UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

MULTI-AGENT BASED SYSTEM FOR REGISTRATION OF DEATHS IN KENYA

BY
KIVINDA, JUSTINA M.

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN COMPUTER SCIENCE OF THE UNIVERSITY OF NAIROBI

AUGUST 2012

Declaration

This report is my original work and has not been presented for any other university award.

Signature : _____ Date : 01/10/2012

Name : Justina Mwikali Kivinda

Reg. No. P58/73083/2009

Supervised by:

This report has been submitted as part of fulfillment of the requirements for the Master of Science in Computer Science with my approval as the University Supervisor.

Signature: _____ Date: 1/10/2012

Dr Elisha T. Opiyo Omulo

School of Computing and Informatics

Dedication

I dedicate this project to my beloved husband David Musyoka and my lovely children Tony, Ann, Mark and my niece Nancy for their support throughout the study period. God bless you.

TABLE OF CONTENTS

Abstract

Registration of deaths as a vital event is not effective in Kenya. The master register of dead persons is not up to date and retrieval of the records is tedious. Many deaths go unreported despite there being governing regulations and procedures for notification and registration of deaths occurring in the country and of Kenyan citizens living in other countries. This is attributed to poor methods and procedures employed in carrying out this process. The problem is compounded by lack of an effective computerized system for managing the registration process. Registration of deaths is a mandatory legal requirement in Kenya and in many countries worldwide. Besides registering to fulfill the legal requirement, a death record has numerous benefits to individuals, society, businesses and international organizations particularly WHO. Additionally, civil registration is one of UN recognized human rights. This study aimed to examine the process of civil registration of deaths and the shortcomings in the current system of registration of deaths in Kenya. From the shortcomings, bearing in mind the importance and benefits of registration of a death event, the study sought to provide a technological solution that is better than the current system.

Accuracy and timeliness are the major challenges in civil registration of deaths. The providers and the users of data are varied and are distributed over wide geographical regions in the country and abroad. Multiagent system technology was identified as the most appropriate to address these challenges. Agents are autonomous, proactive and are able to interact with other agents in the network through agent communication language (ACL). It involved requirement analysis, design and implementation of the system using the Prometheus MaS methodology. In requirements analysis stage, system goals were identified as well as percepts and actions. Architectural design involved determination of agent types and definition of their interactions. Detailed design involved design of the agent internals. The implementation of the system was accomplished using various tools including mysql database management system, Java eclipse IDE and apache- php web server for the web site interface. The implemented system was able to ensure that all captured data was complete as much as possible. The cause of death and age of deceased at time of death are vital parameter in determining the usefulness of vital statistics. The availability of data was assured by way of subscription by interested users. Corrections on erroneous entries were simplified. A search for existing records was made available in the

website. Vital statistics can be computed any time and key fields are available for trend analysis. They system is scalable; it can be adapted to other vital registrations such as registration of births, marriages and adoptions. The agents can be distributed in separate hardware to take advantage of more computing resources.

Acknowledgment

Foremost I would like to acknowledge the support and guidance of my supervisor Dr Elisha T. Opiyo Omulo

I also appreciate the support of the staff at the Department of Civil Registration, particularly for providing insights in to the process of civil registration.

Much appreciation to my colleagues Penina Limo, W. Barasa and James Yogo for their valuable ideas and guidance.

Abbreviations

AI	ARTIFICIAL INTELLIGENCE
ACL	AGENT COMMUNICATION LANGUAGE
DF	DIRECTORY FACILITATOR
FBO	FAITH-BASED ORGANIZATION
FIPA	FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS
IEBC	INDEPENDENT ELECTORAL AND BOUNDARIES COMMISSION
ECK	ELECTORAL COMMISSION OF KENYA
JADE	JAVA AGENT DEVELOPMENT FRAMEWORK
KNBS	KENYA NATIONAL BUREAU OF STATISTICS
KQML	KNOWLEDGE QUERY AND MANIPULATION LANGUAGE
KRA	KENYA REVENUE AUTHORITY
MAS	MULTI AGENT SYSTEM
MIRP	MINISTRY OF IMMIGRATION AND REGISTRATION OF PERSONS
NGO	NON-GOVERNMENTAL ORGANIZATION
NHIF	NATIONAL HOSPITAL INSURANCE FUND
NSSF	NATIONAL SOCIAL SECURITY FUND
SOAP	SIMPLE OBJECT ACCESS PROTOCOL
TSC	TEACHERS SERVICE COMMISSION
WSDL	WEB SERVICES DESCRIPTION LANGUAGE
WSIG	WEB SERVICES INTEGRATION GATEWAY

Definition of Terms

Actors	are the users of the system, and the external systems the software is interacting with.
Agent	an agent is a computer system that is capable of independent action on behalf of its user or owner.
Jade	(java agent development framework) is a software framework to make easier the development of agent applications in compliance with the fipa specifications for interoperable intelligent multi-agent systems.
Directory facilitator	according to the fipa architecture, this is the agent that provides The yellow-page service.
Death	the permanent disappearance of all evidence of life at any time after live birth has taken place (post-natal cessation of vital functions without capability of resuscitation).
Multi-agent system	a multiagent system is one that consists of a number of agents, which Interact with one-another through cooperation, negotiation and coordination.
Registrar	a person appointed to register births and deaths in any area or to register births and deaths occurring outside kenya, and includes a deputy registrar; under cap 149.
Soap	simple object access protocol, is a protocol specification for exchanging structured information in the implementation of web services in computer networks.
Use case	is a view of the functionality (or use) of a system from the user's perspective.
Vital registration	all sanctioned modes of registering individuals and reporting on vital events. These modes can include registration activities through complementary systems that are not done as part of the civil formal registration system and do not produce legal birth or death certificates.
Vital statistics	summary measures of vital events drawn from all of sources of vital events data. Particularly in developing country settings, where civil

registration functions poorly or not at all, the un acknowledges that many data sources and systems are used to derive estimates of vital statistics

Vital statistics system: as defined by the un is the total process of (1) obtaining information by civil registration or enumeration on the frequency or occurrence of specified and defined vital events, and relevant characteristics of the events themselves; and (2) of compiling, processing, analysing, evaluating, presenting, and disseminating these data in statistical form.

Web services integration gateway is a java add-on that provides support for invocation of jade agent services from web service clients.

List of tables

List of figures

CHAPTER1 INTRODUCTION

1.0 Background

An updated database of deceased persons has wide statistical applications in business, social issues and government institutions. The statistical data extracted from the database of deceased persons finds application in discovering increased child mortality rate , inheritance claims, compensation and cleaning up of the electoral rolls and many other uses. Many organizations dealing with registration of persons such as NHIF, NSSF, KRA, TSC, IEBC, insurance companies and financial institutions capture data that is relevant to their operations and hardly create interfaces to other stakeholder databases. They constantly add new clients and remove dormant ones in order to keep their databases up to date. More importantly, they purge dead clients so as to prevent possible fraudulent use of their accounts. Identifying dead persons in their databases is difficulty as a result of non availability of the records of deceased persons in usable format. Many of them learn of the deaths of their clients when premiums are not remitted for a long time or when matured policies are not claimed. They also learn of deaths when clients fail to make transactions in the accounts for a long time. The Department of Civil Registration is mandated to register and preserve data on deceased persons. However, it has not been able to produce usable records in a timely manner to meet the needs of users due to a number of factors. First, there is under coverage of registration as a result captured data is not representative enough to make meaningful statistical conclusions. Secondly, availability of captured records is tedious because the databases are not dynamic. Data is kept in excel templates that do not support complex queries. Efforts to computerize the registration process have been hindered by lack of expertise by staff and inadequate budgetary allocation that can support a roll out the system and sustain intensive training of registrars. Efforts have further been complicated by NGOs who fund temporary surveys to monitor the impact of their projects and do nothing to improve the civil registration system. The UN through the Department of Economic and Social Affairs provides guidelines for computerization of civil registration systems through its periodic publications. It recognizes the resource inadequacies and advises countries to consider phased out implementation (UN,1998) It further emphasizes that civil registration is the only long term comprehensive method of obtaining data on deceased persons(UN , 2001) from which useful vital statistics can be generated. In this study, multi-agent based software technology was used to implement the multi-agent based civil registration system.

1.1 Problem statement

The process of civil registration of deaths in Kenya is semi automated. Death records captured by the assistant registrars on manual forms are transported periodically to the district offices and eventually to the headquarters for verification. This takes a long time for timely consolidation and generation of statistics. There are inaccuracies in data capture, as a result of using many administrative forms for processing the registration and as a result of non availability of the registration materials. Due to the semi automated nature of the system, corrections on erroneous records is tedious and time consuming .Statistical summaries are done in excel sheets. The statistics generated lack usefulness due to non capture of important parameters such as cause of death and age of deceased at the time of death. Attempts to computerize the civil registration system have not been successful largely due to lack of expertise, lack of government and international support and the inherent complex nature of the civil registration system. This study has been conducted to provide a technological solution to the problem.

1.2 Objectives of the study

1. Review multiagent system technology
2. Formulate systems requirements for a system for registration of deaths.
3. Design a multi-agent based system for registration of deceased persons.
4. Implement a multi-agent based system for registration of deaths that will increase timeliness and accuracy of information.
5. Test the multiagent based system for registration of deaths in Kenya

1.3 Significance of the study

This study will benefit many planners and individuals who wish to use the data for deceased persons for various purposes such as health planning and population growth forecasting. The national bureau of statistics will be able to generate vital statistics and population changes much faster and more accurately since updates on new death cases will be available. The electoral process will benefit from increased accuracy of the voters register as the data for deceased persons will be readily available and with the vital details of the deceased persons. The study will also minimize fraudulent claims of inheritance and administration of estates for deceased persons more so in this era of HIV/AIDs as the deceased records will be searchable all the time without having to travel to the registrars' offices.

1.4 Assumptions and delimitations of the study

This study assumes that there are computers and internet or mobile connectivity in all registration centers. It also assumes that subscribers have either mobile phones or email addresses. This study includes civil registration of deaths only. The study does not include registration of marriages, births, judicial separations adoptions, annulments and legitimations.

1.5 Expected outcomes

One of the major aims of this study is to provide quality deaths data through use of multiagent systems technology. We wish to demonstrate that MaS technology can make complex process look simple and are transparent to the suppliers and consumers of data.

a) A new multi-agent based system for registration of deaths which is scalable to other aspects of civil registration.
b) The reporting tool will provide statistical data in various formats as desired.
c) User friendly interface

1.6 Chapter summary

Deceased persons' data has wide applications and numerous benefits to individuals, society, government, the United Nations and NGOs. The Despite the numerous advantages of civil registration of deaths, the data is not easily available as records are either inaccurate or have not been updated in a timely manner. The government and international community have not directed enough resources to this area and as result it has stagnated. In this era of advanced systems technology, the Department of Civil Registration rates poorly. Attempts to computerise have not been successful. This is attributed to numerous impediments ranging from lack of expertise by staff, lack of prioritization of civil registration tasks and huge resources required to implement the system and train staff. Multi-agent systems technology will be used to conceptualise design and implement a civil registration system. The study includes registration of deaths only.

CHAPTER 2 LITERATURE REVIEW

2.0 Civil registration

Registration of persons for occurrence of a vital event is generally referred to as civil registration (KNBS, 2010). According to the UN, a vital event is the occurrence of a live birth, death, foetal death (defined as the death of a foetus before birth or extraction from its mother, irrespective of the duration of pregnancy), marriage, divorce, adoption, legitimation, recognition of parenthood, annulment of marriage, or legal separation(UN, 2001). ''Vital registration'' is sometimes used interchangeably with ''civil registration''. According to United Nations Department of Economic and Social Affairs, civil registration is the continuous permanent, compulsory recording of the occurrence and characteristics of vital events and as provided through decree of regulation in accordance with the legal requirement in each country (UN, 1998). It covers all vital events occurring in each geographical area and all population groups.

Not all countries register the above events. In Kenya, only live births and death events are registered, albeit with serious under coverage. Many people in Asia and Africa are born and die without being recorded in any legal document. This contravenes UN proclaimed right to a recorded name and nationality. Each year, nearly 50 million newborn children are not registered, (United Nations Children's Fund, 2005). Barely a third of countries outside North America and Europe have the capacity to obtain usable mortality statistics, and half the countries in Africa and Southeast Asia record no cause of death data at all (Mathers C, et al ,2005). The root cause of this is stagnation in maintenance of civil records in the last 3 decades largely due to:

i. Lack of prioritization of civil registration systems by governments. As a result, governments have not set aside adequate budgets to improve vital statistics systems. The data collected lacks completeness especially cause-of-death.
ii. Development partners failure to recognise civil registration systems as key components of development infrastructure. It is therefore difficulty to assess the impact of aid funding on poverty and mortality.
iii. Lack of international mandate to strengthen civil registration systems. Instead international mandate has focused on the vital statistics instead of the civil registration system that generates the vital statistics. The body mandated with vital statics, the Statistical Commission, has no authority and resources to enforce improvement of civil

registration systems. It assists countries to improve civil registration systems by periodically publishing principals and recommendations through its Statistics Division.

Apart from civil registration, complementary sources of statistics, although not recommended, are allowed as alternative source of data particularly to capture mortality and cause of death. The three principal alternatives are population censuses, household sample surveys and sample registration. These alternatives have inherent limitations which emanate from assumptions and approximations of the demographic relationships between various characteristics of the population (UN, 2001).

Civil registration data is the main source of vital statistics. In order to generate useful and sensible statistics, the deaths records have to be complete, accurate and captured in a timely manner. The United Nations has developed an assessment framework for vital statistics generated from civil registration system to guide countries on recommended parameters of a good registration system. The cause of death should be recorded to assist countries know the causes of death. For relevance, age of persons at the time of death is necessary for tabulations of 5 year age intervals. The data should also be accessible in a variety of data formats. In addition there should be quality metadata.

2.1. Purpose of registration of deaths

The purpose of registration of death is primarily the legal value of the documents obtained from the process. In Kenya, the act of registering a death provides an individual with a medical certificate ascertaining the cause of death especially in matters of inquest. A burial permit is the only legal document that permits burial, cremation or any other form of disposal. It is illegal to dispose a body without a permit, the penalty being a fine of kshs 500.00 or six months imprisonment or both (Births and Deaths Registration Act_ Cap 149, Revised2010).

