BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **13**, No 3

Sofia • 2013 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2013-0031

Enhancing a Traditional Health Care System of an Organization for Better Service with Agent Technology by Ensuring Confidentiality of Patients' Medical Information

R. Jaya Subalakshmi, Haleema, N. C. S. N. Iyengar

School of Computing Science and Engineering, VIT University, Vellore – 632014, Tamilnadu, India
Emails: jayasubalakshmi.r@vit.ac.in haleema@vit.ac.in nchsniyr@vit.ac.in

Abstract: *Agent technology is one of the widely adapted technologies for developing applications that deliver e-Services. Ensuring confidentiality of the patients' data in e-health care systems remains a serious challenge. Many large enterprises provide in-house health care services free of cost for their employees and their dependents as a competitive benefit to prevent employees turnover and also to maintain healthy and productive human resource. This paper proposes enhancements to the traditional health care system of an organization so that it provides better services with respect to users' satisfaction. The requirements identification of the system proposed and the evaluation of the new system are done using a feedback model. The new system proved to be mutually beneficial to employees and employers in terms of saving time and cost and thus it enhances productivity.*

Keywords: *e-Healthcare, agent technology, confidentiality, JADE, FIPA.*

1. Introduction

The advancement of Internet and Web 2.0 technologies with their immeasurable benefits forces the businesses to adapt to e-Services. The medical field also competes to benefit from the features provided by these modern technologies by developing e-Health care systems that incorporate a complete set of medical services delivered efficiently to their consumers at reduced management cost.

The real power of the computers is realized through distributed, open, dynamic and heterogeneous systems which can interact, span organizational boundaries and operate efficiently within rapidly changing circumstances [1]. These systems must possess some degree of *autonomy* in order to respond dynamically to the changing circumstances. The applications that are built on top of the distributed systems demand flexible and intelligent solutions. In the context of addressing these requirements, agent technologies have been developed with the purpose to provide solutions for the emerging problems and for managing the complexity that arises in this arena.

Agent-based systems have become one of the most vibrant and important areas of research and development in recent years. *Agents* can be defined to be autonomous, problem-solving, computational entities capable of efficient operation in dynamic and open environments. The agents are deployed in environments where they can interact and may cooperate with other agents on user's behalf that have possibly conflicting aims. Such environments are known as *multi-agent systems*.

An agent platform is a technological architecture providing the environment, in which the agents can actively exist and operate to achieve their goals [2]. Two organizations provide standards for agent technology: the Foundation for Intelligent Physical Agents (FIPA) and Object Management Group (OMG). The authors in [3] concluded that the agent platforms conforming to FIPA specifications are suitable for developing multi-agent systems. Table 1 presents a comparison of agent platforms with respect to their major features.

The only agent platform that is FIPA-compliant and continuously maintained upto now, is JADE. It is a software framework for multi-agent systems, written in Java and has been in development since 2001. JADE platform allows the coordination of multiple FIPA-compliant agents and enables communication among them using the standard FIPA-ACL communication language. It is well documented and free for download. JADE is licensed under Lesser General Public License (LGPL), meaning that users can use both binaries and source code of the platform without any restrictions. A light-weighted release of JADE is also available as JADE Lightweight Extensible Agent Platform (LEAP). In [4] the authors list out the salient features of JADE.

With the above mentioned features and advantages of agent technology, agent platforms, FIPA and JADE, the proposed multi-agent based health care system is implemented using JADE.

The rest of the paper is organized as follows. Section 2 describes the related work in this area and the need for this proposal. Section 3 provides a description on the existing system, the architecture of the proposed system, and a description on various agents deployed in this system and their functionalities. Section 4 describes the experimental setup and the implementation procedures. The evaluation of the proposed system is presented in Section 5. Section 6 concludes the paper and also identifies the directions in future work.

Table 1. Comparison of software agent platforms

Platform / Features	Aglets	JADE	Voyager	TACOMA	Grasshopper	SPRINGS
Organization	IBM Tokyo research	Telecom Italia Lab	Object Space	Tromoso & Cornel University	IKV++	Distributed Information Systems Group
Model	Events	Behaviours	Procedural	Procedural	Procedural	Procedural
Compliancy to agent Standard	MASIF	FIPA	---	---	FIPA	---
Programming languages used	JAVA	JAVA	JAVA	C, Perl, Unix, Tcl scripting language	JAVA	JAVA
Elements	Contexts -Agents (aglets) -Tahiti	Containers -Main container -Platforms -Agents	-Servers -Agents	-DATA -CODE -HOST -meet	-Places -Regions -Agents	-Places -Regions (RNSs) -Agents
Communication Technique	Synchronous, Asynchronous	Asynchronous	All methods	Asynchronous	Synchronous	Synchronous, Asynchronous
Messages	Yes	Yes(FIPA)	No	Yes	Yes (FIPA)	Yes
Mobility	Aglet transfer protocol	Inbuilt agent Mobility service	Java object series	Transfer Control Protocol	Dynamic Proxies	Dynamic proxies (location wise)
Available for download	IBM Public license	LGPL	Not free (evaluation version)	---	Not anymore	Yes (binaries)
GUI Based tools	Some	Yes	No	Some	Yes	No
Security	Limited	Strong	Limited secured channel	User firewall agent	Limited	Limited

2. Related work

Software agents are used to implement highly modular e-Services that are inter-operable, flexible, co-operative and autonomous. The desire for higher cost efficiency and less sub-optimal business processes drives the employment of agent technology in e-Business. E-health care system is a new paradigm that offers health care services which can be accessed from anywhere at any time. E-health care systems developed by using various technologies are found in literature. One of these is offered in M o r e n o et al. [5] and implemented using agent technology. The multi-agent system has to be intrinsically multi-threaded for control management. The authors in I a n t o v i c s [6] state that the basic intelligent property of the agents lies in the autonomy of the operation, coordination and negotiation between agents for communication that leads to efficient problem solving. Therefore, the multi-agent approach is very efficient and appropriate for handling the complexity of e-healthcare systems.

