UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

AN AGENT-BASED WIRELESS SENSOR NETWORK SIMULATION FOR DAIRY CATTLE BREED SELECTION IN KENYA

BY

JOSEPH OCHIENG' OUMA

SUPERVISROR: DR. ELISHA T. OPIYO OMULO

Submitted in partial fulfillment for the requirements of Masters of Science in Computer Science

August 2012

DECLARATION

This project as presented in this report is my original work and has not been presented for any other institutional award.

Sign: ____________________ Date: 26/11/2012.

Joseph O. Ouma

P58/61998/2010

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

Sign: ____________________ Date: 26/11/2012

Dr. Elisha T. Opiyo Omulo

School of Computing and Informatics

University of Nairobi

ABSTRACT

With dairy farming an important activity in Kenya, it is logical that there is need to increase productivity of the sector in order boost the country's economy. In an effort to realize this, it is important that the farmers are able to easily make right choices when looking for cattle breeds to rear. This can be achieved by closely monitoring health status of the animals in an environment. With the combination of the agent and wireless sensor networks technologies, such health monitoring applications have been made a reality.

In this study, we designed and built a model for monitoring dairy cattle health with focus on body temperature variations. A wireless sensor network simulation was used to generate temperature readings for input into the model. The model consists of a team of intelligent agents that receive the simulated temperature readings, work collectively to analyze it and then display the health status of the animal to the user.

ACKNOWLEDGEMENTS

First, my appreciations go to the Lord Almighty, for enabling me to get this far with the quest for knowledge. Secondly, I would like to thank my supervisor, Dr. Opiyo and the entire panelists for their continued advice and support during this study. Lastly, much appreciation goes to my family for their support.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

LIST OF BBREVIATIONS AND ACRRONYMS

GDP	Gross Domestic Product
MAS	Multi Agent System
WSN	Wireless Sensor Networks
SA	Sensor Agent
AA	Analysis Agent
VA	Viewer Agent
OSI	Open System Interconnection
GoK	Government of Kenya
KDB	Kenya Dairy Board
THI	Temperature Humidity Index
GPS	Global Positioning Systems
RFID	Radio Frequency Identification
PDA	Personal Digital Assistant
CLDC	Connected Limited Device Configuration
CDC	Connected Device Configuration

CHAPTER 1

INTRODUCTION

1.1 Background

In Kenya, Dairy farming is an important agricultural activity and contributes about 14% of agricultural GDP and 3.5% of total GDP [17]. Dairy cattle are very sensitive to heat stress and when exposed to excessive heat, may lead to a significant economic impact to the farmer [27]. Farmers are therefore faced with a challenge of choosing dairy cattle that will be most suitable for their farming environment in order to increase their profits and reduce costs. The heat stress in cattle can be determined using the temperatures and humidity in an area [30].

There is need to employ agent technology to help farmers in making right choices of cattle breeds to rear. An agent has been described as computer system that exists in some environment and is capable of taking some autonomous actions in this environment in order to achieve its design goal [3]. This proposed agent based system will enable the users to easily select the right dairy cattle breeds and thus maximizing their productivity. The system monitors the variations in body temperatures of different dairy cattle breeds in an environment. Sensors in a wireless sensor network collects physiological data of each animal then sends the data to a mobile gateway. The sensed data is then sent to an agent who analyses it and then classifies it as normal or abnormal. The analyzed data is then presented to the user of the system by the viewer agent. The presented data can be used determine which breed is more adaptive to the environment and thus helps the user to choose the right breeds for their activity area.

1.2 Problem statement

The main goal of this study is to come up with an agent based system that will assist users/farmers in making the right choice between any presented dairy cattle breeds based the adaptability of the animal to the environmental conditions of a farming area. In this study, we'll use the agent based approach to monitor the variations in the body temperature of the cattle breed in an environment. The temperature readings of the animals are sensed and transmitted to the system using wireless sensor networks. The agents will be developed using the java platform after which it will be implemented in a prototype and then tested or evaluated in view to the effectiveness, accuracy and ease of use in trying to choose the right dairy cattle breeds for various environments.

1.3 Objectives

The objectives of this research are:-

- To design and build a model for selecting more profitable dairy breeds in any particular environment using java platform.
- Generate sensor data through simulating wireless sensor network.
- Run simulation experiments to determine appropriate dairy breed selection.
- To analyze the results of the tests and report findings of the study.

1.4 Justification of study

In real life, humans face great challenges in trying to make decisions that they face on a daily basis, all the time. In trying to maximize the outputs or profits earned from their dairy animals, small scale farmers in Kenya, who are the majority [4], need to be provided with a means of identifying dairy breeds that are most suitable for their farming regions (environments). Agent technology is a useful resource in health monitoring and thus provides a great potential in the dairy industry. Agents can be employed to carry out continuous monitoring of dairy cattle body temperatures which is important in investigating the adaptability of the cattle to an environment.

1.5 Limitations and Assumption

There were some assumptions made in the process of this study. These includes:-

- There already exists a Wireless Sensor Network for collecting physiological data of dairy animals
- There already exists mobile phones that can support agent programming and sensor network gateway approach

1.6 Chapter summary

This work presents an agent based system for monitoring the health (body temperature) of Dairy cattle. It is s organized into seven chapters. In the beginning, a review of available literatures is done on Chapter 2 after which the methodology adopted to solve the problem is described in

Chapter 3. Chapter 4 for presents the project management plan and Conclusions follow on chapters 5.

CHAPTER 2

LITERATURE REVIEW

2.1 Agent Technology

2.1.1 An Agent

Weiss [3] defines an agent as a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. An agent can be perceived as taking sensory input from its environment, and producing as output actions that affect it. This interaction with the environment is continuous and non-terminating as can be seen in the diagram below.

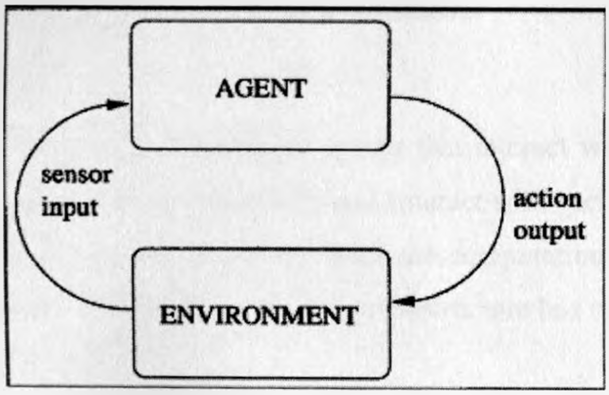


Figure 2-1: An agent in its environment (Source: [3])

2.1.2: Intelligent agent

[3] also defines intelligent agents as those that have got three characteristics. These characteristics include:

- Reactivity: They are able to perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy their design objectives
- Pro-activeness: They are able to generate and attempt to achieve a goal due to an own initiative
- Social ability: Capable of taking each other into account when trying to achieve a goal sometimes through negotiation, cooperation etc.

2.1.3: Agents with states

It is possible to model an agent decision action function model as from a sequence of environmental states /percepts to actions. The behavior of such agent can be summarized as below:-

- The agent starts in some initial internal state i^0 It then observes its environment state s^l and generates a percept see(s).
- The internal state of the agent is then updated via the next function, becoming set to next(i^0, see(s)).
- The action selected by the agent is then action(next(i^0, see(s))). This action is then performed, and the agent enters another cycle, perceiving the world via see, updating its state via next, and choosing an action to perform via action.

2.1.4: Multi Agent Systems

This is a term used to describe a group of intelligent agents that interact with each other in an environment. The agents are able to operate effectively and interact with each other productively [3]. The environment, in which the agents reside, provides the computational infrastructure for the agents to communicate/interact with one another. The infrastructure has two protocols to help the agents communicate and interact.

2.1.4.1 Characteristics of MAS environments:

Weiss presents three characteristics of a multi agent environment namely:

- They provide an infrastructure specifying communication and interaction protocols.
- They are typically open and have no centralized designer
- They contain agents that are autonomous and distributed, and may be self-interested or cooperative.

2.1.4.2: Evaluation Criteria

Six criteria for evaluating protocols are discussed in [3]. These includes:-

a. Social Welfare

This refers to the sum of all agents' utilities in a given solution and helps when measuring the global good of an agent.

This can be used when comparing alternative mechanisms by comparing the solutions got from each of the mechanisms.

b. Pareto efficiency:

A solution X is said to be pareto-efficent if there's no any other solution , X' in which an agent is better off than in X, and there's no agent worse off in 'X' than in X. It measures the global good.

c. Individual Rationality:

This occurs when the payoff after negotiation is better than that achieved without a negotiation. A self-interested agent will participate in a negotiation only if the negotiated solution is individually rational to them

d. Stability:

The mechanism/protocol among self-interested agents should be designed such that they cannot be manipulated. All the agents should be motivated to behave in a particular manner.

i. Nash Equilibrium:

Two strategies s^1 and s^2 are said to be in a Nash equilibrium if:

Under the assumption that agent I plays s^1, agent j can do no better than play s^2; and
Under the assumption that agent j plays s^2, agent i can do no better than play s^1.
Neither agent has any incentive to deviate from a Nash equilibrium.
The equilibrium approach has two limitations in that:

- Some interaction scenarios don't have a Nash equilibrium.
- Some interaction scenarios have more than one Nash equilibrium.

e. Computational Efficiency:

Mechanisms should be designed to minimize the computations done by the agents using them. The lower the computational overhead, the better. This is at times traded off with the quality of the solution.

f. Distribution and Communication Efficiency

Distributed protocols are preferred to centralized ones to avoid single point of failures; however, communication overheads should be minimized.

2.1.5: Agent Communications

In a multi-agent setting, agents need to communicate or interact with each other to achieve their goals. Weiss [3] describes an agent as an active object with ability to perceive reason and react. An agent is perceived to have two abilities namely perception (receive messages) and action (send messages).

Coordination: In order to achieve better goals, either self-interest goals or goal of the society they exist, agent need to communicate. This enables them to coordinate their behavior or actions and hence a more coherent system. Coordination is important when performing action in a shared environment to reduce resource contention, avoid livelock and deadlock and maintain applicable safety conditions [3]. Cooperation: Coordination among nonantagomnistic agents while negotiation is coordination among self-interested agents.

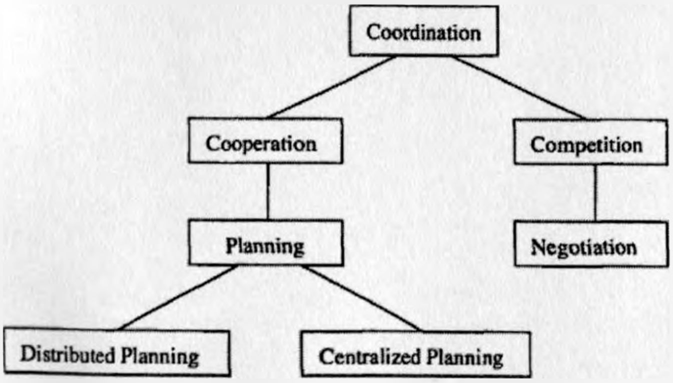


Figure 2-2: Taxonomy of agent's behavior and activities coordination. (Source: [3])

2.1.6: Prometheus Methodology

In this agent development, the agent development process is divided into three phases namely system specification, architectural design and then the detailed design [13].

2.1.6.1 System Specification

At this phase, the system is specified using goals and use-case scenarios. The system's interface to its environment is described in terms of percepts, actions and external data. This phase also involve describing the functionalities of the proposed system.

2.1.6.2 Architectural Design

In this phase, agent types are identified and the overall structure of the system is captured in a system overview diagram. The use-case diagrams from the system specification stage are developed into interaction protocols.

2.1.6.3 Detailed Design

This is the phase where an agent's internals are defined and developed in terms of capabilities, data, events and plans. Process diagrams here serve as a stepping stone between interaction protocols and plans.