2.2. Benefits of registration of deaths

2.2.1 Benefits to individuals

Civil registration provides relevant documents relating to a vital event. In case of a birth event, a birth registration generates a birth certificate. Registration of a death event is the basis for a death certificate. There two documents act as legal prove of birth and death respectively. A death certificate can be used as prove of legal relationship to a deceased parent thereby giving the children right of administration of property. Additionally a birth certificate of a child and the

parent's death certificate can be used to secure sponsorship for the orphaned children by NGOs. For example, a combination of a birth- and a parent's death certificate enables orphaned children in Uganda to be registered for sponsorships by NGOs and FBOs (Plan International, 2008) and in addition by local private companies who can provide orphaned children with university loans: "It is important for us that our orphans are part of scholarships...It helps us to look after them better" -Focus Group Discussion with Early Childhood Care Development caregivers, Uganda, August 2008. A death certificate is useful to individuals who need to claim pensions of their deceased spouses. It also grants surviving spouses right to remarry.

2.2.2 Benefits to the government and Society

Data on deaths is needed to assess the impact of policies at national and local level. The completeness of death data is needed to set priorities so as to formulate policies and monitor their impact. It is also useful in determining population changes. In Kenya, the Central Bureau of Statistics department carries out population census once in 10 years. Between the censuses, projections are done based on number of deaths and births. This is supplemented by surveys in selected areas for verification purposes. A good civil registration system will benefit the department by minimizing on the number of costly surveys as the statistics on deaths will be updated all the time and will be comprehensive. In South Africa, many adult deaths occurred in the early 1990s as a result of HIV/AIDs epidemic, but due to poor quality data on deaths, the government could not determine the cause of the massive deaths. Supplementary registration methods at local level helped to bring out the main cause of death which was identified to be as a result of HIV/AIDs infection. As a result the civil registration system has improved and cause of death is identifiable. The government recently increased the budget on HIV/ AIDs as a result of availability of quality data that could classify deaths related to HIV\AIDs.

2.2.3 Benefits to the international community.

International aid to fight specific diseases can only measure the impact of their efforts by getting complete and timely data on cause of death. Some of the millennium development goals (MDGs), specifically eradication of extreme poverty and hunger, achievement of universal primary education, attaining gender equality, reductions in child mortality, improvements in maternal health, and reductions in prevalence of HIV/AIDS, malaria, and other diseases can only measure their impact by having reliable source of statistical data. The only long term and consistent system to provide this data seems is civil registration system.

2.2.4 Benefits to the democratic process.

One of the major weaknesses of the electoral process in Kenya is non-removal of deceased persons from the register of electors thus giving false statistics on the number of eligible voters. In the 2007 general elections, the voters register had a total of 14,296,180 voters. Out of this, number, an estimated 1.7million voters had died since 1997, the year when the voters roll was initiated. ECK had only managed to delete 513,000 deceased persons. Therefore over 1.2 million dead people were still in the register. This was a major credibility loss for the then Electoral Commission of Kenya as the deceased records were used to manipulate the voter turnout. It also led to inflated voter numbers and subsequently high costs due to overestimation of materials for conducting the election (Kregler Commission, 2008) such as ballot papers.

The non- availability of accurate data on deceased voters hinders the process of cleaning up the voters' roll in a very significant way. It leads to removal of the wrong persons because in some instances, the name of the relative who reported the death to the assistant chief has been registered instead of that of the dead person. The death records also have insufficient information to facilitate successful removal of the deceased persons from the voters' roll. The deceased persons records that are not successfully removed from the voters register are a risk to the credibility of the voters register. In 2007, many fraudulent activities were committed using the records of the deceased voters. Foremost, they contributed to high cost of the election process as the amounts of resources mobilized were based on the number of voters. For example, number of ballot papers produced was based on the number of voters. Secondly, the number of electoral officials hired depended on the number of voting precincts. The number of precincts is calculated by groups of 1,000 voters. Lastly, it was reported that some election officials colluded with some politicians to inflate the voter turnout due to the fact that there were many records of persons who had not voted, including the deceased persons..

2.2.5 Benefits to financial institutions and insurance industry

Financial institutions hardly learn of deaths except by coincidence, nonpayment of loans, claims by dependants or matured unclaimed policies. Even when a report of death of a client gets to them, they want to be sure that it not a false claim. To ascertain the accuracy of the death reports, they have to cross check with assistant chiefs with involvement of their legal personnel for authenticity of the information. This is because the records at the headquarters are not updated quickly enough to give a one stop information source.

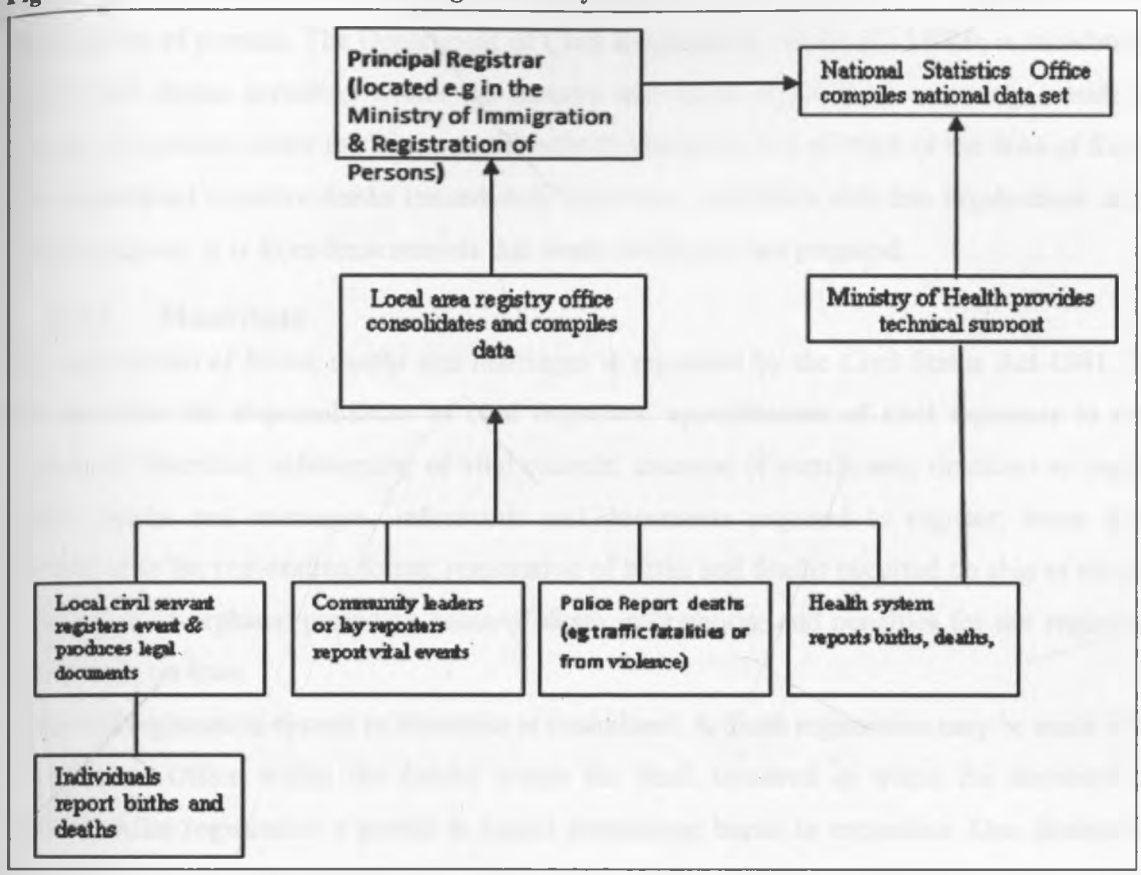
2.3. The process of registration of deaths in Kenya

The process of registration of deaths in Kenya is community based. Communities have the responsibility of notifying the provincial administration of deaths occurring at home. Upon notification, registration forms are completed by the assistant chiefs for onward transmission to the district registrar. The completed forms are the ones used to prepare burial permits by the district registrar. The provincial administration acts as registration assistants on behalf of the department of civil registration. Where deaths occur at the hospitals, the medical officers act as registration assistants and are responsible for completion of specific death registration forms and submitting them to the district registrar. The completed forms are submitted to the district registrar once per month or as may be directed incase of areas with high mortality rate like national hospitals.

Upon receiving the registration forms from hospitals and assistant chiefs, the district registrar verifies them and serializes them alphabetically for compilation of a deaths register for the district. A copy of the deaths register is subsequently transmitted to the headquarters for storage. The Death certificates are prepared from these registers and referenced with the serial number assigned on the register. Under certain circumstances, a registration of a death event may not occur immediately. This is referred to as a late registration. According to The Births and Deaths Registration Act'_Cap.149, late death means the "death of a person which has occurred in Kenya since 23rd January, 1906, the particulars whereof have not been registered in the register of deaths within six months of the date of such death" .In such cases the registrar has to obtain written authority from the Principal Registrar and the person reporting must pay the prescribed fees to process the registration.

The organizational structure of the Department of Civil Registration in Kenya is shown below (figure 1). At the lowest level, individuals report occurrence of a birth or death to the civil authorities and in turn receive legal documents (burial permits, birth notification). Police, health practitioners and local leaders also report events to the district registrar. The District Registrar's function is to authenticate and summarise data for onward transmission to higher authorities. The compiled and summarised data is maintained at the head office central registry.

Figure 1: General structure of civil registration systems.



2.4. Registration of deaths in other countries

The structure of civil registration system varies from country to country. The administrative responsibilities are spelled out in the in the legislations governing civil registration of each country. The source of the death records is usually the heath institutions, police, community leaders and individuals. Many countries have a statistics bureau responsible for generation of vital statistics from the civil registration with technical support provided by the Health system.

Besides the general functional structure of civil registration systems, a further classification can be derived from the governing structure where civil registration system is centralized or decentralized. Under centralized civil registration system, a national agency provides direction to local agencies and provides financial and technical support. The decentralized system is commonly used in countries with federal systems of government like Unites States.

The Kenyan model is the centralized one controlled by the Ministry of Immigration and Registration of persons. The Department of Civil Registration , under the MIRP, is mandated to register all deaths occurring within the country and those of Kenyans occurring outside the country. It operates under the Births and Deaths Registration Act, of 1928 of the laws of Kenya. The department registers deaths immediately they occur and deals with late registrations as per laid procedures. It is from these records that death certificates are prepared.

2.4.1 Mauritius

The registration of births, deaths and marriages is regulated by the Civil Status Act 1981. The Act specifies the responsibilities of civil registrars; appointments of civil registrars in other Islands of Mauritius; safekeeping of vital records; issuance of certificates; timelines to register births, deaths and marriages; informants and documents required to register; items to be completed in the registration forms; registration of births and deaths occurred on ship or aircraft; registration of orphans; providing cause-of-death information; and penalties for not registering vital events on time.

The civil registration system in Mauritius is centralized. A death registration may be made at the Civil Status Office within the district where the death occurred or where the deceased last resided. After registration a permit is issued authorizing burial or cremation. One declarant is required at time of the registration. Documents needed for the registration include medical certificate on the cause of death; birth certificate and marriage certificate of the deceased; and national identity card of the deceased. Death registration is considered to be late after 24 hours. To facilitate timely registration, the main registration office is open every day. As a result of these administrative measures, coverage registration of death is above 90%. Electronic registration began in 2001. One challenge facing the Central Statistics Office is that there are no ICT staff to assist the statisticians with computer problems.

2.4.2 Tanzania

The registration of Births and Deaths in Tanzania regulated through the Births and Deaths Registration Act (Chapter108, last revised in 2002). According to the Act, birth and death registration is compulsory for all persons living in Tanzania at the time that such event occurs. The Act specifies how officials should be appointed but does not include other details on the registration procedure such as the information to be registered and the preparation of a statistical reports. The structure of the registration system in Tanzania is centralized and is administered by

the Registration, Insolvency and Trusteeship Agency (RITA), which is an Executive Agency under the Attorney Generals Chambers in the Ministry of Justice and Constitutional Affairs. For death registration, information on the name, age, sex, residence, occupation, nationality (citizenship) of deceased is required. The date and place of death as well as the cause of death are also covered on the form (BD6). Deaths occurring in hospitals receive burial permits automatically while those that occur at home are processed by the Village Executive Officer or the District Registrar of Births and Deaths. Late registrations (after 30 days) require more documentary proof and attract a fee.

Issues affecting civil registration include low coverage of registration, about 46% in the urban areas and 15% in rural areas. The second problem is low manpower at the Statistics Office for data compilation. A third problem related to information sharing. There is no computer network to facilitate sharing of data.

2.5. Issues affecting registration of deaths in Kenya

The Department of Civil Registration is mandated to register all deaths of Kenyan citizens. Although it is mandatory to register deaths, the process is not effective due to a number of factors. First, there still low coverage of registration, at 38%. The low registration is attributed to several causes including lack of awareness about the importance of registration of deaths, lack of information on what is required when a death occurs, low knowledge of registration procedures by registrars and low interest in improving civil registration systems by the government and international donors. Secondly, the data collected on deceased persons is not complete. The death records lack vital information to use them for analysis and classification. For instance where the age of deceased is unavailable or is estimated, analysis of such a record by age specific death rate is not possible. The cause of death is also very crucial for heath planning because heath policies rely on such data to mobilize appropriate resources. To assess the impact of economic and health policies, mortality data with complete cause of death, age and residential details of a deceased person are important. Thirdly, the civil registration process is manual. There is currently no computerized system for managing the registration process and generating vital statistics. Statistics are summarized using Microsoft Excel templates. In some instances, the Kenya National Bureau of Statistics has to rely on other methods to obtain vital statistics because the civil registration system does not provide records in electronic from which statistics can be extracted.

2.6.Computerization of Civil Registration Systems

Computer technology can facilitate many functions of the civil registration system including data capture, analysis, retrieval and edits. In many countries, civil registration system is not well computerized. It is the process of civil registration that generates vital statistics that a country requires to make critical policy decisions and evaluate the impact of the decisions.

In the last three decades, efforts to streamline civil registration have borne minimal improvements. This is evident from the statistics that the UN gets from its member countries on a yearly basis. There are some member countries that have been consistently submitting yearly statistics, yet there are others who have not reported for over 40 years. Table 1: below shows the percentage of population living in countries with complete civil registration system, by WHO region. Even with regular reporting, other challenges are encountered in analyzing the usefulness of data. Some death statistics lack cause of death. Other causes of death are coded poorly resulting in mismatch of death and corresponding code hence poor quality data. Figure 2 shows the percentage of ill-defined causes of death in selected countries between 1950 and 1998.