A multi agent based efficient healthcare system which provides a pervasive service via mobile devices is proposed by B y u n g-M o H a n et al. [7]. It saves human time and effort by transferring routine tasks from humans to software. The patient's medical information is managed and monitored using multiple agents that are implemented using JADE and LEAP. C h u a n-J u n S u and C h i a-Y i n g W u [8] described the design and development of a mobile multi agent based distributed information platform on top of JADE for health care monitoring. A medical diagnosis system designed by G u p t a and P u j a r i [9] is based on multi agent technology where the knowledge base, collaborative and co-operative intelligent agents reside on the multi-agent platform that provides a communicative task-sharing environment to resolve the patient's queries. To reduce the propagation time in an emergency situation, D o m n o r i et al. [10] devised an Ubimedic2 framework with multi agents. The framework improves the decision making capability of the agents and enhances the communication and collaboration among agents.

Virtual Electronic Patient Record system (VERP) is developed by C r u z-C o r r e i a et al. [11] with two major modules, i.e. a Multi-Agent system for Integration of Data (MAID) and a Visualization module (VIZ). MAID ensures communication among various hospital information systems and their immediate report retrievals and VIZ, available in the hospital intranet, presents the integrated clinical information in VERP. C h e n et al. [12] concise the evolution of the electronic based records and secured transmission of records using Lagrange interpolation formulas in mobile agents. The multi agent method has been utilized for an efficient drug management process and limited resource allocation by B a f f o et al. [13], using combinatorial auction mechanism. I t a b a s h i [14] illustrates the characteristics of the agents which increase the competence of automatic scheduling of home care services and hospital patient scheduling (Z o l l e r et al. [15]).With MAS, D a k n o u et al. [16] has developed a system for patients' treatment in an emergency department with quick data access and scheduling of patients using JADE platform.

In the study carried out by N e a l o n and M o r e n o [17], the agents are considered to be the best approach to handle different health care issues. A novel approach for integrating the healthcare enterprise with MAS has been introduced by Z h e n g et al. [18], also presented the intra structure and implantation algorithms of the agents. W a n g et al. [19] in their work used multi agents to accomplish the task of simulating an intelligent diet management for customers and the efficiency of the system is improved by using a genetic algorithm. With a cooperative agent team Z h a n g et al. [20] designed an efficient knowledge base monitoring system for chronic diseases, such as diabetes and hypertension using Case base reasoning. P a r a n j a p e and G i l l [21] modelled an agent technology for interaction between a Type 2 diabetic patient and a physician and thus demonstrated the clinical and cost efficiency of diabetic intervention.

Regarding the confidentiality and security issues, M o r e n o et al. [22] propose a multi agent based security model for a medical system which establishes a diverse level of security with a set of permissions for each security level. The user authentication is provided by signing the message with a public key. W a l c z - a k [23] derived a set of security practices which are carried out by agents, simulated in MIRRORS (Medical Information Retrieval Robots and Office Reminder System). K e f a R a b a h [24] considered ECC public key cryptosystem to be the best as it provides the same level of security like other public key cryptosystems (RSA) with a smaller key size (160 bits) and thus takes less computation time.

From the literature survey carried out as above stated, it is clear that ensuring confidentiality of patient's medical information is an important concern in any e-health care system and the contributions to address this challenge are not frequently found in literature. Therefore, in this paper a secured e-health care system is proposed, that is designed and developed encrypting patient's medical information using ECC, since it is proved to be the best public key cryptosystem.

3. The system proposed

In this paper an Agent Based Health Care System (ABHCS) is designed and developed as an enhancement to the traditional health care system in an organization. Our VIT University is considered to be the test environment for the proposed system. The various academic departments (called schools) and administrative departments within the university which are scattered geographically, are connected via 100 Mbps Ethernet LAN. Every faculty is provisioned with a laptop and the employees and students can access the intranet in any of the computer laboratories. VIT University has got a health centre which provides free health care service for the students, employees and their dependents.

3.1. Description of the existing system

The health centre at VIT University provides health care services round the clock. The people who have any ailment, visit the health centre in person and all the patient records are maintained manually in it. They are asked to enter the visiting patient details register and the administrative personnel makes an entry into the patient's medical file about the visit and gives the file to the patient. As the physicians are available 24 hours in a day, the patient can consult the available physician and undergo the prescribed medical test if any, and get the prescribed medicine from the pharmacy. The physicians note down the medical test that the patient is supposed to undergo and the medications that the patient must take, in the patient's medical file. The patient carries his/her medical file to each department, such as lab, pharmacy, etc., in the health centre and gives it back to the administrative personnel on exit. The physicians also recommend if the patient needs any consultation with other specialist doctors who are available in the health centre on certain days. Then the patient gets an appointment to meet the specialist doctors. In case the patient needs in-patient treatment, then either the specialist doctor or the physician advice his/her reception in the health centre for treatment. For in-patients, enough beds need to be allotted.

A survey is taken from the users about the existing system, concerning the services provided by the health centre at VIT University, and the need for enhancing these health care services is identified on the base of this feedback.

- Requirement 1: Some users expressed their wish to receive the medicine from the pharmacy for chronic (medicine refilling) and non-serious ailments, such as cold and cough without consulting the doctors because of their busy schedule.
- Requirement 2: Most of the people expressed their difficulty to visit the health centre in person for getting an appointment for medical tests and visits to special doctors, so they wanted these steps to be automated.
- Requirement 3: Some other users felt the need for privacy of their information.

