UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

MULTI AGENT MONITORING AND SUPPORT SYSTEM FOR TUBERCULOSIS TREATMENT

LILY C. SIELE

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENTS OF THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

August 2013

DECLARATION

This project as declared in this report as presented in this report is my own original work and has not been presented anywhere for the purpose of an academic award.

Signature: ___________________________________ Date ______/_______/_________

Lily C. Siele

P58/64583/2011

This project has been submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer science of the University of Nairobi with my approval as the University supervisor.

Signature: ___________________________________ Date: ______/_______/________

Dr. Elisha T. Opiyo Omulo

School of Computing and Informatics

University of Nairobi

ABSTRACT

The prevalence of tuberculosis (TB) has economic and social effect not only to the patient and patient's family but also to the country at large .Tuberculosis is an important and growing problem in many developing countries. New strategies have been developed to combat the disease but require strictly followed treatment regimens and close monitoring of patients' bacteriology results. The new strategies require a long-term relationship with the patient, accurate and accessible records of each patient's history and methods to track his/her progress. This can be achieved by closely monitoring the health status of the patient from when the disease is discovered till when the patient is pronounced cured. An agent based approach has been widely used for monitoring and management of applications in health care setup.

In this study, agent based software engineering approach is used to design and implement agent based software for enrolling patients, tracking follow up patterns and Pattern of sputum results for TB patients. The data and reports can be linked together for decision making. The model consists of a team of intelligent agents that capture, work collectively to analyze it and then display the health status of the patient to the user.

ACKNOWLEDGEMENTS

I wish to express my deep gratitude first and foremost to my God, the creator of heaven and earth who gave me the insight, ability and strength to be able to conceptualized and carry out this study.

Secondly, I would like to thank my supervisor, Dr. Opiyo and the entire panelist for their advice and support during this study. In particular they helped me moderate my ideas and ensured that the study was on track and met the required standards.

To my colleagues, Department of mathematics and computer science, University of Eldoret for their tirelessly encouragement and support throughout the study.

Lastly, much gratitude and appreciation goes to my entire family for their continued support.

TABLE OF CONTENTS

DECLARATION i
ABSTRACT ii
ACKNOWLEDGEMENTS iii
TABLE OF CONTENTS iv
TABLE OF FIGURES vii
LIST OF TABLE viii
LIST OF ABBREVIATION ix
CHAPTER ONE: INTRODUCTION 1
1.1 Background of the Study 1
1.2 Problem Statement 2
1.3 Purpose of the Study 3
1.4 Research Objectives 3
1.5 Research Questions: 4
1.6 Significance of the study 4
CHAPTER TWO: LITERATURE REVIEW 7
2.0 Introduction 7
2.1. The Tuberculosis concept 7
2.1.1 Basic TB Treatment Principles 7
2.1.2 Challenges facing TB treatment 8
2.2 TB Healthcare structure in Kenya 9
2.2 Software Agent Technology 12
2.2.1 Software Agent 12
2.2.2 Agent Architectures 14
2.2.3 Agent Languages 17
2.3 JADE Development Environment 18
2.4 Agent concepts in E- Health 19
2.4.1 Use of agents in health care domain 19
2.5 Related Work 20
2.6 Proposed architecture 22
CHAPTER 3: METHODOLOGY 23

3.0 Overview 23
3.1 Research Design 23
3.1.1 Problem design 23
3.1.2 Use of MAS Methodology 23
3.1.3 Skills and Tool required 24
3.2 Prototype implementation and testing 25
3.3 Data sources and Collection Methods 25
3.4 Data analysis 25
Chapter 4: SYSTEM ANALYSIS AND DESIGN 26
4.0 Overview 26
4.1 Early requirement Analysis 26
4.1.1 Initial System Description 26
4.1.2 Actors 27
4.1.3 Goals of the system 27
4.1.4 Use case scenarios 28
4.2 Late Requirement Analysis 30
4.3 Architectural Design 30
4.3.1 Architectural overview 30
4.3.2 Agents in the system 31
4.2.3 Agent Descriptors 33
4.2.4 System Overview Diagram 34
4.4 Detailed Design 34
4.4.1 Agent capabilities 34
4.5 Data 36
4.5.1 Input Data 36
4.5.2 Output Data 36
4.6 Verification and Validation 37
CHAPTER 5 : SYSTEM IMPLEMENTATION AND TESTING 39
5.1 Overview 39
5.2 Implementation of the system 39
5.2 Testing, Result and Discussion 39

5.2.1: Patient Registration 39
5.2.2: Observation Entry 40
5.2.3: The communication agent 41
5.2.4 Reporting Agent 43
5.3 System testing for user acceptability 43
CHAPTER 6: CONCLUSION AND RECOMMENDATION 47
6.1 Conclusion 47
6.2 Limitations 48
6.3 Recommendation 48
REFERENCES 49
Appendices 53
Appendix I: Sample code 53
Appendix 2: Test case data (Patient Registration) 71
Appendix 3: Observation Test Data 72
Appendix 4: Questionnaire 74
Appendix II: Gantt Charts 77

TABLE OF FIGURES

Figure 2-1: Health care structure source (NHSSP-II., 2005-2010) 11
Figure 2-2 A Generic model of a basic agent 13
Figure 2-3: BDI architecture 15
Figure 2-4: Horizontal Layering 16
Figure 2-5: Vertical Layering 17
Figure 2-6 : Agent management reference model.. 18
Figure 2-7: A proposed architecture 22
Figure 4-1: The stakeholders of the system 27
Figure 4-2: TB Patient Goal diagram... 28
Figure 4-3: Registration Use case 29
Figure 4-4: Patient diagnosis use case 29
Figure 4-5 Mean end analysis for TB patient 30
Figure 4-6 Sub- Actor decomposition... 31
Figure 4-2 System Overview Diagram 34
Figure 4-7: Capability diagram 35
Figure 4-7: Sequence diagram 36
Figure 5-1: Patient Registration 40
Figure 5-2 Patient Observation Form 40
Figure 5-3 Showing notification Information 41
Figure 5-4: Agents in the Environment 42
Figure 5-5 Message passing... 42
Figure 22 Sample Report generated by reporting Agent 43
Figure 5-3: Inventory of system.. 44
Figure 5-4 Acceptable performance Quality... 45
Figure 5-5 Functionality of the system 46
Figure 5-6: JADE Agent Management GUI 46

LIST OF TABLE

Table 2-1 Provision of TB treatment services in 2009 11
Table 4-1Agent Types and their Capabilities 33
Table 5-1 User acceptance 44

LIST OF ABBREVIATION

DOT	Directly observed Treatment
DRS	Drug Resistance Survey
HMIS	Health management information systems
MDR-TB	Multi-Drug Resistance Tuberculosis
MAS	Multi Agent System
MOH	Ministry of Health
NTP	National Tuberculosis Programme
OOC	Out of Control
TB	Tuberculosis
WHO	World Health Organisation
XDR	Extra pulmonary Tuberculosis

CHAPTER ONE: INTRODUCTION

1.1 Background of the Study

The development of the Kenya health system in relation to TB control dates back to 1980. The NLTP was established in 1980, with specialized staff at national, provincial and district level .The NLTP principal objective is to reduce Tuberculosis incidence, to strengthen and sustain accessible, quality assured bio-safe TB bacteriology for early diagnosis, monitoring, surveillance and management of tuberculosis and other lung diseases lung disease and to engage and network all health care providers and stakeholders for provision of standardized quality lung health care. The NLTP is mandated with the responsibility to ensure increase in new case detection, Universal access to quality assured TB bacteriology, Adoption and use of appropriate new technologies in TB diagnosis and improved reporting from all stakeholders. It is currently managing over 2,318 health units offering the TB services. This institution are of different categories (Dispensaries, Health centre, public and private hospitals).This project focuses on the monitoring of TB patient by ensuring their data is kept and status of current patients is known at any one time and the usage of the collected data for reporting.

This study is intended to utilize Multi agents capabilities to solve the TB programme problems whereby more than one person is required to provide the services to the patient and using the communication capabilities to centralized all the data coming from all the centers' in one place. This is important information to NLTP because an early detection of disease, follow up of cases and easy identification of relapse would allow the TB facility to ensure the patient care is known and therefore reduce the defaulters and development of the multi drug TB. The government agencies and researcher would readily access available data hence improve efficiency in terms of curbing disease and finding new clinical ways of dealing with the disease. It has been desirable to accomplish the TB programme process for example identifying a relapse case easily in the shortest time possible but without a concrete monitoring system this can continually remain an illusion. One of components that would hasten this process is to be able to capture all patients' activities and compare with the laid down guidelines hence anomaly can be identified immediately which is the main focus of this study.

1.2 Problem Statement

One of the greatest challenges in today's TB control and management programme is closing the gap between reporting evidence and medical practice. Keeping up with information coming from the entire TB centre has never been easy. Managing information for adequate TB program support usually requires a challenging and complex integration of systems with separate modules developed and managed in separate sectors. According to (WHO, 2009), Monitoring and Evaluation system is not linked to the Health Management Information System (HMIS). The NTP collects data from services delivery points making data analysis and use at point of collection rarely done. Health information required for national and international use is not adequate and often, is not timely for effective and efficient management and decision making. For example, In 2010, a total of 112 MDR TB cases were identified and notified to the WHO, by the end of 2010, the country had cumulatively initiated 180 patients on treatment with 70 of them being initiated within the year and the actual burden is not well understood till planned Drug Resistance Survey (DRS) and TB prevalence survey to be conducted in 2012, again Out of control (OOC) or defaulters is not known and cannot be efficiently approximated. The Failures cannot be accurately determined whether it is due to relapse or exogenous re-infection with a new strain of Mycobacterium tuberculosis complex because the record does not exists or cannot be trace (Anyama et al, 2007). The limited parameters available to the NTP about the TB patient-physician-clinic can be related together and have a system that can efficiently and reliably manage all the parameters.

The intention is to design and implement monitoring and support system making the capture of data faster and more reliable and making it easier to access and analyze individual patient data. A central data bank will make it possible to access data rapidly in order to facilitate planning. This research will make use of intelligent agents which provides features like autonomy and social activity which are very important. Intelligent agents are software or hardware entities that perform a set of tasks on behalf of user with some degree of autonomy (Barley & Kasabov, 2005).The TB clinics are distributed and autonomous to a certain extend, since each TB clinics treats there own patient depending on the TB Guidelines. The social and cooperative aspects is realized when patient transfer out, therefore proper communication between clinics and when extra services are required e.g. the smear tests that some clinics do not have. TB diagnosis involves a wide variety of physician and nurses of different specialties. The particular focus is on tracking patients from initial diagnosis to initiation of effective treatment and then monitoring them for treatment breaks or loss to follow-up and in the process the data storage base can be used for reporting that can be used for future policy, NTP strategies and decisions. It is therefore crucial that this study be conducted so as to get a solution to the problem.

1.3 Purpose of the Study

The purpose of this study was to identify and analyze the components of TB programme and study the associated determinant factors affecting the TB patient. The information was then utilized to develop a monitoring and support prototype that can be used to manage the TB patients and ease the process of reporting. The data captured and reports generated by the system can be used as follows:

- The Treatment management: The case treatment, Monitoring, follow-up, tracks patient transferring in and out, and provides data for treatment adherence and patient contacts evaluation
- Surveillance and information management: It will maps TB and MDR/XDR cases, previous treatment history, treatment results, providing surveillance reports and updated information with ready access online at central and sub-system levels.
- Clinical and operation research: It will provide easy and efficient method for analyzing collected data and exporting data to other programs.
- A storage base from which reports can be produced that can have an impact on future policy and NTP strategies, and decisions

1.4 Research Objectives

i) To identify and analyze the components of TB programme and study the associated determinant factors affecting monitoring and support case management.

ii) To design a multi agent monitoring and support system.

iii) To implement a TB monitoring and support system prototype.

1.5 Research Questions:

- What are the best design options for TB Decision support application as alternative to or extension of existing?
- Assess the potential future impact of the system
- Assess the possible future effect of the system

1.6 Significance of the study

The study will aid the Ministry of Health, Government agencies and Division of National Leprosy and Tuberculosis to have information that is readily available and can be used for decision making. The findings of this study are expected to be significant in offering better management of TB, reduce the time taken and cost incurred in finding the yearly prevalence of the disease which is currently hampered by the non existence of a proper recording and reporting system. The outcome of the study is expected to help the TB clinic staff and the public health counterpart in easily finding the defaulters and non adherence to the regimen and the ultimate outcome will be better health care, reduce new incidence and a proper reporting system.

The data and reports from the system may be use for the following:

- The Treatment management: The case treatment, Monitoring, follow-up, tracks patient transferring in and out, and provides data for treatment adherence and patient contacts evaluation
- Surveillance and information management: It will maps TB and MDR/XDR cases, previous treatment history, treatment results, providing surveillance reports and updated information with ready access online at central and sub-system levels.
- Clinical and operation research: It will provide easy and efficient method for analyzing collected data and exporting data to other programs.
- A storage base from which reports can be produced that can have an impact on future policy and NTP strategies, and decisions.

Academically the study will contribute to research methodology that other scholars and researcher can adopt for future research. It will also form a basis on which others can develop their studies.

1.7 Limitations of the study

- The research depends on the reliability of the data provided by the TB specialists on TB management.
- Defaulters may sometimes be attributed to unforeseen events or be governed by factors that may be difficult or impossible to see in the attributes of the care taker (i.e. Unavailability of finance, existence of communication method (phone, e-mail). The study is not able to capture those aspects.
- The study will rely on the co-operation of the respondents who are mainly TB clinic staff. The researcher may not have control of the attitudes of the respondents which may affect the validity of the responses. This is due to the fact that respondents may at times give acceptable, but not honest answers.
- The information from the available literature might not be authenticated though the research will incorporate as many resources as possible.

1.8 Assumption of the study

The researcher will base the study on the following assumptions:

- All the respondents will be co-operative and that they will give reliable and honest responses.
- The literature used is authentic.
- Health institutions have computers, laptops or mobile phones
- The data collected from TB facility is consistent and hence suitable for the study.
- The TB facility will make use of the results obtained in this study.

1.9 Organization of the study

The study will be organized into five chapters.

Chapter one, which is the introduction will place the context under the following subtopics; background of the study, statement of the problem, purpose of the study, objectives of the study, research questions, significance of the study, limitations of the study and the assumptions of the study.

Chapter two will review the related literature. It will subdivide into various subheadings. Chapter three will comprise of research methodology which will also be divide into various subheadings.

Chapter four will consist of presentation and analysis of data collected, research findings and discussion of the research findings. Chapter five will provide the summary of the findings, conclusions, recommendations and suggestions for further research.

CHAPTER TWO: LITERATURE REVIEW

2.0 Introduction

Agent Technology provides software designers, engineers and developers with a way of structuring an application around autonomous, communicative elements, and lead to the construction of software tools and infrastructure to support the design metaphor (Luck et al., 2004). My research focuses on analyzing, designing and implementing agent-based systems for the Tuberculosis monitoring and support. Before we begin discussing an agent system in an application context, we need to understand the context. In this perspective, I will briefly describe some important properties of the tuberculosis concepts and programme

2.1. The Tuberculosis concept

Tuberculosis (TB) is a chronic infectious disease that kills over 2 million people per year in developing world and therefore is a global health concern. Nearly one-third of the global population is infected with Mycobacterium tuberculosis and at risk of developing the disease (Dye et al., 1999). More than 90% of global TB cases and deaths occur in the developing world, where 75% of cases are in the most economically productive age group (Ahlburg and Initiative, 2000). Kenya is rank fifteen among the world's 22 high-burden tuberculosis countries (WHO, 2009). Patient compliance is a key factor in treatment success. In many countries, a significant proportion of patients stop treatment before completion, for various reasons (WHO, 1997). If not diagnosed, monitor and treated promptly, tuberculosis may be spread via an airborne route to family and community members. Untreated, someone with active tuberculosis will infect an estimated 10 to 15 people per year (USAID).

2.1.1 Basic TB Treatment Principles

The critical responsibility for assuring the cure of every active TB case lies with the local health department. TB disease must be treated with a regimen containing at least two drugs to which the organism is susceptible in order to prevent the development of drug resistance. Each person treated for TB disease must be individually managed based on clinical and bacteriological response to therapy. The duration of anti-TB therapy depends on the drugs utilized, susceptibility test results, and response to treatment.