Table 1: Percentage of population living in countries with complete civil registration system, by WHO region

	Deaths				Births			
	1965-74	1975-84	1985-94	1995-2004	1965-74	1975-84	1985-94	1995-2004
Total	27%	25%	28%	26%	33%	31%	28%	30%
Africa	2%	4%	2%	7%	7%	7%	9%	5%
Americas	69%	66%	64%	61%	58%	55%	53%	53%
Eastern Mediterranean	17%	21%	15%	1%	21%	25%	17%	42%
Europe	62%	61%	92%	86%	95%	94%	93%	92%
South-East Asia	1%	1%	1%	1%	1%	1%	1%	1%
Western Pacific	12%	11%	10%	13%	12%	14%	13%	18%

Complete means that at least 90% of events (births or deaths) are registered by the system. Figures for 1965—94 from the Demographic Yearbook (historical supplement 1948-1997), UN Statistis Division, New York, 2000. Figures for 1995-2004 are based on the Demographic Yearbook 2004, UN Statistics Division, New York.

Figure:2 Trends in percentage of deaths assigned ill-defined codes

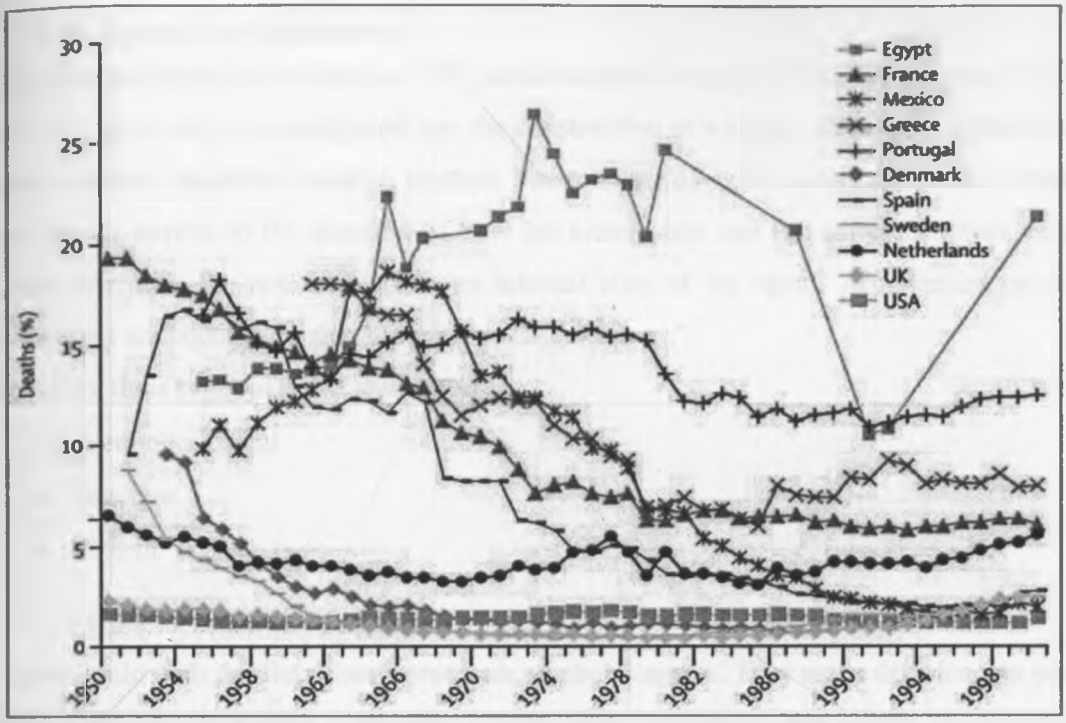


***Figure:* Trends in percentage of deaths assigned ill-defined codes according to the ICD classification in selected countries, 1950–2000**

2.7.Agents

Definition: an agent is a hardware or (more usually) software-based computer system that enjoys the following characteristics:

i. Autonomy: acting independently in an environment.

ii. Social ability: Agents have the ability to interact with other agents through Agent Communication Language, the popular ACL being KQML and FIPA.

iii. Mobility: ability to move around electronic network.

iv. Pro-activiteness: not just being driven by events but recognizing opportunities and talking the initiative.

v. Veracity: avoid communicating false information knowingly

vi. Benevolence: always try to do what is asked.

vii. Rationality: agent will act in order to achieve its goals, and will not act in such a way as to prevent its goals being achieved — at least insofar as its beliefs

viii. Learning/adaptability: agents improve performance over time.

ix. Personality: has distinct personality, behavior, name and role.

2.8. Agent Architectures

Maes defines agent architecture as: '[A] particular methodology for building [agents]. It specifies how 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. Architecture encompasses techniques and algorithms that support this methodology.'

There are three types of agent architectures:

- symbolic/logical
- reactive
- hybrid

2.8.1 Symbolic agents

Agents build with Artificial Intelligence are symbolic agents. They make decisions on what to do using symbolic reasoning. Originally, (1956-1985) all agents build with AI were symbolic reasoning agents. They were supposed to use pure logical reasoning to decide what to do. They were however difficulty to build which led to emergence of reactive agents from 1985 to present. Some key issues with symbolic agents include:

The transduction problem: that of translating the real world into an accurate, adequate symbolic description, in time for that description to be useful.

The representation/reasoning problem: that of how to symbolically represent information about complex real-world entities and processes, and how to get agents to reason with this information in time for the results to be useful.

2.8.2 Reactive agents

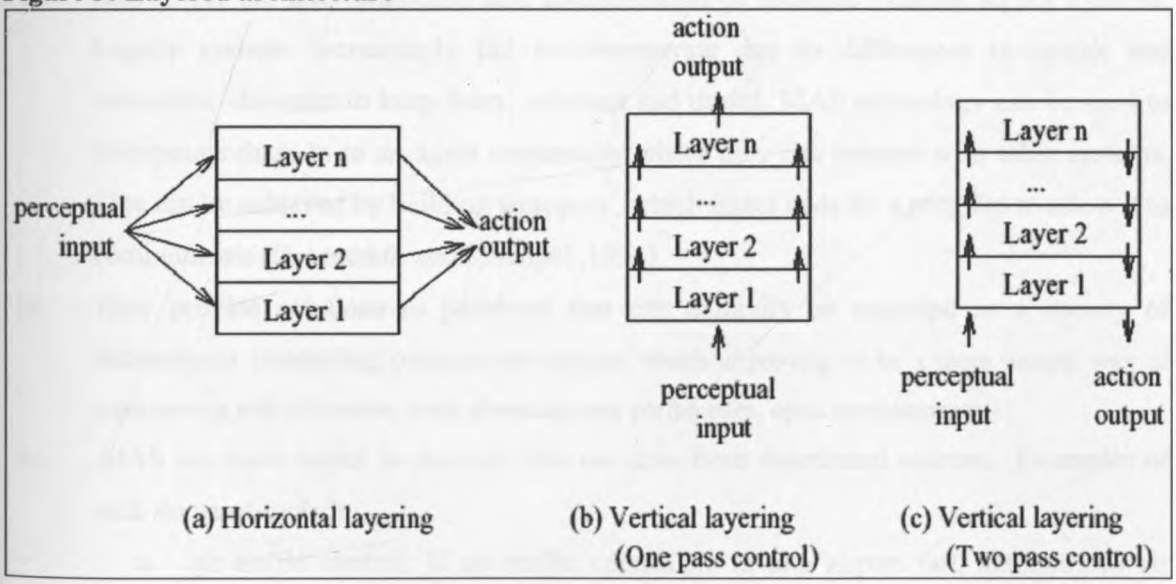
Reactive architectures base their decision making entirely on the present without any reference to the past. They implement a direct mapping of situation to action and are based on a stimulus–response mechanism triggered by sensor data. One of the critics of main stream AI on which symbolic agents are based was Rodney Brooks who argued that intelligent behavior can be generated without explicit representations of the kind that symbolic AI proposes and that intelligent behavior can be generated without explicit abstract reasoning of the kind that

symbolic AI proposes. He also put forward a third thesis that Intelligence is an emergent property of certain complex systems (Brooks, 1991). The advantage of reactive agents is their simplicity to design unlike symbolic agents. This also means they are economical to design therefore robustness can be achieved faster.

2.8.3 Layered (Hybrid) architecture

Many researchers have argued that neither a completely deliberative nor completely reactive approach is suitable for building agents. They have suggested using hybrid systems, which attempt to marry classical and alternative approaches. An obvious approach is to build an agent out of two (or more) subsystems; deliberative one, containing a symbolic world model, which develops plans and makes decisions in the way proposed by symbolic AI. Secondly, a reactive one, which is capable of reacting to events without complex reasoning (Michael Wooldridge M, Wiley. J , 2002). This led to the idea of layered architecture. With layered architecture, an agents control subsystem are arranged into a hierarchy with higher layers dealing with information at increasing levels of abstraction. The layering can be horizontal or vertical as shown in figure 3 below.

Figure 3: Layered architecture



2.9.Multi -Agent based systems technology

Agent-based systems technology is a new paradigm for conceptualizing, designing, and implementing software systems. It has become an attractive technology for building software systems for environments that are distributed, open and complex. Initially agent systems were build with one agent. As complexity of systems increased and agent technology matured incorporation of several agents has been adopted to counter the limitations of a single agent namely, limitation of computational resources and risk of failure.

A MAS is a system of agents which interact with one another through cooperation, competition, coordination or negotiation (Wooldridge M, 2002). In most cases, agents act on behalf of the user with different goals and motivation. To successfully interact, they will require the ability to cooperate, coordinate, and negotiate with each other, much as people do. Compared to a single agent or a centralized system, MAS has several advantages which include the following:

i. Ability to solve problems that are too large for a single agent. Implementation of many agents increases robustness and reliability. If one agent fails, another one can take over.
ii. Allows for the interconnection and interoperation of multiple existing legacy systems. Legacy systems increasingly fail to interoperate due to differences in syntax and semantics. In order to keep them relevant and useful, MAS technology can be used to incorporate them in to an agent community where they can interact with other systems. This can be achieved by building wrappers , which inject code in a program to allow it to communicate (Genesereth and Ketchpel ,1994).
iii. They provide solutions to problems that can naturally be regarded as a society of autonomous interacting components-agents; which is proving to be a more natural way of representing task allocation, team planning, user preferences, open environments.
iv. MAS are more useful in domains that use data from distributed sources. Examples of such domains include:
    a. air traffic control: If air traffic controllers in one airport fail, autonomous air traffic control systems in nearby airports recognize the failure and take over.
    b. Internet agents: search agents look for answers to a search query from different information sources.

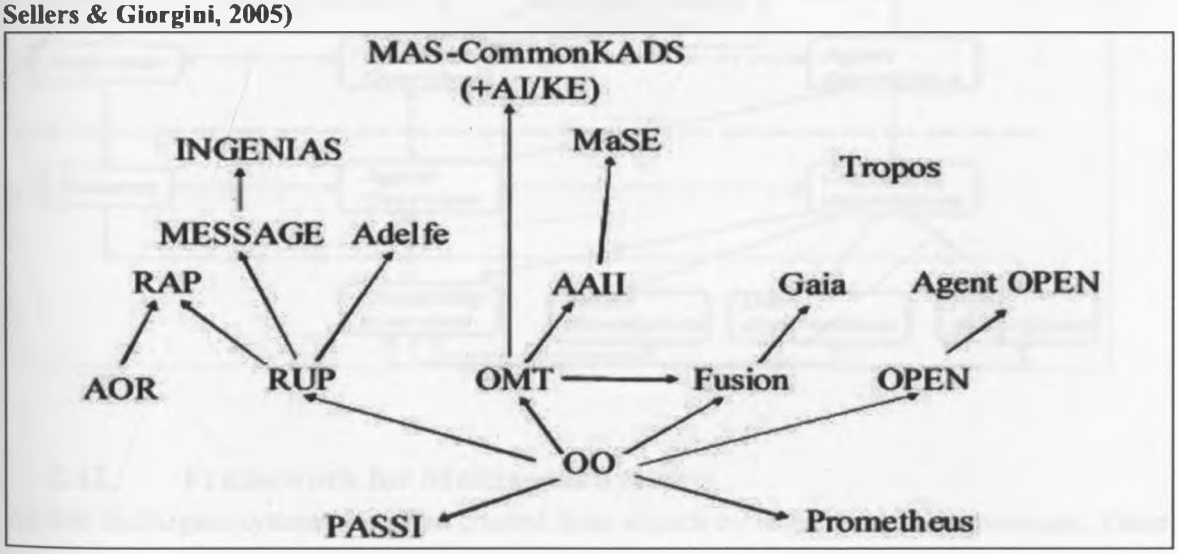
v. to provide solutions in situations where expertise is distributed. Examples of such problems include concurrent engineering (Lewis and Sycara 1993), health care, education and manufacturing.

vi. Finally MAS lead to realization of speed and efficiency. Efficiency is achieved through concurrency of computation.

2.10. MAS methodologies

A methodology refers to a procedure for attaining something. In MAS, an agent building methodology must meet certain criteria. It must be agent oriented, in the sense that it must be geared towards creation of agent software. It also must focus on MAS, an organised society of agents playing roles within an environment. A MAS Methodology generally encompasses a set of concepts, notations for modeling aspects of the software and processes that follow in order to build the software. The agent-oriented methodologies have multiple roots. Some are based on AI originating from (KE). Others are inspired by software engineering and they are extensions of object-oriented (OO) paradigm. Yet others are derived from a mix of other agent -oriented methodologies. Figure 4 below shows these origins.