3.2. Architecture of the proposed system

The proposed system is designed to address the above mentioned requirements. The architecture of the proposed system (ABHCS) is depicted in Fig. 1. As the proposed system is developed using agent technology, a set of software agents is deployed in the system to carry out a variety of tasks shown in Table 2.

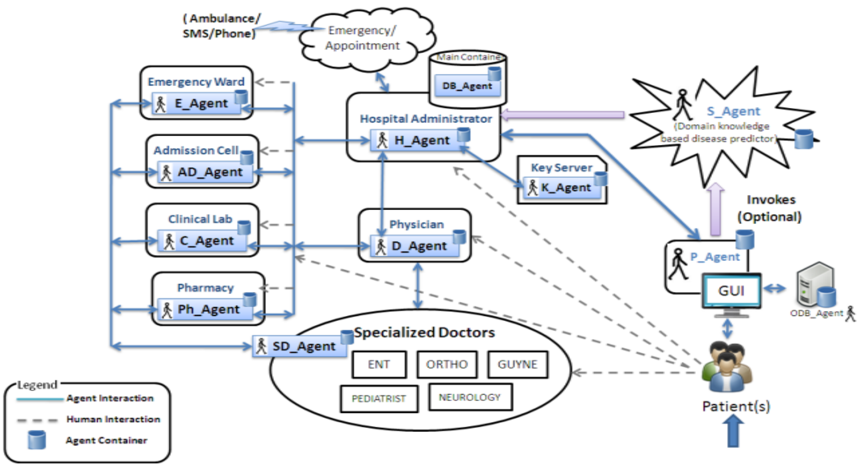

Fig. 1. Architecture of the proposed system (ABHCS)

People with expertise in specific industry are called domain experts. The knowledge they possess is called domain knowledge. In order to address Requirement 1, a Domain Knowledge DataBase (DKDB) is created with support from physicians and specialists in the health centre, that includes a list of ailments and their associated symptoms along with the suggested list of medical tests and medications to be taken for each ailment. This database is frequently updated by the physicians and specialists to include seasonal ailments and newly identified diseases.

The system proposed maintains the privacy of the patient's record by encrypting the data using Elliptic Curve Cryptography which is one of the strongest public key cryptosystem (Requirement 3). An elliptic curve represents a cloud of points that fulfil the curve equation $y^2 = x^3 + ax + b \mod p$ where x, y, a and b are within F_p and are integers modulo p. The parameters of the curve cryptosystem are p, a, b, G, n and h, where p is the prime number which defines the field F_p, in which the curve operates. All point operations are taken modulo p; a, b are the two integer coefficients that define the curve; G is the generator or base point. This is a distinct point of the curve that resembles the "start" of the curve. This is either given in a point form G or as two separate integers g_x and g_y. n represents the order of the curve generator point G. This is the number of different points on the curve which can be gained by multiplying a scalar by G. h is the cofactor of the curve. It is the quotient of the number of curve-points divided by n.

A user friendly interface for the health care system proposed is developed, via which the students/employees can perform any of the following tasks:

- Case 1**:** The users can enter their symptoms and get medicines for non-serious ailments (cold, cough, etc.) and also for their chronic diseases (diabetes, hypertension, etc.) directly from the pharmacy (called medicine refilling) in person in their leisure time without consulting the doctor (using DKDB – Requirement 1).

- Case 2: The users can enter their symptoms and get the list of medical tests to undergo using DKDB. The doctor checks the report of the medical tests and intimates the patient about the medications if the case is non-serious (Requirement 1) or advises the patient to meet a specialist if required. *The users are prevented from choosing options 1 or 2 two consecutive times in order to ensure that they will consult a doctor in person before making such attempts.*

- Case 3**:** The users can fix appointments with any of the specialists. (Requirement 2).

- Case 4: The users can make a request to view their report online. The system transmits the encrypted report (refer to Fig. 2).

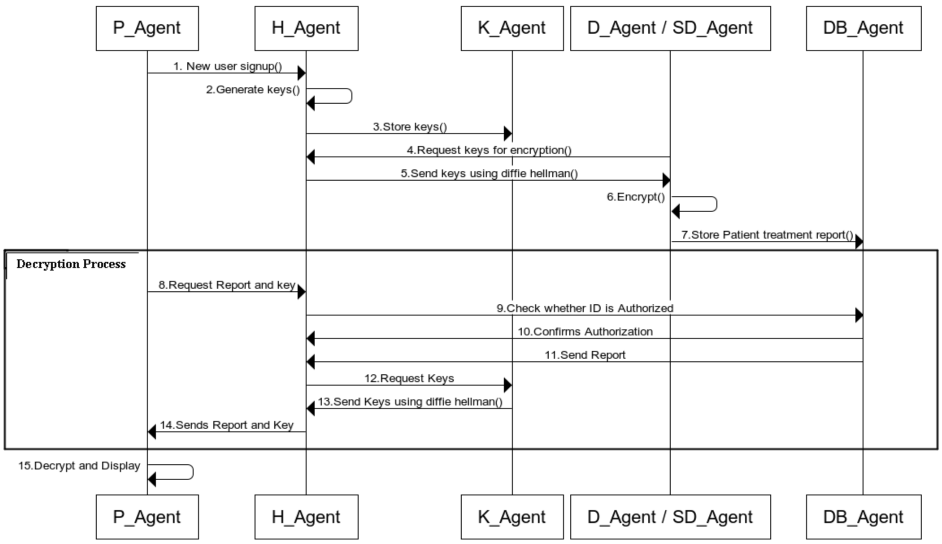

Fig. 2. Sequence diagram depicting key generation, encryption and decryption Processes

In all the above cases, the system acquires information about each consultation of the patient in the patients' treatment details table.