Treatment of active TB disease always requires the use of multiple antibacterial agents [1]. The treatment of TB disease can be divided into two parts.

- Intensive phase typically uses at least four drugs and is designed to drastically reduce the total body burden of TB. [1,9-11]
- Continuation phase is generally after 2 months of intensive treatment, the continuation phase is designed to eliminate the 'persisters'. These are organisms that survive the initial phase of treatment.

For each person with confirmed active TB disease, a specific client centered, case management and treatment plan emphasizing adherence to therapy, should be developed that includes types of drugs to be prescribed, anticipated changes to and duration of therapy, methods of assessing and ensuring adherence (including directly observed therapy (DOT), methods of monitoring for adverse drug reactions, and necessary evaluations with appropriate time frames. The person adherence to prescribed anti-TB therapy and medical follow-up is emphasized.

2.1.2 Challenges facing TB treatment

The required facts that need to be considered when doing follow up on TB treatment include:

- Drugs prescribed: Each person treated for TB disease must be individually managed based on clinical and bacteriological response to therapy. The duration of anti-TB therapy depends on the drugs utilized, susceptibility test results, and response to treatment.
- Interrupted or incomplete treatment: When a client has had interrupted or incomplete treatment for TB disease, the clinician must decide the appropriate duration of a new regimen
- Relapse (Failure): Treatment failure should be suspected in clients whose cultures do not convert to negative after two months of therapy and/or who have clinical deterioration due to TB disease or worsening of the chest x-ray due to TB disease.

- Recurrence: Even after effective short-course chemotherapy for active tuberculosis (TB), some patients experience a recurrent TB episode.
- Potential drug interaction: Anti-TB medications may interact with prescription and over-the counter medications. Because of this, clinicians should closely monitor persons with TB disease for potential drug interactions for example Ethanol and illicit drugs interfere with the metabolism of rifampin (TB first line drug) which may cause a lowering of the therapeutic serum levels of the TB medications, which can lead to treatment failure as well as possible resistance.
- Drug reaction: Anti-TB medications can cause a variety of adverse reactions. All clients undergoing treatment for TB disease should be closely assessed, at least monthly, to identify and address potential adverse reactions.

A solution to these challenges is an information system to track all patients with the disease, keep records of critical data such as laboratory tests and medication, and provide continual updates about treatment status. Kizito describes the various barriers that can prevent a patient from returning for follow-up in a poor settlement areas like Kibera in kenya including frequent need for migration due to poverty and social disruption, and they point out that responsibility must be placed on the health system to find missing patients and provide treatment. Failure to do so TB is often a life-threatening situation (Kizito et al., 2011).

An essential component of the information system is a master patient list, regularly updated and used to find missing or failing patients. They emphasize that the list should include patient unique identifier and should be computerized(Fraser et al., 2007).

2.2 TB Healthcare structure in Kenya

The TB diagnosis and treatment are currently available at all level of health care that is from Hospitals to Community Health Centres. TB diagnosis in these centers is based on presenting signs and symptoms, the results of a sputum test and sometimes X-rays results. If required, e.g. in case of presumed recurrence, culture examination is performed at higher health facilities Tuberculosis health care system depicts at least one of the following characteristics:

i) Distributed System

The tuberculosis health care system is considered to be distributed, since it possesses the selected characteristics given by (Enslow, 1978) that is the computational resources are dynamically assigned to specific tasks. The computational resources in the tuberculosis health care system refer to both the medical resources (doctors, nurses, medical devices, medicine) and the patients and These resources are physically distributed with the help of a communication network.

ii) Cooperative System

By cooperation means a type of activity existing among two or more elements of the system when they are engaged in a mutually beneficial exchange (Brooks, 1991).' The Tuberculosis health care mainly includes two parts: primary care and home care, which are provided by hospital and community. The two levels must be able to cooperate to ensure the patient health care is of high quality.

iii) A Social System

The Tuberculosis health care is considered as one kind of social practice. According to (Chaiklin et al., 1999), 'social practice is defined as structured human traditions for interaction around specific tasks and goals'. By social system, I mean that the Tuberculosis society is grouped into structures that are associated with different functions and goals.

iv) A Complex System

The distribution of resources, cooperation and sociality lead to the complexity of the TB health care system. (Flood and Jackson, 1991) summarized a list of characteristics of complex systems which include large number of elements.This is depicted by the hierarchy of TB facilities composes of departments which interact with each other and form the whole system i.e. from a National hospital to a dispensary hospital and Sub-systems pursue their individual goals. The TB health care system is not under one authority but several.

In Kenya, the TB health care system providers are structured in a four-level pyramidal hierarchy see figure 1 below. In this structure, healthcare centers are organized as

- Level 1- The community level is the foundation of the service delivery priorities. Important gains can be reached to reverse the downward trend in health status at the interface between the health services and the community.

- Levels 2 and 3 – This level includes the dispensaries, health centre, and maternity/nursing homes. It handles activities related to promotion, preventive care and some of curative care.
- Levels 4-6 – It includes primary, secondary and tertiary hospitals. It is responsible for curative and rehabilitative activities of their service delivery package. Also address to a limited extent preventive/promotive care.

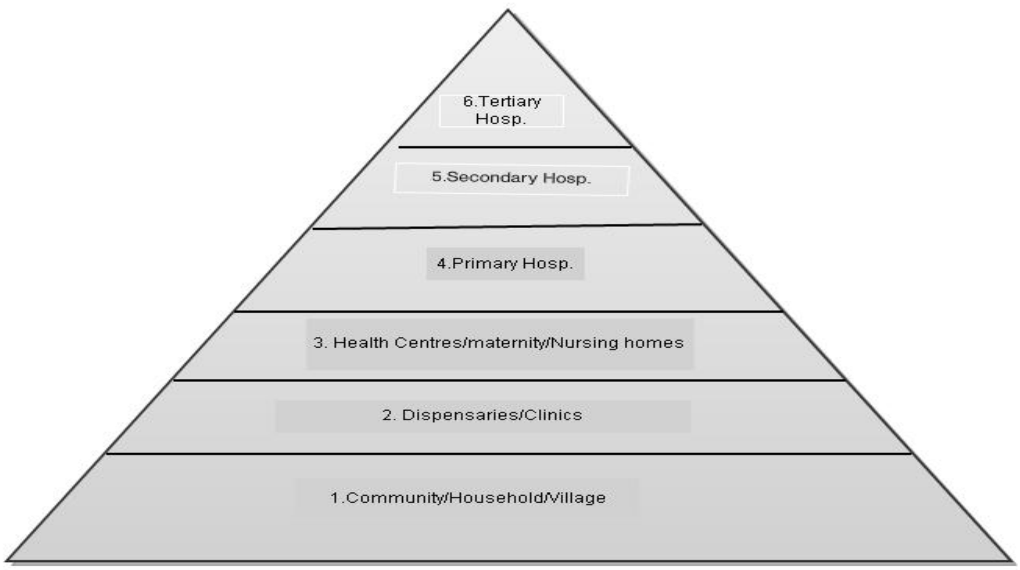


Figure 2-1: Health care structure source (NHSSP-II., 2005-2010)

By 2009 TB and Leprosy services were delivered through 2,318 health units and it is distributed as follows. Smear microscopy services were available at 1,030 of these units

Table 2-1 Provision of TB treatment services in 2009

	GOK	**NGO**	**PR (Private)**	**Total**
Hospitals	199	105	82	386
Health C.	544	118	60	722
Dispensaries	915	139	37	1091
AFB Lab facilities	753	172	105	1030
Other	8	20	53	81
Total	**1704**	**382**	**232**	**2318**

2.2 Software Agent Technology

Software agent technology is a promising approach for the analysis, design, and implementation of complex software systems. It provides a modeling abstraction for different levels in a development process. The agent technology provides the system designers with a natural abstraction in modeling a whole system or components of the system (Wooldridge and Jennings, 1999).

2.2.1 Software Agent

An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives (Wooldridge, 2008).

Computer agent has various characteristics that are given by various researchers from various fields. The following are a list of some characteristics based on (Shoham, 1999) :

- Autonomy: agents do not require constant human control or supervision.
- Environment awareness: agents model the environment in which they operate, track it, and react to change in it.
- Ongoing execution: unlike software routines that are invoked to achieve particular tasks and then disappear, agents function continuously for a long period of time
- Adaptiveness: over time, agents adapt their behavior to suit the preferences and behavior of individual user.
- Mobility: agent can migrate in a network.

(Wooldridge, 1995) believes that an agent may expect to have three levels of intelligent behaviors. These are:

- Reactivity: agents are able to perceive their environment, and respond in a timely fashion to the changes that occur in the environment in order to satisfy their design objectives.
- Proactivity: agents are able to exhibit goal-directed behaviors by taking the initiatives in order to satisfy their design objectives.
- Social activity: agents are capable of interacting with other agents in order to satisfy their design objectives.

Additional characteristics can be found, for example, in (Franklin and Graesser, 1997).

If a system does not need any of these properties, the system is better to design and implement without agents. For example, a program that reads an input and translates that to some other representation and then terminates, does not benefit from using software agent technology. In (Wooldridge and Jennings, 1998) give an overview of when to employ software agent technology, and when one should use more conventional techniques to solve the problem.

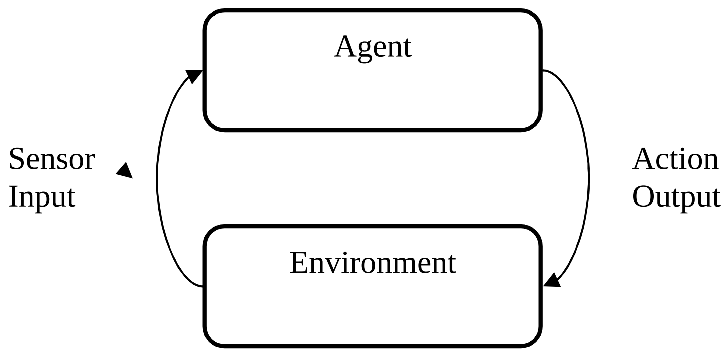


Figure 2-2 A Generic model of a basic agent

Figure 2-2 depicts the basic architecture of an agent and its interactions with its environment. The agent uses its sensors to receive input from the environment. Based on the input and its internal state, the agent decides what to do, that is, what actions it needs to perform, and then commands its effectors to carry out the selected actions.

Sometimes one agent is enough for solving a problem but at times, multiple agents are required to solve a problem, especially when the problem involves distributed data, knowledge, or control. When this is a case, a multi agent system (MAS) could be used to solve the problem. A MAS is a collection of several interacting agents in which each agent has incomplete information or capabilities for solving a problem (Jennings et al., 1998). As the inter-agent communication is an essential part of this thesis, we are more interested in multi-agent systems than single agent systems.

2.2.2 Agent Architectures

Agent architectures are design solutions, which describe agent's modules and capabilities, and how these operate together. An agent architecture helps to explain and to predict the agent behaviour, that is, it helps to understand how an agent's internal state affects its decisions, and how perceptions affect the agent's internal state. In addition, agent architecture may support the design of multi-agent systems, by providing tools and methodologies for designing agents and their interactions (Wooldridge et al., 1996). (Maes, 1991) defines agent architecture as " a particular methodology for building an autonomous agent. It specifies how the overall problem can be decomposed into sub problems, i.e., how the construction of the agent can be decomposed into the construction of a set of component modules and how these modules should be made to interact. The total set of modules and their interactions has to provide an answer to the question of how the sensor data and the current internal state of the agent determine the actions and future internal state of the agent".

Three types of agent architectures have been identified (Wooldridge and Jennings, 1995a): deliberative, reactive, and hybrid. Deliberative agents are based on the sense-plan-act problem-solving paradigm of classical AI planning systems. These agents model their world symbolically and make their action decisions using logical reasoning. The essential parts of a deliberative agent are a world model and a planner. The world model contains the agent's internal representation of its environment and the domain knowledge. The planner uses this information in planning how the agent can accomplish its goal. The BDI (Belief, desire, intention) architectures are probably the most popular agent architectures (Rao and Georgeff, 1995) and fall in the deliberative group. They have their roots in philosophy and offer a logical theory which defines the mental attitudes of belief, desire and intention using a modal logic. One of the most well-known BDI architectures is the Procedural Reasoning System (PRS) (Georgeff and Lansky, 1987). This architecture is based on five key data structures as depicted in the fugure below: beliefs, desires, intentions and plans, and an interpreter. In the PRS system, beliefs represent the information an agent has about its environment, which may be incomplete or incorrect. Desires represent the tasks allocated to the agent and so correspond to the objectives, or goals, it should accomplish. Intentions represent desires that the agent has committed to achieving. Finally, plans specify some courses of action that may be followed by an agent in order to achieve its intentions.

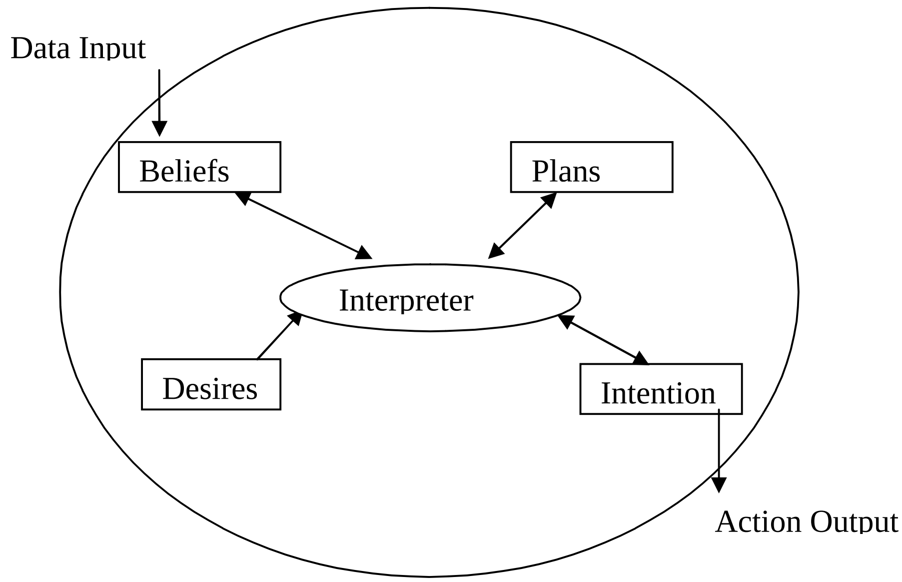


Figure 2-3: BDI architecture

BDI metaphor is a good choice for the computer agents to make to realize its practical reasoning. However, the weakness of this model exists in its disability to represent and interpret the relations between its components in agent's design and implementation as (Thangarajah et al., 2002) pointed out.

Brooks (1991) developed a new architecture called subsumption architecture. These architectures are often called reactive, behaviour-based, or situated architectures, and have a simple internal representation of the world, but a tight coupling between perception and action. The underlying idea of these architectures builds on the observation that many every-day routines are not based on abstract reasoning, but more on generic tasks. And especially, these generic tasks can be performed without sophisticated reasoning. Since reactive agents do not need time consuming reasoning, they can be used to build autonomous systems that need to react to environmental changes quickly. The reactive approaches have been demonstrated to be successful, however they have their own disadvantages (Jennings et al., 1998) including requires the agent to have enough information in its current state to choose an adequate action. Again it is difficult to design purely reactive agents that learn from experience and improve their performance over time.

Hybrid (layered) architectures is a blend of both deliberative and reactive behaviour. It allows the agent to respond quickly to the agents in the environment, but it also allows symbolic reasoning. The deliberative layer carries out the symbolic reasoning and based on the reasoning results, guides the reactive layer. The architecture can be arranged either vertically or horizontally (Müller et al., 1995). In the vertical layering approach (see figure), two different ways of arranging the data flow can be identified; the one pass control and two pass control. In one pass control the data passes through all the layers once while in two pass architecture, the data is directed back through the layers in the reverse order when reaching the top most layers. In horizontal layering (see figure), the layers are directly connected to the sensory input and action output, which essentially has each layer acting like an agent. The main advantage of this is the simplicity of design since if the agent needs different types of behaviors. However, since each layer is in effect an agent, their actions could be inconsistent prompting the need for a mediator function to control the actions. The drawback of hybrid approach is the problem of balancing the reactive and deliberative behaviour. (Kinny and George, 1991) concluded that if the world change rate is low, agents that do more reasoning perform better than those that employ a more straightforward behaviour. On the other hand, if the world change rate is erratic, a straightforward behaviour out performs those agents that waste too much time on thinking. Examples of hybrid agent architectures include Ferguson's Touring Machine (Ferguson, 1992) and others' Procedural Reasoning System (PRS).