The figure 2-3 below shows an overview of the Prometheus Methodology

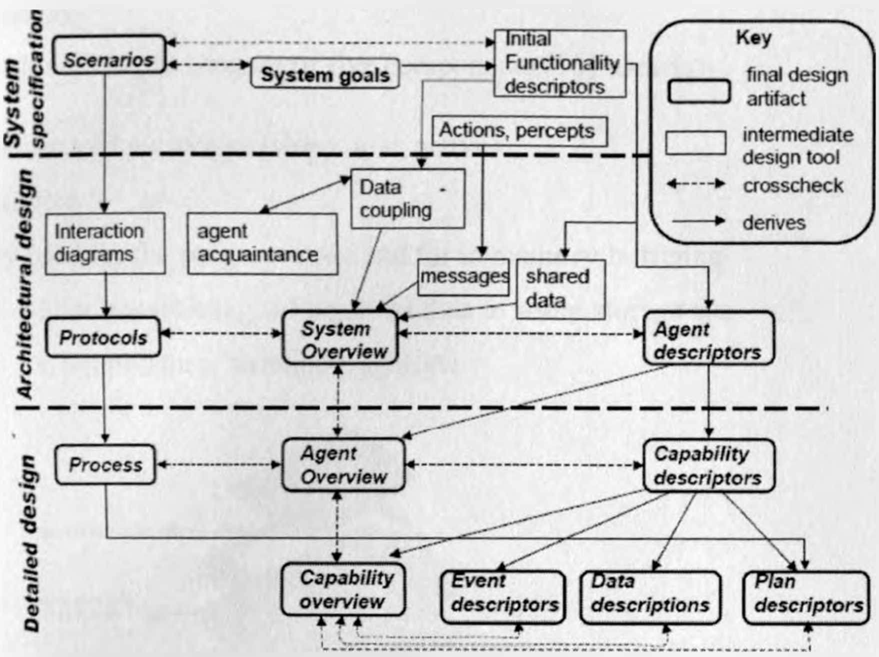


Figure 2-3: Overview of the Prometheus Methodology (Source: [13])

2.2 Wireless Sensor Networks

2.2.1 Introduction

A wireless sensor network (WSN) is as a wireless network which consists of equally distributed autonomous devices using sensors capable of monitoring the physical or environmental conditions such as temperature, sound, vibration, pressure, motion or pollutants, at various different locations [5]. The sensor nodes are often small in size and have got limited individual computing capabilities [6].

With the current technological advances, each of the node are tiny, inexpensive, have got limited/low processing power, permit very low energy consumption and perform limited and specific monitoring and sensing functions [8].

WSNs are used in smart environments to gather information [9]. Smart environments mark the next revolutionary development in the automation of computerized systems and are particularly important in cases where wired connections are impractical.

2.2.2: Node Anatomy

A sensor network node/mote consists of five components [10] namely:-

- Processor in various modes (sleep, idle, active)
- Power source
- Memory used for the program code and for in memory buffering
- Radio used for transmitting the acquired data to some storage site
- Sensors for temperature, humidity, light, etc

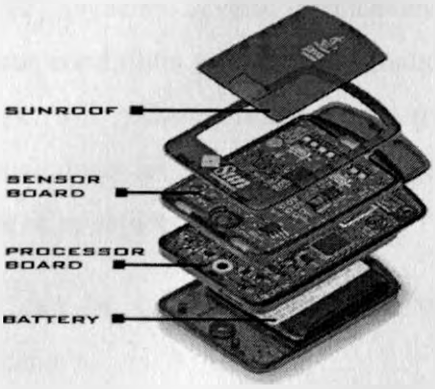


Figure 2-4: Anatomy of Sunspot (Source: [16])

2.2.3: Multihop Wireless Sensor Network Architecture

In multihop routing protocol, each of the sensor nodes of a WSN generates a data packet of fixed length and transmits towards the base station. Each is able to aggregate the incoming data packet with its own and then transmits towards the base station. The base station receives data from each sensor at the end of a data gathering round [11].

Obidul and Hajid [11] present a genetic algorithm based multi-hop routing protocol that is able of maximizing the lifetime of the network in terms of first node death. A transmission schedule is generated consisting of a collection of transmission rounds. The schedule denotes how data is collected from each sensor and eventually propagated to the base station. It is a collection of routing paths to be used by the network in order to maximize its lifetime.

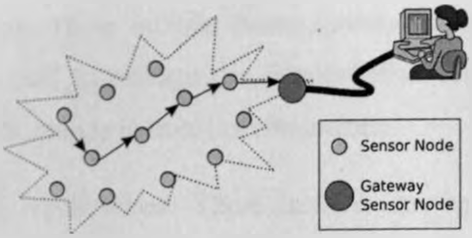


Figure 2-5: Typical Multihop WSN Architecture (Source: [14])

2.2.4: Some Applications of Sensors Network

Akyildiz et al [12] presents several applications of the sensor networks. These includes monitoring ambient conditions such as temperature, humidity, vehicular movement, lightning condition, pressure, soil makeup, noise levels, the presence or absence of certain kinds of objects, mechanical stress levels on attached objects, and the current characteristics such as direction, and size of an object.

The applications can be categorized into military, environment, health, home and other commercial applications.

a. Health Application: WSNs have got a wide range of use in this field. Some of these are Telemonitoring of human physiological data where the installed sensor can monitor and detect a patient's physiological data e.g. temperature. WSN and wearable sensor devices are also used to monitor animal health.

b. Military Applications: In this respect, the networks have been used in command, control, communications, computing, intelligence, .surveillance, reconnaissance and targeting (C4ISRT) systems. Since sensors are densely deployed, low-cost and disposable, destruction of some sensor cannot hinder an ongoing military operation. The sensor in military are used to monitor friendly forces, equipment and ammunition; battlefield surveillance ; reconnaissance of opposing forces and terrain; targeting; battle damage assessment; and nuclear, biological and chemical (NBC) attack detection and reconnaissance.

c. Environment Applications: In this category, the WSN have been used in several areas including animal, bird or insect movement tracking.

d. Home Applications: These include home automation where sensor embedded in home appliances can be used to manage the devices either remotely or locally via the internet. Another home application is in smart environments.

e. Other commercial Applications: These includes among others environmental control in office buildings, Interactive museums, Detecting and monitoring car thefts, Managing inventory control and Vehicle tracking and detection

2.2.5: Sensor Network Communication Architecture:

[12] presents communication architecture for the sensor networks with sensor nodes are scattered in a sensor field. Each of the nodes collects data and route it to the sink node and the end users. This can be shown in the figure 2-6 below:

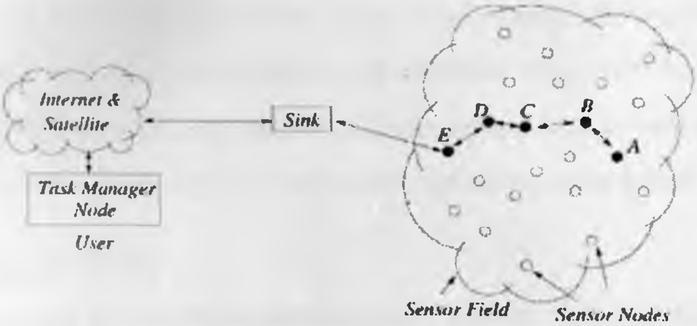


Figure 2-6: Sensor nodes scattered in a sensor field (Source: [12])

2.2.5.1: Communication Protocol Specification:

Pillay and Muddenahalli [6] present a four layer OSI model of communication protocol. The four layers are the physical layer, MAC layer, Network and the Application Layer. These function as discussed below:

a. Physical layer: The layer has a transceiver that determines how information is transmitted across the medium. This layer is responsible for inter-node communications

b. MAC Layer: This layer coordinates when nodes are to transmit/receive packets and when they become active or go to sleep.

i. Node identification: There is a unique identification for each node in the network. The ID is stored within the node memory.

ii. Node discovery and synchronization: All the nodes are mobile and thus the network topology is not fixed. Nodes are therefore expected to perform node discovery so as to find the next hop/node for data transmission. Whenever an inactive node becomes active, it tries to find or discover its neighbours.

iii. Node discovery by base station: The network base stations send out a broadcast message regularly and nodes within the range will receive the packet and try to communicate with the base station(handshaking

c. Network Layer: Three packets are exchanged at this layer namely the broadcast, acknowledgement and the data packets.

Broadcast packets are sent out by nodes time to time to alert other nodes of their presence. Base station can also send out broadcasts at certain time intervals to alert nodes that that are within range its range. Acknowledgements allow nodes to determine whether they are in communication lock with each other or if their last sent message was received correctly. Data packets include sensed data as well as the time-stamp and transmitter node ID to assist in correct storage of the sensed data.

2.2.6: Technologies of WSN

Luis et al [23] presents two technologies commonly used by Wireless Sensor Network. These are the ZigBee and Bluetooth. These operate at Industrial and medical band of 2.4GHz and have got license-free operation, huge spectrum allocation and worldwide compatibility. Another possible technology is the Wi-Fi technology but it suffers limitations like high power consumption and has short autonomy of battery supply.

a. ZigBee: This is built on top of the IEEE 802.15.4 standard which defines the physical and MAC (Medium Access Control) layers for low personal wireless networks. It's low tier, terrestrial, wireless standard and similar to Bluetooth in some way (Ref9). Zigbee operates in the 868Mhz, 915Mhz and 2.4GHz ISM Bands.

Zigbee supports functionalities for channel selection, link quality selection, energy measurement and channel assessment. The network layer is in charge of organizing and providing routing over a multi-hop network, specifying different topologies. Applications include the agriculture and food industry, building control and automation, security, consumer electronics, personal computer peripherals, medical monitoring and toys.

b. Bluetooth: This was developed as a wireless protocol for short range communication in wireless personal area networks as a cable replacement for mobile devices. It is a low tier, ad hoc, terrestrial, wireless standard for short range communication and was designed for small and low cost devise with low power consumption. [24]. Bluetooth technology operates with three classes namely Class 1, Class 2 and Class 3 with ranges 100, 10 and 1 metres respectively

2.2.7: Mobile Phone as Sensor Network Gateway

There has been research on how to use mobile phone as a gateway in a sensor network. Mobile phones have been identified as a suitable gateway for a body sensor network [29]. In their paper, [25], Frank et al presents how sensor nodes can use mobile devices as data gateway or as an interface for retrieving data from homogenous sensor clusters. They advocate for the convergence of the mobile devices and the wireless sensor network technologies to create context aware services. Fummi et al [28] states that the mobile phone acts as the gateway between a wireless sensor network and the traditional communication network. Most commercially available mobile phones use Bluetooth to connect with the sensor [29].

2.2.7.1: Convergence Forms

[25] states that convergence of mobile devices with sensors has already been realized at some integration level by some phone manufacturing firms e.g Nokia and Apple. This convergence is shown in the figure 2-7 below.

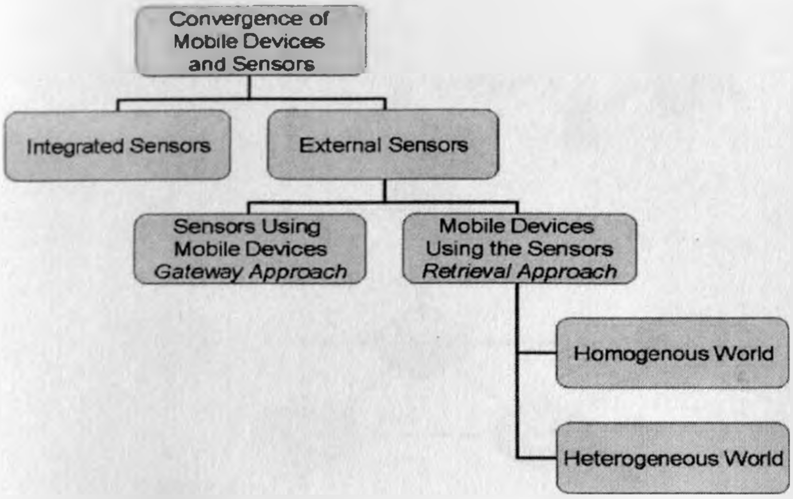

Convergence of Mobile Devices and Sensors
Integrated Sensors
External Sensors
Sensors Using Mobile Devices Gateway Approach
Mobile Devices Using the Sensors Retrieval Approach
Homogenous World
Heterogeneous World

Figure 2-7: Classification of convergence of mobile devices and sensor networks.

2.2.7.2: Gateway approach architectures

As depicted in the figure above, if sensors are using mobile devices as a gateway to transmit their sensed data, it is referred to as the gateway approach. This approach enables the sensors on a WSN to make use of the functionalities of the mobile device that the sensors lack.

Two architectures shown in figures 2-8 and 2-9 below are used by the sensors in communicating with the mobile gateway in the gateway approach [25].

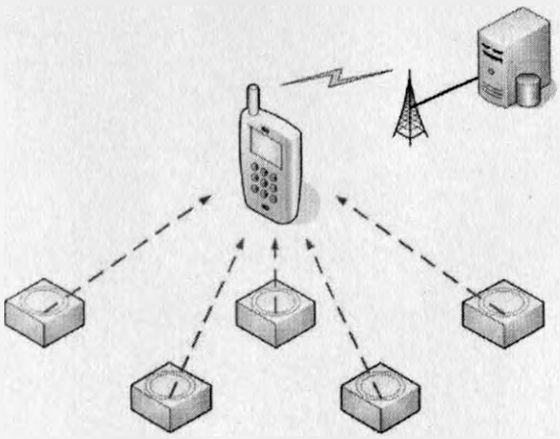


Figure 2-8: Gateway approach with heterogeneous sensors

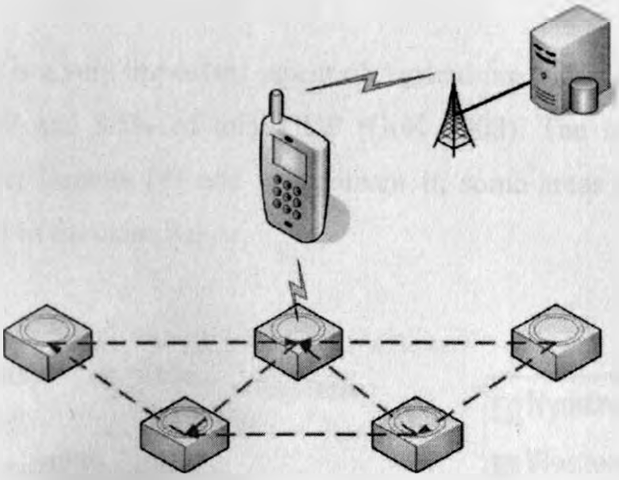


Figure 2-9: Gateway approach with homogenous sensors

In the first architecture, the sensors send their sensed data directly to the mobile gateway. This suffers drawback like huge network traffic, larger messages conveyed and smaller area coverage. The other approach has sensors communicating their sensed data among themselves and then a selected sensor send a request to the mobile device. This is generally better approach.