3.3. Description of the agents deployed in ABHCS and their functionalities

Table 2. Agents deployed in ABHCS and their functionalities

Agent Name	Functionality
P_Agent (Patient Agent)	1. Transmits user's sign up details to ODB Agent for identity verification 2. Generates<Patient Id, Password> Pair for valid users 3. Reads the user's preference (Case 1, 2, 3 or 4) and contacts the appropriate agent (S_Agent for case 1 & 2, H_Agent for case 3 & 4) to process the user's request further 4. Receives the response of those agents and displays them to the users
S_Agent (Symptom Checker Agent)	1. Receives user's preference and symptom details from P_Agent 2. Compares user's symptoms with the available data in Domain Knowledge Database and identifies the medications for case 1 or the medical tests for case 2 and informs the same to H_Agent for further processing
C_Agent (Clinical Lab Agent)	1. Receives medical test details from H_Agent 2. Gives an appointment date and time for the patient to undergo the required medical test. This information is passed to H_Agent 3. Once the patient gives the medical tests, generates the test reports and passes it on to H_Agent
H_Agent (Hospital Administrator Agent)	1. Once a patient is signed up successfully, H_Agent generates a Public/Private Key pair to ensure confidentiality of patients' medical reports using Elliptic Key Crptography (ECC) and stores it in the Key Store with the help of K_Agent 2. Handles user's preferences **Case 1:** Receives patient and medication details from S_Agent for case 1 and makes an entry into the patient's treatment details table about the patient's consultation. Also transmits the medication details to Ph_Agent and P_Agent for further processing **Case 2: a)** Receives patient and medical test details from S_Agent for case 2 and transmits it to C_Agent to fix an appointment date & time for the patient to undergo the prescribed medical tests **b)** Receives C_Agent's response for the appointment date & time and sends it back to P_Agent and also makes an entry into Patients' treatment details table about the patient's consultation **c)** Receives Medical Test Report of a patient from C_Agent, updates the patient record and forwards the report to D_Agent for further processing **Case 3:** Receives Patient id and Specialist doctor name from P_Agent, checks the Doctors Details table for availability of the required doctor and sends the appointment date and time for the patient with the particular specialist to P_Agent and DB_Agent **Case 4:** Receives Patient Id and Consultation date from P_Agent, fetches the appropriate encrypted medical report from the database and sends it to P_Agent. It also retrieves the private key of the patient from the key store and sends it using diffie – hellman algorithm 3. When D_Agent or SD_Agent request for a particular patient's key to encrypt or decrypt his/her medical report, the H_Agent response to the request by supplying the key to the D_Agent / SD_Agent 4. Creates a record in Patients' Treatment Details table about the emergency case patients and transfers the patient details to E_Agent for further processing 5. For in-patients, confirms the bed allotment by consulting AD_Agent

Table 2 (c o n t i n u e d)

Agent Name	Functionality
D_Agent / SD_Agent (Doctor Agent / Specialist Doctor Agent)	1. Doctors and Specialists make entries into the Domain Knowledge database about well-known diseases along with their symptoms, medication and medical test details via D_Agent and SD_Agent respectively 2. Encrypts patient's medical report by requesting the encryption key of that patient from H_Agent 3. Decrypts patients' medical report by getting the decryption key from H-Agent when a doctor/specialist wants to view them to diagnose the ailment
K_Agent (Key Agent)	Stores and maintains the key pair and supplies when H_agent requests for the key of a particular patient
ODB_Agent (Organization DB Agent)	Receives user's sign up information from P_Agent, checks VIT main database for identity of the user and returns the result to P_Agent
Ph_Agent (Pharmacy Agent)	1. Maintains drugs stock details in Medication Stock Details table 2. Updates Medication Issue Details table when the medications are issued to a patient
DB_Agent (Database Agent)	Creates all database tables needed for this system and maintains them according to the instructions from H_Agent, S_Agent, C_Agent and Ph_Agent
AD_Agent (Admission Cell Agent)	Receives In_Patient details from H_Agent, identifies a bed allotment for them and passes the information to H_Agent for further processing
E_Agent (Emergency Ward Agent)	Receives emergency case details from H_Agent, performs all the activities of H_Agent on priority basis to handle the emergency cases

4. Implementation

The proposed system is implemented using JADE technology. JADE 4.2.0 is the latest version of JADE which is installed as a plug in software in Net beans IDE 6.9.1 for Java. In the following section the implementation procedures of the significant agents from the former list and their main functionalities are described.

4.1. Implementation of P_Agent

The algorithm for generation of P_Agent and implementation of P_Agent functionalities are shown in Table 3.

Table 3. Algorithm for implementation of P_Agent functionalities

Algorithm	Generation of P_Agent & Implementation of P_Agent Functionalities
Input	Agent Name(P_Agent), Agent Class that describes agent behaviour
Output	AID of P_Agent and user performs an action

Procedure	Begin
	1. Create a Jade class that inherits jade.core.Agent class
	2. Override setup() method
	3. Display AID of the created agent
	4. Add an agent behavior (Simple behavior)
	validateUser();
	generate_UserName_Password_Pair()

Table 3 (c o n t i n u e d)

	5. Add another behavior for P_Agent (Handling User's Preference)
	i) displayOptions();
	ii) switch UserPreference {
	case 1|| case 2 // Domain Knowledge based
	If Mode_of_Consultation_of_ last_consultation is Case1 or Case 2
	Display "You cannot choose this option"; break;
	Else Read symptoms
	Send user's preference and symptoms to S_Agent
	Receive medication details from H_Agent for Case 1
	Receive Medical test details and appointment date & time from H_Agent for Case 2.
	Display the H_Agent response to the user; end if; break;
	case 3
	Display specialist details
	Read Specialist Name
	Send Patient_Id and Specialist name to H_Agent
	Get Appointment date & time with the specialist from H_Agent; break;
	case 4
	Read Date_Of_Consultation
	Send Patient_Id and Date_Of_Consultation to H_Agent
	Receive medical report and decryption key from H_Agent
	Decrypt the report and display it on the screen. }
	End

The following snap shots depict the agent creation process in JADE and some of the functionalities of P_Agent.