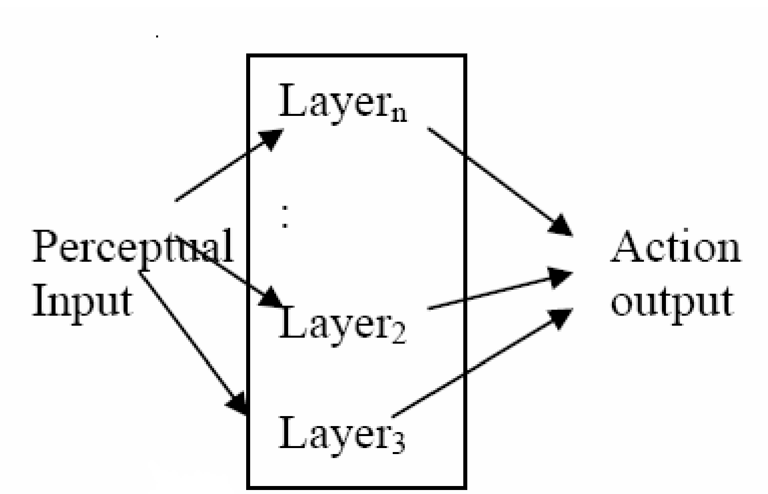


Figure 2-4: Horizontal Layering

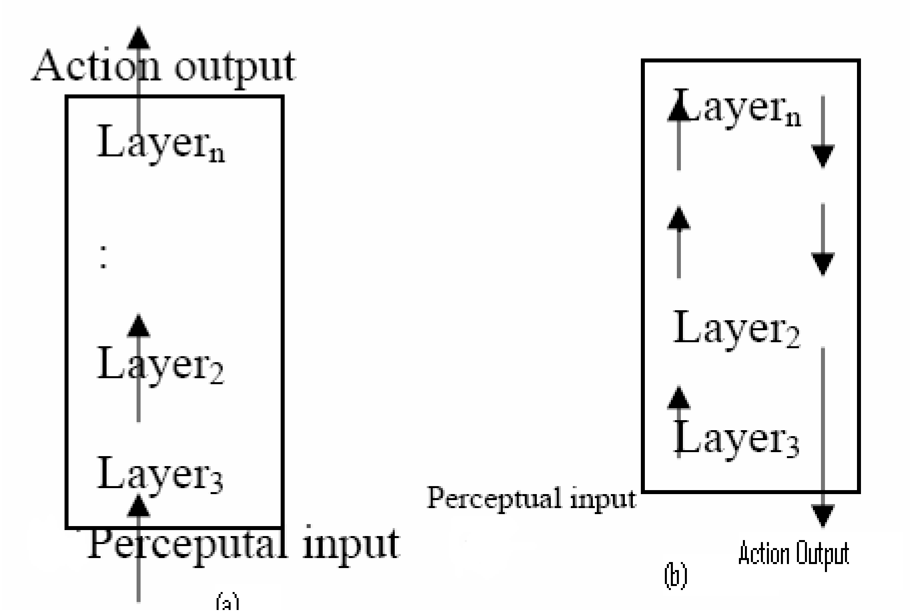


Figure 2-5: Vertical Layering

2.2.3 Agent Languages

Agent languages are programming languages designed for programming agents. While using any programming language to program agents is possible, using a language especially designed for agent programming may simplify implementing agents. The following are some of the languages used for programming agents: Agent0, April, Telescript, and Java e.t.c. I will only discuss the Java language used in the thesis.

Java - Java (Gosling and McGilton, 2003) is not strictly an agent programming language, but a general-purpose interpreted object-oriented programming language. Yet, it is widely used for implementing agents, and thus we cannot disregard it when discussing agent-programming languages. Java has become the de-facto standard for Internet programming, and is becoming that of the agent development (Wooldridge and Jennings, 1999). Given that Java compilers produce platform-independent bytecode , it is fairly easy to create applications based on the mobile agent technology. Also, Java is used for implementing intelligent agents and their platforms. For example, there are several FIPA compliant agent platforms implemented using Java, the most well-known examples being JADE (Bellifemine et al., 1999) and FIPA-OS (Poslad et al., 2000).

FIPA specification

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications (IEEE Standards Committee, 2004). FIPA specifications consist of five parts, i.e. abstract architecture, agent management, agent message transportation, agent communication, agent-based application. Agent management provides the normative framework within which agents exist and operate. It establishes the logical reference model for the creation, registration, communication, migration and retirement of agents, as shown in figure.3

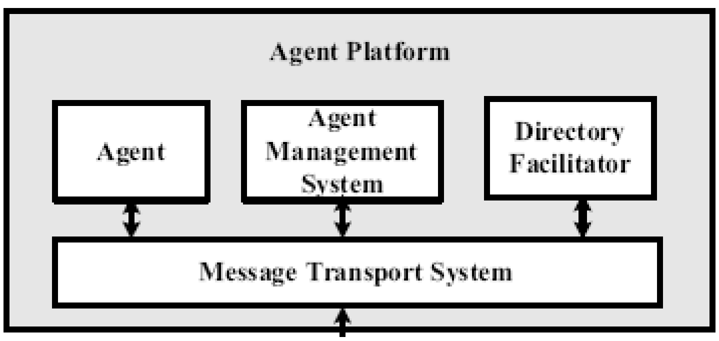


Figure 2-6 : Agent management reference model

2.3 JADE Development Environment

JADE (Java Agent Development Framework) is a software development environment aimed at developing multi-agent systems and applications conforming to FIPA standards for intelligent agents. It includes two main products: a FIPA-compliant agent platform and a package to develop Java agents. JADE is written in Java language and is made of various Java packages, giving application programmers both ready-made pieces of functionality and abstract interfaces for custom, application dependent tasks. Java was the programming language of choice because of its many attractive features, particularly geared towards object-oriented programming in distributed heterogeneous environments; some of these features are Object Serialization, Reflection API and Remote Method Invocation (RMI). In our study, we integrate JBuilder with JADE to facilitate developing agents in the system. The only necessary thing is make the JADE libraries known to JBuilder by means of adding jade.jar, jadeTools.jar, iiop.jar, and base64.jar into the configuration libraries of JBuilder. JADE provides a comprehensive set of API functions, which allow easy development of a MAS system for example:

- Built-in XML codec which implements the Agent Communication Language (ACL) format specified by FIPA. However, user defined communication language codec is also supported (Caire, G., 2002).
- Common interaction protocols (such as FIPA contract net protocol) are also built-in and ready for direct utilization (Bellifemine et al, 2004).

There are many other add-on components developed for JADE. For example, support for J2ME to enhance the usability of JADE in terms of integrating mobile applications, as well as its portability to embedded platforms

2.4 Agent concepts in E- Health

E-health systems are large scale complex systems characterized by distribution, coordination and sociability. The activities involved in a health care system require transfer and coordination of complex and diverse forms of information distributed among providers in the target healthcare system. According to (Nealon and Moreno, 2003) health care is a vast open environment characterized by shared and distributed decision making and management of care, requiring the communication of complex and diverse forms of information between a variety of clinical and other settings, as well as the coordination between groups of health care professionals with very different skills and roles. This characteristics presented by e health provides an opportunity to delegates this tasks to automated assistants called agents. The consensus in the agent research fields believes that agents' intelligent capabilities includes, at least but not limited, reactivity, proactivity and social activity (Wooldridge and Jennings, 1995).The properties of intelligent agents match quite precisely with our needs in this health care basically with the requirement of having autonomous intelligent proactive collaborative entities in a distributed environment.

2.4.1 Use of agents in health care domain

Use of agents in e-health involves (Nealon and Moreno, 2003)

- *Patient scheduling*: scheduling the activities to be carried out on a patient in a hospital(Paulussen et al., 2004).

- *Organ transplant management*: coordinating the management of organ and tissue transplants among different medical centres (Calisti et al., 2003).
- *Community care*: coordinating all the activities that have to be performed in order to provide an efficient health care to the citizens of a community (Pang, 2012).
- *Information access*: deploying information agents that gather, compile and organise medical information available in Internet, or providing mobile users with information about the medical centres or the doctors available in a particular town .
- *Decision aid systems*: monitoring the status of a hospitalised patient, and helping to diagnose the state of the patient (Laleci et al., 2008).
- *Internal hospital tasks*: for instance monitoring the application of medical, or controlling the usage of restricted use antibiotics (Godo et al., 2003).

2.5 Related Work

Various studies have been carried out on how MAS can be used in health domain applications. In this section we look at a few examples that put our study into practicability.

The fundamental component to the continuity of care is maintaining a record which summarizes patient's care history and shares with other health provider or counselor at the point of care irrespective of the location. On other side, access to Health Information Technology may enable healthcare providers to better understand and manage the conditions as well as assist patients with self-care management and permit timely sharing and access of accurate information between various healthcare providers.

Coffin et al (Coffin et al., 2004) describe a reminder system to repurpose administrative data in order to foster a program of disease prevention in an outpatient context. The paper describes architecture for determining who should be given the pneumonia vaccination based on the administrative data as opposed to clinical data. It uses intelligent agents to monitor looking for circumstances that require notifying healthcare professionals about various clinical events. These events may include giving shots, vaccinations, surgeries, follow-up checks and other important clinical events.

Corchado et al (Corchado et al., 2008) present an Intelligent environment for monitoring Alzheimer patients using agent technology .The proposed planning agent called AGALZ is a deliberative one and uses a case-based reasoning technique. The architecture allows it to respond to events, to take the initiative according to its goals, to communicate with other agents, to interact with users, and to make use of past experiences to find the best plans to achieve goals. The architecture utilizes the deliberative agent that works at a high level with the concepts of Believe, Desire, and Intention (BDI) .The main goal of the systems aim to support the Alzheimer patients in all aspects of daily life, predicting potential hazardous situations and delivering physical and cognitive support.

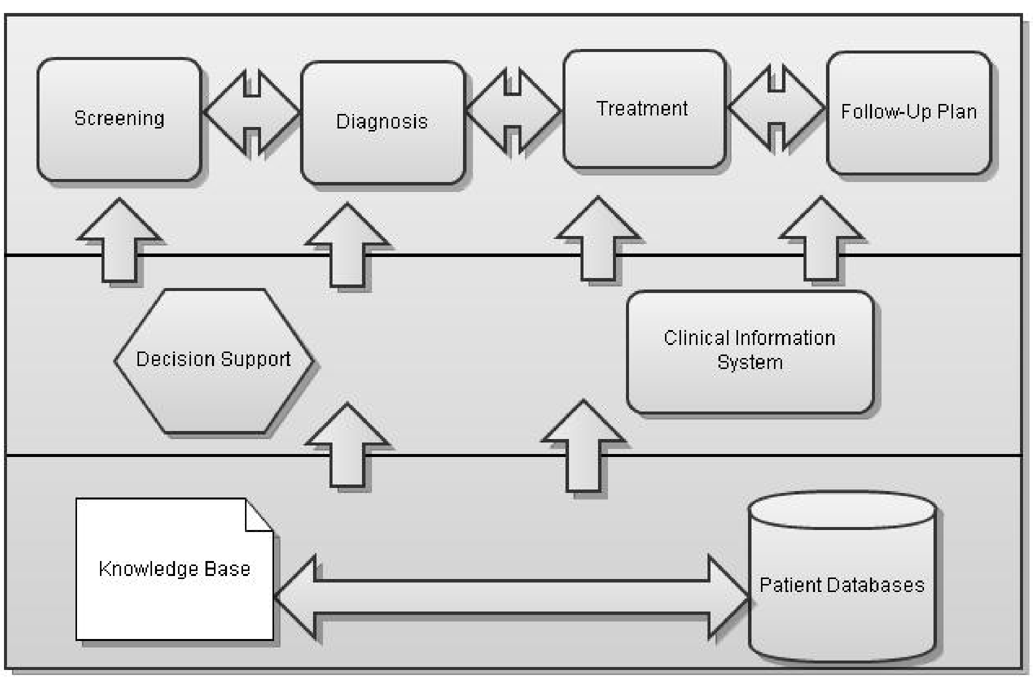
Zhang et al (Zhang et al., 2010) presents a multi agent based system for enhancing collaboration in Diabetic Healthcare for Children. The system is called IMAS and they presented the major key factors that the system was to solve and these are communication problem and intelligent decision making. The IMAS system purpose was to support the daily healthcare activities of patients and healthcare staff. It provides the patient with real time glucose monitoring and management, intelligent decision supports, user task delegation. The stakeholders (Parents, caretakers, nurse, doctor) each is assign an agent that is used to present the data at any one time. The system generates alarms i.e. when the patient is in an acute state, necessary actions must be taken as soon as possible. To do these the decision support module where by the patient history is scrutinized to help in arriving at a favourable decision to help the patient.

MADIP is an multi agent based system for wide area monitoring system (Su and Wu, 2011). In their architecture, it consists of hosts running the agent platform and hosts running supporting services such as a directory agent and a naming agent. The directory agent provides supervisory control over services that other agents provide in the environment. MADIP was built on top an already existing environment which is suitable to operate in a heterogeneous, networked environment such as the Internet to provide wide-area health monitoring service or other types of monitoring services. The architecture consists of two types of agents; Static agents that provide resources and facilities to mobile agents and Mobile agents that move between network domains gathering resources that are used to fulfill their goal.

2.6 Proposed architecture

For this study, we proposed architecture in which particular focus is placed on tracking patients from initial diagnosis to initiation of effective treatment and then monitoring them for treatment breaks or loss to follow-up. The architecture consists of the User interface layer, the Decision support layer and the data/information layer. The layering approach here is used to simplify the design complexity of the overall system by decomposing it into manageable pieces. On the interface layer, the physician and user agent exist to capture data from patient and user agent used to query the system. In the decision support layer the resource agent and database update agent exists. The decision is based on the underlying knowledge base. Knowledge captured from clinical practices is embedded into healthcare applications to assist healthcare providers' decision making. In the information layer, the reporting agent exists for reporting purposes.This is depicted in the following figure.

Figure 2-7: A proposed architecture



CHAPTER 3: METHODOLOGY

3.0 Overview

In this chapter, we describe a systematic step by step guide used to solve the research problem. We all explain the research design, sources of data, methods and tools to be used in data collection and the methods used in data analysis.

3.1 Research Design

Kothari define research design as the arrangement of conditions for collection and analysis of data in a manner that aims to combine relevance to the research purpose with economy in procedure. Using this definition, we intend to collect relevant data, analyze it and interpret the results.

3.1.1 Problem design

We conducted a simple survey using observation, interview and existing literature on how the NLTP is monitoring and managing patients with active TB and MDR-TB. We came to a conclusion that the evaluation and monitoring of TB patient is done using paper based systems. When the government wants to know the prevalence of the TB, they usually carry out a survey that requires a lot of resources and manpower to carry out. The monitoring aspects like follow ups and defaulters are not done. It only depends on the discipline of the patient and the efficiency of the public officers in charge of defaulters. We therefore decide to use the multi agent approach to enhance monitoring and support activities and make it easy to monitor and access information.

3.1.2 Use of MAS Methodology

The research is guided by Tropos methodology. Tropos is a methodology, for building agent-oriented software systems, tailored to describe both the organisational environment of a system and the system itself, employing the same concepts throughout the development stages. Tropos adopts the i* modeling framework (E. Yu,1995), which uses the concepts of actors, who can be (social) agents (organisational, human or software), positions or roles, goals and social dependencies (such as soft goals, tasks, and resources) for defining the obligations of actors (dependees) to other actors (dependers). This means the system (as well as its environment) can been seen as a set of actors, who depends on other actors to help them fulfill their goals. Tropos covers four phases of software development as follows:

- Early requirements – It is concerned with the understanding of a problem by studying an organizational setting; the output of this phase is an organizational model which includes relevant actors, their respective goals and their inter-dependencies. Early requirements include two main diagrams: the actor diagram and the goal diagram.
- Late requirements -The proposed system is described within its operational environment, along with relevant functions and qualities. The proposed system is represented as one actor which has a number of dependencies with the other actors of the organization. These dependencies define the system's functional and non-functional requirements.