2.3 Dairy Farming

In Kenya, Dairy farming is a very important aspect of Agriculture and is said to contribute about 14% of agricultural GDP and 3.5% of total GDP (GoK 2008). The sector however remains dominated by smallholder farmers [4] and is dominant in some areas in Kenya compared to others as presented by [2] in the chart below.

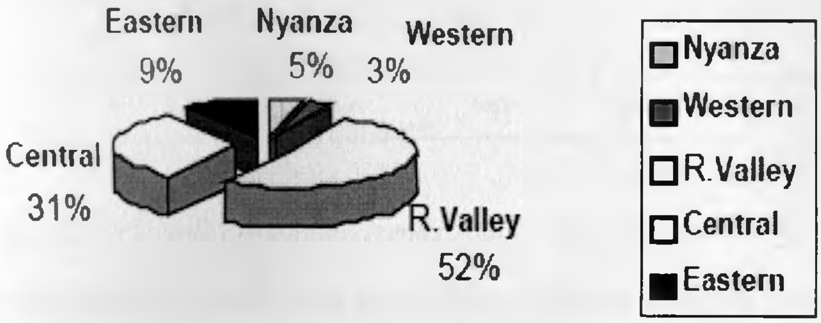


Figure 2-10: Dairy Cattle Distribution in Kenya.

The dairy sector is responsible for the production of milk consumed by the populations and therefore ensures nutritional needs of the population is catered for and also act as a source of income to the farmers as well as those employed in the sector. The sector also helps in earning foreign exchange as the products are exported to other countries. Smallholder dairy production contribute over 56% and 70% of total and marketed milk production, respectively , the productivity per animal in these farms, however still remains low [4].

[17] presents a steadily increasing milk production between 2000 – 2007. There was however a downward trend between 2007 and 2010 as shown in figure 2-11 below.

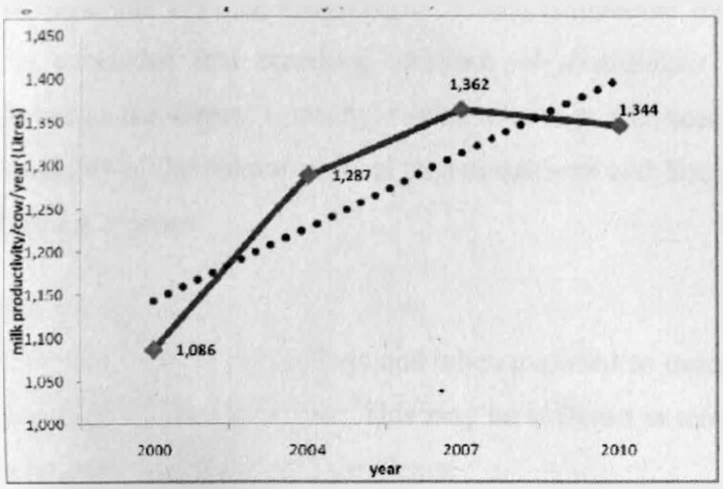


Figure 2-11: Milk production trends 2000 – 2011 (Source: [17])

The downward trend was mainly attributed to the prolonged droughts between the years 2008 and 2010. Kenya Dairy Board [7] presents a number of setbacks facing the sector. These includes poor quality and costly feed supplements , lack of an organized and reliable market, low quality breeding stock , high cost of animal health and diseases control services , over reliance on rains resulting to uneven milk supply throughout the year, lack of affordable credit facilities and high cost of Artificial Insemination services.

2.3.1 Dairy Breed Preference in Smallholder Farmers in Kenya

A study on the smallholder Central highlands farmer [19] reveals that, two dairy cattle breeds, Friesian and Ayrshire are the most preferred breeds for high milk yield. Ayrshire is however the most preferred over Friesian as it is more resistant to diseases and feeding behavior but not for market value and body weight. The other breeds preferred in that order are the Guernsey and Jersey.

The other factors considered while choosing Dairy breeds as presented in are the Lack of choice of semen, better looking of the animal and availability of extension service.

Bebe et al [20] finds out that the smallholder farmers considered milk production for feeding the families and cash income as the most important reasons for choosing particular breeds. Those keeping Friesian and Ayrshire breeds gave slight higher priority to milk production for Cash income whereas those that kept the Guernsey or Jersey considered feeding their families most important. It is concluded that breeding decisions of smallholder dairy farmers in Kenya highlands conforms to the farmer's multiple objectives e.g. the need to have increased milk production. adaptability of the animal to local feed conditions and diseases and provision of non market production e.g. manure

2.3.2: Heat Stress in Dairy Cows

Dairy cattle are very sensitive to heat stress and when exposed to excessive heat, may lead to a significant economic impact to the farmer. This may be suffered in terms of loss of productivity and milk quality as well as health related problems.

2.3.2.1: Defining Heat Stress

Larry, [30] present the impact of temperatures and the humidity levels on dairy cattle. He states that the conditions that induce heat stress in the cattle can be calculated using the temperature humidity index (THI). THI has been calculated using a number of equations. He present the following in his paper:

$$\text{THI} = (\text{Dry bulb temperature } ^{o}\text{C}) + (0.36^{*} \text{ dew point temperature } ^{o}\text{C}) + 41.2$$

Heat stress starts in cattle if the THI is greater than 72 as presented in the table 1 below. The characteristics exhibited by the affected animal are also stated against each THI level.

THI	Stress Level	Comments
<72	None	
72-79	Mild	Dairy cows adjust e.g. by moving to a shade, increasing their respiration rate and dilation of the blood vessels. Slight impact on milk production
80-89	Moderate	Increased saliva production and respiration rates. Possible low feed consumption with increased water intakes. Increased body temperatures. Decreased milk production and reproduction .
90 – 98	Severe	Cows very uncomfortable as body temperatures are high, higher respiration rates /panting and excessive saliva production. Production of milk and reproduction will be markedly decreased.
>98	Danger	Possible cow deaths can occur

Table 2-1: Effect of Heat Stress on Dairy Cattle

[30] also present a number of changes that occur in the Dairy animals as a result of heat stress. These includes:-

- Increased body temperatures (> 102.5 °F , normal is 101.5 °F)
- Increase rates of respirations (70-8/min)
- Increase energy requirement for maintenance – the cattle activates mechanism to get the extra energy needed to maintain body temperatures
- Feed nutrient utilization - Increased loss of sodium and potassium
- Decreased dry matter intake
- Decreased milk production
- Decreased reproductive performance

It has also been suggested the heat stress and its behavioral consequences increases risk of contracting diseases in dairy cattle e.g. rumen acidosis

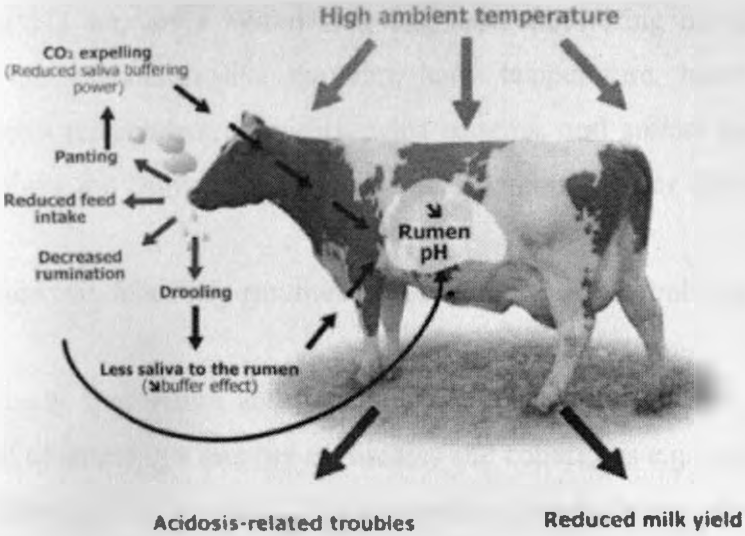


Figure 2-12: Effect of heat stress on dairy cattle (Source: [27])

2.4 Related Work

Various studies have been carried out on how MAS and WSNs can be used in health monitoring applications. In this section, we look at a few examples that put our study into practicality:

2.4.1: Intelligent Agent based routing in Wireless Body Sensor Mesh Network

Kevin Miller, et al [18] presents an agent based patient health monitoring that uses wireless sensor networks to collect real time patient physiographic data (Temperature).

In their architecture, wireless body sensor based mesh networks consists of spatially distributed autonomous devices that make use of cooperating body sensors in monitoring a patient's physiological data in various wards of a hospital. The wearable devices/sensors transmit the data to the nearest access point /mesh nodes at the wards. The Mesh nodes use multi-hop routing to inform mobile devices e.g. PDA. In this architecture, the intelligent agent is mainly used as a middleware in acquiring and transmitting the sensed information. The goal is to inform health professional/doctor of the patient's condition wherever they are on their hand held devices e.g. a PDA.

2.4.2: Ambulatory Instrumentation Suitable for Long-Term Monitoring of Cattle Health

Schoenig et al [1] presents a system for long term monitoring of cattle health. Taken into consideration are parameters like the core body temperature, heart rate, respiratory rate, behavior, ambient temperature, humidity, wind patterns, and animal identification. These data are collected from the animals using a wearable wireless sensor device that is worn on the animal.

[1] explains how the following parameters can be used in determining the health status of an animal:-

a. Core body temperature and Thermographic profiles: This has been investigated as a method of detecting a number of diseases and conditions e.g. estrus, respiratory diseases, and thermal stress in cattle. Thermographic investigations show animals with higher surface temperature tend to be healthier and higher weight gain. This method is also useful when screening for respiratory diseases that deteriorates animal health and hence performance.

b. Ambient Temperature, Humidity, and Wind: The environment of an the cattle is important and determines the animal stress. Heat and cold stressors lead to diseases, decreased performance, and the readiness to mate and conceive.

c. Heart Rate: Variance of the heart rates can symbolize stress or change in health state of an animal. The thickness of the cattle hide makes in uneasy to obtain heart rate in cattle by using electrodes and thus leaving implantable electrodes as the viable solution. This however may lead to infections and at time biological reaction to the materials used.

d. Respiratory Rate: This can be used to recognize illness and stress in cattle. This can be obtained through use of pressure transducers that measures changes in thoracic volume.

e. Food and Water Consumption: Health animal usually spend most time at the feed bunk as compared to ill ones. They also tend to consume more water regularly. By monitoring this behavior, ill animals can be identified from the healthy ones.

f. Electronic Identification: Radio Frequency Identification (RFID) tags are ideal for identifying animals and maintaining the animal/data associations.

g. Movement: The GPS (Global Positioning Systems) devices can be used to detect reduced movements of cattle and thus early detection of morbidity.

2.5 Jade Leap

2.5.1 Overview

Jade Leap is an environment for developing multi-agent systems. It can be defined as a set of Java classes that allow a developer to build a FIPA-compliant multi-agent system quite easily [21]. Jade provides a set of graphical tools that facilitate the complex task of implementing a multi-agent system.

Jade leap when combined with jade forms a modified runtime environment that can be deployed in a wide range of devices ranging from servers to java enabled mobile phones [22]. To achieve this, jade-leap is configured in three different ways as per the three java environments namely:-

- **j2se**: runs on PCs and servers in the fixed network running JDK1.2 or later.
- **pjava**: runs on handheld devices supporting Personal Java such as most of today PDAs.
- **midp**: runs on handheld devices supporting MIDP1.0 only such as the great majority of Java enabled cell phones.

The jade-leap runtime environment is shown in the figure 2-13 below. In the figure there is a multi-agent system with six agents. Two of them are running on the main container of the platform, which is connected through Internet with a container on a PC that holds another two agents. These four agents can communicate wirelessly with two agents running on mobile devices (PDA and a mobile phone). Each mobile device has one container.