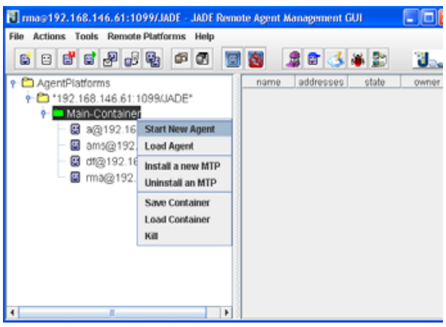

Fig. 3. JADE remote agent management GUI – start new agent

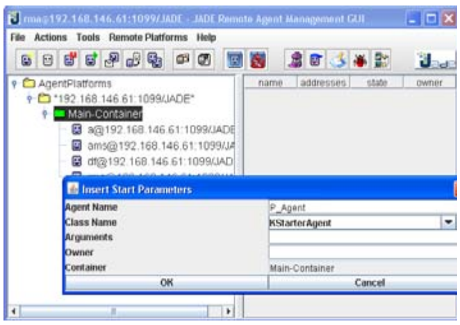

Fig. 4. JADE remote agent management GUI– agent parameters

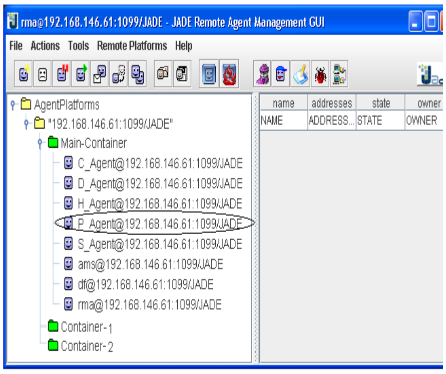

Fig. 5. JADE remote agent management GUI – P_Agent

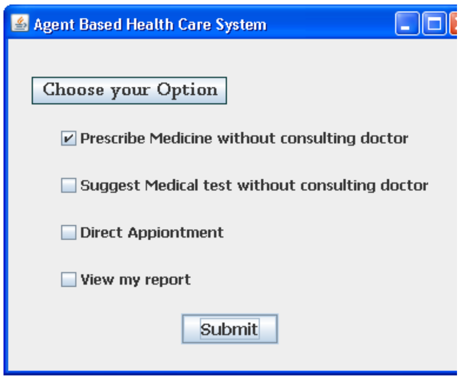

Fig. 6. Choices for ABCHS users

4.2. Implementation of S_Agent

The following algorithm (Table 4) describes S_Agent implementation procedure. The snapshots demonstrate the same.

Table 4. Algorithm for S_Agent Functionalities

Algorithm	Implementation of S_Agent functionalities
Input	User's preference and symptoms
Output	Suggestions for medication for case 1 and medical test for case 2

Procedure	1. Db_Connect() // connect to Domain_Knowlege_DB
	2. Receive_Symptom_Details_P_Agent()
	3. Compare_Symptoms_with_DKDB();
	4. Switch UserPreference {
	Case 1
	Send_Medication_Details_H_Agent(); break;
	Case 2
	Send_MedicalTest_Details_H_Agent(); break;
	}

The following snapshots demonstrate the functionalities of S_Agent.

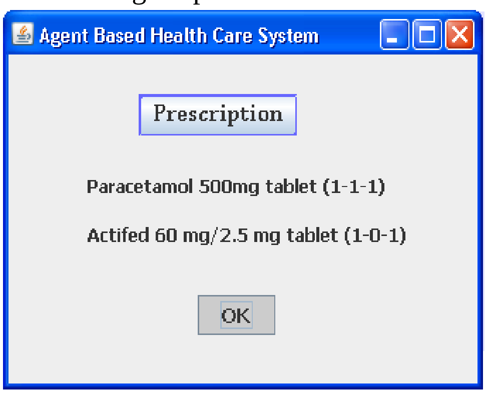

Fig. 7. Medication prescription based on domain knowledge (Case 1)

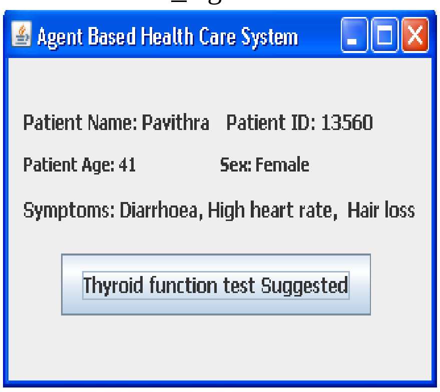

Fig. 8. Medical test prescription based on domain knowledge (Case 2)

4.3. Implementation of C_Agent

Table 5 describes an algorithm for implementation of C_Agent functionalities and the snapshots in Figs 9 and 10 demonstrate the same.