- Architectural design – The proposed system’s global architecture is defined in terms of subsystems, interconnected through data, control and other dependencies.

  This phase is articulated in three steps:

    - Definition of the overall architecture
    - Identification of the capabilities the actors require to fulfill their goals and plans
    - Definition of a set of agent types and assignment to each of them one or more capabilities

  - Detailed design – The behaviour of each architectural component is defined in further detail. Each agent is specified at the micro-level. Agents' goals, beliefs and capabilities are specified in detail, along with the interaction between them.

3.1.3 Skills and Tool required

To accomplish the goal of this study, the following tools and skills are required:

- Java programming skills
- Software engineering skills

- Java JDK 1.6ul
- Windows Operating system (XP and above)
- Reference journals

3.2 Prototype implementation and testing

The implementation of the prototype was done using the Java eclipse and JADE. The Java eclipse is for building the main environment where agents reside and JADE for developing the agents of the system.

To enable testing of the system, Java runtime environment need to be installed on the computer. The prototype is run by invoking the agents and the outcome is observed and recorded.

3.3 Data sources and Collection Methods

In the study we made use of both primary and secondary data sources. The primary data was collected through observation and interview using open ended question with the TB staff. Interview schedule was the major research instruments used in the study. Secondary data has been researched from published and unpublished literature, academic papers, journals, internet and special reports.

3.4 Data analysis

After recording the outcomes of the prototype run, they are put in the tables (tabulated) and the results represented in the form of bar graphs. The reason for using bar graph is that they are easily interpreted by the users. This will then be evaluated to determine the importance and relevance of the study in relation to finding a solution for the stated problem

Chapter 4: SYSTEM ANALYSIS AND DESIGN

4.0 Overview

The analysis and design of the system was guided by the tropos methodology which has been discussed in chapter 2.This chapter describes in details how methodology was used in analysis and design

4.1 Early requirement Analysis

The requirement phase in Tropos consists of two steps; the early and late requirement. During the early requirements analysis the analysis engineer models the goals and the dependencies between the actors. To be able to models the goal, the initial description of the system is important.

4.1.1 Initial System Description

The TB health care fraternity requires the best systems that can optimally manage the TB patients from initial stage all through to the day when the patient is release (cured).This therefore requires a system that can monitor the patients and report on the best action on the interest of the patient. In the conventional system, The TB personnel would perform the following when managing the TB patient

- Record the arrival of new patient in a black book.
- When the patient comes for follow up, the TB personnel will look for the records and update it.
- The TB personnel need to coordinate with the doctor in ensuring the changes made by the doctor on patient status are recorded against the patient record.
- Every morning the TB personnel need to know the patient that are required to come for follow up and they are on what phase of treatment.
- Upon request, the TB personnel are required to write a transfer out, or receive transfer in and update the black book accordingly.
- It is upon the doctor to diagnosis the presence of MDR-TB.
- Weekly the TB personnel need to scrutinized the patient black book for defaulters and passed the information to the public Health officers.

4.1.2 Actors

The main actors indentified includes among others:

- TB patient is the patient that is suffering from any type of TB.
- Physician is the people who diagnosis/ treat the TB patients.
- Ministry of Health (MoH) is the government department that must provide TB patient with healthcare and support
- TB researchers /Agencies who wish to obtain information about TB patient for research

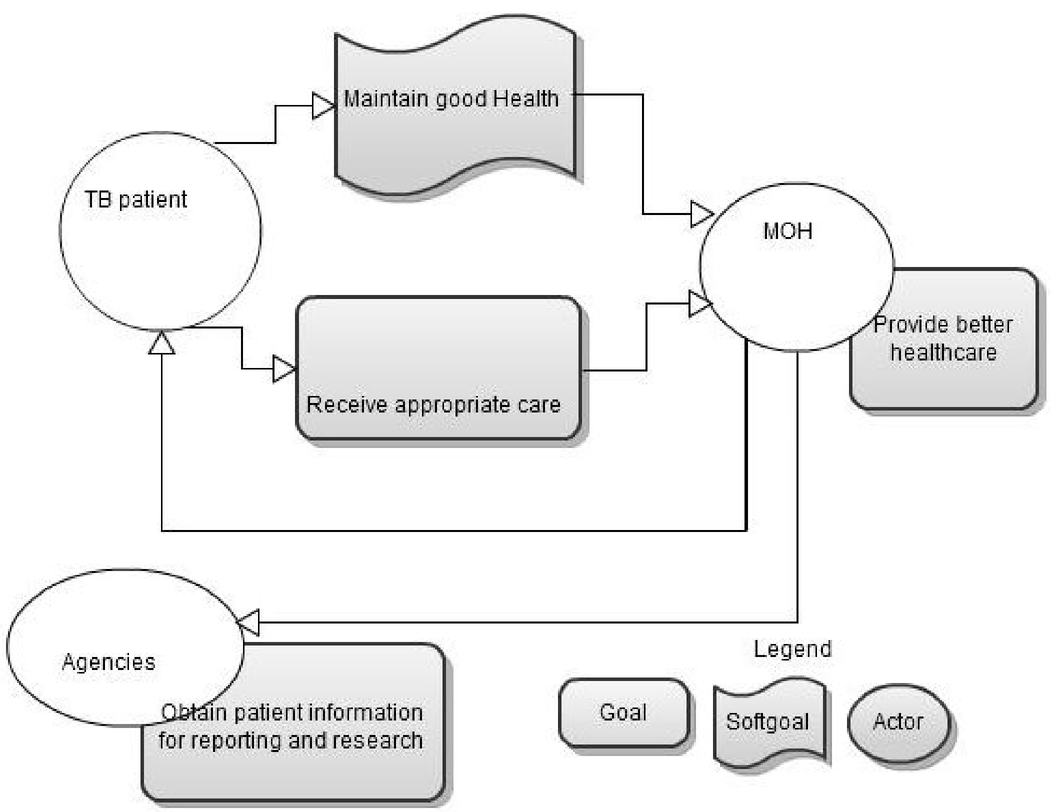


Figure 4-1: The stakeholders of the system

When the stakeholders, their goals and the dependencies between them have been identified, the next step of this phase is to analyze in more depth each goal relative to the stakeholder who is responsible for its fulfillment.

4.1.3 Goals of the system

When the stakeholders of the system have been identified, the next step is to identify goals associated with each actor. The TB patient actor has a goal to receive appropriate TB care and a soft goal to maintain good health.

To receive appropriate care goal is fulfilled by among others the following tasks:

- Undertake TB Assessment e.g. undertaking sputum test periodically
- Follow-up Care Plan,
- Obtain Medical Information
- Have Appointment with Physician.

The TB patient depends on the Ministry of Health department of Tuberculosis to make technology infrastructure available and easy to use.

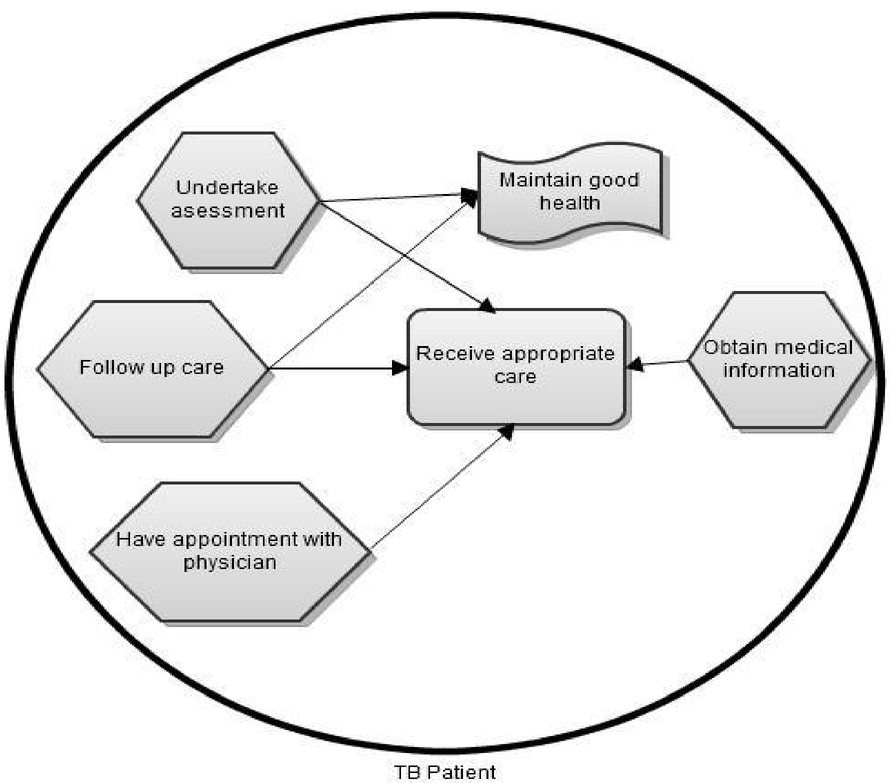


Figure 4-2: TB Patient Goal diagram

4.1.4 Use case scenarios

A medical clinical officer records details of a patient in the public TB system if the patient does not exist in the registration database. This is a scenario that happen when there is a new incidence. The use case diagram is as shown below:-

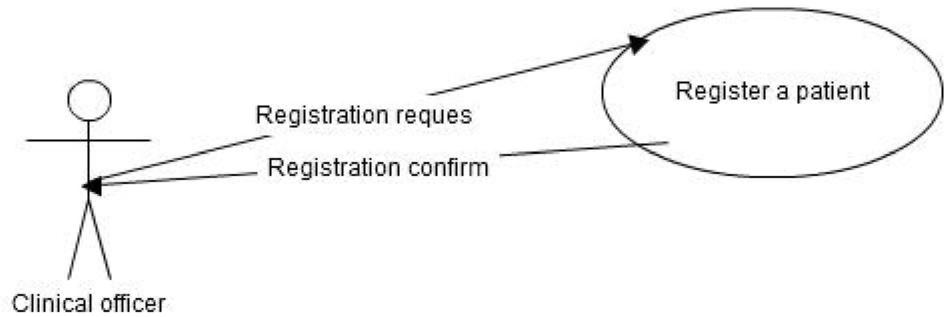


Figure 4-3: Registration Use case

A medical clinical officer examines a patient for a TB ailment and can recommend further analysis to be done on the patient by way of medical laboratory (Either X-ray or Sputum test (AFP)) , if the doctor is not sure about the disease the patient is suffering from. The use case diagram is as shown below:-

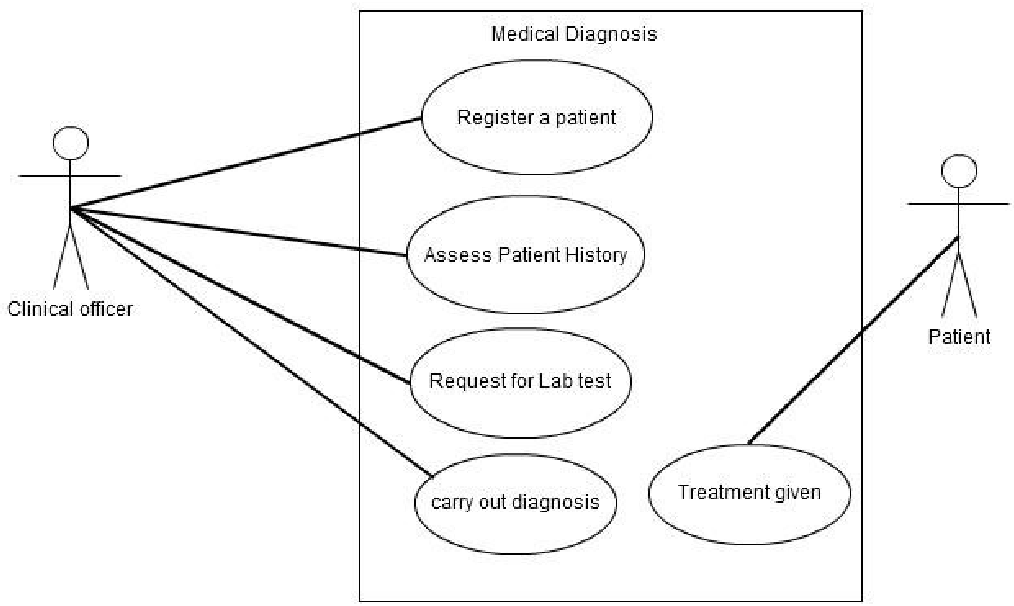


Figure 4-4: Patient diagnosis use case

4.2 Late Requirement Analysis

The system is described within its operation environment, along with relevant functions and qualities. The system is presented as one or more actors, who have a number of dependencies with the other actors of the organization. These dependencies define all functional and non-functional requirements for the system. The following diagrams depict the relationship between the actor TB patient the Health care provider where TB facilities falls using a mean end analysis diagram.

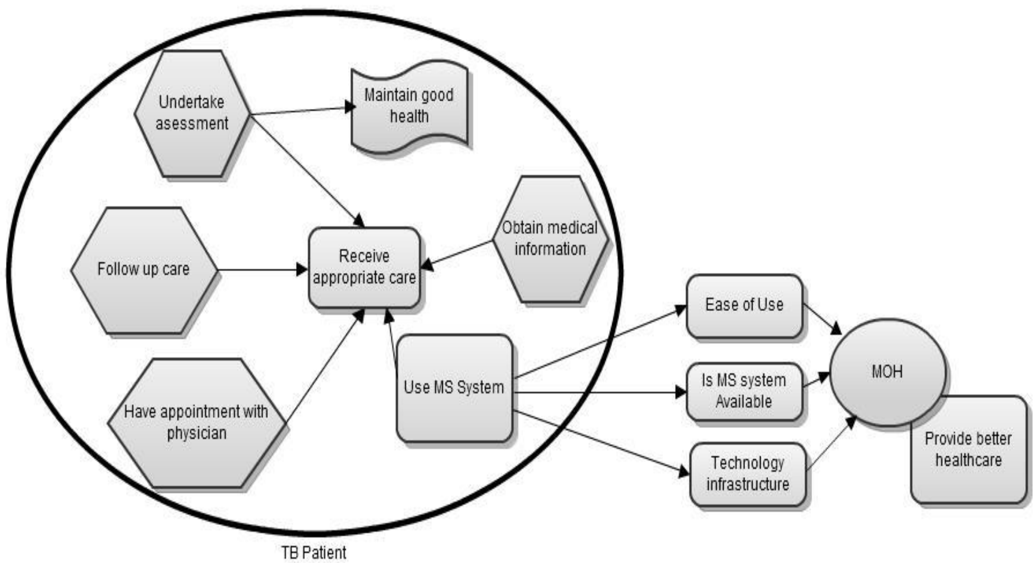


Figure 4-5 Mean end analysis for TB patient

4.3 Architectural Design

The actor decomposition and actor capabilities are described here.

4.3.1 Architectural overview

The following is sub Actor decomposition of monitoring and support system.

