ISSN 2250-3153

Implementation Scheme for Online Medical Diagnosis System Using Multi Agent System with JADE

Mr.Shibakali Gupta, Arindam Sarkar, Indrani Pramanik, Banani Mukherjee

Computer Science & Engg. , University Institute of technology, Burdwan University, Golapbag, Burdwan (W.B)

Abstract- In this proposed methodology we are building agent based information services for internet users. In our work we are considering the use of agents that will provide Medical Care related services. In this paper we report the design and construction of a Multi-Agent System (MAS) that is composed of agents that provide medical services. The implemented system contains agents developed using JADE that allow the user to search for medical centres satisfying a given set of symptoms, to access his/her medical record or to make an appointment to be visited by a particular kind of doctor in his/her location. He/She can also chat or mail to a doctor. And can get various tips about health care and diseases.

Index Terms- Multi Agent System, JADE, Health Care, Agents.

I. INTRODUCTION

As an initial problem in the domain of Health care service [1] and diagnosis, we choose online medical service system .Now a days internet has expanded in everywhere .And this proposed methodology is concerned about the online treatment and creating a Medical Diagnosis System [2] consisting of agents developed using JADE [8][9] constituting a Multi Agent System. Imagine that a user is going for a tour in an unknown city. It is the first time that he goes to that city, so he only knows the location of the airport, the hotel and the tour sites. After visiting a few sites and having a heavy dinner in the night he doesn't feel very well, for two hours. The user is a diabetic patient, so he is quite worried about this condition. Fortunately, the user carries his laptop with him. He now opens the site OnlineDocs.com. He enters his symptoms in the site and in a few seconds he have on the laptop's screen the details of the closest and best hospital, nursing home, clinics and the doctor's details. Then, the site provides information about the timetables of the doctors of that medical centre that are diabetic specialists and with a list of three doctors; one of them performs examinations on the afternoon, and another one on the evening (the third one is on holidays). He confirms to book a visit to one of the two available doctors as soon as possible. OnlineDocs tries to book a visit to the first doctor by contacting the doctor agent, but he is told that the doctor has a full schedule. However, it manages to book a visit to the second doctor at 18:00. Then, he can see in the laptop's screen a confirmation of the booking. At 18:00 he goes to the hospital .After explaining the problem to the doctor. And then the doctor logins as an administrator to our site and asks the site to retrieve his medical record (which can be done with his medical ID code and a password that he introduce in the computer's keyboard). In a few seconds he can see the previous health conditions of user, the treatments and operations he might have had in the past, the medicines to which he is allergic, etc. After a detailed examination, the doctor concludes that the problem is very serious, as the user's sugar level has increased very much due to excessive sugar consumption and so he gives the patient medicines and asks him to take some rest. He also adds the information gathered in this examination to the user's medical record, so that it can be taken into account in the future. And after that the site will give him information about where the prescribed medicines by the doctor are available in that city. Now after taking medicines for few days he can also report his progress to the doctor through the chatting and mailing functionalities of the website.

Now suppose there are a large number of medical records of users which have cases of malaria then the system will generate an epidemic alert to the *media and government agent.*
This scenario is not as far-fetched and futuristic as it may seem at first glance. If agent technology delivers the results it is promising, the facts depicted in this example could indeed happen in a real setting. The challenges involved are many and come from five main areas:

1. Communication between services - e.g. sharing ontology and semantics which could be heterogeneous even within the health care domain.
2. Security/Authorization-accessing/editing sensitive medical data, such as the medical records of the patients.
3. Creating an environment where agents can discover one another and access one another's services.
4. Communication between users and agents -e.g. graphical user interfaces to communicate with personal assistants.
5. Co-ordination between distributed services, i.e. making complex distributed decisions (for instance, the medical centers could get in touch with one another so that patients are evenly distributed between them).

The work reported in this paper is a first step towards these future scenarios, with contributions to primarily the first four challenges. The final aim of the project is to design and implement a system which could be run in an environment such as the *Agent Cities* network, which is attempting to create a technical environment in which some of the above topics may be studied. In this paper we describe the design and implementation of a MAS that offers some of the services which have been requested in the previous paragraph. *Main agent* of the system can provide information about the medical centers that are available in a given city. The MAS also contains an *Doctors agent* for each medical centre 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 and can also access a database in which the medical records of the users are stored. The developed system also contains *Medicine agents* for each drug centre in the city which provides information about the medicines available. Therefore, when agent (and telecommunication) technology is a little bit more mature, we will hopefully be able to access much useful health- related information and facilities all around the world.