Figure 4: Influences of Object Oriented Methodologies on Agent Oriented Methods (Henderson-Sellers & Giorgini, 2005)



2.11. Prometheus methodology

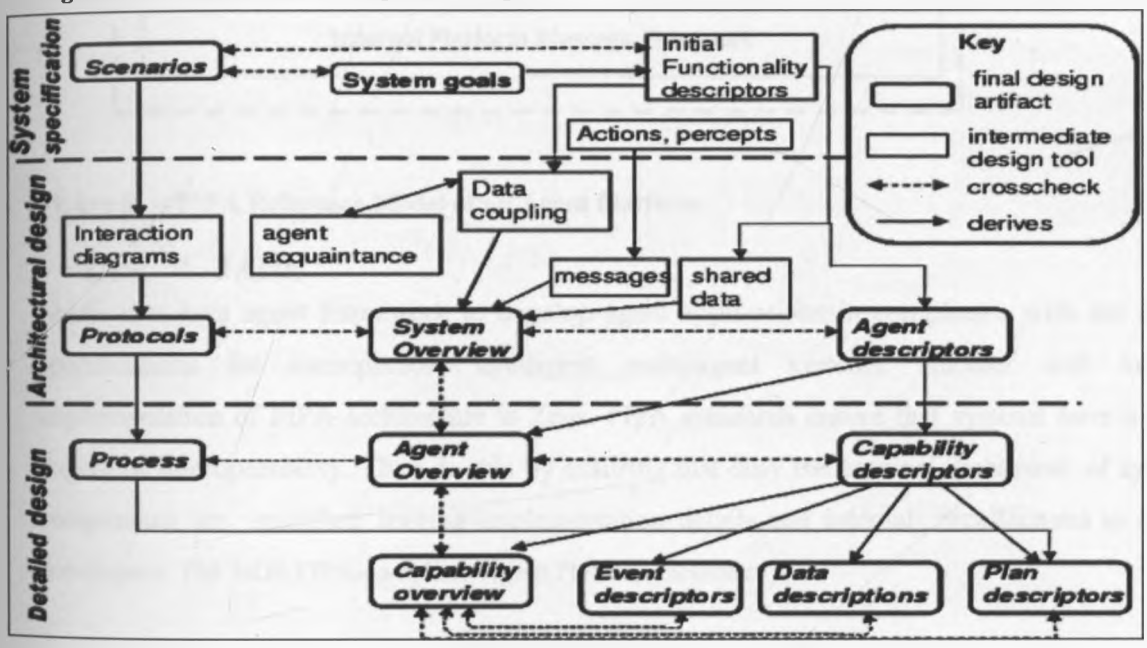
The Prometheus methodology is a detailed process for specifying, designing, and implementing intelligent agent systems which we have been developing over the last several years, and is supported by several tools in the stages of development and deployment. Its hierarchical structuring can scale to large designs easily. It involves three phases:

1. System specification phase
2. Architectural design phase and
3. Detailed design phase.

In the first phase, system specification phase, the focus is on the systems requirement. The system is specified using goals and scenarios. The interface to the environment is specified in terms of actions, percepts and external data.

The architectural design phase utilizes the products of the first phase to develop the system architecture. The agent types are identified and scenarios are developed into interaction protocols. In the third phase, detailed design, the details of each agent's internals are developed.

Figure 5: Prometheus Development Stages.



2.12. Framework for Multiagent Systems

Ad hoc multiagent systems are often created from scratch by researchers and developers. There is however a framework that has arisen that implements standard reference architecture for MAS known as FIPA (*jade.tilab.com*). A Framework for agent systems saves developers time and also helps in standardization of MaS development. In fact, advances in agent technology depend on improving the frameworks. In this study, the FIPA framework will be used to implement the multiagent based system for reporting of deaths in Kenya. The FIPA standard specifies the agent

abstract architecture, agent management system, agent message transportation and agent communication

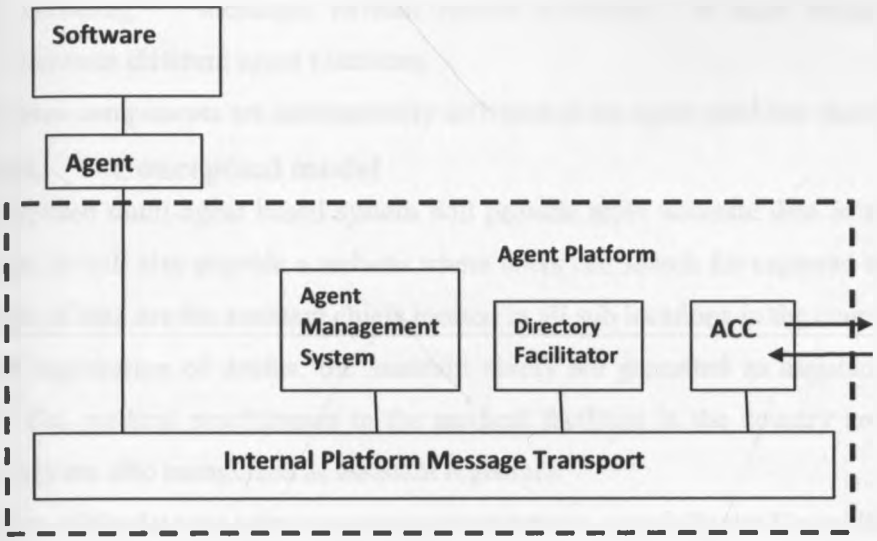


Figure 6: – FIPA Reference Model of an Agent Platform

2.13. JADE

JADE is a Java agent framework to develop agent applications in compliance with the FIPA specifications for interoperable intelligent multi-agent systems. Another well known implementation of FIPA architecture is Zeus. FIPA standards ensure that systems have a high degree of interoperability. They do this by ensuring that only the external behaviour of system components are specified, leaving implementation details and internal architectures to agent developers. The JaDE FIPA-compliant Agent Platform includes:-

1. AMS (Agent Management System). This is the agent that exerts supervisory control over access to and use of the platform; it is responsible for authentication of resident agents and control of registrations.

2. The DF (Directory Facilitator), the agent that provides a yellow page service to the agent platform.

3. The Message Transport System, also called Agent Communication Channel (ACC), is the software component controlling all the exchange of messages within the platform, including messages to/from remote platforms. It must support interoperability between different agent platforms.

These three components are automatically activated at the agent platform start-up.

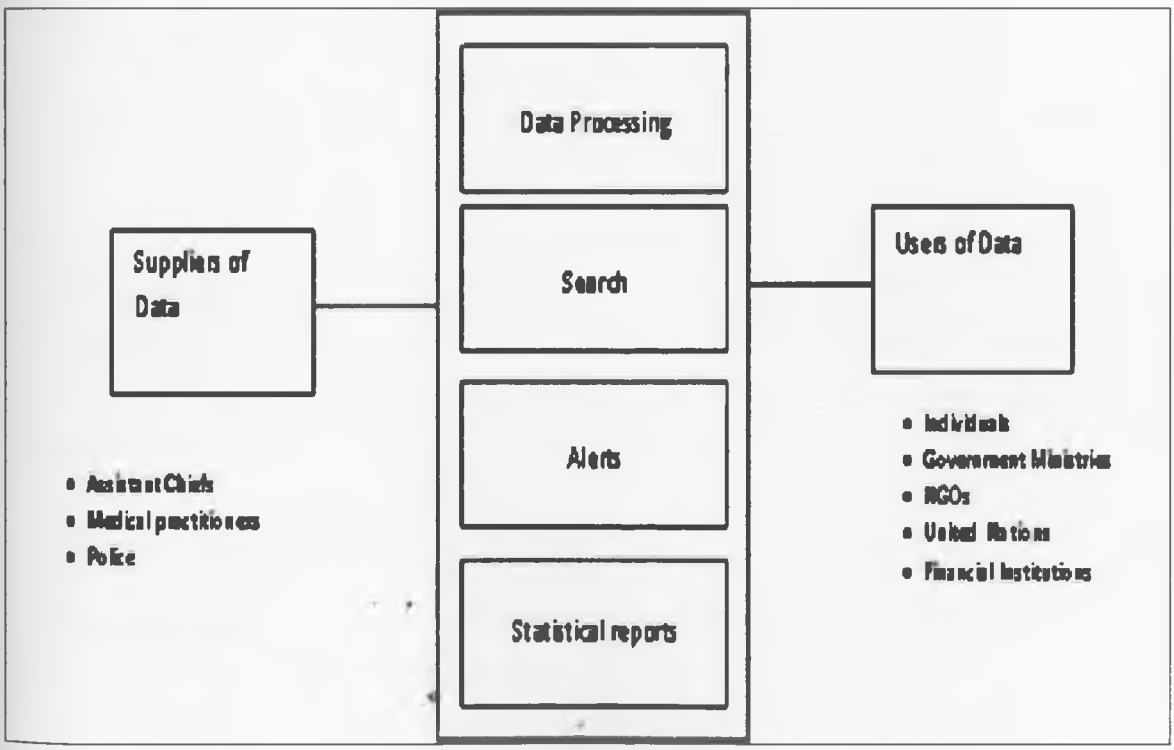
2.14. Conceptual model

The proposed multi-agent based system will provide more accurate data in a timely manner to the users. It will also provide a website where users can search for captured death records. The suppliers of data are the assistant chiefs located in all sub locations in the county. For the purpose of civil registration of deaths, the assistant chiefs are gazzetted as assistant registrars by the MIRP. The medical practitioners in the medical facilities in the country are also suppliers of data. They are also recognized as assistant registrars.

The users of the data are other government institutions especially the Kenya Bureau of Statistics, individuals, Financial institutions and international organizations especially the United nations.

Multiagent technology is best suited for providing a solution to the challenges facing the civil registration of deaths.

Figure 7: Conceptual model.



2.15. Chapter summary

The chapter has examined the purpose and benefits of civil registration of deaths to individuals, society and government as well as NGOs. It has also looked at the process of registration of death event in Kenya and other countries. The purpose and process for registration of deaths is similar in many countries, yet the level of coverage differs. Low coverage is directly related to the administration of the process. Effective computerization of civil registration system in Kenya has not been successful. The government has approached the process from the point of computerizing all aspects of civil registration making it costly. In this chapter multi-agent based systems development technology has been studied to justify it as a powerful technology for implementing systems with distributed sources of data such as the civil registration system.

CHAPTER 3 METHODOLOGY

3.0 Introduction

In order to achieve the overall objectives set out in chapter one, a number of activities were carried out as outlined below:

i. Review of MaS technology.
ii. System specification
iii. System design
iv. System implementation
v. Testing

3.1 Literature review

Extensive literature review of the civil registration system for deaths was carried out with a view to gain thorough understanding of the underlying needs, challenges and desired operation level of civil registration and vital statistics systems. This review guided the process of compiling the requirements of the system. A study on multi-agent systems technology was also conducted to justify its appropriateness in providing a solution to problems in civil registration system. The study identified the use of Prometheus methodology to specify, design and implement the system.

3.2 Prometheus methodology.

Prometheus methodology was used in the system specification, system design and implementation. It is preferred over other MaS methodologies in this study due to the fact that it is a complete methodology (from start to end). It has detailed and elaborate process for system specification, implementation, testing and debugging. It is also supported by Prometheus Design Tool that allows the user to design overview diagrams. Further, it has examples which help to better understand what is required in each stage of development. It has three phases which were implemented as described below in to accomplish activities set out in 3.0 (ii) to (v).

3.2.1 System specification

This is the first step in Prometheus methodology. The following activities were undertaken.

a) Identification of system goals and sub-goals because proactiveness is desired. What is the system being build for? What are the subsidiary goals?

b) Develop use case scenarios. Scenarios show a particular instance of a system. There is no branching. Scenarios consist of a sequence of steps which are themselves scenarios.
c) Identification of the agent system's interface to the environment in terms of actions, percepts, and external data.
d) Identify functionalities.
e) Identify data read and written by functionalities
f) Prepare functionality schemas (name, description, actions, percepts, data used/produced, interaction (with other functionalities), and goals)

3.2.2 System design

In this phase, the system specification artefacts were be used to build the system architecture. The following steps were followed:

a) Specify agent types.
b) Define agent types and develop agent descriptors
c) Produce a system level overview diagram describing the overall structure of the system
d) Develop interaction protocols from use case scenarios (via interaction diagrams)

3.2.3 System implementation

During the Detailed Design phase agent overview diagram was produced and the agent capabilities and interactions produced

3.3.1 Data collection process

Data collection was done to understand the civil registration of deaths and obtain copies of statutory documents used in the data capture.

3.3.2 Data collection Tools

Primary data was collected through face to face interview-Annex II. Secondary data was obtained from reports and statistical publications by KNBS and UN Yearbook and from the departments website.

3.3 Chapter summary

Extensive literature review was undertaken to gain understanding of the process and challenges of civil registration in Kenya. A study of technological solutions lead to the adoption of MAS technology. The process of registration of deaths has distributed sources of data and the users are varied. Data collection was carried out to understand the requirements of the system.

CHAPTER 4: ANALYSIS, DESIGN AND IMPLEMENATION

4.0 SYSTEM SPECIFICATION

This is the initial phase of Prometheus methodology for building MAS. The phase involved specifying the system requirements using goals and scenarios and identifying the actions and percepts. Use case scenarios were developed together with the agents interface to the environment. .

Records of dead persons in Kenya is required for health planning, inheritance issues and population statistics. The goals of the system and use cases were used to analyse the requirements of the system.

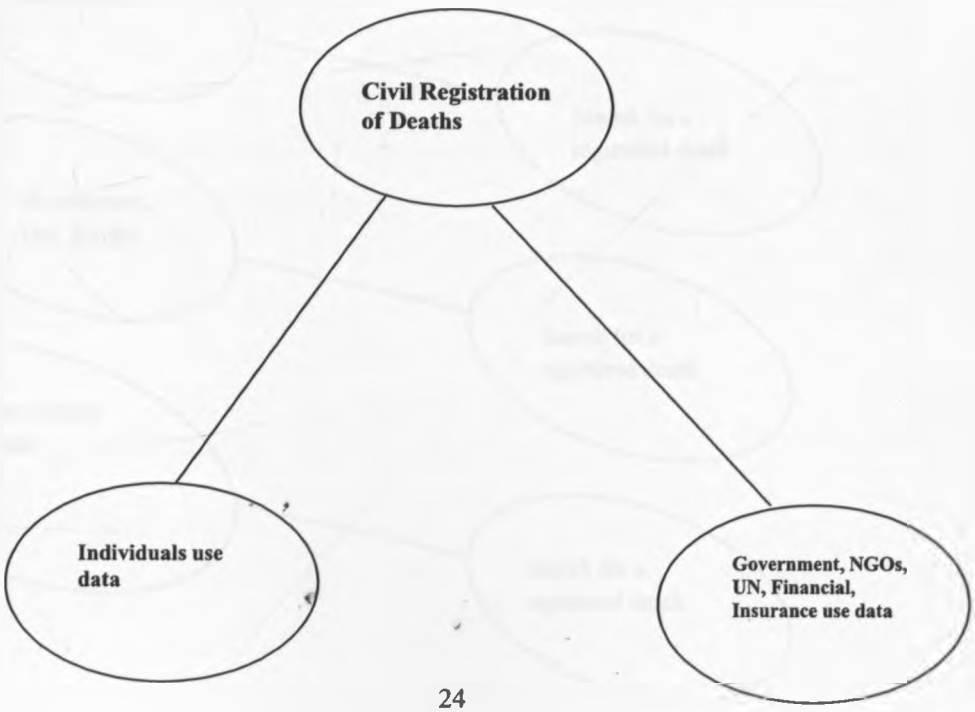
4.1 System Goals

What is the system build for?