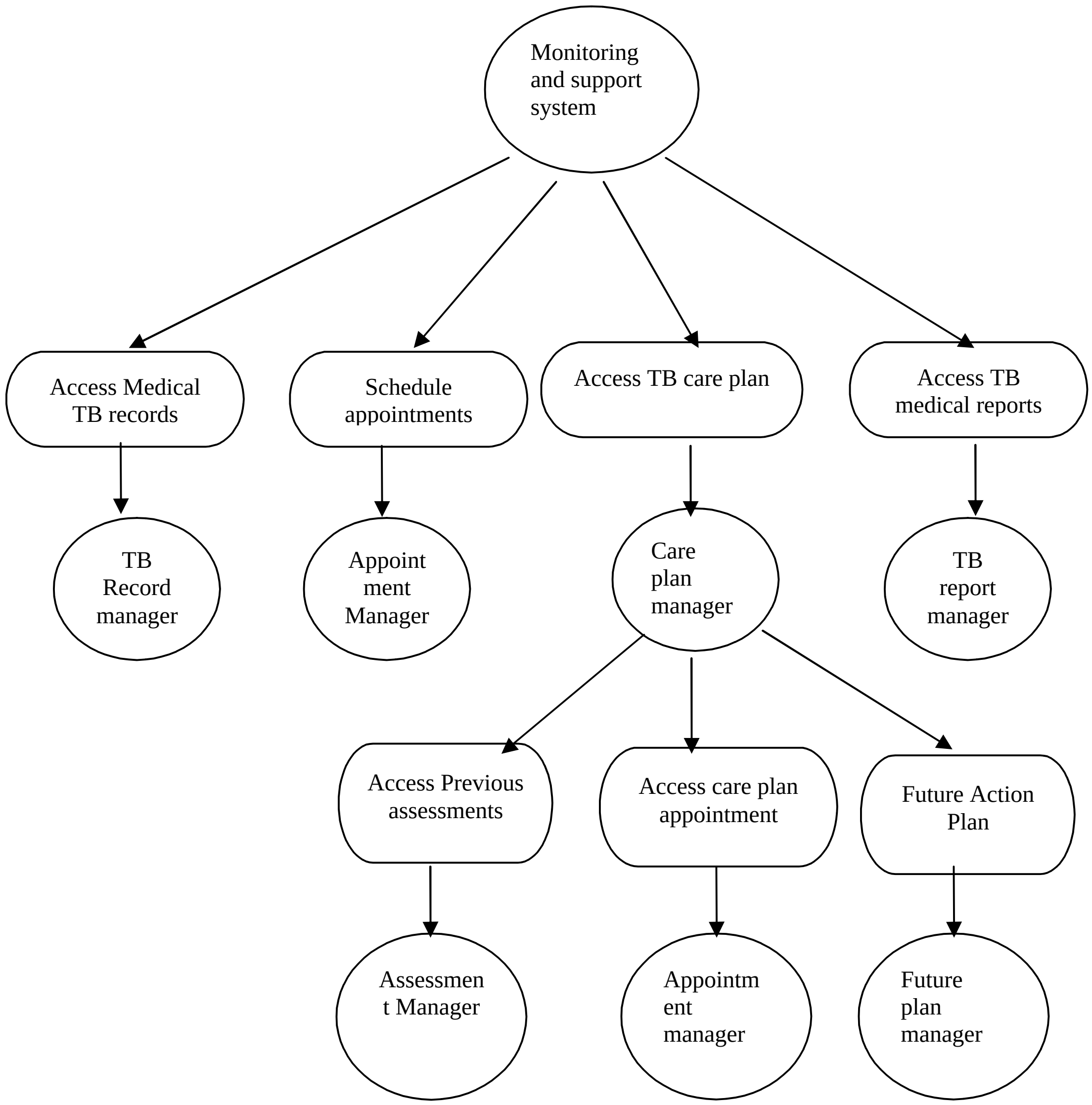


Figure 4-6 Sub- Actor decomposition

4.3.2 Agents in the system

In the system, the software agents will act on behalf of TB physicians and TB staff as the users. We therefore identified five types of agents in the system. These agents are namely

1) Patient agent
2) Database update agent
3) Reporting agent
4) Patient Observation agent
5) Resource agent
6) Communication Agent

i) Patient agent (PA)

This is the end used friendly agent. It is located where there is the reception of patients or retrieving already existing data. This agent is also used by the physician to archive or update the patient data. Its main duty is to receive patient data either on new enrollment or making changes and send it to the update agent which will then send it to backing storage after analysis e.g checking for existence of the same patient in the system.

ii) Reporting agent (RA)

The reporting agent resides on the web site. The main purpose of this agent is to receive queries from the visitors through some authorization, analyze the query and present the results of the query to the user in form of report.

iii) Database update agent (DA)

The database update agent updates the local database and the central database whenever there is a change .It also ensures that the records are not repeated.

iv) Communication Agent (CA)

Each of these agents is responsible for continuously monitoring the evolution of the patient. These agents can make sure that the patient follow up concerning his health status; they can also send reminders to the patient if they forget to appear on the schedule date. If the Agent detects any problem (e.g. the persistent of positive sputum test after the forth month), it can send a warning to the physician Agent responsible for that patient, so that appropriate measures are taken.

v) Patient Obs Agent (POA)

It has an interface used to input the observation whenever a patient visits the physician. This where the data used for follow up is capture.

vi) Resource Agent (RA)

The resource agent usually resides on the host to provide expert advice or services locally. It grant access the resources of database

4.2.3 Agent Descriptors

Table 4-1Agent Types and their Capabilities

No	Agent Type	Capabilities
1	PatientObs Agent	Application interface to the outside work (Used to input patient observations)
2	Database Update Agent	Get new records data Store data Update existing records Update central database
3	Patient Agent	Act on Behalf of Physician/TB staff Get user specification Provide Information about the Patient Get Physician query
4	Reporting Agent	Get user query Present query results to the user. Basically for generating reports
5	Resource Agent	Parses Incoming messages, and formats outgoing messages
6	communication Agent	Application interface to the outside work

4.2.4 System Overview Diagram

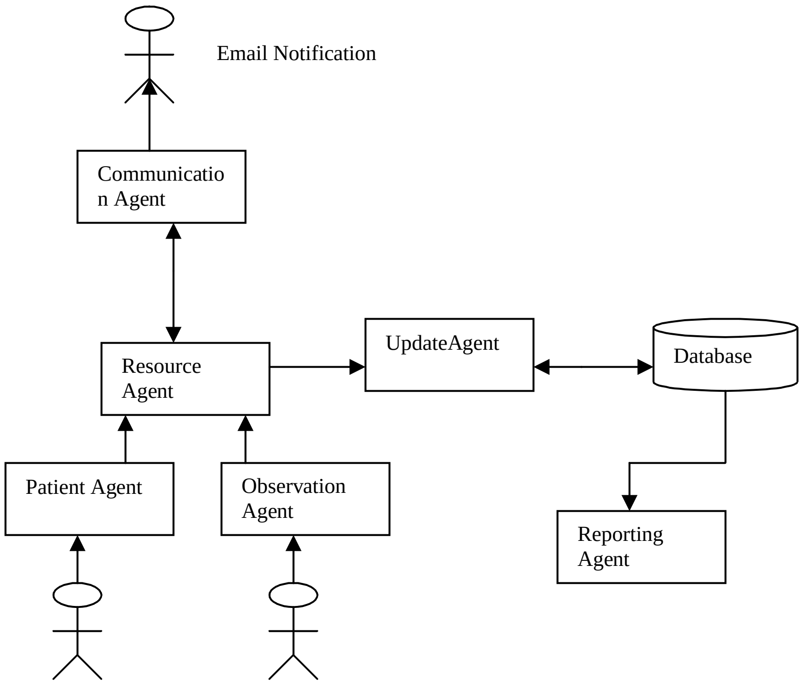


Figure 4-2 System Overview Diagram

4.4 Detailed Design

Detailed design stage aims at specifying agent capabilities and interactions. Thus,
Internal structures of individual agents are identified by the capability of each agent and are realized through the generation of capability Diagrams and sequence diagram.

4.4.1 Agent capabilities

From the agent and their capabilities diagram, each agent has at least one capability. Figure shows a simple part of the communicative interaction among the system's agents and the user. In particular, the diagram models the interaction among the user (doctor), the user interface (PA), the database and the query handler. The interaction starts with an information request by the user to the User interface, and ends with the results presentation by the User interface to the user.

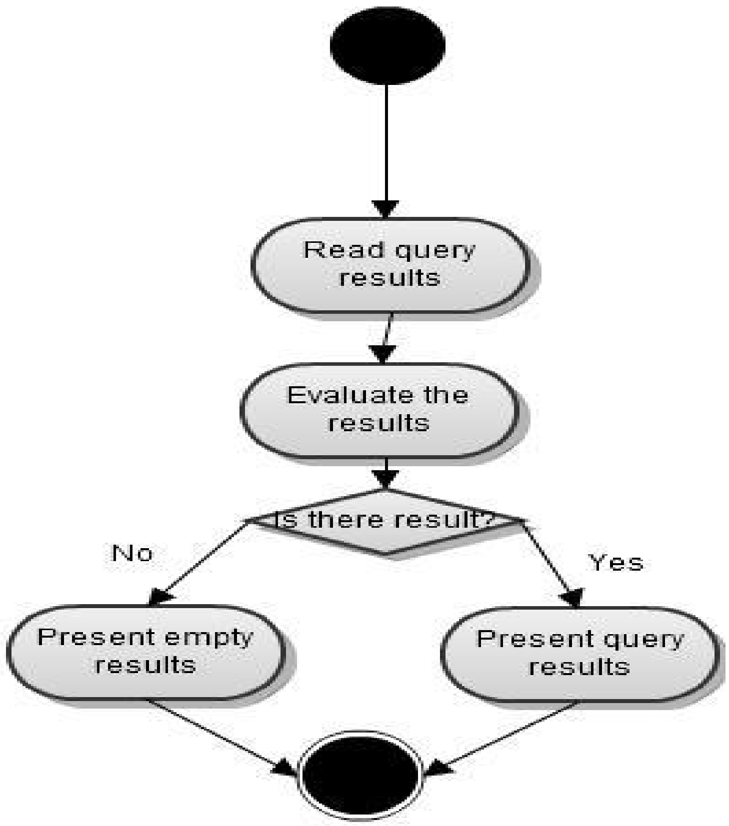


Figure 4-7: Capability diagram

4.4.2 Interaction between agents and the user

Agents in the system generally interact in their environment. The physician agent passes data to the database agent who analyses the data then store. The physician request information by submitting query through physician interface. The query handler will evaluate the query and send back the results. The agent interaction can be model using sequence diagram as shown in the below diagram

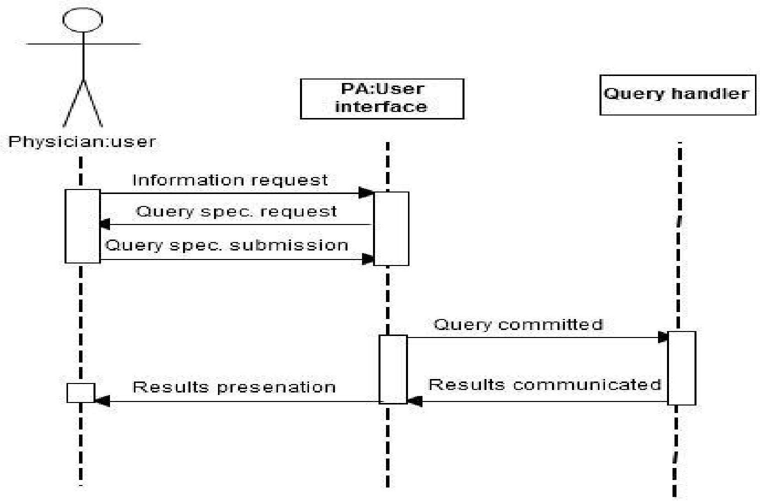


Figure 4-7: Sequence diagram

4.5 Data

The program was tested using test cases(appendix 2 and appendix 3) which simulate the actual happenings in a real life situation. Test cases were used for the TB management and were drawn with the assistance of the TB clinic staff Ministry of Health.

4.5.1 Input Data

The input data are of two types. The patient registration details and the observation details. This data is inputted as per patient arrival. When a patient is diagnose for the first time, the personal details is entered into the system once but the observation data depend on the number of time the patient went for consultation and follow ups. Anytime the patient goes for check ups the person responsible should update the patient records accordingly.

Each record has associated unique identifiers and TB clinic identifier. This identifier are used together to uniquely identified the patient.

4.5.2 Output Data

The agents in the system receive data from the GUI and update the underlying database. Using this data and a knowledge based, the agents can be able to monitor the patient details. The Phase can be in one of the following phases: initial or continuous. This fact is used to calculate the number of days the patient need to come for follow up and check ups. Basically follow up allows the doctors to monitor patient based on the bacteriological status (Sputum status.

In phase one, the patient need to see a doctor after every two weeks plus a grace period of three days that makes 15 days (WHO). For the continuing phase, the follow up days is one month. The communication phase uses this value to generate notifications for decision making and case management. The Agent compare the maximum observation date and the system date, and if it is more than stipulated above, the communication agent generate an alarm to the specified personnel in charge of the patient.

4.6 Verification and Validation

As (WHO), a TB treatment should take a comprehensive 6 months. In our test data the observation date was twitched so that we can be able to test whether the communication agent is working. The date was backdated by more than one month to check the functionality of the alarm/notification. As was discussed above, the agent compares the maximum observation date and the system date and the phase the patient is in. When the difference of dates is more than 15 days and patient is in phase one a notification is raised and if it is more than 31 days and patient is in phase two, the notification is raised.The notification is channeled to the person in charge of the patient. The physician in charge and the public officer in charge of the patient will receive a notification asking them to do something to ensure patient is followed.

Again, during data entry, the system generate new unique identifier of a new case, but before it checks the existing records using the identification number to ensure the record doe not exists.

4.7 Experimentation

To carry out the experimentation, we setup a scenario consisting of three computers. Each computer represents a centre. The TB stakeholders (Physician, nurses,clerks) was asked to test the system .Each one was to write down what they entered in a paper. This are then documented and you can find it in (appendix 2).

CHAPTER 5 : SYSTEM IMPLEMENTATION AND TESTING

5.1 Overview

For ease of implementation the system is broken down into smaller modules and then coded separately. The modules are then integrated to come up with the final system.

5.2 Implementation of the system

The system has been implemented using a combination of frameworks under the Java Enterprise Edition i.e

The following tools have been used

- ✓ J2ME – J2ME framework was used as a guiding framework for the development of the application
- ✓ The back end has been implemented using the eclipse enterprise
- ✓ The Multi agent component has been implemented using JADE which is an add-in to the above components
- ✓ The MySql database is used to store patient data.

5.2 Testing, Result and Discussion

For testing purposes, we subjected the multi agent system to various conditions to confirm whether it works. The structure of the test cases was defined by the TB clinician and they gave us the required boundaries for follow up. Using their guideline we did enter the data and test whether the system can be able to send notification for follow up misses and default. The outcomes were discussed in the following section

5.2.1: Patient Registration

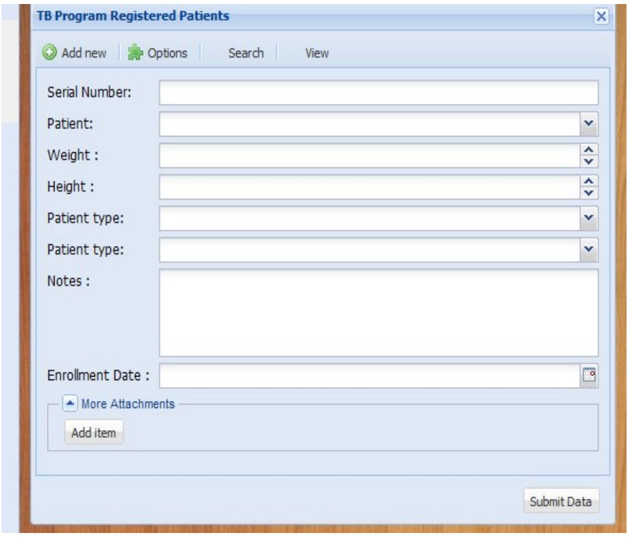


Figure 5-1: Patient Registration

This is a screenshot showing the form for patient registration. The hospital clerks used the form to input the patient personal details into the system

5.2.2: Observation Entry

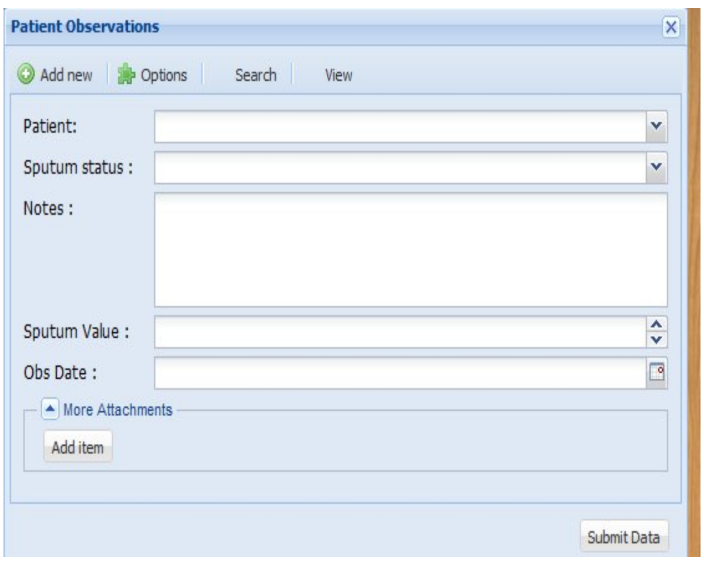


Figure 5-2 Patient Observation Form

The observation entry is used to input the observation details of the patient. During each visit the data is enter through this form. The patient is assigned a unique identifier during registration. The system is search using the combination of both unique identifier and

5.2.3: The communication agent

The communication agent is used for notification purposes. When a patient exceed the time he/she is required to go for follow up, the notification is message sent via mail to the concern party. This is the physician in charge of the patient and the public officer for that particular area.