The rest of the paper is organized as follows. First of all, the *Agent Cities* project, which is the framework in which this work has been made, is briefly introduced, and we argue why multi-agent systems offer an appropriate platform for developing health-related applications. After that, a detailed description of the MAS that has been designed and implementation scheme is given. We explain the architecture of the MAS, the services offered by the different types of agents and the ontology used in this work. The paper finishes with a discussion of the work done and an outline of some potential future lines of research.

II. BACKGROUND

The *Foundation for Intelligent Physical Agents* (*FIPA*) [10] is an organization that defines standards for agent interoperation. *Agent Cities* [11] is an ambitious project whose main aim 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 (e.g. facilities, amenities, attractions, information and commercial services). The agents that are planned to be initially deployed in this network will focus on information and booking services in domains such as cinemas and theatres, restaurants, hotels, taxis, magazines, tourist information, weather information and a meeting scheduling service. It is expected that, once these initial services have been deployed, it will be possible to implement intelligent complex compound services (e.g. agents that are able to help a user to plan a weekend away, including tasks such as booking air tickets, selecting and booking a room in an appropriate hotel, buying tickets for the theatre and reserving a table in a restaurant that is near the theatre and offers the user's favourite meal). Our group has been working in the last twelve months in the application of AI techniques (especially agent-based technology) to the medical domain [5]. When the *Agent Cities* initiative was made public, we immediately imagined the potential development of agents that could offer not the usual leisure-oriented services but health-care related services. Therefore, we decided to design and construct a multi-agent system with the following characteristics:

- The user may request information about all the medical centres available in a particular city by giving his/her symptoms.
- If the user is aware of a specific medical centre in the area, he/she may request information about the medical services and doctors in that centre.
- It is possible to book a visit to a doctor.
- The user may access his/her medical record.
- The user can also find medical stores in the city.
- The user can chat or mail to the doctors.
- Media and Government user will get a possible epidemic alert.

The decision of using a Multi-Agent System in this medical setting (and not other more traditional AI techniques such as an expert system or a decision support system) is motivated by the following reasons:

• The information that must be dealt with is geographically distributed, because each hospital or medical centre will keep its own data, each doctor will have his/her personal information (e.g. an up-to-date daily schedule) in a personal computer, the medical records of the potential users of the system may be located in different databases, etc. Therefore, a distributed AI approach (such as the one offered by multi-agent systems) seems suitable in this case.

• There must be a fluent communication between the user and the medical centres. For instance, the user's agent should be able to ask for a booking with a certain doctor, and be able to react quickly if the doctor's schedule is full (so another doctor has to be chosen). Agents are not only reactive but also endowed with social abilities, so they are able to communicate with other agents in order to negotiate and co-ordinate their activities.

• Existing systems (such as databases containing medical records) may be easily included in a multi- agent system. The standard way of *agentifying* a database is to put a *wrapper* around it. A *wrapper* is an agent that receives the queries to be made to the database in a standard agent communication language (such as FIPA-ACL or KQML) and is able to translate these requests into queries in SQL to the database. After receiving the reply from the database, the wrapper may translate the answer to the common agent communication language and send it to the requesting agent.

• Health-oriented agents could perform pro-active tasks in order to have ready all the information that the user may need at a particular time. For instance, the user's agent might keep, in the user's health profile, the information that he/she has had heart conditions in the past. When the user travels to a foreign city, the personal assistant could immediately (without an explicit demand from the user) search all those medical centres in which there are heart specialists, and have this information ready in case the user needs it. This property has not been implemented in the agents described in this paper, though.

• The user should be able to access his/her medical record from anywhere. For instance, the user may be on holiday in a rural area, far away from medical centres, and need to be examined by a local doctor. If the user may access his/her medical record from a mobile phone or a personal computer connected to the Internet, the doctor may take it into account in order to make a more precise diagnosis.