Overall Goal

The overall goal of the system is to capture deaths as soon as they are reported by the handlers of the deceased, in a timely and accurate way and make the information available to stakeholders and interested individuals.

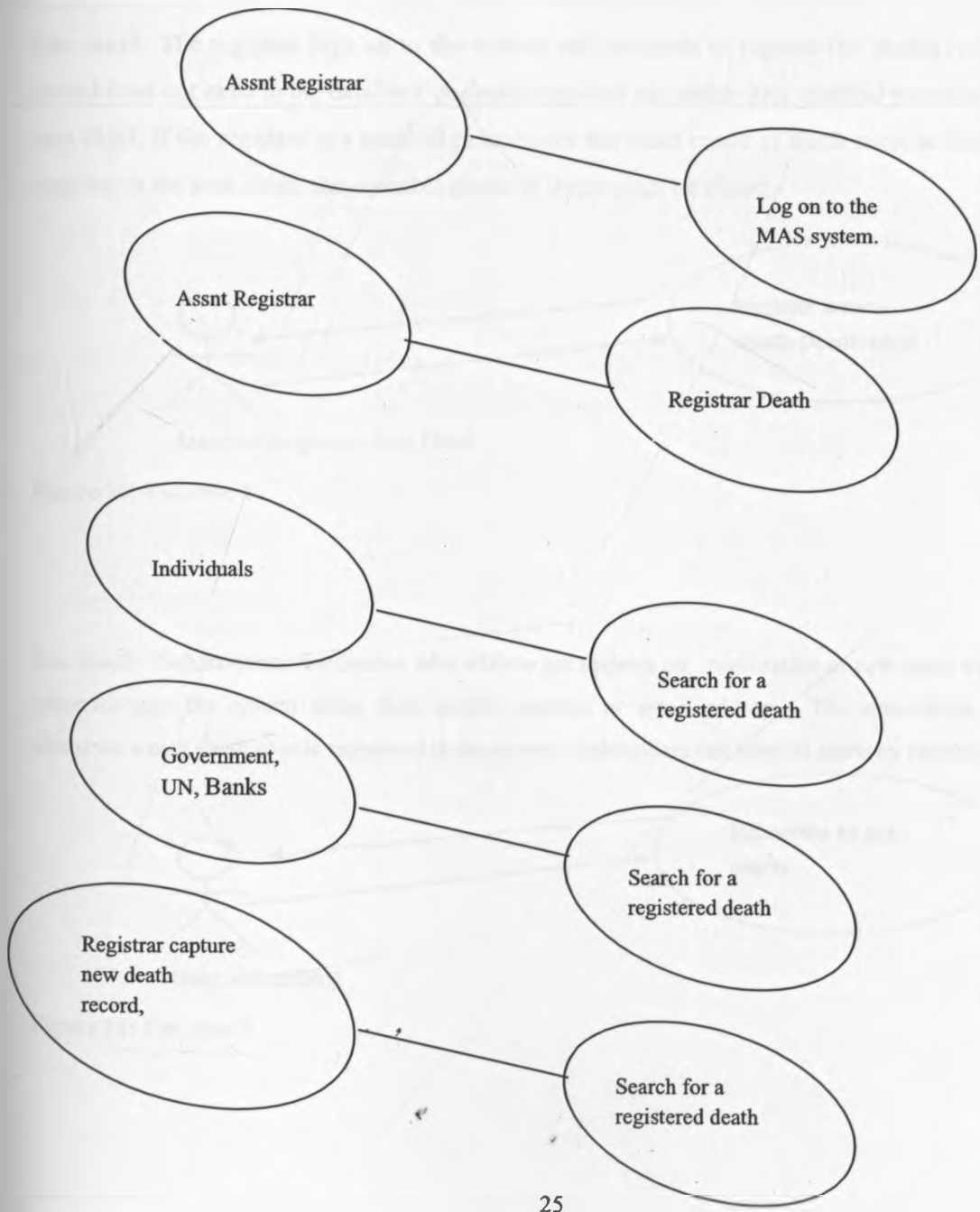
Figure 8: Goal Diagram



Subsidiary goals

Sub goals are important in implementing agent proactiveness. Actions change states and states are goals. "A goal is a condition or state of affairs in the world that an actor would like to achieve". (Yu, E , 1995),(Tomáš Šalamon,2011).

Figure 9: Subgoals



4.1.1 Use Case Scenarios

A use case is a view of the functionality (or use) of a system from the user's perspective. Use case scenarios represent a particular instance of the system without branching. Use case analysis was used for requirements specification of the multi-agent system and the behaviour specification. (Michael Papasimeon and Clinton Heinze, 2000)

Use case1: The registrar logs on to the system and proceeds to capture the deaths record if the record does not exist in the database. A deaths registrar can either be a medical practitioner or the area chief. If the registrar is a medical practitioner the exact cause of death must be stated. If the registrar is the area chief, the probable cause of death must be stated.

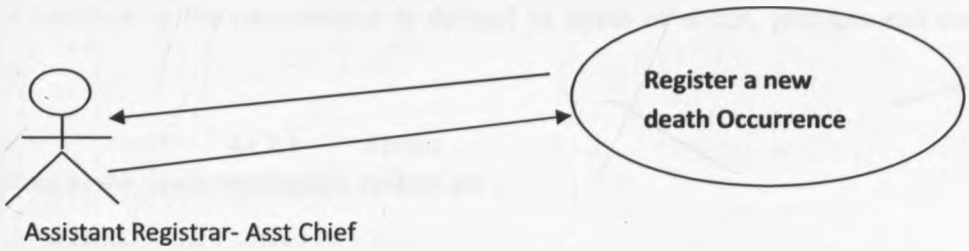


Figure 10: Use case 1

Use case2: Subscription: the entities who wish to get updates on registration of new death events must subscribe into the system using their mobile number or email address. The subscribers get alerts whenever a new death case is registered in the system. Subscribers can turn off alerts by unsubscribing.

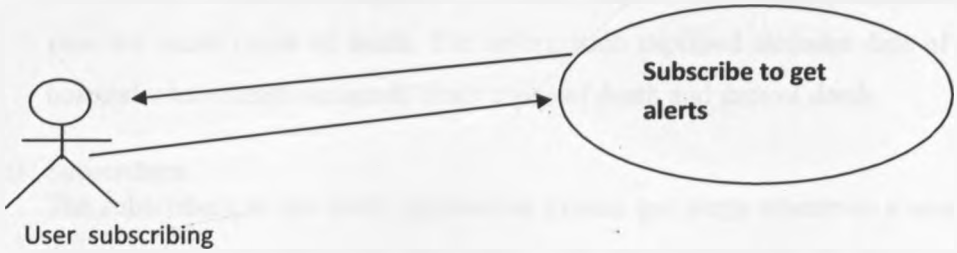


Figure 11: Use case 2

Use case 3: Search for existing registered deaths. A user who is subscribed searches the system for a particular record using specified parameters.

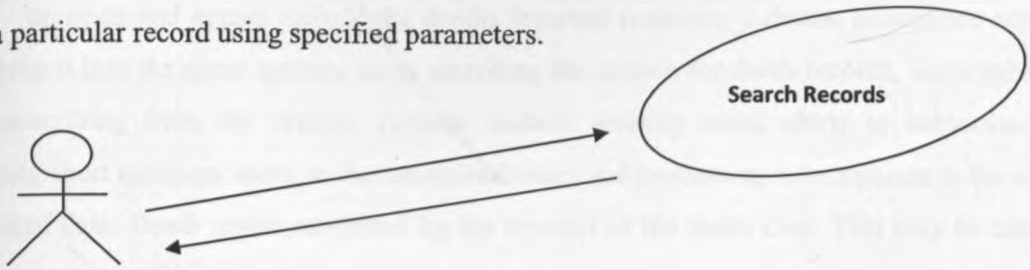


Figure 12: Use case 3

4.1.2 Identification of the Systems Interface to the Environment

The systems interface to the environment is defined in terms of actors, precepts and external data.

4.1.2.1 Actors

The main actors in the death registration system are :

a) Registrar (area chief)
A person approaches the area chief with information about a death occurrence. The area chief records the information in the system. The information captured includes names, date of birth, date of death, place of death, name of person reporting the death and probable reason for death.

b) Registrar (Medical practitioner)

For deaths that occur in hospitals and health centres, the medical practitioner must state the exact cause of death. The information captured includes date of death, hospital where death occurred, exact cause of death and date of death.

c) Subscribers
The subscribers to the death registration system get alerts whenever a new death case has been registered in the system. The subscribers get the name, date of death, place of death. They also search registered deaths in the system.

4.1.2.2 Percepts and Actions

The precepts and events include the deaths registrar receiving a deaths occurrence report and entering it into the agent system, users searching the system for death records, users subscribing /unsubscribing from the system. Actions include sending email alerts to subscribed users, sending short message, alerts to the subscribed users and processing subscriptions to the system. External data: Death report submitted by the reporter of the death case. This may be citizens or security personnel

4.1.3 System Functionalities

The system functionaries identified are: registration of a death occurrence, sending alerts to subscribers, registration of subscribers, sending emails, searching of death record.

a)DEATH REGISTRATION FUNCTIONALITY

NAME: Death Registration
Description: Register a new death occurrence in the system
Percepts/events/messages: Death record registered
Message send : Registration completed successfully ; customised message
Actions : display customised message
Data used: Data supplied by the person reporting the death
Interactions : data reception agent via communication agent

b) SENDING SMS ALERTS

Name : Sending Short message to mobile phone
Description: Alert subscribers that there is a new death occurrence.
Percepts/events/messages: new death record in deaths DB(customised message)
Messages sent: new death occurrence (customised message)
Actions: Display death record
Data used: Deaths DB, Subscribers' details
Interactions: Communications manager via sms alerts

C) SENDING EMAIL ALERTS

Name: Sending Email alerts

Description: Sending email alerts when a death event gets added into the system

Percepts/events/messages: new death occurrence in DB (customised message)

Messages sent: Email alert (message)

Actions: Display death record

Data used: Death DB, subscribers' details.

Interactions: Communications manager via email alerts

D) SEACHING DEATH RECORDS

Name: Search record of registered deaths

Description: perform search

Percepts/events/messages: record found in DB (customised message)

Messages sent: search results

Actions: Display death record

Data used: Death DB, subscribers' details.

Interactions: Communications manager via search

E) SUBSCRPTION

Name: subscription

Description: register users in the system so as to get alert messages

Percepts/events/messages: successful (un)subscription (customised message)

Messages sent: subscription activated.

Actions: subscribe/activate subscription/deactivate

Data used: subscribers' details.

Interactions: Communications manager via subscription

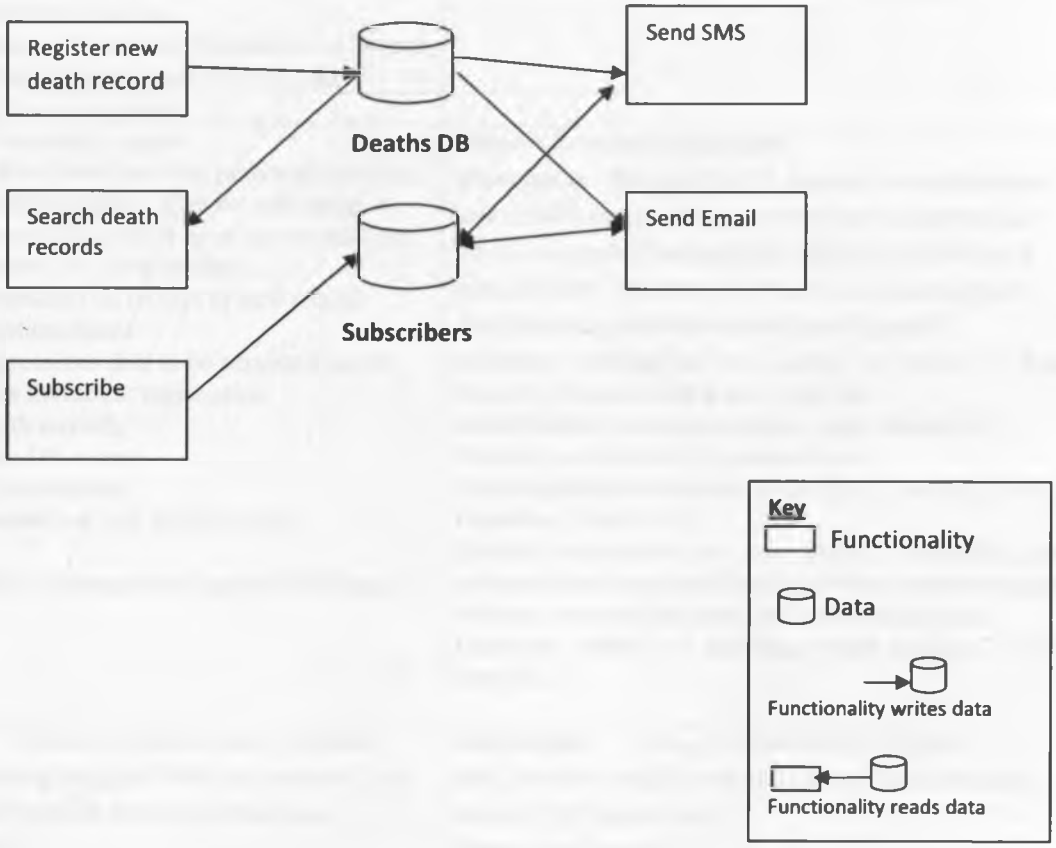
4.2 ARCHITECTURAL DESIGN

4.2.1 Determine Agent Types

Agent types are determined from the system functionalities. The process of identifying agents involved grouping functionalities using the data coupling diagram below: A data coupling diagram consists of the functionalities and all identified data.