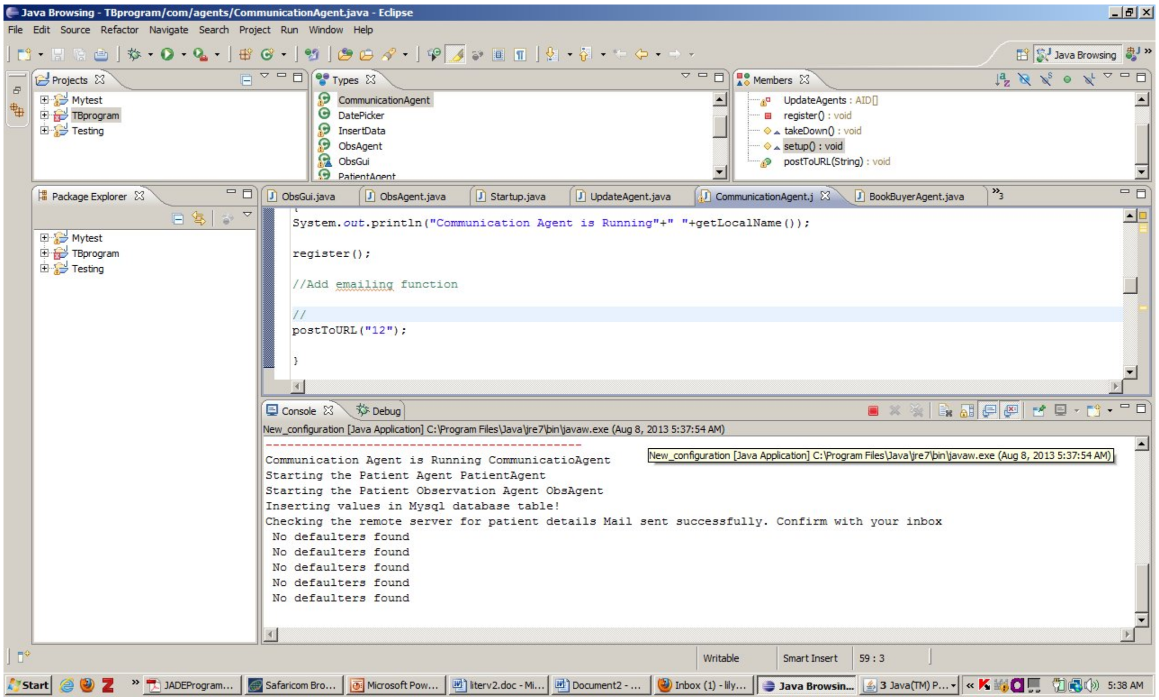


Figure 5-3 Showing notification Information

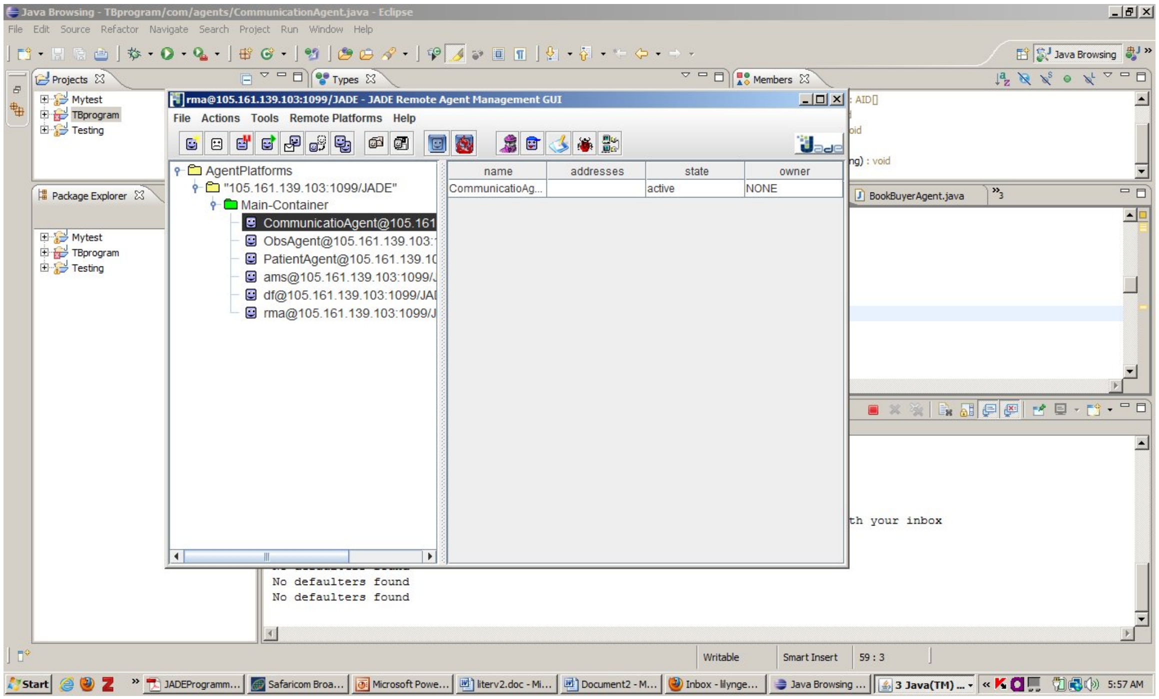


Figure 5-4: Agents in the Environment

Using sniffer window, the interaction between agents in the system are observe.This can be seen in the sniffer agent diagram below.

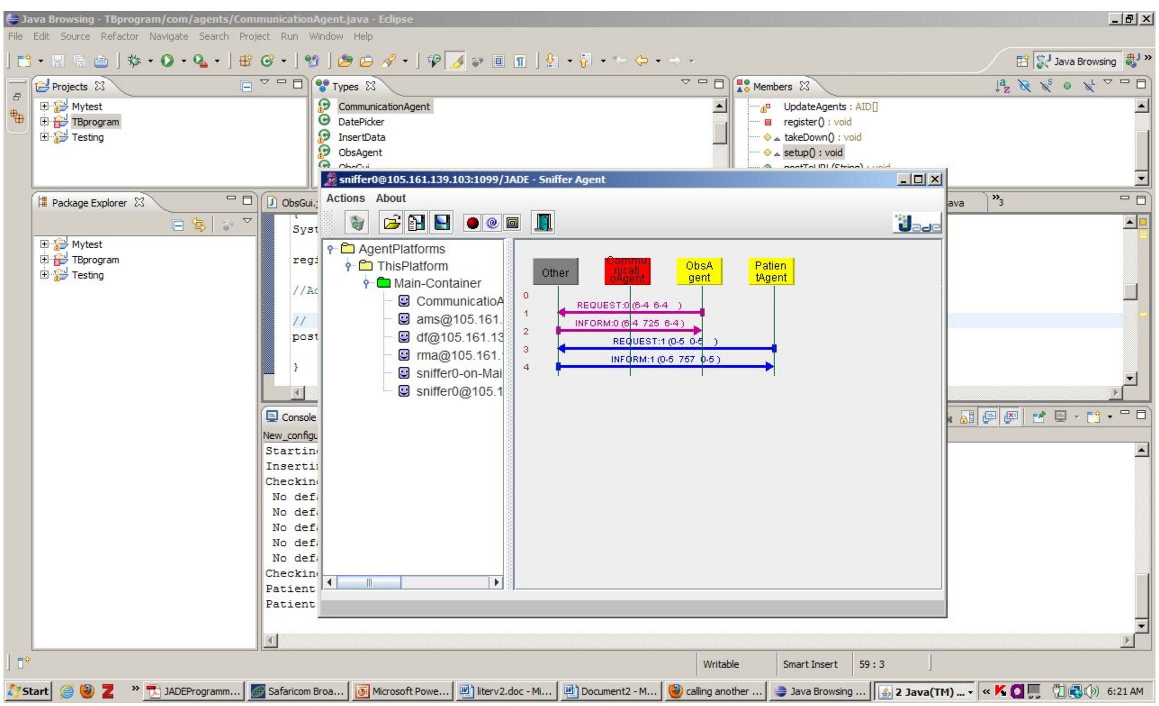


Figure 5-5 Message passing

5.2.4 Reporting Agent

Using the data that has been entered a reporting agent will generate a general report for all the existing patients. The following is a sample report generated for testing purposes.

Patient Registration Number

Direct Observation Treatment

Type of patient

Patient Observation Results

PID	Center	Name	Date	Sex	Age	Weight	Height	BMI	Address	DOT	Type TB	Type Patient	Culture	X-ray	Results
TB/001/2013	1	Ambrose Ochieng	12-May-2013	M	67	114	179	34	782 Mosoriot	H	P	N	Yes	Yes	Positive
TB/002/2013	2	Linnah Maina	13-Jun-2013	F	34	49	174	16	2100 Eldoret	H	P	N	Yes	Yes	Positive
TB/003/2013	1	Grace Ogot	14-Mar-2013	F	54	46	162	18.5	243 Eldoret	HCW	P	N	Yes	Yes	Positive
TB/004/2013	3	John Kimani	10-Aug-2013	M	66	57	172	18.6	80 Eldoret	H	P	N	Yes	Yes	Positive
TB/005/2013	2	Francis Ngaira	16-Jun-2012	M	16	46	167	16.4	8090 Eldoret	H	P	N	Not Done	Yes	Positive

Center	DOT		Type of patient		Culture
1=MTRH	Health care worker	=HCW	New	=N	Yes
2=UG District Hospital	Household Member	=H	Smear Positive relapse	=R+	Not Done
3=Huruma Clinic	Community Health Care Worker	=CHW	Smear Negative relapse	=R-	Done No Results
	Not Done	=ND	Failure	=F	

Figure 22 Sample Report generated by reporting Agent

5.3 System testing for user acceptability

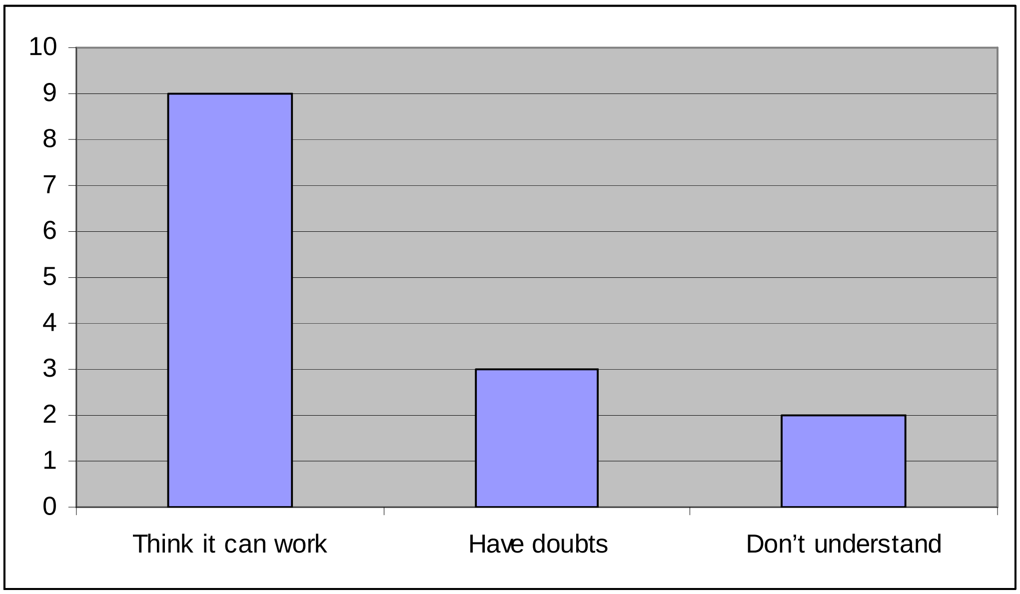
After implementation, we engaged sample users to verify the acceptability of the proposed system. We sample fourteen potential users and demonstrated to them how the system works. We then carried out a survey to determine their perspective of the system. Nine out of the fourteen sample users think the system can improve TB healthcare if implemented .Three have doubts about the system and two did not understand the importance of the system.

Total number of test users	Think it can work	Have doubts	Don’t understand
14	9	3	2

Table 5-1 User acceptance

Out of the fourteen sample users, only three said they have seen or used a monitoring system. The monitoring systems is based on the recording of the patient in the black book and also stress out that most follow ups occur if there is a pressing needs like an occurrence of an outbreak.

This can be represented graphically as below: -



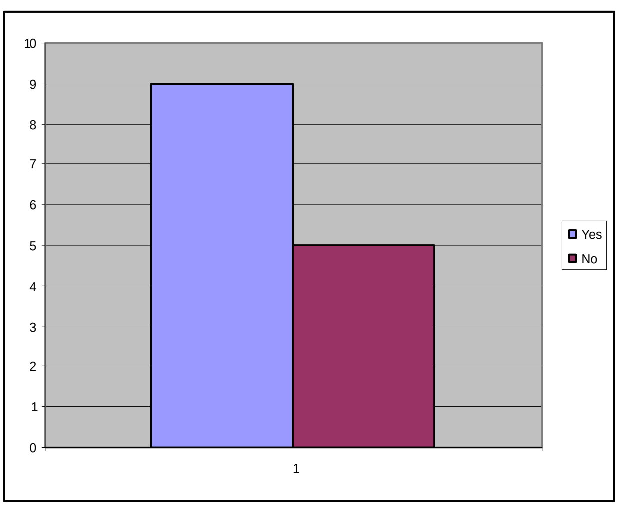


Figure 5-3: Inventory of system

The following figure 5-3 and figure 5-4 shows the rating of users based on the system functionality and ease of use.