• Finally, perhaps the most important reason for capturing services as agents in this way is to enable the individual medical services to interact with each other at a high enough level to ensure that they can all interoperate. Agent communication languages (such as FIPA-ACL), content languages (such as SL) and formal ontologies are very useful in describing communication between different services at the application level – i.e. in a way which relates to the domain of discourse rather than to any single implementation. Without formal specifications for domain ontologies it would be almost

impossible for diverse medical systems to interact with each other correctly.

III. ARCHITECTURE OF THE MASS AND IMPLEMENTATION

This section describes the Multi-Agent System [3] [4] which have been developed. The primary design objectives of the work were the following:

• To provide a decomposition of the problem that matched agents to entities which could be realistic players in such a domain (e.g. Medical centres, personal agents, etc.) And to take care in who had access to which information.

• To provide an ontology for the domain (a detail often ignored in demonstrations).

• To make the developed agent services as reusable as possible by using standard languages- in this case FIPA-ACL for communication and FIPA-SL for content - and providing detailed service models to describe the individual functioning and objective of each agent including descriptions of actions, protocols used and example messages.

These points are particularly important since the aim is to make re-usable service components which could be used in many applications (not all of them necessarily medical). The aim of the multi-agent system is to provide access to the basic health-care services in a given city to the users. The basic architecture of the MAS which has been developed in this work is shown in figure 1.

This multi-agent system contains six different types of agents: the *main agent* (MA), the *user agent* (UA), the *medical store agent* (MSA), a *database wrapper* (DW) and some *doctor agents* (DA), *media and govt. agent* (MGA).

User Agent:

- User agent enters his/her symptoms and gets the list of available doctors in clinic, hospital, nursing homes in the city.
- The user agent can chat or mail to the doctors.
- The user agent can view his/her health profile.

Main Agent:

- The main agent searches the best available doctors in the city.
- The main agent fixes appointment to the doctors.
- The main agent searches the medicines available in the different medical stores in the city through medical store agent.
- The main agent send alert about possible epidemic situation to the media and government agent in the city.

Doctor Agent:

- The doctor agent prescribes the medicine.
- The doctor agent updates the medical records.

Media and Government Agent:

- The media and government agent can view the alert about the possible epidemic situation in the city.

Database Wrapper:

Database wrapper stores the data and performs analysis on the data to generate the required result.