Table 5. Algorithm for implementation of C_Agent functionalities

Algorithm	Implementation of C_Agent Functionality.
Input	Medical Test Prescription
Output	Appointment date and time, Medical Test Report.
Task 1: Fixing Appointment for Medical Test	1. Db_Connect() //Medical Lab Appointment Details 2. tday = today; 3. Loop If Resource_Available() on tday Fix Appointment for tday; break; Else tday = tday + 1; end if; end loop; 4. Send Appointment Details to H_Agent
Task 2: Generation of Medical Report	1. Db_Connect() //Connect to Lab Test Details table. 2. For each test loop insert a record into the table end loop; 3. Send Report Number to H_Agent

Fig. 9. Appointment for a medical test

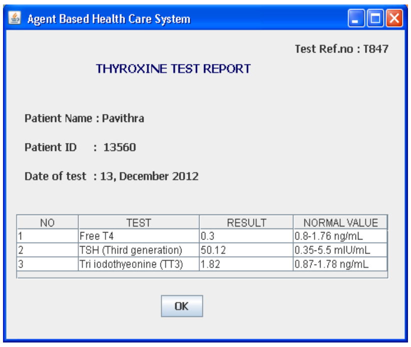

Fig. 10. Medical test report

4.4. Implementation of D_Agent / SD_Agent

The following snapshots show some of the functionalities of SD_Agent.

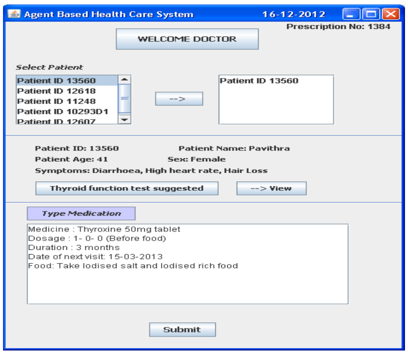

Fig. 11. Prescription window for doctors

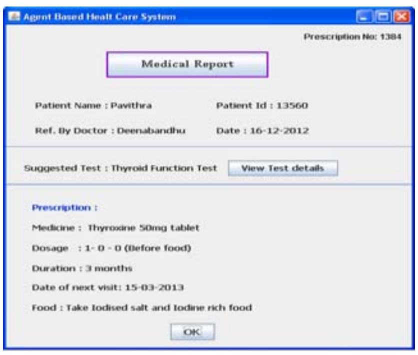

Fig. 12. Patient's medical report

4.5. Implementation of H_Agent

Table 6 shows an algorithm describing the important functionalities of H_Agent and the snapshots in Figs 13, 14 and 15 demonstrate the same.

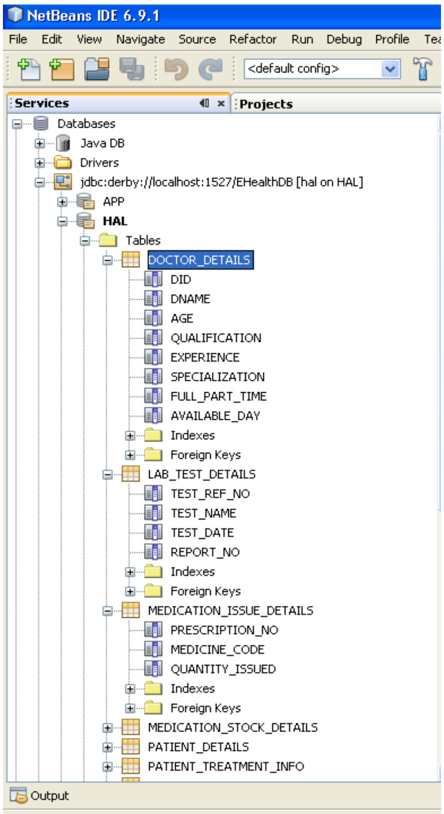

Fig. 13. ABHCS database

Fig. 14. Read the medical speciality

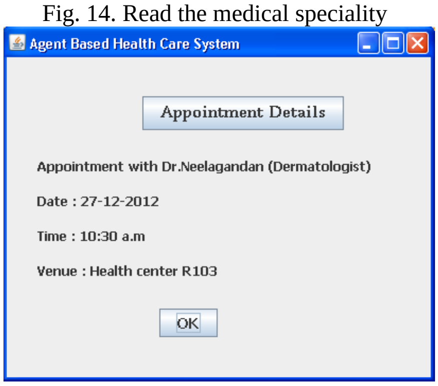

Fig. 15. Appointment with a specialist

Table. 6. An algorithm describing H_Agent implementation procedure

Algorithm	Implementation of H_Agent Functionalities

Task 1: Generation of Key to ensure Confidentiality	*Input:* a prime number *p*
	Output:<Private Key, Public Key> Pair
	Procedure:
	Begin
	Generate_Coefficients() // Generation of *a*, *b*
	Generate_BasePoint() // Generation of *G*
	Find_Order_Of_G() // Find *n*
	Find_CoFactor_Of_Curve //Find *h*
	Generate_Private_Key() // Private key d_A such that $0 < d_A < n$
	Genertae_Public_Key() //Public key $Q_A = d_A . G$
	End;

Task 2: Handling User's Preferences	*Input:* Patient Id, Consultation date, Medication details, Medical test details, Specialist Name, User's Preference. *Output:* Prescription Number, Appointment Date and Time, Medical Report

Table. 6 (c o n t i n u e d)