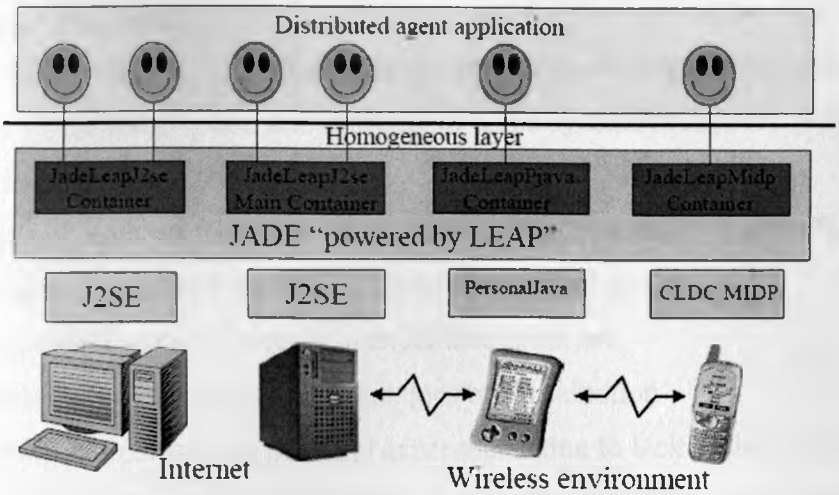


Figure 2-13: Jade-Leap Runtime environment [22]

The three environments offer the same API to the developers and therefore and therefore a programmer is able to deploy their Jade agent on Jade-Leap and vice versa, without changing a single line of code.

2.5.2 Java J2ME/Java ME

This is a java technology that was created to deal limitations associated with developing applications for small devices [32].

2.5.2.1 Configurations in J2ME

Two configurations are supported by this technology, namely Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC) [26].

a. CLDC

The CLDC targets the low end of consumer electronics range. A typical platform used by CLDC is the cell phone or PDA with limited memory. This technology is closely related to Wireless Java.

b. CDC

This is used by devices that lie between those used by CLDC and the full desktop systems running J2SE. The devices have more memory as compared to those of CLDC. Examples of supported devices high-end PDAs and smart phones

2.6 Modeling and Simulation

According to Anu Maria [39], modeling is the process of producing a model whereas a model is a representation of the construction and working of some system of interest. Models are similar to but simpler than the system they represent.

A simulation of a system on the other hand, refers to the operation of a model of the system. Models can be reconfigured and are usually used for experimentation.

The reasons for using simulation models in experimentation are

- Unavailability of real time system that does breed selection
- Impossibility of performing physical experiments due to lack of the system
- The cost of acquiring wireless sensors and collecting data from actual cattle is too high
- The limited time scales available

Steps in developing a simulation model are as follows:

- Identify the problem.
- Formulate the problem.
- Collect and process real system data.
- Formulate and develop a model.
- Validate the model.
- Document model for future use.
- Select appropriate experimental design.
- Establish experimental conditions for runs.
- Perform simulation runs.
- Interpret and present results.
- Recommend further course of action.

2.6.1 Steps of a simulation study

As Perros [38] states, simulation model is, in general, used in order to study real-life systems which do not currently exist. He presents the steps of carrying out such a study as shown in the figure below.

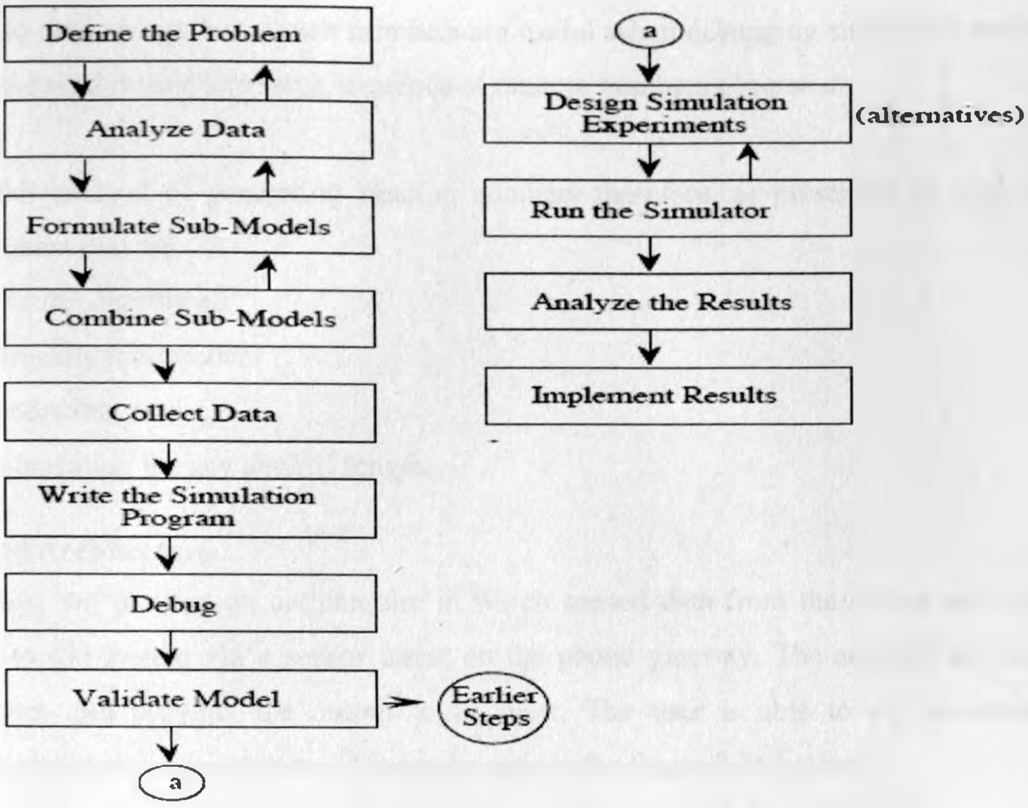


Figure 2-14: Steps of simulation study (Source [38])

2.6.2 Variable types

[38] presents two variable types. Those that are affected and those that are not affected during a simulation run. A variable whose value is not affected is called exogenous. A variable having a value determined by other variables during the course of the simulation is called endogenous.

2.6.3 Generation of pseudo random numbers

Random numbers are used in order to introduce randomness in the model. [38] also indicates that when physical phenomenon is used as a source of randomness from where random numbers can be generated, then such random numbers are called true random numbers. Such generator requires a completely unpredictable and nonreproducible source of randomness.

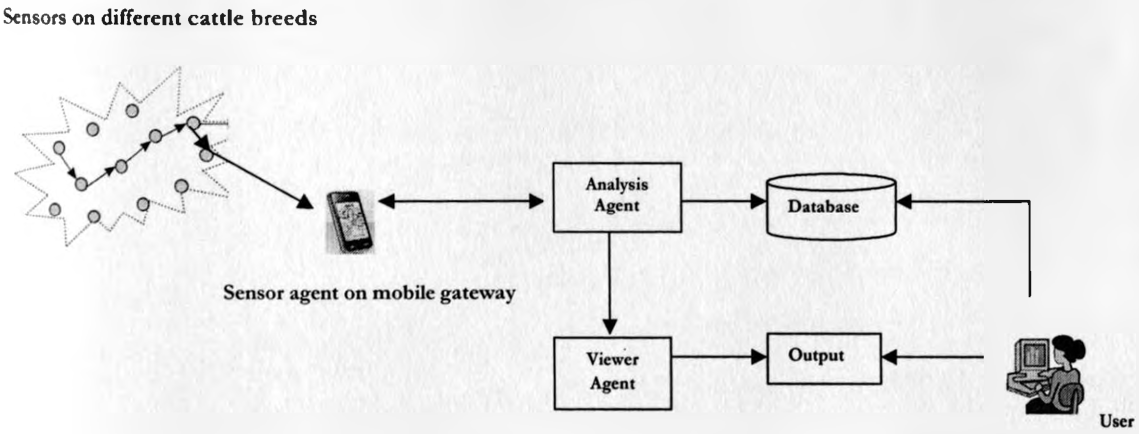
Another way of generating random numbers is the use of mathematical algorithm. Efficient algorithms have been developed that can be easily implemented in a computer program to generate a string of random numbers. Such algorithms use a seed to generate random values called pseudo-random numbers. Such numbers are useful when debugging simulation programs since the same seed is used and same sequence of random numbers generated.

An acceptable method of generating random numbers therefore, as presented by [38], must produce numbers that are

- Uniformly distributed
- Statistically independent
- Reproducible, and
- Non-repeating for any desired length.

2.7 Proposed Architecture

For this study, we propose an architecture in which sensed data from the sensor networks is transmitted to the system via a sensor agent on the phone gateway. The analysis and viewer agent analyses and presents the output to the user. The user is able to set the threshold temperature readings for the system. This can be seen in the figure 2-14 below.



Sensors on different cattle breeds
Sensor agent on mobile gateway
Analysis Agent
Database
Viewer Agent
Output
User

Figure 2-15: The proposed architecture

2.8 Chapter summary

In this chapter, we have presented a review of the available literature on the technologies available as well as the dairy farming practices in Kenya. The section starts by looking at the agent technology then wireless sensor technology followed by the dairy farming in Kenya and then a review of past related studies available and lastly, the Jade and Leap Addon development environments.

CHAPTER 3

METHODOLOGY

3.1 Overview

In this chapter, we present a step by step guide on how the study will be carried out i.e. the way we will systematically solve the research problem [33]. In this section, we'll explain the research design, sources of data, methods and tools to be used in data collection and the methods to be used in data analysis.

To realize the main goal of the study, a prototype of the system will be developed and used for experimentation. Data collected from the prototype is analyzed.

This chapter is divided into four sections namely:-

- Research Design – The different types of designs that will be used in the project e. the system design.
- Tools Design and skills needed – Involves looking at the tools and skills necessary in carrying out the research.
- System Design – The design of the agent based prototype and demonstrating how it works.
- Implementation and Evaluation – The implementation and evaluation of the agent based approach.

3.2 Research Design

Here, we present how we intend to collect the relevant data, analyze it and interpret the observations. As Kothari [33] states, this is the "the arrangement of conditions for collection and analysis of data in a manner that aims to combine relevance to the research purpose with economy in procedure"

3.2.1 Problem Design

We conducted a simple literature survey on how farmers decide on which dairy cattle breeds to rear in their farms. It was realized that farmers lack enough appropriate information concerning the animal breeds adaptability to help the make such important decisions. We therefore decided to use the agent based approach to ensure such information is available to the farmers.

3.2.2 Tools and Skills Required

To accomplish the goal of this study, the following will be needed in terms of tools and skills:

- Windows OS (At least XP SP2)
- Java JDK 1.6.0.10 or later
- Java EE SDK 6ul
- Java Net Beans
- Java Programming skills
- System analysis and design skills
- Reference materials (Books, journals etc)

3.2.3 System Design

Design of the agent based prototype will be guided by the Prometheus Methodology (discussed in Chapter 2 above). This will involve three steps namely system specification, architectural design then the detailed design.

System Specification, will involve the four activities namely: - initial description of the system, outlining the goals of the system, identification of the actors involved, designing use-case diagram to show the interaction between the actors and the system.

In Architectural Design, we'll identify the agents in the multi-agent system, their roles and also the interactions between them. The percepts, action and goals of the various agents will be described here. The systems' overview diagram showing the sensor input, multi-agent system and human user interaction will be designed here as well as the technological infrastructure.

Detailed Design will involve detailed description of each agent's capabilities. In this section, we'll present the capabilities of each of the agents in the system in details.

3.2.4 Experimental and data sources design

To set up the prototype for testing, we send sensed data from the sensor simulator to the analysis agent on the main pc. The sent data will be similar to body temperature reading that can be obtained from a healthy or heat stressed (unhealthy) dairy cattle.

3.3 Data Sources and Collection

In the study, we will make use of both primary and secondary data sources. Primary data will be mainly collected through experiment observations. Observing the outcomes of the prototype based on the corresponding sensor data inputs will be critical in achieving the goal of the study. Observing the outcomes of the prototype based on the corresponding sensor data inputs will be critical in achieving the goal of the study. Secondary data will mainly be input to the simulator obtained from past surveys of cattle temperatures.

3.4 Prototype Implementation and Testing

The implementation of the prototype shall be done using the Java Net beans and Jade and the Leap add-on as described below:

- Java Net Beans:- For building the main environment where agents reside
- Jade and Leap Add-on: For developing the agents in the system.

To enable testing of the system, java runtime environment has to be installed on the computer used. We then run the prototype by invoking the agents and the outcome observed and recorded or tabulated.

3.5 Data Analysis

After recording the outcomes of the prototype run, they are tabulated and the results represented in the form of bar graphs that are easily interpreted by the users. These will then be evaluated to determine the relevance of the study in relations to finding a solution for our problem.