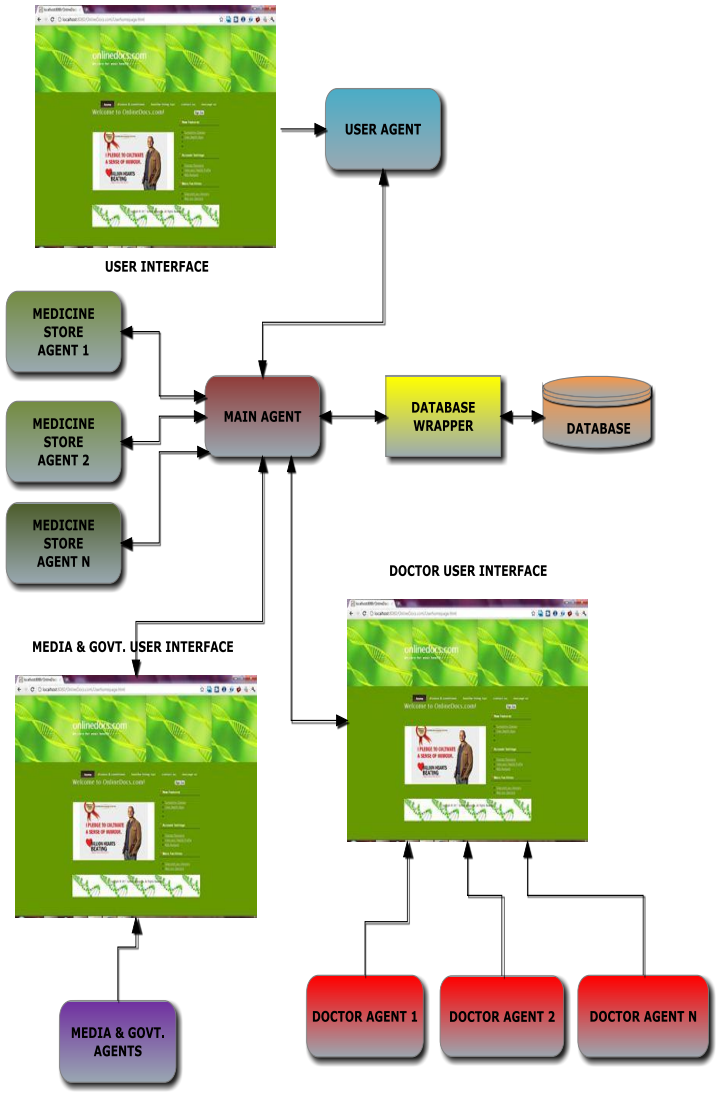


Fig.1 Architecture of the MAS

A. *FIPA Agent Platform*

The standard model of an agent platform, as defined by FIPA, is represented in the following figure 2,

- **Agent Management System**

The Agent Management System (AMS) is the agent who exerts supervisory control over access to and use of the Agent Platform. Only one AMS will exist in a single platform. The AMS provides white-page and life-cycle service, maintaining a directory of agent identifiers (AID) and agent state. Each agent must register with an AMS in order to get a valid AID. The Directory Facilitator (DF) is the agent who provides the default yellow page service in the platform.

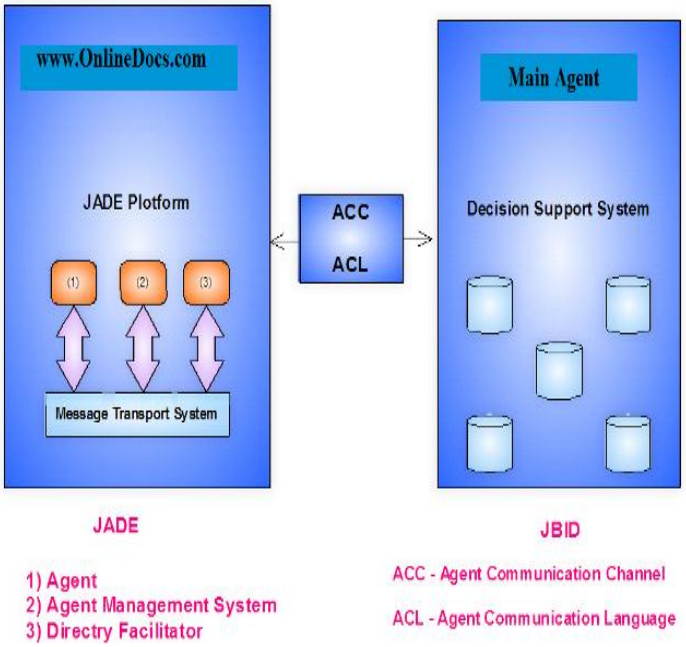


Fig.2 Reference of FIPA Agent Platform.

- **Agent Communication Channel**

The Message Transport System, also called Agent Communication Channel (ACC), is the software component controlling all the exchange of messages within the platform, including messages to/from remote platforms. JADE fully complies with this reference architecture and when a JADE platform is launched, the AMS and DF are immediately created and the ACC module is set to allow message communication. The agent platform can be split on several hosts. Only one Java application, and therefore only one Java Virtual Machine (JVM), is executed on each host. Each JVM is a basic container of agents that provides a complete run time environment for agent execution and allows several agents to concurrently execute on the same host.

The main-container, or front-end, is the agent container where the AMS and DF lives and where the RMI registry, that is used internally by JADE, is created. The other agent containers, instead, connect to the main container and provide a complete run- time environment for the execution of any set of JADE agents. According to the FIPA specifications, DF and AMS agents communicate by using the FIPA-SL content language and the FIPA-agent-management ontology, and the FIPA-request interaction protocol. JADE provides compliant implementations for all these components. The following messages are used by agents to communicate with other agents:

a. Accept Proposal - One Agent accepts the proposal of another agent
b. Agree - The process of mutual agreement between agents for request and response
c. Inform - One agent informs to the agent about the process
d. Failure - The communication failure is send as the response
e. Propose - One agent submits some task to the other agent
f. Refuse - The process of denying some task
g. Request - Request for some action to be performed

B. Experimental Design and Implementation of Software Architecture

The user is authenticated, and he/she has been taken into website user part. The user may now give his/her symptoms to the User agent (UA). After that expert advice i.e., recommended doctors or appropriate medicine stores are provided of the city by the Main agent (MA).Once the expert advice has been selected by the User agent (UA) a XML file through the J2EE parser is created and is sent to the Main agent (MA).