The system functionaries identified in the previous chapter are: registration of a death occurrence, sending alerts to subscribers, registration of subscribers, sending emails, searching of death record. The agents required to achieve these functionalities include, general communication agent, data reception agent, data processing agent, data retrieval agent, web interface agents.

Figure 13: Data coupling diagram



4.2.2 Agent Descriptors

The **agent descriptors** for the identified agents are described below.

Name: WSIG – Container:control **Description:** receive all requests from users of the web interface and sends the responses back after	**Name: Data reception agent** **Description:** receive data from the WSIG control container

agents have worked on them. Directs the requests to the appropriate agent(s). **Lifetime** : instantiated when system starts. **Initialization:** Reads user input **Demise:** Closes requests **Functionalities included**: Registration, search, sms sending , email sending **Uses Data**: Deaths DB, SubscribersDB, usersDB **Goals**: respond to web requests, direct requests to the appropriate agents, output results. **Events responded to**: new death record,search request, sms and email alerts. **Actions**: display customized responses to search requests and successful entry of new records. **Interacts with**: data reception	**Lifetime** : instantiated when a new death record is entered into the system) **Initialization:** receives data from WSIG container **Demise:** Closes requests **Functionalities included**: Registration, **Uses Data**: deaths data captured **Goals**: receive message from WSIG , return response to WSIG, **Events responded to**: new death record **Actions**: **Interacts with**:WSIG, Data processing agent
Name: Data Processing agent **Description**: Receives data to be persisted into DB. Does data validation ;Enters data into db;Initiates communication to subscribers upon successful data persistence;Generates serial numbers. **Lifetime**: instantiated on receipt of new record. **Demise** application closes **Initialisation**: receives data to be persisted in DB. **Functionalities included**: registration **Uses data**: death records, **Produces data**: DB record **Goals**: enter data into db, **Events responded to:** new death record; **Actions**: **Interacts with**: communication agent,WSIG agent	**Name** :Communication agent **Description**: Responsible for general communication tasks.;SMS and email agents work under instruction from this agent.;Converts data objects into SMS and email objects.;Validates SMS email and data object data;Sends requests to sms and email agents **Lifetime**: Instantiated on receipt of SMS or Email request. Demise when a user logs out **Initialisation**: receives request , reads deaths DB **Demise**: on close of DB connections **Functionalities included**: sending sms, sending email **Usesdata**: deaths DB, **Events responded to:** new arrival; customer query; customerpurchase;creditcheckresponsecustomerresponse; **Actions**: convert data into sms and email objects. **Interacts with**: sms sending, email sending , record searching
Sms agent Receives SMS objects;Validates SMS data ensuring required fields are present ;Sms sending;Retrieves valid sms recipients from subscriber table. **Name**: Sms agent **Description**: retrieve valid sms recepints from subscriber table, send sms **Lifetime**: Instantiated on receipt of sms object. Demise when sms is send. **Initialisation**: receives sms object. **Demise**: regenerated. **Functionalities included**: sms sending, **Usesdata**: subscriber table, sms data. **Producesdata**: sms message. **Goals**sends sms, validates sms recepients **Events responded to:** new sms object;	**Email agent** Acts on email object;Validates data.;Retrieves valid email recipients from subscriber data in DB.;Sends email **Name**: email agent **Description**: retrieve valid email recepints from subscriber table, send email **Lifetime**: Instantiated on receipt of email object. Demise when application closes **Initialisation**: receives email object. **Demise**: regenerated. **Functionalities included**: email sending, **Usesdata**: subscriber table, email data. **Producesdata**: email message. **Goals**:sends email, validates email recipients **Events responded to:** new email object; **Actions**: Display email alert ,

Actions: Display sms alert , **Interacts with**: subscibers	**Interacts with**: subscibers
Search agent Receives search requests ;Performs data search;Formats output to html;Sends back results ;Validates search request.	

4.2.3 Interaction Diagram

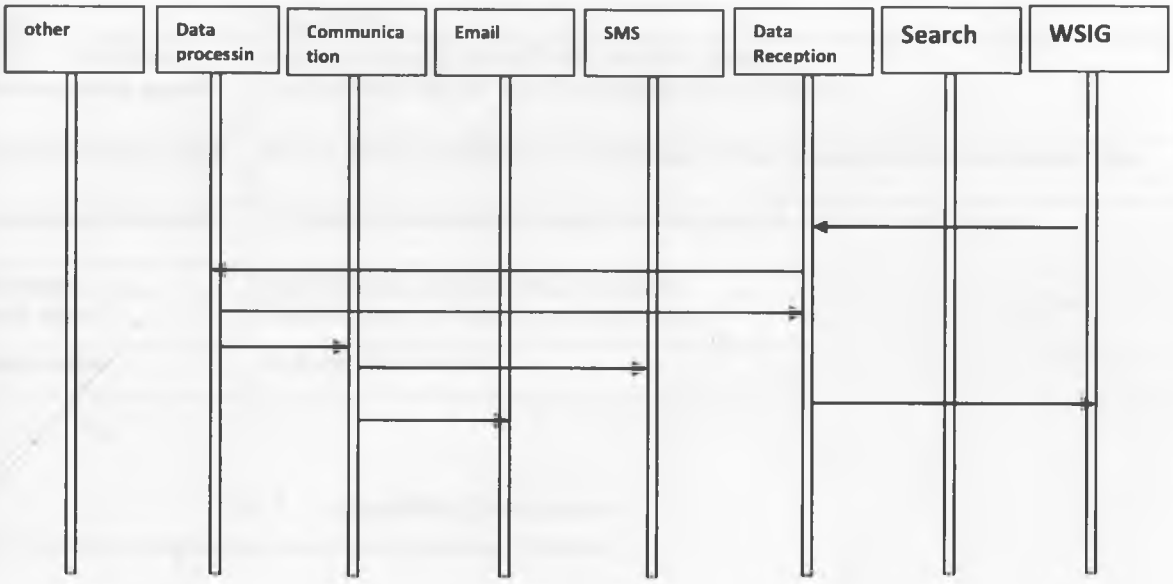


Figure 14: Interaction Diagram

4.3 DETAILED DESIGN

Detailed design focused on developing the internal structure of each of the agents and the behaviour of the agents. The agent's internal events, plans and detailed data structures.

4.3.1 Agent Capabilities

The capabilities of each agent were identified as follows:

Agent	Capabilities
WSIG – Container	Interface between the web and the other agents
Data reception agent	Receive data entered into the database and validates it.
Data processing agent	Ensure data is persisted into the DB and initiate communication with subscribers.
Communication agent	For general communication and send requests to sms and email agents.
Sms agent	Sms sending, validate sms subscribers
Email agent	Email sending ,validate email recipients
Search agent	Perform data search

4.3.2 Capability Descriptors

The capability descriptors have been generated below:

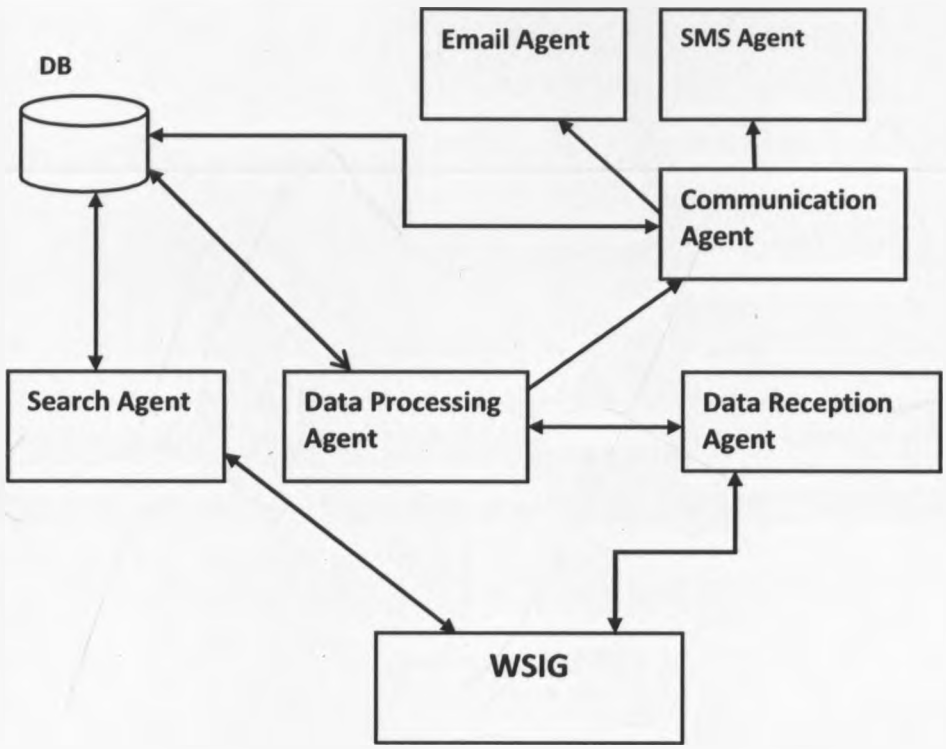
Name: handle search requests
External interface to the capability: user enters search string
Natural language description: produce search results
Interaction with other capabilities: data reception capability
Data used/produced by the capability: search results
Inclusion of other capabilities: None

Name: email sending
External interface to the capability: new record found.
Natural language description: send sms msg to subscribed users.
Interaction with other capabilities: communication
Data used/produced by the capability: sms object
Inclusion of other capabilities: None

Name: data reception
External interface to the capability: Data entry
Natural language description: enter data into the deaths data base
Interaction with other capabilities: communication agent
Data used/produced by the capability: database record

Inclusion of other capabilities: None

Figure 15: Agent Overview diagram



4.4 Chapter summary

The overall goal of the multiagent based system for registration of deaths is capture deaths as soon as they are reported and sending alerts to subscribers. Several subgoals have been identified which include logging on to the system, registration of a death event, searching of records , subscribing to receive alerts, generating statistics and updating records. Use case scenarios were used to identify the functionalities of the system. The systems interface to the environment was defined interms of actors,percepts and external data.From the system functionalities defined in system specification stage, the agent types determined are communication agent, data reception agent, data processing agent and data retrieval agent .Data

coupling diagram (consisting of functionalities and data) was used to determine the agent types. An interaction diagram showing how the agents interact was developed.

CHAPTER 5: VERIFICATION AND VALIDATION

5.0 Overview

The main function of this system is registration processing, data retrieval, alerts and generation of statistics. The multiagent technology implemented in this system brings out the following aspects :

a. Scalability: agents can join or leave the system.

b. Distribution: the agents can be installed on other servers and made to interact. Moreover, the users and suppliers are distributed in different physical environments.

c. Communication: the agents communicate using ACL .

5.1 Agent types and communication

Java agents used in the implementation are:

i. WSIG agent (Web Services Integration Gateway)- .acts as interface between the web and agents. It converts http requests data into Soap requests. SOAP requests are converted to data objects and forwarded to the appropriate agent. After processing the agent sends back to WSIG to convert to http. At start up, the WSIG sends REQUEST and INFORM ACL messages. It then SUBSCRIBES with the controller agent.

ii. Reception agent: receives data from WSIG (the new record); validates data and send back message to WSIG.

iii. Processing agent: receives data from Reception agent. It does data validation and enters data into the database. It generates serial numbers and initiates communication to subscribers.

iv. Communication agent: General communication tasks. Converts data objects into email and sms objects

v. Email agent: retrieve valid email recipients , send email.

vi. Search Agent : receive search requests , validate request perform data search , format output to html , send back .

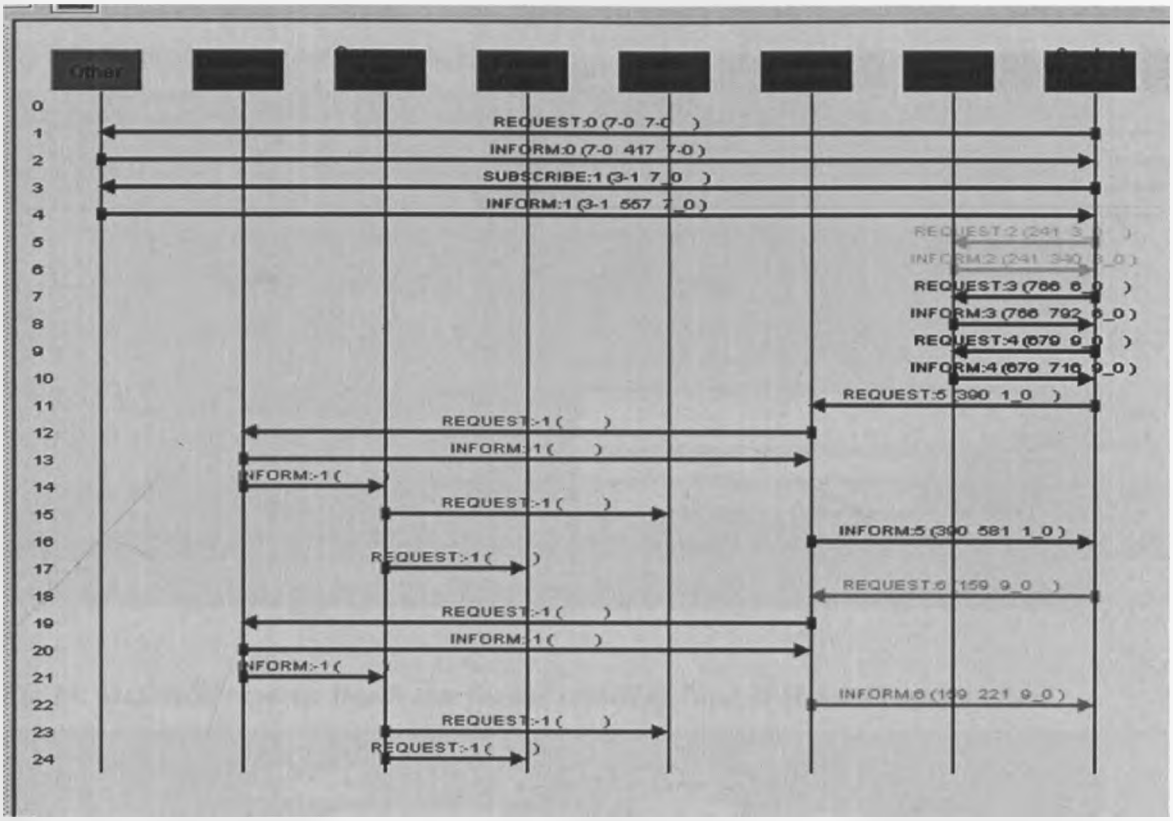
Generic agent classes can be derived from the abstract base class.

```
public class [redacted] extends AbstractAgent {

    /**
     *
     */
```

```
private static final long serialVersionUID = 6366801477888787791L;
private Logger log = Logger.getLogger(CommunicationAgent.class.getName());
```

Figure 16: Agent Communication



5.2 Hardware and Software tools

The agent classes , users of the system and the web environment have been realised using the following tools.