	Procedure:
	Begin
	Switch (Case){
	Case 1:
	Db_Connect() //Connect to Patient Treatment Details
	Receive_PDetails()_S_Agent // Receives Patient Details
	Receive_MD()_S_Agent // Receives Medication Details
	Generate_Prescription() // Prescription Number
	Store_Prescription() // Stores Prescription number in table.
	Send_Prescription_P_Agent() //Agent sends message
	Send_Prescription_Ph_Agent() //Agent Sends message
	Insert() // create an entry for patient's consultation
	Break;
	Case 2:
	Db_Connect() //Connect to Patient Treatment Details
	Receive_PDetails_S_Agent() // Receives Patient Details
	Receive_MTD_S_Agent() // Receives Medical Test Details
	Generate_MTRN() // Medical Test Reference Number(MTRN)
	Store_MTRN() //Store MTRN in table
	Send_MTD_C_Agent()// Sends Medical Test Details
	Receive_App_Details_C_Agent() //Appointment Details
	Send_MTD_P_Agent() // Agent Sends Message
	Send_App_Details_P_Agent() // Agent Sends Message
	Insert() // create an entry for patient's consultation
	Break;
	Case 3:
	Db_Connect() //Connect to Patient Treatment Details
	Db_connect() // Special Doctor Appointment Details Table
	Receive_PDetails_P_Agent() // Receives Patient Details
	Receive_SName_P_Agent() // Receives Specialist Name
	Fix_Appointment() // by checking the total no of patients.
	Store_App_Details() // in Special Doctor Appointment Details
	Send_App_Details_P_Agent() // Agent Sends Message
	Send_App_Details_SD_Agent() //Agent Sends Message.
	Insert() // create an entry for patient's consultation
	Break;
	Case 4:
	Db_Connect() // connect to Patient Treatment Details,
	//Lab Test Details & Medication Issue Details
	Receive_PID_CD_P_Agent()// Patient Id, Consultation Date
	Generate_MR() //Generates Medical Report of the Patient
	Request_DKey_K_Agent() // Requests Decryption key
	Receive_DKey_K_Agent() // Agent Receives Message
	Send_MR_P_Agent() // Agent Sends Message
	Send_DKey_P_Agent() //Agent Sends Message } End;

5. Evaluation of the proposed system

As there are no standard metrics to evaluate multi-agent based systems, a feedback model is used to evaluate the efficiency of ABHCS. The feedback is taken from fifty patients/users of the system, out of which twenty are students, fifteen are teaching staff and other fifteen are a non-teaching staff. The feedback questionnaire is based on 5 points of scale satisfaction [Level of Satisfaction: 1 – Poor, 2 – Average, 3 – Good, 4 – Very Good, 5 – Excellent]. The result of evaluation is shown in Fig. 16. The questionnaire includes the following questions:

1. Does the system provide user-friendly interfaces?
2. Are you satisfied with getting medications re-filled without seeing a doctor?
3. Are you satisfied with doing vital medical tests without consulting a doctor?
4. Do you really save time by using this system?
5. Do you feel convenient about the options provided?
6. Is the system efficient in terms of the response time?
7. In emergency cases, does the system respond more quickly than in other cases?
8. Does the process help in fixing an appointment with specialists and realizing medical tests?

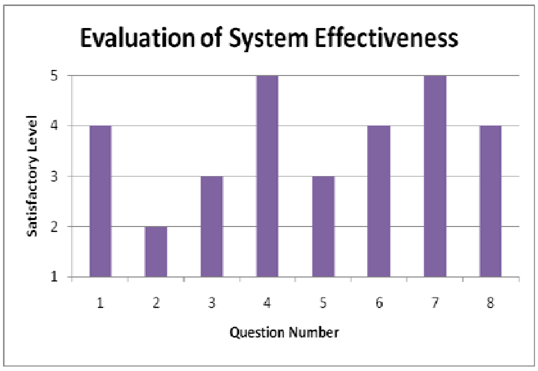

Fig. 16. Evaluation of the system efficiency

The feedback results show that the efficiency of ABHCS is rather high.

6. Performance analysis

The performance of a system is evaluated based on the following parameters:

- Average Waiting Time – the waiting time represents the amount of time the user waits before getting the service.
- Average Service Time – the service time represents the amount of time the user spends in getting the service.
- Average Response Time – the response time presents the sum of the waiting time and the service time for a user.

When multiple users access ABHCS simultaneously, then the load at H_Agent increases drastically. H_Agent manages this situation by immediately deploying some helper agents to handle the load. The average waiting time and response time of the traditional health care system and ABHCS for a load of 23 users is compared and shown through a graph depicted in Figs 17 and 18.

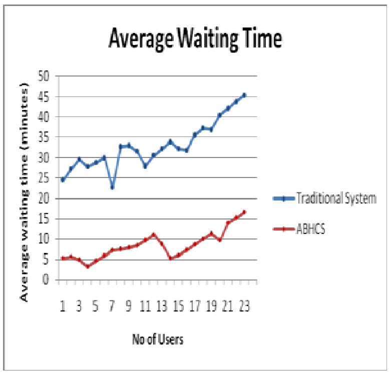

Fig. 17. Average waiting time

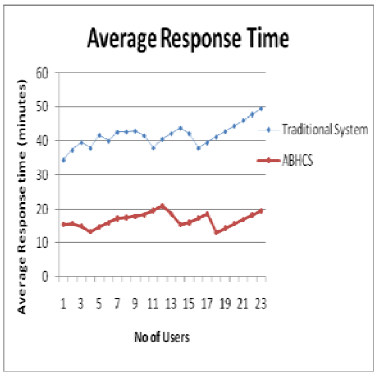

Fig. 18. Average response time

7. Conclusion and future work

This paper has made an attempt to envisage the performance of an agent based secured health care system that provides confidentiality of the users' medical information along with a support for automation of the traditional health care system activities. The performance of the system can be enhanced further by deploying the patient specific mobile agents and by cloning the H_Agent (Hospital Administrator Agent) to speed up the administrative activities. Key Agent can be enhanced to check the authenticity of the agent storing and retrieving the key from the key store. The agents deployed in ABHCS are assumed to be reliable. In the future, certain measures can be taken to identify the possible security attacks on the agents and methodologies must be devised to prevent, detect and recover after such attacks.