The specific requirement of XML file is that, in near future JADE [6] may be moved to some other server. It gives flexibility to transfer the data from one server to another server. The XML file is parsed using J2EE Parser and data is transferred to JADE agents. Intern it is forwarded to Main agent (MA). Main agent (MA) is connected with Decision Support System and the MYSQL database. Main agent by communicates with the Doctor Agent (DA) and make intelligent analysis and returns the best doctors and best medicine centres available in the city. The above steps are carried out in backward direction and Media and Govt. Agents gets a possible alert in case of an outbreak. The software architecture is shown in figure 3,

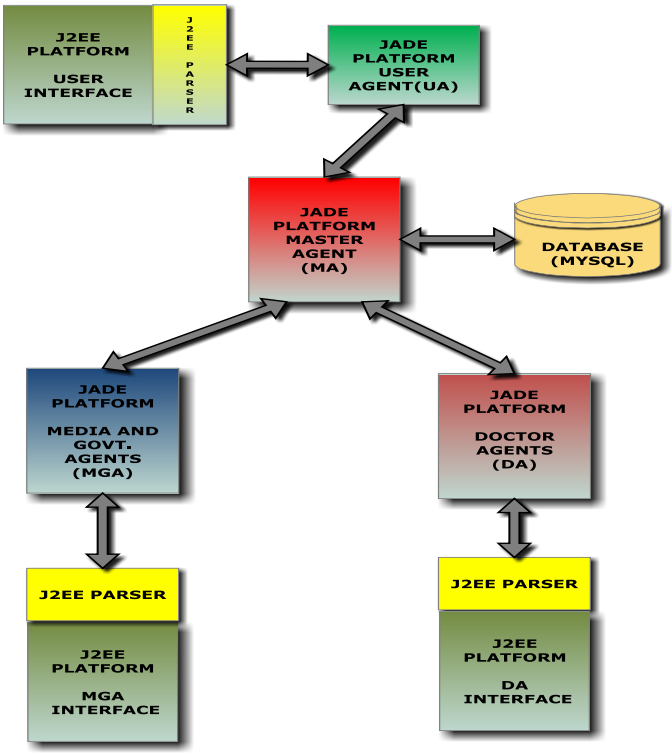


Fig.3 Software Architectural design for the System.

C. Experimental Result Screenshots

The different sections of the above proposed system is shown in the figure below,

Fig. 4 User Interface (Webpage) for the Medical Diagnosis System developed using JSP and HTML.

Fig. 5 A JADE platform starter Agent System which will start the Agents on clicking a link on the user homepage.