Microsoft Windows 7 for the operating system.

Mysql : for database management.

PhpMyAdmin 3.4.9: a graphical interface for mysql database.

Java EE Eclipse IDE for the development environment.

Apache Tom cat 5.5.35 Servlet Container

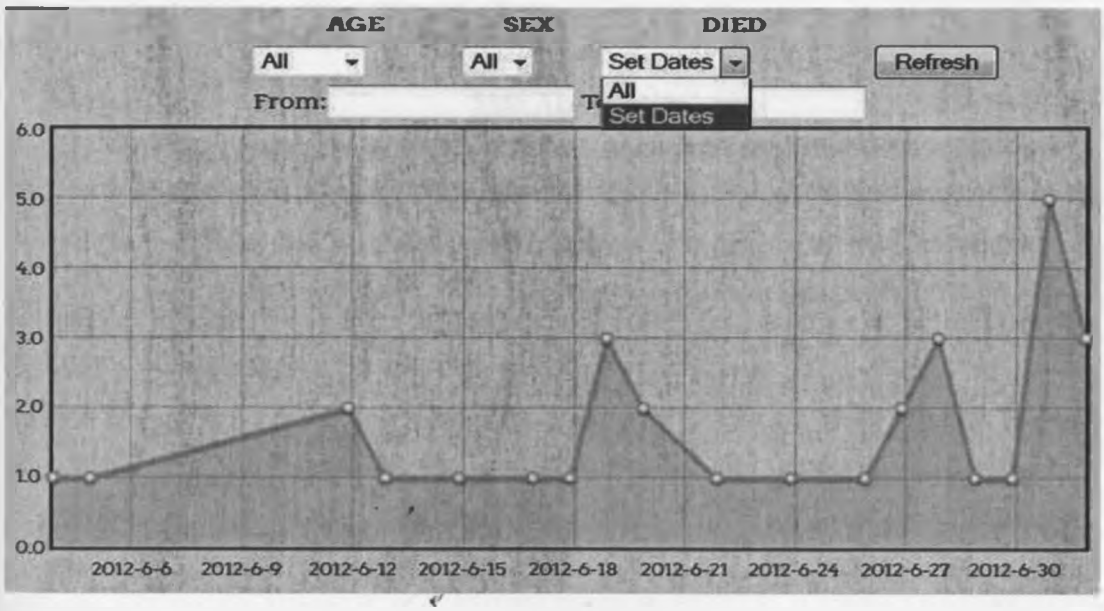
Apache, php web server to host the website.

5.3 Analysis of the implementation results

The evaluation of the multiagent based system is based on the results obtained from registration of deaths, retrieval of records, alerts and edits in comparison to the existing system of registration of deaths. The figures below shows the content of the tests obtained from the system.

Fig 17: Subscription to receive updates

Fig 18: Statistical reports: Death rate for the month of June 2012 for all ages and sex.



EMAIL ALERTS

From: Deaths Registrar <cmutindak@gmail.com>

To: gim_yogo@yahoo.com; jkivinda@utalii.co.ke; justina.mwikali@yahoo.com;

justina.mwikalig@gmail.com; plimo@yahoo.com; tindase@gmail.com

Sent: Tuesday, July 3, 2012 12:25 PM

Subject: New Case

New case reported. ID: 98766789 Names: lukas ongolo omollo Born: 1945-04-01 Died: 2012-06-26

Deaths Registrar cmutindak@gmail.com

Jun 28 to jkivinda, justina.mwikali, me, tindase

New case reported. ID: 123123 Names: timothy Elias Bonny Born: 2012-06-01 Died: 2012-06-28

Deaths Registrar cmutindak@gmail.com

Jun 28 to jkivinda, justina.mwikali, me, tindase

New case reported. ID: ID: NVA Names: mulli peter oscar Born: 2008-08-04 Died: 2012-06-28

Deaths Registrar cmutindak@gmail.com

Jun 28 to jkivinda, justina.mwikali, me, tindase

New case reported. ID: NVA Names: cartis mambo m Born: 2012-04-09 Died: 2012-06-28

SMS ALERT MSG

- New case reported. ID: NVA Names: mbithe m Tony Born: 1945-08-26
- New case reported. ID: 4567980 Names: hellen peter m Born: 1978-08-01
- New case reported. ID: NVA Names: mulli peter oscar Born: 2008-08-04
- New case reported. ID: NVA Names: cartis mambo m Born: 2012-04-09
- New case reported. ID: 908765 Names: moris james R Born: 1968-10-27
- New case reported. ID: NVA Names: purity m oloo Born: 2011-12-09
- New case reported. ID: NVA Names: muloo D Benard Born: 2012-01-06

5.4 System testing and verification

Civil registration of deaths is an important process for providing statistical reports and legal documents to the relatives of the deceased persons. While the current process of registration of deaths is slow and does not produce timely and accurate data, the multiagent based system can efficiently carry out this process and avail data to all interested stakeholders mostly the Kenya Bureau of Statistics and UN, who require the data for economic reports and health planning.

The system was tested with random records. Transmission of alerts for new records was tested with mobile numbers and email addresses of selected participants. The system was able to transmit updates to subscribed users.

Testing subscription: The recipients of the alerts first subscribed to receive update. Incorrect emails and mobile numbers did not receive alerts.

Testing data capture: Only the registered data entry personnel are able to add new records to the database.

Testing editing: If there are errors in the captured records, the assistant registrar can update the records with the correct details.

Testing vital statistics: The system was able to produce statistical reports by age , date of death , or gender in a graph.

CHAPTER 6: CONCLUSION

Civil registration of deaths is a vital process that ensures that a country can account for its citizens and make major policy decisions based on statistical data generated from the system. Civil registration is a universal human right recognised by the United Nations. However, the government of Kenya has not been able to implement an efficient technological solution to achieve universal coverage and provide useful statistical summaries to users.

This study proposed a muiltagent based system designed for registration of dead persons in Kenya. The multiagent based system for registration of deaths consists of agents that interact to achieve set goals. The study has several aspects including analysis of civil registration of deaths, evaluation of multiagent system technology, architectural design, implementation, testing and verification of the system. An attempt has been made to capture the essence of agents and the distinctive properties between them and other software technologies. The MaS framework, FIPA that guide the development of MaS systems should justify the choice of implementation tools and environment.

The multiagent based system for registration of deaths implements a set of agents that serve as intermediaries among the suppliers of data and the consumers. The WSIG container is the main agent that converts agents into web services. The communication agent is responsible for general communication tasks. SMS and Email agents are responsible for sending SMS and email respectively with instruction from the communication agent. The search agent receives http requests, validates and formats it for output. The data processing agent receives data and enters it into the database. The multiagent system is essentially developed to process registration of deaths and provide statistical information on death occurrences. The system simplifies updates by making retrieval and correction of errors easy and transparent to the user. The system offers the advantage of open architecture since it allows new agents representing new functionalities, users or data sources to join and leave the environment thus allowing dynamism and scalability.

Use of the system encounters the limitation of using only android mobile phones for sending the alerts. The SMS application runs on an android enabled phone to send alerts to SMS subscribers. Other versions of mobile telephone operating systems such as Palm OS , BlackBerry, Windows Mobile, Symbian, and iPhone OS cannot send SMS but can receive.

Further Work

However, some aspects of the multiagent based system for registration of deaths must be reconsidered in future in order to improve the system design and prevent bottlenecks on resources. Firstly, it will be necessary to distribute the agents in different servers so as to permit interconnection of different agent environments that provide more computing power. Secondly the system can be interconnected to the database for medical practitioners' board and that of public service in order to validate the suppliers of data.

REFERENCES

1. Bellifemine, F., Caire, G., Greenwood, (2007). Developing Multi-Agent systems with JADE. John Wiley & Sons

2. Births and Deaths Registration Act_Cap.149. www.Kenyalaw.org. Last accessed 15 March 2012.

3. Brian Henderson-Sellers, Paolo G, (2005), Agent-oriented methodologies, Idea Group Publishing.

4. Caire, G. (2002). JADE Tutorial: Application- Defined Content Languages and Ontologies. Telecom Italia Laboratory.

5. Civil Registration Department Kenya.http://www.freerepublic.com/focus/ ,Last Accessed 11 March 2012.

6. Compendium. www.knbs.or.ke/pdf/COMPENDIUM.pdf . last accessed 13 February 2012.

7. Gascue˜na,J and Fern´andez-Caballero,A(2009). Prometheus and INGENIAS Agent Methodologies: A Complementary Approach. Springer-Verlag Berlin Heidelberg .

8. Genesereth, M., Ketchpel, S. (1994). Software Agents. *CACM - Special Issue on* Intelligent Agents. *37:7.*

9. IEEE Standards Committee, "FIPA specifications,", http://www.fipa.org/, 2004.Last accessed 16 March 2012.

10. Hill, K and Preston S (2011). Estimating the completeness of death registration. PubMed web source.

11. In the context of HIV and AIDS-English. http://plan-international.org/birthregistration. Last accessed 9 March 2012.

12. JADE Board, Telecom Italia (2007) JADE WEB SERVICES INTEGRATION GATEWAY (WSIG) GUIDE Last update: 12-Jul-2011.

13. Kaufman, J.S et al (1997). The absence of adult mortality data for sub-Saharan Africa: a practical solution. Bull World Health Organ. USA.

14. Mahapatra P, Shibuya K (2007). Civil registration systems and vital statistics: successes and missed opportunities. TheLancet

15. Mathers C, Ma Fat D, Inoue M, Rao C, Lopez AD(2005). Counting the dead and what they died of: an assessment of the global status of cause of death data. Bull World Health Organ; 83: 171–77.

16. Michael , P and Clinton H (2000) Specifying Requirements in a Multi-Agent System with Use Cases Aeronautical and Maritime Research Laboratory ,Defence Science and Technology Organisation ;Melbourne, Victoria, 3001,Australia

17. Padgham, L., Winikoff, M.,et al (2005). Adding debugging support to the Prometheus methodology. Engineering Applications of Artificial Intelligence 18(2),173–190 .

18. Padgham,L. and Winikoff, M (2002). Prometheus: A methodology for developing intelligent agents. In Third International Workshop on Agent-Oriented Software Engineering, July 2002, RMIT University.

19. Registration Process. http://www.crd.go.ke/Registration Process . Last accessed 12 January 2012.

20. SADIO Electronic Journal of Informatics and Operations Research, vol. 3, no. 1, pp. 1-12 (2000) A Java Framework for Multi-agent Systems. .http://www.dc.uba.ar/sadio/ejs . Last accessed 23/06/2012

21. Tomáš Šalamon (2011) , Design of Agent-based Models: Developing Computer Simulations for a Better Understanding of Social Processes Academic Series Tomas Bruckner , Revin Zivonin , 2011 Czech republic.

22. UN (1998).Handbook on Civil Registration and Vital Statistics Systems Computerization Studies in Methods Series F, No. 73. United Nations, New York.

23. UN (2001). Principles and recommendations for a vital statistics system. Revision 2. United Nations. New York.

24. UN (2010). Status of Civil Registration and Vital Statistics in the SADC Region .Technical Report. United Nations, New York.

25. UNICEF (2005). The 'rights' start to life. A statistical analysis of birth registration. United Nations, New York.

26. Why counting births and deaths is important Fact sheet N°324 October 2007.http://www.who.int/mediacentre/factsheets/fs324/en/index.html . Last accessed 11 March 2012.

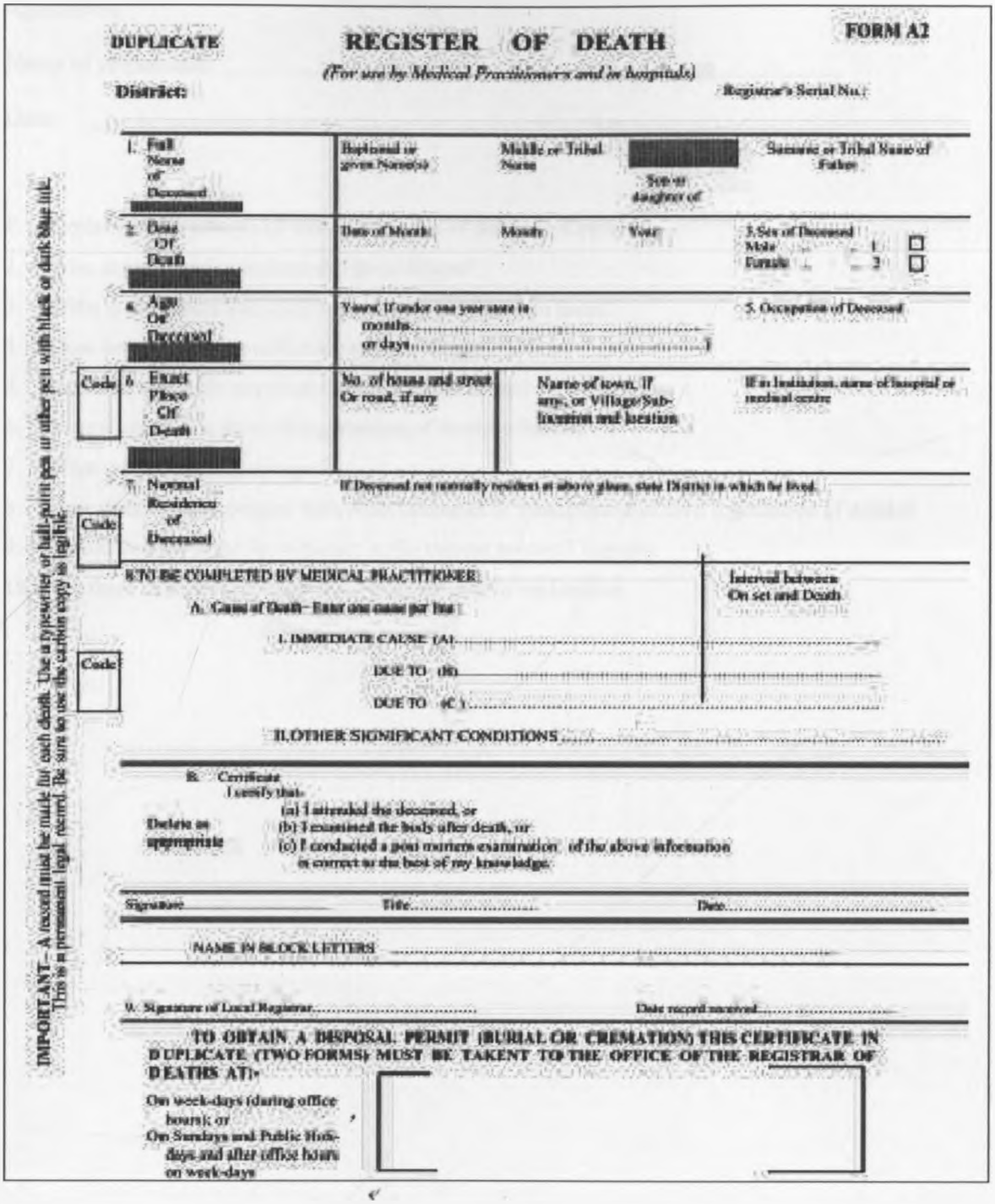
27. Wooldridge, M (2002). Introduction to multi-agent systems. John Wiley and Sons Ltd.