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 Overview

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

4.2. System Specification

As outlined by Michael and Lin [13], our system specifications involved three activities, namely:-

- Determining the systems interface to the environment
- Determining the systems goals and functionalities
- Determining scenarios that captures the system's usage

4.2.1 Initial Description of the System

Farmers need to be aware of dairy cattle breeds that can perform best in their areas of farming. They therefore need a system that can monitor the animal breeds and report on the most suitable ones for a given environment. In a conventional system, a farmer would perform the following two tasks when looking for a dairy animal to purchase:

- Get information on the best affordable breeds available
- Purchase the cattle for rearing

4.2.2 Goals of the system

The main goal of our system is to enable the users/farmers to make the best choice when looking for a dairy cattle breed to rear in an environment and thus high productivity. The sub goals of the system are to increase productivity of the dairy animals and reduce operating costs of the farmers as shown in the goal diagram below.

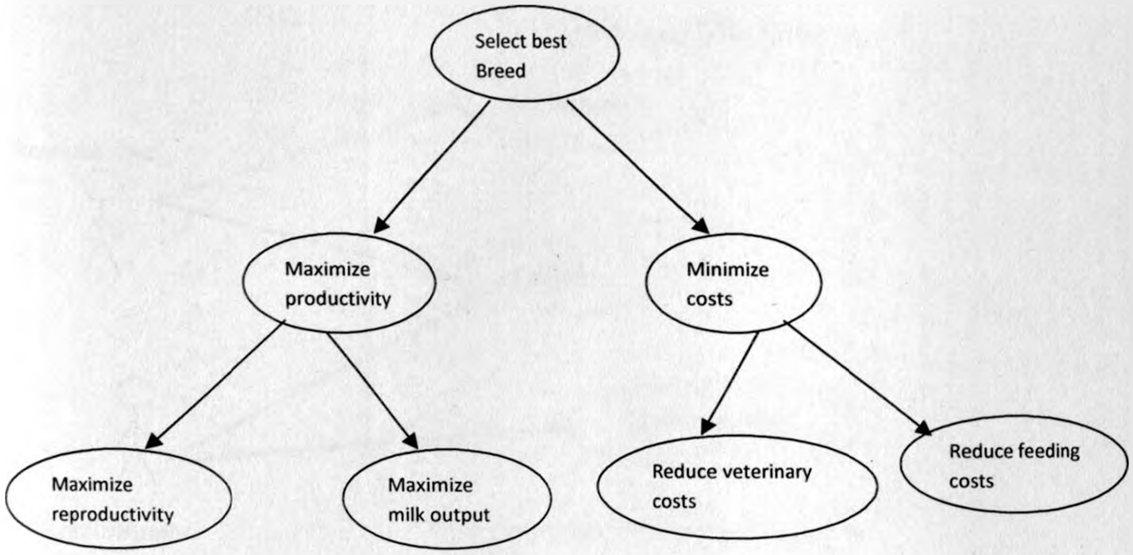


Figure 4-1: System Goal Diagram

4.2.3 Actors

The main actors identified in this multi agent based system are the farmers and Dairy research institutions. In this system, we'll refer to all of them as users.

4.2.4 Use-case diagram and scenario

The multi agent based system receives data from the sensors through mobile gateway and makes appropriate decision based on this data. It then passes the decision made to the end user for appropriate decision making. The system basically receives input from wireless sensor networks and then carries out some data analysis after which it passes the outcome to the end-user (human) who uses the information to make appropriate decisions.

The system's interaction with its data collecting sensors and users (environment) is depicted in the use-case diagram below.

a. System use-case diagram

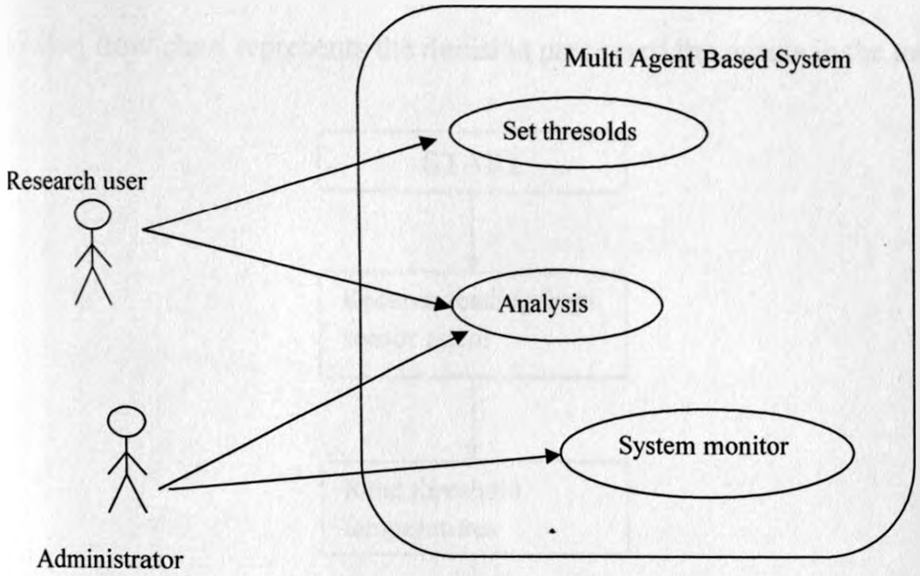


Figure 4-2: Use-case diagram

b. System use case scenario

Scenario: Breed Selection

Overview: The user aims to choose the most suitable dairy cattle breed for their farming regions. The various breeds available are monitored for body temperature variations and the user selects the best one.

Context: Assumes the various breeds are available and in the same environment.

Steps:

1. EVENT: Breed Selection
2. Monitor the available dairy animal breeds for body temperature variations
3. Obtain preferred breed option
4. Acquire the selected breed

c. System Flow Chart

The following flow chart represents the decision process of the agents in the multi agent system.

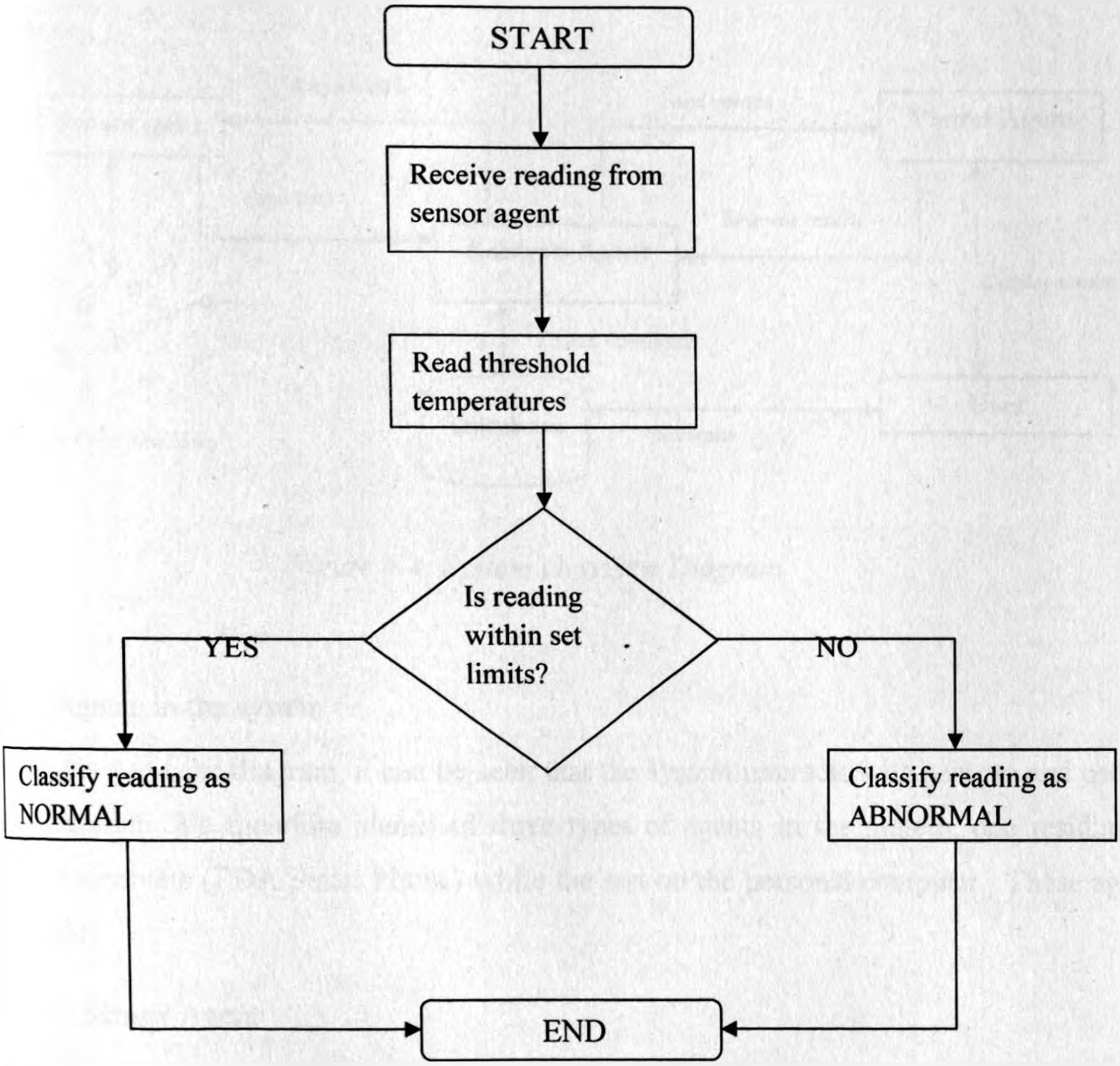

START
Receive reading from sensor agent
Read threshold temperatures
Is reading within set limits?
YES
NO
Classify reading as NORMAL
Classify reading as ABNORMAL
END

Figure 4-3: Agent decision process flow chart

4.3 Architectural Design

4.3.1 System Overview Diagram

The system overview diagram below represents the interactions between agents and the system user.

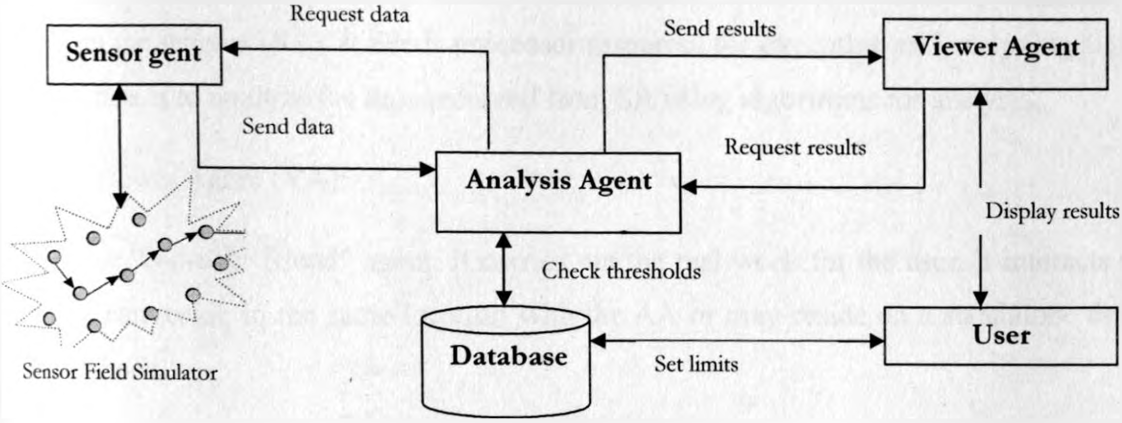


Figure 4-4: System Overview Diagram

4.3.2 Agents in the system

From the use-case diagram, it can be seen that the system interacts with sensors and users in its environment. We therefore identified three types of agents in the system, one residing in the sensor receivers (PDA/Smart Phone) while the rest on the personal computer. These agents are namely:-

- Sensor Agent
- Analysis Agent
- Viewer Agent

i. Sensor Agent (SA):

This agent is located where there is a device which can measure data from sensor(s). It resides in a container inside a mobile device. Its main duty is to receive the measurement data from the sensor and then send it to any other agent who could be interested in receiving such data e.g. the Analysis Agent.

ii. Analysis Agent (AA):

This agent resides where there is enough bandwidth available for collecting huge amounts of data from the sensors (SA). It needs processor resources for executing and analyzing algorithms. Its main task is to analyze the data received from SA using algorithms for analysis.

iii. Viewer Agent (VA):

This is the "end-user friend" agent. It carries out the real work for the user. It interacts with the user and can reside in the same location with the AA or may reside on a standalone device e.g. mobile phone.

The table below describes the percepts, action and goals of the various agents described above.

Agent Type	*Percept*	*Actions*	*Goal*
Sensor Agent	Sensor temperature readings.	Receive and buffer sensor readings Send received sensor readings to the Analysis Agent	Send sensor temperature readings to Analysis Agent
Analysis Agent	Data from the Sensor Agent.	Queue data from Sensor Agent for processing Execute an analyzing algorithm to classify data received Send analysis results to the Viewer Agent	Produce analyzed data for Viewer Agent
View Agent	Analyzed data from the Analysis Agent	Request analyzed data from the Analysis Agent Present the analysis results to user	To fulfill user requests by showing current state of required measurement

Table 4-1: Table showing the agents in the system

4.3.3 Interaction between agents and the user

Agents in the system generally interact in their environment. The Sensor Agent passes sensor data to the Analysis Agent which analyzes the data and then passes the results to the Viewer Agent which presents the results to the end-user of the system. Communication between agents involves more than just simple messages [15]. The agents make use of message passing in a peer to peer architecture.