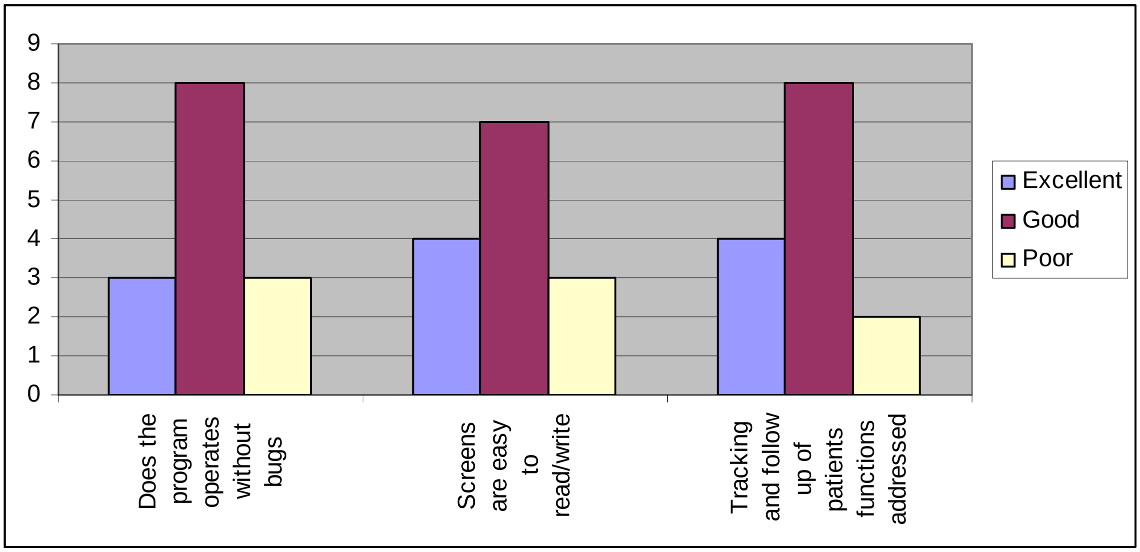


Figure 5-4 Acceptable performance Quality

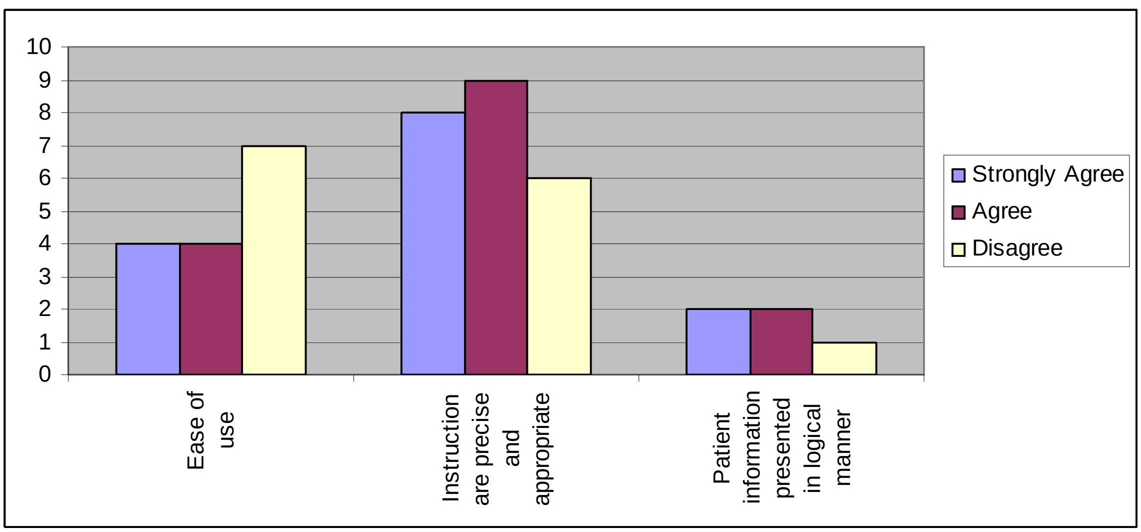


Figure 5-5 Functionality of the system

The monitoring and support activities were realized using eight agents as shown in Figure below:-

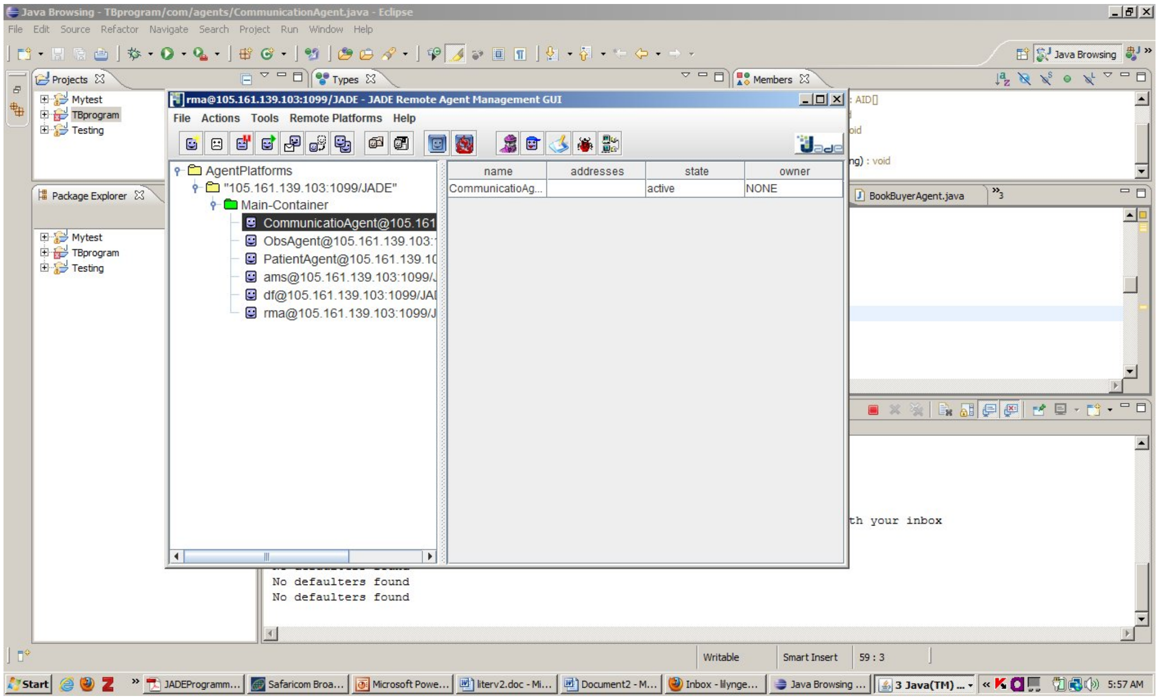


Figure 5-6: JADE Agent Management GUI

The execution of the communication agent and the results of agents sniffing the environment are shown in figures

The results show that the agents sense the environment on their own without any other human intervention and give the desired results as specified in the agent's behavior. The communication aspect that is provided well in agent behaviour protocol is importance compare to other software technologies

The results have shown that the JADE add-on works seamlessly with any Java Enterprise Edition components and can be integrated with any other technology so long as the libraries are specified in the class path.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Effective information systems in developing countries are a recent innovation but will need to play an increasing role in supporting and monitoring TB and MDR projects as they scale up from hundreds to thousands of patients. A particular focus should be placed on tracking patients from initial diagnosis to initiation of effective treatment and then monitoring them for treatment breaks or loss to follow-up. The process is however not very easy to carry out as it is currently majority of it is manually and the lack of coordination between the various centers. To effectively treat TB and MDR-TB we must use medications and laboratory tests that is well established but more so the patient monitoring to ensure correct regimen. The recent development and initiatives have made the TB facilities available almost to the lowest level of healthcare and this have shown that patient treated can have excellent outcomes (Fraser et al., 2002).

In this study, we have proposed an agent based system to monitor and support the TB patient with the aim of ensuring that the patient follows the correct regimen of drugs. The main goal of the study was to develop a multi agent based model that can assists the TB stakeholders (physician, public health officers) in ensuring that the patient are track/follow which will ensure the regimen and reduce relapse cases and defaulters. Kizito et al discusses the use of information system for Patient Follow-Up and Chronic Management of HIV and Tuberculosis and he called it a Life-Saving Technology in Resource-Poor Areas. In his study, he describes the challenges that are faced in terms of patient follows ups and monitoring. The agent in our model interactively performed the function of collecting, analyzing and presenting the monitoring information to the target user in this case the physician in charge of the patient or the public healthcare officer. Thus, this was a clear demonstration of how the agent technology can be utilized to improve the TB healthcare.

6.2 Limitations

The implementation of relatively high technology systems in a developing country context is not straightforward. Several concerns arise including the cost of deploying a health care system putting into consideration the limited resources. Data security, confidentiality and authorization also are an issue since the patient data must always be secure. The Data entry is a challenge since supervision and training may be required to ensure the data entered are complete and accurate.

6.3 Recommendation

The time available was not adequate to do a full analysis of the system. For tuberculosis treatment to be successful, patient-centered case management and close collaboration between health care professionals and local public health programs are imperative. Some areas that need to be covered if this system was to be implemented fully developed are the security and confidentiality of data, have a standard data entry form that must be strictly followed and design the best methods of sending notification.

The system also has to be implemented for more detailed testing and verifying that all components work together.

REFERENCES

Ahlburg, D.A., Initiative, S.T., 2000. The economic impacts of tuberculosis.

Bellifemine, F., Poggi, A., Rimassa, G., 1999. JADE–A FIPA-compliant agent framework, in: Proceedings of PAAM. p. 33.

Brooks, R.A., 1991. Intelligence without representation. Artif. Intell. 47, 139–159.

Calisti, M., Funk, P., Biellman, S., Bugnon, T., 2003. A multi-agent system for organ transplant management, in: Applications of Software Agent Technology in the Health Care Domain. Springer, pp. 199–212.

Chaiklin, S.E., Hedegaard, M.E., Jensen, U.J.E., 1999. Activity theory and social practice., in: Congress of the International Society for Acitivty Theory & Cultural Research, 4th, Jun, 1998, Aarhus, Denmark; The Contributions in This Book Are Revised Versions of Keynote Lectures and Invited Panel Presentations from the Aforementioned Meeting.

Coffin, C., Saunders, C., Thomas, C., Loewen, A., Campbell, N., Ghali, W., 2004. Using intelligent agents to repurpose administrative data in fostering disease prevention in an outpatient context: the case of pneumococcal vaccination, in: System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference On. p. 7–pp.

Corchado, J.M., Bajo, J., De Paz, Y., Tapia, D.I., 2008. Intelligent environment for monitoring Alzheimer patients, agent technology for health care. Decis. Support Syst. 44, 382–396.

Dye, C., Scheele, S., Dolin, P., Pathania, V., Raviglione, M.C., 1999. for the WHO Global Surveillance and Monitoring Project. Global burden of tuberculosis: estimated incidence, prevalence, and mortality by country. Jama 282, 677–86.

Enslow, P.H., 1978. What is a“ distributed” data processing system? School of Information and Computer Science, Georgia Institute of Technology.

Ferguson, I.A., 1992. TouringMachines: An architecture for dynamic, rational, mobile agents. University of Cambridge, Computer Laboratory Cambridge CB2 3QG, England.

FIPA, The Foundation for Intelligent Physical Agents. 2001, see http://www.fipa.org.

Flood, R.L., Jackson, M.C., 1991. Creative problem solving: Total systems intervention. Wiley Chichester.

Franklin, S., Graesser, A., 1997. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, in: Intelligent Agents III Agent Theories, Architectures, and Languages. Springer, pp. 21–35.

Fraser, H.S., Allen, C., Bailey, C., Douglas, G., Shin, S., Blaya, J., 2007. Information Systems for Patient Follow-Up and Chronic Management of HIV and Tuberculosis: A Life-Saving Technology in Resource-Poor Areas. J. Med. Internet Res. 9.

Fraser, H.S., Jazayeri, D., Mitnick, C.D., Mukherjee, J.S., Bayona, J., 2002. Informatics tools to monitor progress and outcomes of patients with drug resistant tuberculosis in Peru., in: Proceedings of the AMIA Symposium. p. 270.

Georgeff, M.P., Lansky, A.L., 1987. Reactive reasoning and planning., in: AAAI. pp. 677–682.

Giovanni Caire , JADE PROGRAMMING FOR BEGINNERS.pdf

Godo, L., Puyol-Gruart, J., Sabater, J., Torra, V., Barrufet, P., Fabregas, X., 2003. A multi-agent system approach for monitoring the prescription of restricted use antibiotics. Artif. Intell. Med. 27, 259–282.

Gosling, J., McGilton, H., 2003. The Java language environment: A white paper, 1995. Sun Microsystems.

Jennings, N.R., Sycara, K., Wooldridge, M., 1998. A roadmap of agent research and development. Automation. Agents Multi-Agent Syst. 1, 7–38.

Kinny, D., George, M., 1991. Commitment and Effectiveness of Situated Agents, in: IJCAI-91. pp. 82–88.

Kizito, K.W., Dunkley, S., Kingori, M., Reid, T., 2011. Lost to follow up from tuberculosis treatment in an urban informal settlement (Kibera), Nairobi, Kenya: what are the rates and determinants? Trans. R. Soc. Trop. Med. Hyg. 105, 52–57.

Laleci, G.B., Dogac, A., Olduz, M., Tasyurt, I., Yuksel, M., Okcan, A., 2008. SAPHIRE: a multi-agent system for remote healthcare monitoring through computerized clinical guidelines, in: Agent Technology and E-health. Springer, pp. 25–44.

Luck, M., McBurney, P., Preist, C., 2004. A manifesto for agent technology: Towards next generation computing. Auton. Agents Multi-Agent Syst. 9, 203–252.

Müller, J.P., Pischel, M., Thiel, M., 1995. Modeling reactive behaviour in vertically layered agent architectures, in: Intelligent Agents. Springer, pp. 261–276.

Nealon, J., Moreno, A., 2003. Agent-based applications in health care, in: Applications of Software Agent Technology in the Health Care Domain. Springer, pp. 3–18.

Organization, W.H., 1997. Global Tuberculosis Programme Treatment of Tuberculosis: Guidelines for National Programmes. World Heal. Organ. 2.

Organization, W.H., 2009. Global tuberculosis control 2009: epidemiology, strategy, financing. World Health Organization.

Pang, G.K., 2012. Health monitoring of elderly in independent and assisted living, in: Biomedical Engineering (ICoBE), 2012 International Conference On. pp. 553–556.

Paulussen, T.O., Zöller, A., Heinzl, A., Pokahr, A., Braubach, L., Lamersdorf, W., 2004. Dynamic patient scheduling in hospitals. Coord. Agent Technol. Value Networks Gito Berl.

Poslad, S., Buckle, P., Hadingham, R., 2000. The FIPA-OS agent platform: Open source for open standards, in: Proceedings of the 5th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents. p. 368.

Rao, A.S., Georgeff, M.P., 1995. BDI Agents: From Theory to Practice., in: ICMAS. pp. 312–319.

Shoham, Y., 1999. What we talk about when we talk about software agents. Intell. Syst. Their Appl. Ieee 14, 28–31.

Su, C.-J., Wu, C.-Y., 2011. JADE implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring. Appl. Soft Comput. 11, 315–325.

Thangarajah, J., Padgham, L., Harland, J., 2002. Representation and reasoning for goals in BDI agents, in: Australian Computer Science Communications. pp. 259–265.

Wooldridge, M., 1995. This is MYWORLD: the logic of an agent-oriented DAI testbed, in: Intelligent Agents. Springer, pp. 160–178.

Wooldridge, M., 2008. An introduction to multiagent systems. Wiley. com.

Wooldridge, M., Jennings, N.R., 1995a. Agent theories, architectures, and languages: a survey, in: Intelligent Agents. Springer, pp. 1–39.

Wooldridge, M., Jennings, N.R., 1995b. Intelligent agents: Theory and practice. Knowl. Eng. Rev. 10, 115–152.

Wooldridge, M., Jennings, N.R., 1998. Pitfalls of agent-oriented development, in: Proceedings of the Second International Conference on Autonomous Agents. pp. 385–391.

Wooldridge, M., Müller, J., Tambe, M., 1996. Intelligent Agents II-Agent Theories, Architectures, and Languages: IJCAI'95-ATAL Workshop, Montreal, Canada, August 19-20, 1995 Proceedings. Springer.

Wooldridge, M.J., Jennings, N.R., 1999. Software engineering with agents: Pitfalls and pratfalls. Internet Comput. Ieee 3, 20–27.

Zhang, P., Carlsson, B., Johansson, S.J., 2010. Enhanced collaboration in diabetic healthcare for children using multi-agent systems. Int. J. Agent-Oriented Softw. Eng. 4, 155–173.