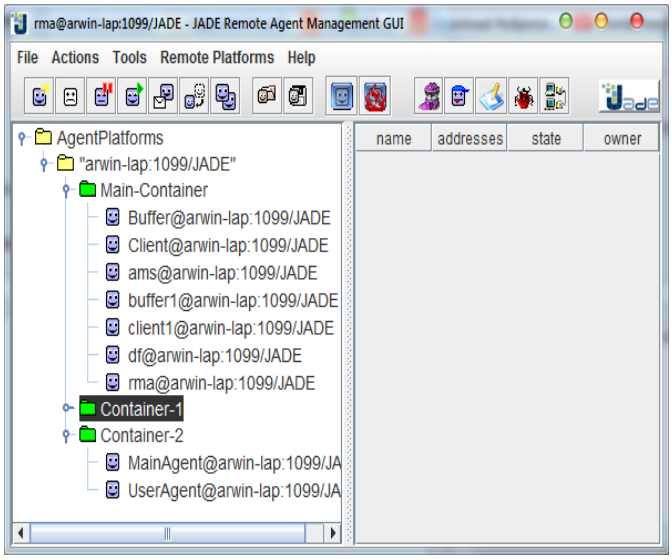
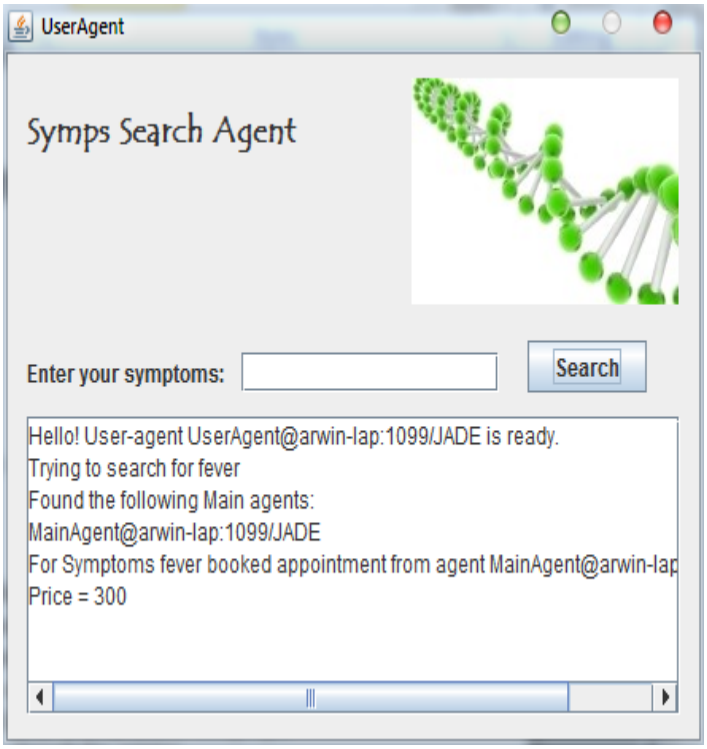


Fig. 6 A JADE platform containing the Main Agent, User Agent and Doctor Agent



.

Fig. 7 The User Agent taking the symptoms fever as given by the user as an input and give it to the Main Agent.

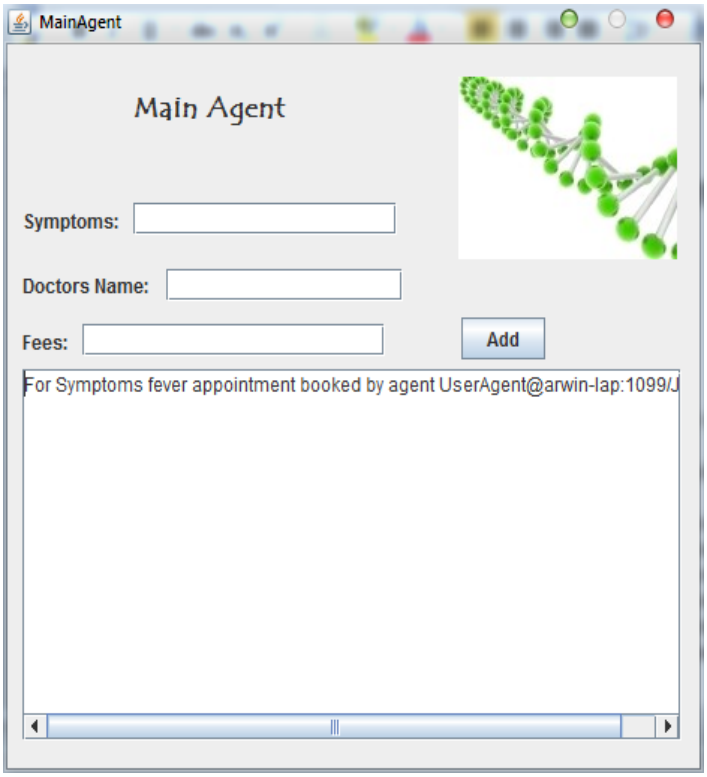


Fig. 8 The Main Agent searching for the symptoms fever given by the User Agent and fixing the appointment to the best doctor available in the city.