Messages passed in the system carries ID of the sender which is useful in managing the data exchanged. The user of the system sets the threshold or limits within which readings from the SA are expected to fall. This is used by the Analysis Agent.

Interaction between agents and the user can basically be modeled as shown in figure 15 below:

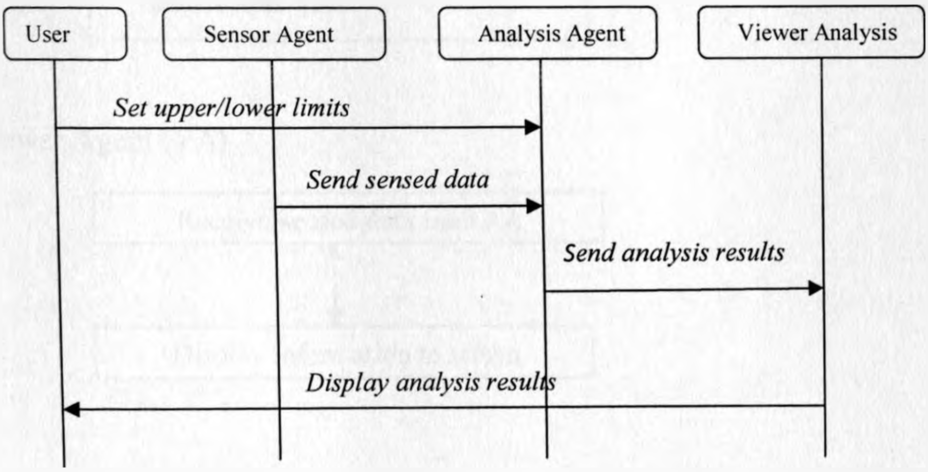


Figure 4-5: Agent Interaction

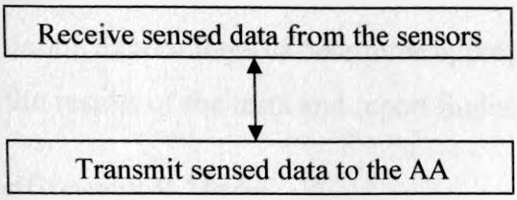
4.4 Detailed Design

4.4.1 Agents capabilities

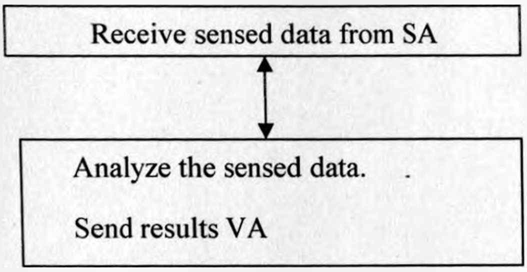
Each of the agents in the proposed system has distinct capabilities defined for them. In this section, we'll present the capabilities of each of the agents in the system.

As depicted in table 1 above, each of the agent capabilities can be portrayed diagrammatically as follows:

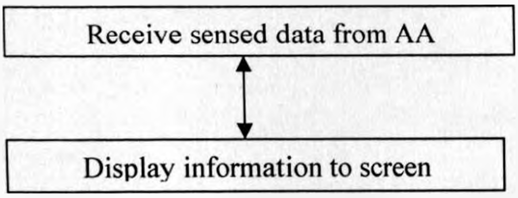
i. Sensor Agent (SA)


Receive sensed data from the sensors
Transmit sensed data to the AA

ii. Analysis Agent (AA)



iii. Viewer Agent (VA)



4.5 Conceptual Model of the Simulation

4.5.1 Reasons for simulations

- Unavailability of real time system that does breed selection
- Impossibility of performing physical experiments due to lack of the system
- The cost of acquiring wireless sensors and collecting data from actual cattle is too high
- The limited time scales available

4.5.2 Modeling Objectives

From the reason for simulation above, we have the following modeling objectives

- To create an experimental environment that reflect the real time system for experimentation
- To run simulation experiments to determine appropriate dairy breed selection
- To analyze the results of the tests and report findings of the simulation study.

4.5.3 The Simulator Conceptual Model

The following chart represents the conceptual model of the simulation:

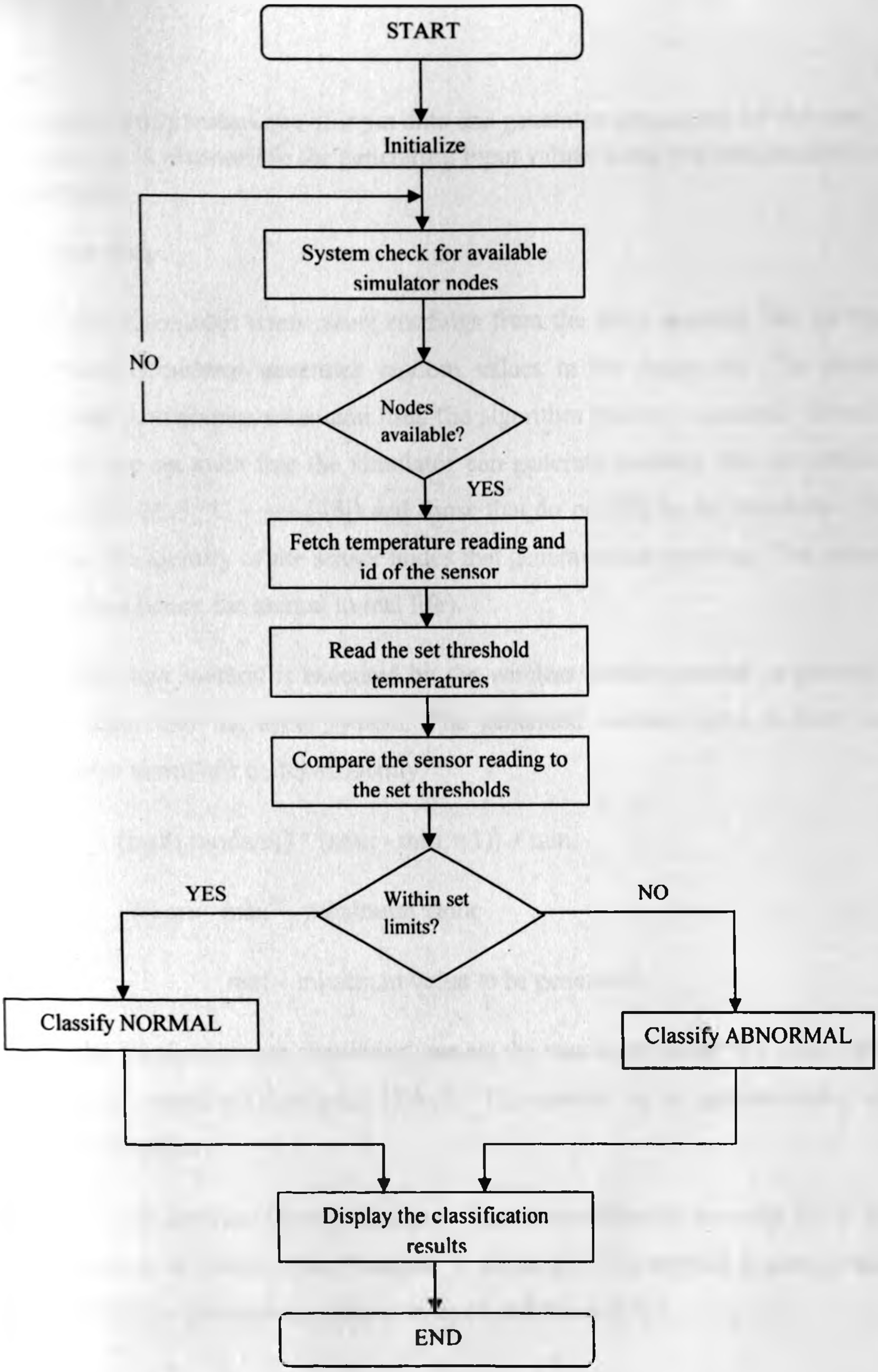

START
Initialize
System check for available simulator nodes
NO
Nodes available?
YES
Fetch temperature reading and id of the sensor
Read the set threshold temperatures
Compare the sensor reading to the set thresholds
YES
Within set limits?
NO
Classify NORMAL
Classify ABNORMAL
Display the classification results
END

Figure 4-6: Conceptual Model of the Simulator

4.6 Data

Our simulation study makes use of input data and generates output data for the user. The wireless sensor simulator is responsible for generating input values using the math.random() algorithm as discussed below.

a. Input data

The input data resembles temperature readings from the dairy animals. For the simulation, the sensor network simulator generates random values in the range set. The simulator takes a maximum and a minimum value and uses the algorithm below to generate values in the range. These values are set such that the simulator can generate readings that are within the allowed thresholds (38.0-39.5 °C – see [35]) and those that do not fall in the threshold. The input data also contains the identity of the sensor nodes that generated the readings. This is used to identify the sensors (and hence the animal in real life).

The following java method is executed by the wireless sensor network to generate the random values for input into the agent system. The generated random value is then attached to the wireless sensor simulator nodes randomly.

```
(math.random() * (max - min + 1)) + min;
```

Where max - maximum value

min - minimum value to be generated.

To set up the wireless sensor simulation, we set the maximum value to a value (40.0^0C) slightly higher than the upper set threshold, 39.5 °C. This enables us to generate some values that fall outside the set limits.

The method math.random() generates pseudo-random numbers in the range [0,1). Every time the simulator calls it, it creates a new number in the range. The method is advised for use as basis for randomness for simulations, games, images, and more [37]

b. Output Data

The agents in the system receive the input readings above and output a classification of the readings. The model classifies readings based on (38.0-39.5 °C).

If the received reading falls in the range, the agent classifies it as NORMAL. If it does not fall in the range (greater than 39.50C), the agents classify it as ABNORMAL, indicating the animal is heat stressed.

c. Results Analysis

The output data from the simulation are collected and analyzed in form of tables and graphs. The simulation is run several times and all output tabled. The results are then summarized to get a more accurate outcome as shown in chapter 5 section 5.3 on table 5-2.

4.7 Verification and Validation

As [35] states, a healthy dairy cow has a normal body temperature reading of between (38.0-39.5 °C). In our simulation, the random values are generated and input into the agents system from the wireless sensor simulator using the method discussed in section (a) above.

Any such values that are within that range are classified by the analysis agent in the model as NORMAL and anything above 39.5 °C as ABNORMAL.

[30] states that, when the temperatures go beyond the upper limit (39.5^0C), the animal suffers heat stress and its productivity is lowered. Our simulation therefore classifies such readings as ABNORMAL.

4.8 Experimentation

To carry out the experimentation, about 100 simulation runs were performed and the outcomes obtained, tabled and analyzed (see chapter 5 sections 5.3). Each simulation run consisted of 10 steps and lasted about 8 minutes. The steps followed in performing experimentation have been included in the appendix III.

CHAPTER 5

SYSTEM IMPLEMENTATION AND TESTING

5.1 Overview

For ease of implementation, the system was broken down into smaller modules which were then coded separately. These were then integrated to come up with the final system. Samples of the code have been included in the appendix section.

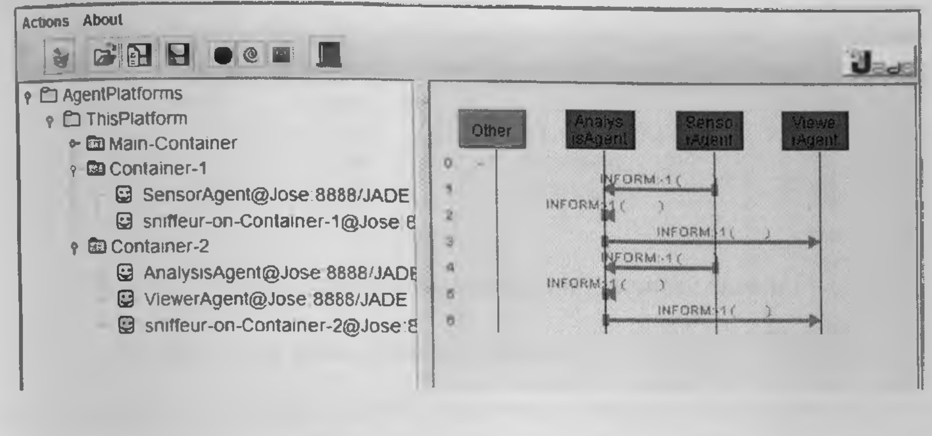
5.2 Testing, Result and Discussions

For testing purposes, we subjected the multi agent based system to various conditions to confirm if it works properly. The sensor inputs depicted temperature readings that can be obtained from dairy animals that are healthy or unhealthy. The outcomes were discussed in the following sections.