Appendices

Appendix I: Sample code

Patient Agent

```
*******************
package com.agents;
import java.io.BufferedReader;

import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import java.util.Hashtable;

import javax.swing.JFrame;

//import examples.bookTrading.BookSellerAgent.OfferRequestsServer;

//import examples.bookTrading.BookSellerAgent.OfferRequestsServer;
//import examples.bookTrading.BookSellerAgent.PurchaseOrdersServer;
import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.behaviours.OneShotBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
```

```
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;
public class PatientAgent extends Agent {

        private static final int EXIT_ON_CLOSE = 0;

   private PatientNGui myGui;

        protected void setup(){
                System.out.println("Starting the Patient Agent"+" "+getLocalName());
                myGui = new PatientNGui(this);
           myGui.show();

           //Registering service in yellow pages
           DFAgentDescription dfd = new DFAgentDescription();
                dfd.setName(getAID());
                ServiceDescription sd = new ServiceDescription();
                sd.setType("Patient Registration");
                sd.setName("Tb program");
                dfd.addServices(sd);
                try {
                        DFService.register(this, dfd);
                }
                catch (FIPAException fe) {
                        fe.printStackTrace();
                }
                addBehaviour(new ReceiveMessage());
                }

        // Put agent clean-up operations here
        protected void takeDown() {
```

```
// Deregister from the yellow pages
try {
        DFService.deregister(this);
}
catch (FIPAException fe) {
        fe.printStackTrace();
}
// Close the GUI
myGui.dispose();
// Printout a dismissal message
System.out.println("Patient Registration Agent "+getAID().getName()+" terminating.");
}


private class ReceiveMessage extends CyclicBehaviour {

        public void action() {

                MessageTemplate mt = MessageTemplate.MatchPerformative(ACLMessage.CFP);
                ACLMessage msg = myAgent.receive(mt);
                if (msg != null) {
                ACLMessage reply = msg.createReply();
                msg.addReceiver(new AID ("PatientObs", AID.ISLOCALNAME));
reply.setPerformative( ACLMessage.INFORM );
reply.setContent(" Pong" );
 myAgent.send(reply);
 send(msg);
                }
```

```
				else {
					block();
				}
			}
		}
		private class MessageConfirm extends CyclicBehaviour {
			public void action() {
		MessageTemplate                    mt                    =
MessageTemplate.MatchPerformative(ACLMessage.ACCEPT_PROPOSAL);
				ACLMessage msg = myAgent.receive(mt);
				if (msg != null) {
					// ACCEPT_PROPOSAL Message received. Process it
					ACLMessage reply = msg.createReply();
					myAgent.send(reply);



				}
				else {
					block();
				}
			}
		}  // End of inner class OfferReque

}



package com.agents;

import jade.core.AID;

import java.awt.*;
```

```
import java.awt.event.*;
import javax.swing.*;
import util.RemoteSave;


/**
  @author Lily
 */
class PatientNGui extends JFrame {
	private PatientAgent myAgent;

	private JTextField nameField, middleField , lastField, ageField,genderField,
addressField;
	static final String gdList[] = {"Female", "Male"};
	JComboBox genderComboBox;
	PatientNGui(PatientAgent a) {
		super("Patient Registration");//a.getLocalName());


		myAgent = a;
		genderComboBox = new JComboBox(gdList);
		JPanel p = new JPanel();
		p.setLayout(new GridLayout(6, 2));
		p.add(new JLabel("First Name:"));
		nameField = new JTextField(15);
		p.add(nameField);
		p.add(new JLabel("Middle Name:"));
		middleField = new JTextField(15);
		p.add(middleField);

		p.add(new JLabel("Last Name:"));
```

```
lastField = new JTextField(15);
p.add(lastField);

p.add(new JLabel("Age:"));
ageField = new JTextField(15);
p.add(ageField);



p.add(new JLabel("Gender:"));
p.add(genderComboBox);

p.add(new JLabel("Address:"));
addressField = new JTextField(15);
p.add(addressField);

getContentPane().add(p, BorderLayout.CENTER);

JButton addButton = new JButton("Save");
addButton.addActionListener( new ActionListener() {
        public void actionPerformed(ActionEvent ev) {
                try {
                        String nameFieldtxt=nameField.getText().trim();
                        String middleFieldtxt=middleField.getText().trim();
                        String lastFieldtxt=lastField.getText().trim();
                        String ageFieldtxt=ageField.getText().trim();
                        //String
genderFieldtxt=genderField.getText().trim();
                        String
addressFieldtxt=addressField.getText().trim();
                        RemoteSave rsave=new RemoteSave();
```

```
String
valuesToSave="&middle_name="+middleFieldtxt+"&first_name="+nameFieldtxt+"&las
t_name="+lastFieldtxt;
                                //"&gender="+genderFieldtxt;
                                        rsave.saveValues(valuesToSave,"admin_person");
                                        addressField.setText(" ");
                                        nameField.setText("");
                                        middleField.setText("");
                                        lastField.setText("");
                                        ageField.setText("");

                                }
                                catch (Exception e) {
                                        e.printStackTrace();
                                        JOptionPane.showMessageDialog(PatientNGui.this,
"Invalid values. "+e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
                                }
                        }
                } );
                p = new JPanel();
                p.add(addButton);
                getContentPane().add(p, BorderLayout.SOUTH);

                // Make the agent terminate when the user closes
                // the GUI using the button on the upper right corner
                addWindowListener(new       WindowAdapter() {
                        public void windowClosing(WindowEvent e) {
                                myAgent.doDelete();
                        }
                } );
```

```
        setResizable(false);
    }

    public void show() {
        pack();
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        int centerX = (int)screenSize.getWidth() / 2;
        int centerY = (int)screenSize.getHeight() / 2;
        setLocation(centerX - getWidth() / 2, centerY - getHeight() / 2);
        super.show();
    }
}
```

2. Communication Agent

```
************************
package com.agents;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;

import jade.content.lang.Codec;
import jade.content.lang.sl.SLCodec;
import jade.content.onto.Ontology;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.OneShotBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
public class CommunicationAgent extends Agent {
private Codec language = new SLCodec();



private void register()//Registration


{
DFAgentDescription dfd = new DFAgentDescription();
dfd.setName(getAID());
ServiceDescription sd = new ServiceDescription();
```

```
sd.setType("Communication Agent");
sd.setName(getLocalName());
dfd.addServices(sd);

try {
DFAgentDescription list[] = DFService.search(this, dfd);
if (list.length > 0) { DFService.deregister(this);
}

dfd.addServices(sd); DFService.register(this, dfd);
} catch (Exception e) {
}
}

@Override
protected void takeDown() {
try { DFService.deregister(this);
} catch (Exception e) {
}
}
@Override
protected void setup()
{
System.out.println("Communication Agent is Running"+" "+getLocalName());

register();

//Add emailing function

//
postToURL("12");
```

```
}

public void postToURL(String receiverid){

        System.out.print("Checking the remote server for patient details "  );

        try {

  String stringToReverse = URLEncoder.encode("pi", "UTF-8");

  //URL url = new URL("http://intelibizafrica.com/tb/sendtbemail.php");
  URL url = new URL("http://localhost/tbprogram/home/sendtbemail.php");
  URLConnection connection = url.openConnection();
  connection.setDoOutput(true);

  OutputStreamWriter out = new OutputStreamWriter(
  connection.getOutputStream());
  //+"&string2="+"Uganda"
  out.write("pi=" + receiverid);
  out.close();

  BufferedReader in = new BufferedReader(
                  new InputStreamReader(
                  connection.getInputStream()));
  String decodedString;
  while ((decodedString = in.readLine()) != null) {
     System.out.println(decodedString);
   }
  in.close();
```

```
    }
    catch(IOException e){
        e.printStackTrace();
    }

}
}
```

3. Patient Observation Agent

****

```
package com.agents;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import java.util.Hashtable;

import javax.swing.JFrame;


import jade.core.Agent;
import jade.core.behaviours.OneShotBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
public class ObsAgent extends Agent {
```

```
        private static final int EXIT_ON_CLOSE = 0;

    private ObsGui myGui;

        protected void setup(){
                System.out.println("Starting the Patient Observation Agent"+"
"+getLocalName());
                myGui = new ObsGui(this);
            myGui.show();
            //Registering service in yellow pages
            DFAgentDescription dfd = new DFAgentDescription();
                dfd.setName(getAID());
                ServiceDescription sd = new ServiceDescription();
                sd.setType("Patient Observation");
                sd.setName("Tb program");
                dfd.addServices(sd);
                try {
                        DFService.register(this, dfd);
                }
                catch (FIPAException fe) {
                        fe.printStackTrace();
                }

                }

        // Put agent clean-up operations here
        protected void takeDown() {
                // Deregister from the yellow pages
                try {
                        DFService.deregister(this);
```

```
		}
		catch (FIPAException fe) {
			fe.printStackTrace();
		}
		// Close the GUI
		myGui.dispose();
		// Printout a dismissal message
		System.out.println("Patient  Observation  Agent  "+getAID().getName()+"
terminating.");
	}


	}
```

```
package com.agents;

import jade.core.AID;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import util.RemoteSave;
```

```
/**
  @author Lily
 */
class ObsGui extends JFrame {
	private ObsAgent myAgent;

	private JTextField sputumStatusField, sputumValueField , obsDateField, notesField;
    static final String gapList[] = {"Negative", "Positive"};
	InsertData insertData=new InsertData();
	JComboBox patientComboBox, sputumStatusComboBox;
	ObsGui(ObsAgent a) {
		super("Patient Observation");//a.getLocalName());



		myAgent = a;
		patientComboBox = new JComboBox(insertData.getRegistered("active").toArray());

		sputumStatusComboBox = new JComboBox(gapList);

		JPanel p = new JPanel();
		p.setLayout(new GridLayout(6, 2));

		p.add(new JLabel("Patient:"));
		p.add(patientComboBox);

		p.add(new JLabel("Sputum Status:"));
		p.add(sputumStatusComboBox);

		p.add(new JLabel("Sputum Value:"));
```

```
sputumValueField = new JTextField(15);
p.add(sputumValueField);

p.add(new JLabel("Obs Date:"));

obsDateField=new JTextField(15);
p.add(obsDateField);
p.add(new JLabel("Notes:"));
notesField = new JTextField(15);
p.add(notesField);







getContentPane().add(p, BorderLayout.CENTER);

JButton addButton = new JButton("Save");
addButton.addActionListener( new ActionListener() {
    public void actionPerformed(ActionEvent ev) {
        try {
            String
sputumStatusFieldtxt=(String)sputumStatusComboBox.getSelectedItem();
            String
personname=(String)patientComboBox.getSelectedItem();
            String
sputumValueFieldtxt=sputumValueField.getText().trim();
            String
obsDateFieldtxt=obsDateField.getText().trim();
            String notesFieldtxt=notesField.getText().trim();
```

```
                    RemoteSave rsave=new RemoteSave();

                String
valuesToSave="&person_name="+personname+"&notes="+notesFieldtxt+"&sputum_va
lue="+sputumValueFieldtxt+"&sputumstatus_id="+sputumStatusFieldtxt+"&obs_date="
+obsDateFieldtxt;
                //"&gender="+genderFieldtxt;
                    rsave.saveValues(valuesToSave,"patient_obs");
                    sputumValueField.setText("");
                    obsDateField.setText("");
                    notesField.setText("");
                }
                catch (Exception e) {
                    e.printStackTrace();
                    JOptionPane.showMessageDialog(ObsGui.this,
"Invalid values. "+e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        } );
        p = new JPanel();
        p.add(addButton);
        getContentPane().add(p, BorderLayout.SOUTH);

        // Make the agent terminate when the user closes
        // the GUI using the button on the upper right corner
        addWindowListener(new    WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                myAgent.doDelete();
            }
```

```
		} );

		setResizable(false);
	}

	public void show() {
		pack();
		Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
		int centerX = (int)screenSize.getWidth() / 2;
		int centerY = (int)screenSize.getHeight() / 2;
		setLocation(centerX - getWidth() / 2, centerY - getHeight() / 2);
		super.show();
	}
}
```

Appendix 2: Test case data (Patient Registration)

S.No	PID	Name	ID/Passport	DoB	PIN	Gender	Status
1	PID/000001/2013	Emmanuel Ken Were	2010129	7/30/1920	20192925t	M	Active
2	PID/000002/2013	Lurambi Shihundu Martin	212134342	7/27/1980	A2019202M	M	Active
3	PID/000003/2013	Imani Em Emily	121212	7/12/2012	A9304493M	F	Active
4	PID/000004/2013	Jacob Suleimani Selemani	3433433	4/4/1987	A9202982M	M	Active
5	PID/000005/2013	glify onyango g	20102019	10/12/1998	A920298M	F	Active
6	PID/000006/2013	dfff wdas sdsa	73490	8/7/2013	W65458453	F	Active
7	PID/000007/2013	husein Muhad Kenya	13223	12/15/1980	G79843575	M	Active
8	PID/000008/2013	Hezron sdasdas sdas	989586	8/26/1960		M	Active
9	PID/000009/2013	chege Maina Kimani	7684434	2/17/1984		M	Active
10	PID/000010/2013	henry hdh ddd	958443	1/30/1978		M	Active
11	PID/000011/2013	peter K Samwel	789032	8/25/1971		M	Active
12	PID/000012/2013	yaya ken	784853	6/20/1986		M	Active
13	PID/000013/2013	musa james	984523	6/9/1989		M	Active
14	PID/000014/2013	Rosa mathes	76342	4/12/1979		F	Active
15	PID/000015/2013	reech mikes	4758539	3/29/1962		M	Active
16	PID/000016/2013	sandra ondiek	237334	9/24/1986		F	Active
17	PID/000017/2013	muli moses	65213948	10/25/1979		M	Active
18	PID/000018/2013	terer geoffrey	968780546	1/18/1980		M	Active
19	PID/000019/2013	mathe kioo kim	6347562384	1/12/1967	76523746b	M	Active
20	PID/000020/2013	jack suleiman james	93774	12/17/1980		M	Active
21	PID/000021/2013	Lilian j. koskei	1848284	11/14/1978		F	Active
	Owner						
	1=MTRH						
	2=UG District Hospital						
	3=Huruma Clinic						

Appendix 3: Observation Test Data

S.No	PID	Sputum ID	Sputum Status	Notes	Obs Date	Phase	Owner
1	PID/000007/2013	896	Negative	Good	5/13/2013	1	1
2	PID/000007/2013	234	Negative		5/28/2013	1	1
3	PID/000007/2013	456	Negative		6/14/2013	1	1
4	PID/000007/2013	235	Negative		6/28/2013	1	1
5	PID/000002/2013	9512	Positive	df	5/1/2013	1	2
6	PID/000002/2013	1567	Positive	sds	5/16/2013	1	2
7	PID/000002/2013	765	Negative	Hellow	6/11/2013	1	2
8	PID/000002/2013	450	Negative	juma	6/14/2013	2	2
9	PID/000004/2013	340	Negative	cvfcf	7/21/2013	1	3
10	PID/000005/2013	732	Negative	saasddf	7/21/2013	1	3
11	PID/000006/2013	546	Negative	kktkt	7/21/2013	1	2
12	PID/000013/2013	233	Negative	cvfcf	6/23/2013	1	1
13	PID/000017/2013	2432	Positive	ytereqryew	7/21/2013	1	3
14	PID/000013/2013	2654	Positive	sdsds	7/21/2013	1	1
15	PID/000017/2013	895	Negative		7/15/2013	1	3
16	PID/000017/2013	980	Negative	ghgh	6/1/2013	1	3
17	PID/000014/2013	2345	Positive	Look at the drugs resistance	5/2/2013	1	2
18	PID/000012/2013	2145	Positive	Improving	4/12/2012	1	2
19	PID/000011/2013	567	Negative	Good	4/13/2012	1	3
20	PID/000011/2013	543	Negative	Healing	4/14/2012	1	3
22	PID/000012/2013	1090	Positive	Improving	6/6/2012	1	2
23	PID/000012/2013	1290	Positive	Improving	8/17/2012	2	2
24	PID/000011/2013	8456	Positive	the status not good	6/8/2012	1	3
24	PID/000002/2013	450	Negative	juma	7/14/2013	2	2

Phase
1 - Initial phase
2 - Continuos Phase

Appendix 4: Questionnaire

Evaluation Form: Multi agent monitoring and support system for TB prevalence
Is there any monitoring and support system used in your TB clinic?

- No
- Yes
- Other

Is there a system for tracking and managing TB patients?

- No
- Yes

For each of the following basic question, indicate the level of acceptance.
Mark only one per row.

	Strongly disagree	disagree	agree	Strongly agree
1) Ease of use				
2) Instruction are precise and appropriate				
3) Patient information presented in logical manner				
4) Promotes fast entry of patient data				
5) Is free of racial, ethical, and sexual stereotypes				
6) Feedback is effective and appropriate				
7) Provides monitoring and follow-up activities				

For each of the following functionality skills, indicate the level of importance.
Mark only one per row.

	Not Applicable (N/A)	Does not exist (None)	Poor	Good	Excellent
Are the systems easy to use?					
Do the systems provide enough information for case management					
Are the systems web based?					
Is Notification/alarm understandable?					
Have all the tracking and follow up of patients function addressed					
Do the system generate reports					

The following questions ask about the acceptable performance quality of the system
Mark only one per row.

	Strongly Disagree	Disagree	Agree	Strongly Agree
Does the program operates without problems(bugs)				
Screens are easy to read/write				
Menus are easy to access				
Does the program branch to remedial routines?				

Would you recommend this program to others?

- o No
- o Yes

If you were to recommend this program, what positive features would you stress?

____

____

____

____

Are there any weaknesses with the program? If so, what are they?

Any other comments

Appendix II: Gantt Charts

	ACTIVITY	Nov 12	Jan 13	Feb 13	Mar 13	Apr 13	May 13	Jun 13	Jul 13
1	Preliminary study								
2	Literature Review								
3	User requirements gathering								
4	Proposal Writing(Milestone 1)								
5	Analysis and first round design								
6	Evaluation testing(milestone 2)								
7	Second round design								
8	Result analysis								
9	Thesis writeup and defence(mil 3)								

Milestone One

Milestone Two

Milestone Three