28. Wooldridge, M , Jennings,N et al (2000). The Gaia methodology for agent oriented analysis and design. Autonomous agents and multi-agent systems. Kluwer Academic Publishers. Boston.

29. Yu, E.(1995) ,Modelling Strategic Relationships for Process Reengineering. PhD Thesis, Graduate Department of Computer Science, University of Toronto, Toronto, Canada

APPENDIX

APPENDIX I: Form A2 - Register of Death



DUPLICATE

REGISTER OF DEATH

FORM A2

(For use by Medical Practitioners and in hospitals)

District:

Registrar's Serial No.:

1. Full Name of Deceased [redacted] | Baptismal or given Name(s) | Middle or Tribal Name | [redacted] Son or daughter of | Surname or Tribal Name of Father

2. Date Of Death [redacted] | Date of Month | Month | Year | 3. Sex of Deceased: Male .. 1 ☐ Female .. 2 ☐

4. Age Of Deceased [redacted] | Years, if under one year state in months ..... or days ..... | 5. Occupation of Deceased

Code | 6. Exact Place Of Death [redacted] | No. of house and street Or road, if any | Name of town, if any, or Village/Sub-location and location | If in Institution, name of hospital or medical centre

Code | 7. Normal Residence of Deceased | If Deceased not normally resident at above given, state District in which he lived

8. TO BE COMPLETED BY MEDICAL PRACTITIONER

A. Cause of Death- Enter one cause per line

Code | I. IMMEDIATE CAUSE (A) ..... | Interval between Onset and Death

DUE TO (B) .....

DUE TO (C) .....

II. OTHER SIGNIFICANT CONDITIONS .....

B. Certificate

I certify that-

Delete as appropriate

(a) I attended the deceased, or

(b) I examined the body after death, or

(c) I conducted a post mortem examination of the above information is correct to the best of my knowledge.

Signature ..... Title ..... Date .....

NAME IN BLOCK LETTERS .....

9. Signature of Local Registrar ..... Date record received .....

TO OBTAIN A DISPOSAL PERMIT (BURIAL OR CREMATION) THIS CERTIFICATE IN DUPLICATE (TWO FORMS) MUST BE TAKENT TO THE OFFICE OF THE REGISTRAR OF DEATHS AT:-

On week-days (during office hours); or

On Sundays and Public Holidays and after office hours on week-days

IMPORTANT – A record must be made for each death. Use a typewriter or ball-point pen or other pen with black or dark blue ink. This is a permanent legal record. Be sure to use the carbon copy is legible.

APPENDIX II: Questionnaire

The questionnaire below was used to collect primary data from the department of civil registration.

Name of respondent: _____

Date: _____

1. Explain is the process of civil registration of deaths in Kenya?
2. Who is mandated to register deaths in Kenya?
3. What is the deaths data used for and who are the main users?
4. How do you generate statistics and how often?
5. How and where do you store the registration forms?
6. In your opinion is the civil registration of deaths effective?
7. What are the main challenges?
8. How does Kenya compare with other countries in management of civil registration of deaths?
9. Would you advocate for a change in the current process? Explain.
10. Are there universal recommended practice in civil registration

APPENDIX III : Agent Algorithms

EMAIL AGENT

```
	public EmailAgent() {
		// TODO Auto-generated constructor stub
	}

	/*
	 * (non-Javadoc)
	 *
	 * @see multi.agent.reporter.core.AbstractAgent#setup()
	 */
	@Override
	protected void setup() {
		// TODO Auto-generated method stub
		register(CommunicationVocabulary.EMAIL_SERVICE);
		this.addBehaviour(new CyclicBehaviour(this) {
			/**
			 *
			 */
			private static final long serialVersionUID =
1715824436050556157L;
			private MessageTemplate template = MessageTemplate
					.MatchOntology(DeathsReporterOntology.getInstance()
							.getName());
			private HashMap<String, ACLMessage> savedMessages = new
HashMap<String, ACLMessage>();

			public void action() {
				ACLMessage msg = myAgent.receive(template);
				if (msg != null) {
					Action actExpr;
					try {
						actExpr = (Action)
myAgent.getContentManager()
								.extractContent(msg);
						AgentAction action = (AgentAction)
actExpr.getAction();
						if (action instanceof Email) {
							Email email = ((Email) action);
							String receivers = "";
							String query = "select email from
subscribers where email is not null group by email";
							try {
								Pattern p = Pattern
										.compile("[a-zA-Z0-
9\\._\\-]{4,}@[a-zA-Z0-9]{3,}\\.[A-Za-z0-9\\.]{2,}");

								ResultSet executeQuery = MySQL

	.executeQuery(query);
								while (executeQuery.next()) {
```

```
                                                String em =
executeQuery.getString(1);

                                                Matcher m = p.matcher(em);
                                                if (m.matches())
                                                      receivers += em +
",";
                                                }
                                          } catch (Exception e) {
                                                // TODO: handle exception
                                          }
                                          try {
                                                receivers=receivers.substring(0,
receivers.length()-1);
                                          } catch (Exception e) {
                                                // TODO: handle exception
                                          }
                                          email.setReceipients(receivers);
                                          if (receivers != null
                                                      &&
!receivers.trim().equals(""))
                                                EmailUtility.send(email);
                                          else
                                                log("No subscribers to email");
                                    }
                              } catch (Exception e) {
                                    log.error("Exception: " + e.getMessage(), e);
                              }
                        } else {
                              block();
                        }
                  }

            });
      }

}
```

SMS AGENT

```
public class SMSAgent extends AbstractAgent {

      /**
       *
       */
      private static final long serialVersionUID = 4676815256949739181L;
      private Logger log = Logger.getLogger(SMSAgent.class.getName());

      /*
       * (non-Javadoc)
       *
       * @see multi.agent.reporter.core.AbstractAgent#setup()
       */
      @Override
      protected void setup() {
```

```
            // TODO Auto-generated method stub
            register(CommunicationVocabulary.SMS_SERVICE);
            this.addBehaviour(new CyclicBehaviour(this) {
                  /**
                   *
                   */
                  private static final long serialVersionUID =
1715824436050556157L;
                  private MessageTemplate template = MessageTemplate
                              .MatchOntology(DeathsReporterOntology.getInstance()
                                          .getName());
                  private HashMap<String, ACLMessage> savedMessages = new
HashMap<String, ACLMessage>();

                  public void action() {
                        ACLMessage msg = myAgent.receive(template);
                        if (msg != null) {
                              Action actExpr;
                              try {
                                    actExpr = (Action)
myAgent.getContentManager()
                                                .extractContent(msg);
                                    AgentAction action = (AgentAction)
actExpr.getAction();
                                    if (action instanceof SMS) {
                                          SMS sms = ((SMS) action);
                                          String query = "select phone from
subscribers where phone is not null group by phone";
                                          try {
                                                ResultSet executeQuery = MySQL

      .executeQuery(query);
                                                while (executeQuery.next()) {
                                                      String no =
executeQuery.getString(1);

                                                      sms.setReceiver(no);
                                                      if (!sms.send())
                                                            log("SMS not sent
to " + no);
                                                }
                                          } catch (Exception e) {
                                                // TODO: handle exception
                                          }

                                    }
                              } catch (Exception e) {
                                    log.error("Exception: " + e.getMessage(), e);
                              }
                        } else {
                              block();
                        }
                  }

            });
      }
```

```
}
```

SEARCH AGENT

```
public class SearchAgent extends AbstractAgent {

	/**
	 *
	 */
	private static final long serialVersionUID = 4676815256949739181L;
	private Logger log = Logger.getLogger(SearchAgent.class.getName());

	/*
	 * (non-Javadoc)
	 *
	 * @see multi.agent.reporter.core.AbstractAgent#setup()
	 */
	@Override
	protected void setup() {
		// TODO Auto-generated method stub
		register(SearchVocabulary.SEARCH_SERVICE);
		this.addBehaviour(new CyclicBehaviour(this) {
			/**
			 *
			 */
			private static final long serialVersionUID =
1715824436050556157L;
			private MessageTemplate template = MessageTemplate
					.MatchOntology(DeathsReporterOntology.getInstance()
							.getName());

			public void action() {
				ACLMessage msg = myAgent.receive(template);
				if (msg != null) {
					Action actExpr;
					try {
						actExpr = (Action)
myAgent.getContentManager()
								.extractContent(msg);
						AgentAction action = (AgentAction)
actExpr.getAction();
						if (action instanceof SearchParameters) {
							SearchParameters search =
((SearchParameters) action);
							String q =
search.generateSearchQuery();
							ResultSet rs = MySQL.executeQuery(q);

							String s = "<div><span
class='error'><font size = '+2' color='red'><center>No records match the given
criteria</center></font></span></div>";
							if(rs.next())
							{
```

```
                                                s="<table width='100%'
align='center'>" +
                                                      "<tr>" +

      "<th>No.</th><th>Names</th><th>ID/Passport No.</th>" +
                                                      "<th>Serial
Number</th><th>Date of Birth</th><th>Date of Death</th>" +
                                                      "<th>Link</th>" +
                                                      "</tr>";
                                                rs.beforeFirst();
                                          }
                                          int i=0;
                                          while (rs.next()) {
                                                try {
                                                      String rowHtml = "<tr>";
                                                      rowHtml+="<td
align='center'>"+rs.getString("id")+"</td>";
                                                      String names =
getNames(rs.getString("firstname"),rs.getString("surname"),rs.getString("middlename")
);
                                                      rowHtml += "<td
align='center'>"+names+"</td>";
                                                      String  id =
rs.getString("idno");
                                                      if(id==null)
                                                            id="N/A";
                                                      rowHtml += "<td
align='center'>"+id+"</td>";
                                                      rowHtml += "<td
align='center'>"+rs.getString("serialnumber")+"</td>";
                                                      rowHtml += "<td
align='center'>"+rs.getString("dateofbirth")+"</td>";
                                                      rowHtml += "<td
align='center'>"+rs.getString("dateofdeath")+"</td>";
                                                      rowHtml += "<td
align='center'><a href='record.jsp?serial="+rs.getString("serialnumber")+"'>More
details</a></td>";
                                                      rowHtml+="</tr>";
                                                      s+=rowHtml;
                                                      i++;
                                                } catch (Exception e) {
                                                      // TODO: handle exception
                                                      i++;
                                                      continue;
                                                }
                                          }
                                          if(i!=0)
                                          {
                                                s+="</table>";
                                          }
                                          search.setResults(s);
                                          actExpr.setAction(search);
                                          replyMessageWithResult(actExpr, msg,
                                                      ACLMessage.INFORM, s);
```

```
                        }
                    } catch (Exception e) {
                        log.error("Exception: " + e.getMessage(), e);
                    }
                } else {
                    block();
                }
            }

            private String getNames(String first, String last,
                    String mid) {
                // TODO Auto-generated method stub
                String n="";
                if(first!=null)
                    n+=first+ " ";
                if(mid!=null)
                    n+=mid+ " ";

                if(last!=null)
                    n+=last+ " ";
                return n.trim();
            }

        });
    }

}
```

COMMUNICATION AGENT

```
public class CommunicationAgent extends AbstractAgent {

    /**
     *
     */
    private static final long serialVersionUID = 6366801477888787791L;
    private Logger log = Logger.getLogger(CommunicationAgent.class.getName());

    /*
     * (non-Javadoc)
     *
     * @see multi.agent.reporter.core.AbstractAgent#setup()
     */
    @Override
    protected void setup() {
        // TODO Auto-generated method stub
        register(CommunicationVocabulary.COMMUNICATION_SERVICE);
        this.addBehaviour(new CyclicBehaviour(this) {
            /**
             *
             */
```

```
                private static final long serialVersionUID =
1715824436050556157L;
                private MessageTemplate template = MessageTemplate
                        .MatchOntology(DeathsReporterOntology.getInstance()
                                .getName());
                                public void action() {
                        ACLMessage msg = myAgent.receive(template);
                        if (msg != null) {
                                Action actExpr;
                                try {
                                        actExpr = (Action)
myAgent.getContentManager()
                                                        .extractContent(msg);
                                        AgentAction action = (AgentAction)
actExpr.getAction();
                                        if (action instanceof MainDataForm) {
                                                MainDataForm form = ((MainDataForm)
action);
                                                SMS sms = form.generateSMS();
                                                actExpr.setAction(sms);
                                                sendMessage(actExpr,
ACLMessage.REQUEST, new AID[]{new AID("SMS-Agent", AID.ISLOCALNAME)});
                                                Email e = form.generateEmail();
                                                actExpr.setAction(e);
                                                sendMessage(actExpr,
ACLMessage.REQUEST, new AID[]{new AID("Email-Agent", AID.ISLOCALNAME)});
                                        }
                                } catch (Exception e) {
                                        log.error("Exception: " + e.getMessage(), e);
                                }
                        } else {
                                block();
                        }
                }

        });

    }

}
```