5.2.1 Sensor Network Simulator

For testing the system, we developed a simulator of a wireless sensor network (based on [34]) that generates random temperature readings in the range that can be obtained from a healthy and heat stressed animal. This is a freeware that is used to develop senor network simulators that generate random nodes and assign them values (readings). A healthy animal's body temperature ranges about 38.0-39.5 °C [35]. Any reading generated by the sensor within this range would be classified as normal whereas anything outside the range is classified as abnormal.

To carry out our experiments, we ran the simulator and then invoked the agents in the system. Using sniffer agent window, interactions between agents in the system are observed. This can be seen in the sniffer agent diagram below.



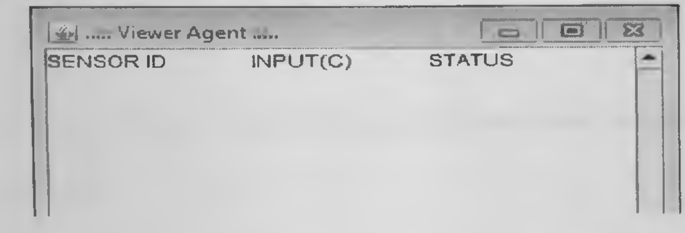
```
AgentPlatforms
  ThisPlatform
    Main-Container
    Container-1
      SensorAgent@Jose 8888/JADE
      sniffeur-on-Container-1@Jose 8
    Container-2
      AnalysisAgent@Jose 8888/JADE
      ViewerAgent@Jose 8888/JADE
      sniffeur-on-Container-2@Jose 8
```

Figure 5-1: Sniffer Agent Window

The Viewer Agent will then display the results of the received sensor readings in a window that is friendlier to the user as will be shown in the following sections.

5.2.1 Testing with no sensor reading

With no sensor inputs, the system delivers no outputs. This can be shown in then figure below.

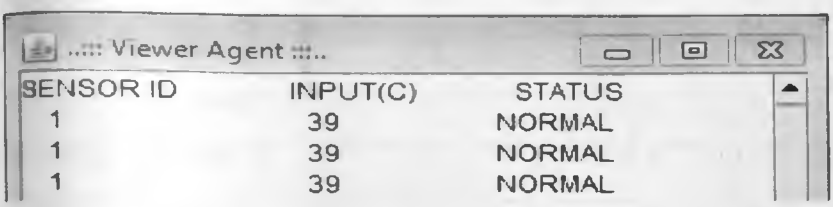


```
SENSOR ID          INPUT(C)         STATUS
```

Figure 5-2: Viewer Agent with no input from Sensors

5.2.2 Testing with sensor reading from a healthy animal

For a healthy animal, the sensors deliver temperature readings that fall within the set thresholds. In our example, the sensor node of id 1 delivered a reading of 39^{0}C which fall in the defined temperature range. The system therefore delivers a NORMAL output. This can be shown in then figure below.

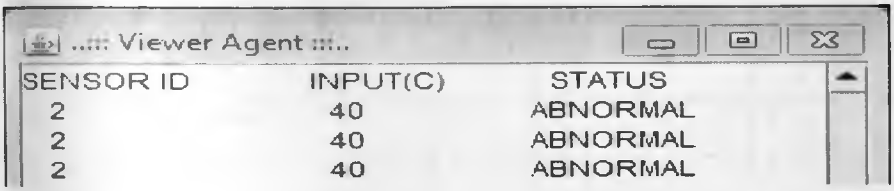


```
SENSOR ID          INPUT(C)        STATUS
   1                  39          NORMAL
   1                  39          NORMAL
   1                  39          NORMAL
```

Figure 5-3: Viewer agent with output from a healthy animal

5.2.3 Testing with sensor reading from unhealthy animal

For a unhealthy animal, the sensors deliver temperature readings that do not fall within the set thresholds. In our testing example, node 2 of the simulator relayed a reading of 40^0C which is outside the set range. The system therefore delivers an ABNORMAL output as can be shown in the figure below.



```
SENSOR ID          INPUT(C)        STATUS
   2                  40          ABNORMAL
   2                  40          ABNORMAL
   2                  40          ABNORMAL
```

Figure 5-4: Viewer agent with output from a heat stressed animal

5.3 Simulation Results

The simulator was tested for accuracy by exposing it to a wide range of readings and then analyzing the outputs in table form. The table below represents outcomes obtained from simulation runs of 10 steps, each lasting about 5 minutes. A normal reading represents readings that fall in the range 38.0-39.5 °C [35]. Abnormal readings fall outside the range.

Run	Reading	Classification	Sensor ID							
			0	1	2	3	8	10	11	12
Run 1	38.0-39.5 °C	Normal	2	4	3	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 2	38.0-39.5 °C	Normal	2	0	0	0	0	0	0	0

	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 3	38.0-39.5 °C	Normal	2	2	1	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	1	0	0	0	0	0
Run 4	38.0-39.5 °C	Normal	0	2	0	0	0	4	4	0
	> 39.5 °C	Abnormal	0	0	0	0	0	3	0	0
Run 5	38.0-39.5 °C	Normal	2	0	0	0	0	6	2	0
	> 39.5 °C	Abnormal	0	0	0	0	0	5	0	0
Run 6	38.0-39.5 °C	Normal	4	7	0	0	0	4	3	0
	> 39.5 °C	Abnormal	0	0	0	0	0	3	0	0
Run 7	38.0-39.5 °C	Normal	0	0	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 8	38.0-39.5 °C	Normal	8	1	1	4	0	0	0	0
	> 39.5 °C	Abnormal	0	0	1	0	0	0	0	0
Run 9	38.0-39.5 °C	Normal	1	6	0	0	8	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 10	38.0-39.5 °C	Normal	0	0	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 11	38.0-39.5 °C	Normal	0	2	1	5	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 12	38.0-39.5 °C	Normal	4	2	0	0	0	1	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 13	38.0-39.5 °C	Normal	3	4	0	0	8	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 14	38.0-39.5 °C	Normal	0	1	1	5	0	0	0	0
	> 39.5 °C	Abnormal	0	0	1	0	0	0	0	0
Run 15	38.0-39.5 °C	Normal	0	4	3	0	0	0	3	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 16	38.0-39.5 °C	Normal	4	2	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 17	38.0-39.5 °C	Normal	4	4	0	0	0	3	5	0

	> 39.5 °C Abnormal	0	0	0	0	0	2	0	0
Run 18	38.0-39.5 °C Normal	0	6	4	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	2	0	0	0	0	0
Run 19	38.0-39.5 °C Normal	2	4	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 20	38.0-39.5 °C Normal	9	2	2	1	0	0	0	0
	> 39.5 °C Abnormal	0	0	2	0	0	0	0	0
Run 21	38.0-39.5 °C Normal	3	2	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 22	38.0-39.5 °C Normal	4	4	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 23	38.0-39.5 °C Normal	0	6	0	0	0	3	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	3	0	0
Run 24	38.0-39.5 °C Normal	5	1	4	0	1	1	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 25	38.0-39.5 °C Normal	4	5	0	0	0	7	0	3
	> 39.5 °C Abnormal	0	0	0	0	0	5	0	0
Run 26	38.0-39.5 °C Normal	0	8	2	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 27	38.0-39.5 °C Normal	2	0	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 28	38.0-39.5 °C Normal	4	2	4	4	0	0	0	0
	> 39.5 °C Abnormal	0	0	3	0	0	0	0	0
Run 29	38.0-39.5 °C Normal	0	6	3	2	0	0	0	0
	> 39.5 °C Abnormal	0	0	3	0	0	0	0	0
Run 30	38.0-39.5 °C Normal	4	0	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 31	38.0-39.5 °C Normal	0	0	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 32	38.0-39.5 °C Normal	4	1	6	0	0	0	3	0

	> 39.5 °C Abnormal	0	0	5	0	0	0	0	0
Run 33	38.0-39.5 °C Normal	0	3	2	3	0	0	3	0
	> 39.5 °C Abnormal	0	0	0	5	0	0	0	0
Run 34	38.0-39.5 °C Normal	4	1	3	0	0	4	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 35	38.0-39.5 °C Normal	0	4	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 36	38.0-39.5 °C Normal	0	5	4	2	0	0	0	0
	> 39.5 °C Abnormal	0	0	3	0	0	0	0	0
Run 37	38.0-39.5 °C Normal	0	6	5	4	0	0	0	0
	> 39.5 °C Abnormal	0	0	4	0	0	0	0	0
Run 38	38.0-39.5 °C Normal	0	4	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 39	38.0-39.5 °C Normal	4	0	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 40	38.0-39.5 °C Normal	4	1	4	2	0	0	0	0
	> 39.5 °C Abnormal	0	0	3	0	0	0	0	0
Run 41	38.0-39.5 °C Normal	7	0	0	0	8	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	4	0	0	0
Run 42	38.0-39.5 °C Normal	3	0	0	0	4	3	0	0
	> 39.5 °C Abnormal	0	0	0	0	3	0	0	0
Run 43	38.0-39.5 °C Normal	0	2	0	0	0	4	4	0
	> 39.5 °C Abnormal	0	0	0	0	0	3	0	0
Run 44	38.0-39.5 °C Normal	2	4	0	0	3	2	0	0
	> 39.5 °C Abnormal	0	0	0	0	2	2	0	0
Run 45	38.0-39.5 °C Normal	0	3	1	0	0	3	8	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 46	38.0-39.5 °C Normal	0	8	0	0	0	0	0	0
	> 39.5 °C Abnormal	0	0	0	0	0	0	0	0
Run 47	38.0-39.5 °C Normal	2	6	3	0	4	0	6	0

	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 48	38.0-39.5 °C	Normal	1	6	0	0	0	4	6	0
	> 39.5 °C	Abnormal	0	0	0	0	0	3	0	0
Run 49	38.0-39.5 °C	Normal	4	2	3	3	0	0	0	0
	> 39.5 °C	Abnormal	0	0	3	0	0	0	0	0
Run 50	38.0-39.5 °C	Normal	0	1	0	3	0	6	2	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	2	0
Run 51	38.0-39.5 °C	Normal	0	2	5	1	0	3	1	1
	> 39.5 °C	Abnormal	0	0	4	1	0	1	1	1
Run 52	38.0-39.5 °C	Normal	0	3	2	3	0	0	0	0
	> 39.5 °C	Abnormal	0	0	2	0	0	0	0	0
Run 53	38.0-39.5 °C	Normal	4	8	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 54	38.0-39.5 °C	Normal	3	1	8	1	0	0	0	0
	> 39.5 °C	Abnormal	0	0	5	0	0	0	0	0
Run 55	38.0-39.5 °C	Normal	0	0	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 56	38.0-39.5 °C	Normal	4	2	3	0	0	0	4	0
	> 39.5 °C	Abnormal	0	0	3	0	0	0	0	0
Run 57	38.0-39.5 °C	Normal	0	10	0	2	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	2	0	0	0	0
Run 58	38.0-39.5 °C	Normal	0	2	0	7	0	0	0	3
	> 39.5 °C	Abnormal	0	0	0	5	0	0	0	0
Run 59	38.0-39.5 °C	Normal	0	7	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 60	38.0-39.5 °C	Normal	4	1	3	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	3	0	0	0	0	0
Run 61	38.0-39.5 °C	Normal	4	0	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 62	38.0-39.5 °C	Normal	0	5	4	0	0	0	4	0

	> 39.5 °C	Abnormal	0	0	3	0	0	0	0	0
Run 63	38.0-39.5 °C	Normal	0	3	2	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	1	0	0	0	0	0
Run 64	38.0-39.5 °C	Normal	0	4	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 65	38.0-39.5 °C	Normal	4	7	2	2	0	0	0	0
	> 39.5 °C	Abnormal	0	0	2	0	0	0	0	0
Run 66	38.0-39.5 °C	Normal	1	2	0	0	0	6	0	4
	> 39.5 °C	Abnormal	0	0	0	0	0	4	0	0
Run 67	38.0-39.5 °C	Normal	0	5	0	0	0	4	4	0
	> 39.5 °C	Abnormal	0	0	0	0	0	3	0	0
Run 68	38.0-39.5 °C	Normal	0	2	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 69	38.0-39.5 °C	Normal	4	3	5	0	0	0	3	0
	> 39.5 °C	Abnormal	0	0	4	0	0	0	0	0
Run 70	38.0-39.5 °C	Normal	4	4	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 71	38.0-39.5 °C	Normal	4	0	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 72	38.0-39.5 °C	Normal	4	4	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 73	38.0-39.5 °C	Normal	2	2	0	0	2	2	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	4	0	0
Run 74	38.0-39.5 °C	Normal	4	1	3	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	3	0	0	0	0	0
Run 75	38.0-39.5 °C	Normal	8	8	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 76	38.0-39.5 °C	Normal	4	3	2	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	1	0	0	0	0	0
Run 77	38.0-39.5 °C	Normal	0	2	1	3	0	0	0	0