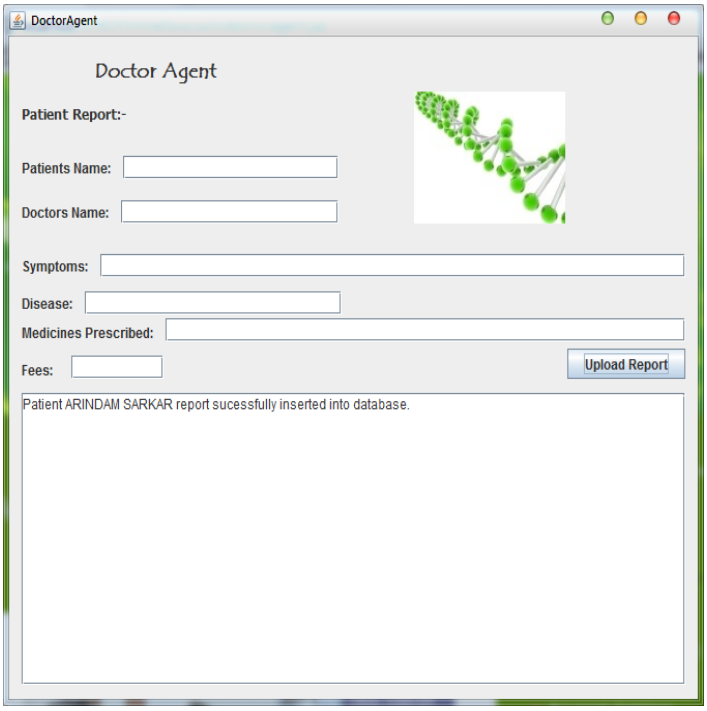


Fig. 9 The Doctor Agent is uploading the User prescription to the database and then the medicine agent working in the background find those medicines and give it to the User Agent.

IV. CONCLUSION & FUTURE WORK

The proposed implemented version of Medical Diagnosis System (MDS) can take care of every stage of patients such as initial checkup, treatment and report for the patients. We believe that the implemented scheme can bring a revolution in our Medical Diagnosis System mainly for Indian environment. In future correspondence the incorporation of KQML (Knowledge Query Manipulation Language) [7] can be done to have more efficient data mining or knowledge mining.

REFERENCES

[1] Pricewater House Coopers, Emerging Market Report: Health in India 2007,2007

[2] Shibakali Gupta, Shiladitya Pujari, A multi agent system (MAS) based scheme for health care and medical diagnosis system, IAMA 09, ieeexplore, ISBN 978-1-4244-4710-7,2009

[3] Michael Wooldridge, "An Introduction to Multi agent Systems", Department of Computer Science, University of Liverpool, Uk JOHN WILEY AND SONS, LTD.

[4] Katia P Sycara Multiagent Systems, AI magazine Volume 19, No.2 Intelligent Agents Summer 1998.

[5] Antonio Moreno, Medical Applications of Multi Agent Systems, Computer Science and Mathematics Department, Universitat Rovira, Virgili, ETSE. Campus Sescelades.Av.delsPasosCatalans, 26, 43007-Tarragona Spain.

[6] Developing Multi-agent Systems with JADE,Fabio Bellifemine1, Agostino Poggi2, and Giovanni Rimassa21,CSELT S.p.A.,Via G. Reiss Romoli, 274, 10148, Torino, Italy,Dipartimento di Ingegneria dell'Informazione, University of Parma Parco Area delle Scienze, 181A, 43100, Parma, Italy.

[7] KQML as an agent communication language Tim Finin, Yannis Labrou, and James Mayfield, in Jeff Bradshaw (Ed.), "Software Agents", MIT Press, Cambridge, to appear, (1997)

[8] JADE:Java FrameWork For Agent Development, http://sharon.cselt.it/projects/jade

[9] JADE:Java framework for Agent Development,http://www.jade.tilab.com

[10] FIPA:Foundation for Intelligent Physical Agent,http://www.fipa.org

[11] Agent Cities: http://www.agentcities.org.

AUTHORS

First Author – Mr.Shibakali Gupta,
Computer Science & Engg. ,
University Institute of technology, Burdwan University,
Golapbag, Burdwan (W.B). Email id - skgupta.81@gmail.com

Second Author - Arindam Sarkar, Computer Science & Engg.,
University Institute of technology, Burdwan University,
Golapbag, Burdwan (W.B). Email id – arsoftech@gmail.com

Third Author - Indrani Pramanik, Computer Science & Engg.,
University Institute of technology, Burdwan University,
Golapbag, Burdwan (W.B).
Email id - pramanikindrani@gmail.com

Fourth Author-Banani Mukherjee, Computer Science & Engg.,
University Institute of technology, Burdwan University,
Golapbag, Burdwan (W.B). Email id – rajtukun@gmail.com