and Moreno, A. Agent-Based Applications in Health Care. In Applications of Software agent technology in the health care domain. Whitestein Series

in Software Agent Technologies. Birkhauser Verlag, Basel. 2003.

11. Riano, D., Prado, S., Pascual, A. and Martin, S. June. A Multi-Agent System to Support Palliative Care Units. Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems (CBMS 2002). 2002.

12. Hashmi, ZI., Abidi, SSR. and Cheah, YN.. An Intelligent Agent-Based Knowledge Broker for Enterprise wide Healthcare Knowledge Procurement, Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems (CBMS 2002). June 2002.

13. Petrie, C. Agent-based software engineering. In: Agent-Oriented Software Engineering. Lecture Notes in Artificial Intelligence 1957, (2001), 58-76.

14. Jennings, N. On agent-based software engineering. Artificial Intelligence 117, (2000), 277- 296.

15. Riano, D., Prado, S., Pascual, A. and Martin, S. June. A Multi-Agent System to Support Palliative Care Units. Proceedings of the 15th IEEE Symposium on Computer Based Medical Systems (CBMS 2002). 2002