Acknowledgements. The authors would like to express their thanks to the reviewer for his valuable suggestions in refining this paper.

References

1. Luck, M., P. McBurney, C. Preist. Agent Technology: Enabling Next Generation Computing. A Roadmap for Agent Based Computing, AgentLink, 2012.
2. Leszczyna, R. Evaluation of Agent Platforms – JRC Scientific and Technical Reports. Version 2.0, 2008.
3. Bellifemine, F. L., G. Caire, D. Greenwood. Developing Multi-Agent Systems with Jade. John Wiley Sons, Ltd., England, 2007.
4. Bellifemine, F., A. Poggi, G. Rimassa. JADE – A FIPA-Compliant Agent Framework. – In: Proceedings of 4th Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, 1999, 97-108.
5. Moreno, A., A. Valls, D. Isern, D. Sanchez. Applying Agent Technology to Healthcare: The GruSMA Experience. – In: Intelligent Agents in Healthcare, IEEE, 2006, 63-67.
6. Iantovics., B. L. Agent-Based Medical Diagnosis Systems. – Computing and Informatics, Vol. **27**, 2008, 593-625.
7. Han, Byung-Mo, Seung-Jae Song, Kyu Min Lee, Kyung-Soo Jang, Dong-Ryeol Shin. Multi-Agent System Based Efficient Healthcare Service. – In: 8th International Conference on Advanced Communication Technology, Vol. **1**, 2006, 47-51.

8. S u, C h u a n - J u n, C h i a - Y i n g W u. JADE Implemented Mobile Multi Agent Based, Distributed Information Platform for Pervasive Health Care Monitoring. – Applied Soft Computing, Vol. **11**, 2011, 315-325.
9. G u p t a, S., S. P u j a r i. A Multi-Agent System (MAS) Based Scheme for Health Care and Medical Diagnosis System. – In: International Conference on Intelligent Agent and Multi Agent Systems, Chennai, 2009, 1-3.
10. D o m n o r i, E., G. C a b r i, L. L e o n a r d i. Designing and Implementing Intelligent Agents for e-Health. – In: International Conference on Emerging Intelligent Data and Web Technologies (EIDWT'11), 2011, 78-85.
11. C r u z-C o r r e i a, R., P. V i e i r a-M a r q u e s, P. C o s t a, A. F e r r e i r a, E. O l i v e i r a-P a l h a r e s, F. A r a ú j o, A. C o s t a-P e r e i r a. Integration of Hospital Data Using Agent Technologies – A Case Study. – AI Communications – Agents Applied in Health Care, Vol. **18**, 2005, 191-200.
12. C h e n, T.-L., T.-L. C h e n g, F. Y. L i n. A Study on Agent-Based Secure Scheme for Electronic Medical Record System. – Journal of Medical Systems, Vol. **36**, 2012, 1345-1357.
13. B a f f o, I., G. S t e c c a, T. K a i h a r a. A Multi Agent System Approach for Hospital's Drugs Management Using Combinatorial Auctions. – In: 8th IEEE Conference on Industrial Informatics (INDIN'10), 2010, 945-949.
14. I t a b a s h i, G., M. C h i b a, K. T a k a h a s h i, Y. K a t o. A Support System for Home Care Service Based on Multi-agent System. – In: 5th International Conference on Information, Communications and Signal Processing, 2005, 1052-1056.
15. Z ö l l e r, A., L. B r a u b a c h, A. P o k a h r, F. R o t h l a u f, T. P a u l u s s e n, W. L a m e r s d o r f, A. H e i n z l. Evaluation of a Multi-Agent System for Hospital Patient Scheduling. – In: International Transactions on System Science and Applications (ITSSA) , 2006, No 1, 375-380.
16. D a k n o u, A., H. Z g a y a, S l i m, H. H u b e r t. Toward a Multi-Agent for the Care of Patients at the Emergency Department. – In: Proceedings of 10th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems, USA, 2008, 264-269.
17. N e a l o n, J., A. M o r e n o. Agent-Based Applications in Health Care–Applications of Software Agent Technology in the Health Care Domain. Whitestein Series in Software Agent Technologies, Birkhauser Verlag, 2003.
18. Z h e n g, S., W. L i, X. P e n g, H. Z h a o, C. Z h a n g. Research on Integrating the Healthcare Enterprise Based on Multi-Agent. – In: International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE'10), Vol. **1**, 2010, 493-496.
19. W a n g, F., Y. Y u a n, Y. P a n, B. H u. Study on the Principles of the Intelligent Diet Arrangement System Based on Multi-Agent. – In: 2nd International Symposium on Intelligent Information Technology Application, Vol. **1**, 2008, 264-268.
20. Z h a n g, Y., M. L e e, T. M. G a t t o n. Agent Based Web Healthcare Systems for Real Time Chronic Disease. – In: Proceedings of the 2009 Congress on Services – I, USA, 2009, 14-21.
21. P a r a n j a p e, R., S. G i l l. Agent Based Simulation of Healthcare for Type II Diabetes. – In: 2nd International Conference on Advances in System Simulation, 2010, 22-27.
22. M o r e n o, A., D. S a n c h e z, D. I s e r n. Security Measures in a Medical Multi-Agent System. – Research Group on AI, Multi-Agent Systems Group (GruSMA), University RoviraVirgili (URV).
23. W a l c z a k, S. Information Security for Agent-Based WWW Medical Information Retrieval. – Logistics Information Management, Vol. **15**, 2002, 393-399.
24. K e f a, R. Using Elliptic Curve Cryptography to Secure Online Data & Content. – A Global Open Versity Technical Research Publication, Vol. **1**, 2009.