	> 39.5 °C	Abnormal	0	0	1	0	0	0	0	0
Run 78	38.0-39.5 °C	Normal	0	6	0	0	0	5	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	4	0	0
Run 79	38.0-39.5 °C	Normal	4	2	0	0	0	5	5	0
	> 39.5 °C	Abnormal	0	0	0	0	0	4	0	0
Run 80	38.0-39.5 °C	Normal	0	6	0	0	0	2	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	2	0	0
Run 81	38.0-39.5 °C	Normal	4	2	6	1	0	0	0	0
	> 39.5 °C	Abnormal	0	0	4	0	0	0	0	0
Run 82	38.0-39.5 °C	Normal	0	2	5	0	0	0	3	0
	> 39.5 °C	Abnormal	0	0	4	0	0	0	0	0
Run 83	38.0-39.5 °C	Normal	2	0	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 84	38.0-39.5 °C	Normal	0	1	2	0	0	0	0	5
	> 39.5 °C	Abnormal	0	0	2	0	0	0	0	0
Run 85	38.0-39.5 °C	Normal	1	3	0	0	2	0	9	4
	> 39.5 °C	Abnormal	0	0	0	0	2	0	6	5
Run 86	38.0-39.5 °C	Normal	0	4	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 87	38.0-39.5 °C	Normal	8	1	1	3	0	0	0	0
	> 39.5 °C	Abnormal	0	0	1	0	0	0	0	0
Run 88	38.0-39.5 °C	Normal	12	2	4	2	6	0	0	0
	> 39.5 °C	Abnormal	0	0	0	2	0	0	0	0
Run 89	38.0-39.5 °C	Normal	0	3	3	0	0	0	4	0
	> 39.5 °C	Abnormal	0	0	3	0	0	0	0	0
Run 90	38.0-39.5 °C	Normal	0	1	3	0	0	0	3	0
	> 39.5 °C	Abnormal	0	0	3	0	0	0	0	0
Run 91	38.0-39.5 °C	Normal	0	5	2	0	0	0	3	0
	> 39.5 °C	Abnormal	0	0	2	0	0	0	0	0
Run 92	38.0-39.5 °C	Normal	4	2	0	0	0	0	0	0

	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 93	38.0-39.5 °C	Normal	4	12	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 94	38.0-39.5 °C	Normal	0	0	0	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	0	0	0
Run 95	38.0-39.5 °C	Normal	1	6	0	0	2	0	5	0
	> 39.5 °C	Abnormal	0	0	0	0	0	3	3	0
Run 96	38.0-39.5 °C	Normal	0	14	1	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	1	0	0	0	0	0
Run 97	38.0-39.5 °C	Normal	0	3	0	0	0	5	5	0
	> 39.5 °C	Abnormal	0	0	0	0	0	3	0	0
Run 98	38.0-39.5 °C	Normal	0	5	0	0	0	7	0	0
	> 39.5 °C	Abnormal	0	0	0	0	0	5	0	0
Run 99	38.0-39.5 °C	Normal	0	5	2	0	0	0	0	0
	> 39.5 °C	Abnormal	0	0	2	0	0	0	0	0
Run 100	38.0-39.5 °C	Normal	1	5	0	0	0	5	3	0
	> 39.5 °C	Abnormal	0	0	0	0	0	4	0	0

Table 5-1: Simulation Results

Summary of the outputs for sensors 0, 1 and 2 are as shown in the table below:

Sensor ID	Readings in normal range (38.0-39.5 °C)	Readings above normal range (> 39.5 °C)	Percentage readings in normal range (%) (38.0-39.5 °C)	Percentage readings above normal range (%) (> 39.5 °C)	Total Readings
0	207	0	100	0	207
1	320	0	100	0	320
2	129	83	73.4	26.6	312

Table 5-2: Simulation Results Summary

The graph in the figure below shows summarizes the output data for the three sensors (sensors id 0, 1 and 2) in the given simulation environment.

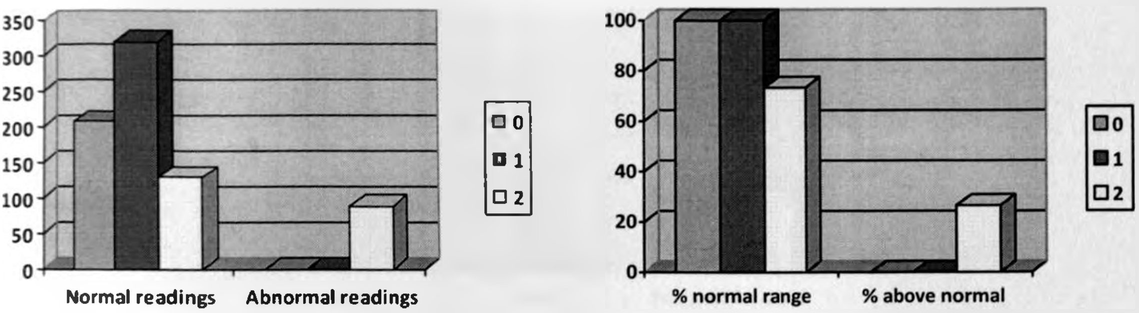


Figure 5-5: Graphs comparing the simulation outcomes

The results reveal that the breeds with sensors of id 0 and 1 are more favorable in this case compared to 2. The breed with sensor id 2 has less chances of doing well in the environment. In real life, a sensor attached to an animal would be identified using their ids (RFID).

5.5 System testing for user acceptability

After implementation, we engaged sample users to verify the acceptability of the proposed system. We sampled eleven potential users and demonstrated to them how the proposed system works. We then carried out a survey to determine their perspective of the system.

Six out of the eleven users think the system will improve their productivity. Two did not understand while three doubted the importance. The main concern of the users is the funds needed and the knowledge required using the system. Six users were concerned that the system is not as user friendly.

Total test users	Think it can work	Doubt it can work	Don't understand
11	6	3	2

Table 5-2: User acceptance

These samples are presented graphically as below: -

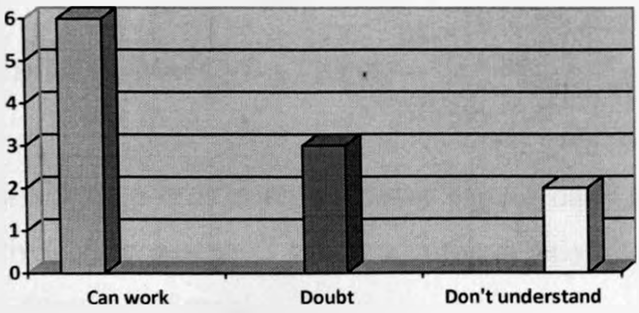


Figure 5-6: Graph showing system acceptability

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusion

Selecting suitable dairy cattle breeds is an important part of dairy farming, if farmers are to increase their productivity. The process is however not very easy to carry out as it is currently fully reliant on human guesses in Kenya.

In this study, we have proposed an agent based system to monitor dairy cattle body temperature variations in an environment with an aim of aiding in selection of the right breeds for an environment. The main goal of this study was to develop a multi agent based model that can assist in selection of the suitable dairy cattle breeds for a given farming environment by closely monitoring their health – body temperature variation. Schoenig [1] discusses the use of core body temperature and thermographic profile to detect various diseases and heat stress in dairy animals. In his study, the animals have wearable wireless sensor devices worn on the animals (nodes). The agents in our model interactively performed the functions of collecting, analyzing and presenting data from the sensor networks nodes. Thus, this was a clear demonstration of how the agent technology can be utilized to increase our dairy sector productivity.

6.2 Limitations

The main limitation of the solution proposed by this simulation study is the high cost of deploying sensor network. Also, the short battery life of the sensors is a real issue.

6.3 Future work

The time available was not adequate to do a full analysis of the system. Therefore, some future work is left that has to be done. Some areas that will need to be covered if this system was to be fully developed are the security of data, loosing of messages. The system also has to be implemented for more detailed testing and verifying that all the components work together. Future development should also incorporate automatic data export to an analysis tool e.g. excel for easy output data analysis.

EFERENCES

1] S. A. Schoenig, T. S. Hildreth, L. Nagl, H. Erickson, M. Spire, D. Andresen, and S. Warren, "*Ambulatory Instrumentation Suitable for Long-Term Monitoring of Cattle Health*" 2004.
Url: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1403689 – Last visited Feb 5th, 2011

2] Export Processing Zones Authority Report, *"Dairy Industry in Kenya"*, 2005, pp 3
Url: http://www.epzakenya.com/UserFiles/File/DairyReport.pdf - Last visited Feb 1th, 2011

3] G. Weiss, "*Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence"*. MIT Press, 1999, pp 29 – 31, 81 -84, 201 – 204.

4] A. M. Karanja, "*The dairy industry in Kenya: the post-liberalization agenda"*,
In proceedings of dairy industry stakeholders workshop held in Nairobi, Kenya, 2003, pp 1
Url: http://www.aec.msu.edu/fs2/kenya/o_papers/dairy_sector_color.pdf - Last visited Feb 5th, 2011

[5] A. Victor, A. Khader, C. Rao and A. Mehta, "*Build an IEEE 802.15.4 Wireless Sensor Network for Emergency Response Notification for Indoor Situations"*, 2008, pp 1

[6] N.V. Pillay and T. Muddenahalli N., " *Design and Implementation of Wireless Sensor Network for Wildlife Monitoring (May 2009)" , 2009*
Url: http://www.satnac.org.za/proceedings/2009/papers/access/Paper%204.pdf – Last visited 01/03/2012

[7] http://kenyadairy.com/m/report/1279104320-377335-KENYA_DAIRY_BOARD.pdf - Last visited Mar 3rd, 2012

[8] W. Dirk, G. Joao, S. Amardeo *"Security Solutions for Wireless Sensor Networks"*, 2006, pp 2
Url: http://www.ist-ubisecsens.org/publications/SecuritySolutionsWSN.pdf - Last visited 25th Feb, 2012

[9] F. L. Lewis " *Wireless Sensor Networks"* To appear in Smart Environments: Technologies, Protocols, and Applications, John Wiley, 2004, pp1

Url: http://arri.uta.edu/acs/networks/WirelessSensorNetChap04.pdf - Last visited Feb 25th, 2012

Z. Marco, *"Introduction to wireless sensor networks"*, 2011
Url: http://www.6deploy.eu/workshops2/20111010_guayaquil_ecuador/Guayaquil-IP.pdf - last visited Jan 25th, 2012

] I. Obidul and H. Sajid, *"An Intelligent Multi-hop Routing for Wireless Sensor Networks"*, 2006
Url: http://cold.fisk.edu/sajid/publications/shussain-intelligent-wsn.pdf - last visited Feb 25th, 2012

] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, *" Wireless sensor networks: a survey"*, Elsevier Science B.V., 2002, pp 395 - 408

] W. Michael and P. Lin, *"The prometheus methodology"*,
Url: http://www.cs.rmit.edu.au/agents/www/papers/mseas04-wp.pdf - Last visited Feb 25th, 2012

4] M.Reddy, "File:WSN.svg" in Wikipedia, Free Encyclopedia , 2007. [Online].

Url: http://en.wikipedia.org/wiki/File:WSN.svg - Last visited Feb 14th , 2012

5] S. Viro, *"Multi-Agent System for Distributed Data Fusion in Peer-to-Peer Environment*", Master thesis, University of Jyvaskyla, 2002.

6] http://ecosoc.sdu.dk/coe/Image:SunSPOTAnimatedGif_27716_013.gif – Last visited

Jan 23rd, 2012

17] W. Stella, K. Lilian and O. Joseph, *"Productivity trends and performance of dairy Farming in kenya", 2011,* Tegemeo Institute of Agricultural Policy and Development. pp 1, 15
Url: http://www.tegemeo.org/documents/work/WP43-Productivity-Trends-and-Performance-of- Dairy-Farming-in-Kenya.pdf - 25/02/2012 – Last visited Jan 30th, 2012

[18] M. Kevin, S. Suresh, *"Policy based Agents in Wireless Body Sensor Mesh Networks for Patient Health Monitoring"*, International Journal of u- and e- Service, Science and Technology Vol. 2, No. 4, December, 2009.

Url: http://www.sersc.org/journals/IJUNESST/vol2_no4/4.pdf - Last visited Feb 20th, 2012

[19] B. O. Bebe, W. Thorpe, H.M.J. Udo and W. Mulinge, "*Breed preferences and breeding practices in smallholder dairy systems of the central highlands of Kenya*" Url: http://mahider.ilri.org/bitstream/handle/10568/1576/Bebe%20et%20al-2000-Breed%20preferences-KARI.pdf?sequence=1 - Last visited Feb 1st, 2012

[20] B.O. Bebe, H.M.J. Udo, G.J. Rowlands, W. Thorpe, "*Smallholder dairy systems in the Kenya highlands: breed preferences and breeding practices*", Elsevier B.V., 2003

[21] M. Antonio,V. Aïda, V. Alexandre, "*Using JADE-LEAP to implement agents in mobile devices*"
Url: http://jade.tilab.com/papers/EXP/02Moreno.pdf - Last visited Mar 7th, 2012

[22] C. Giovanni. Leap 3.1 User Guide, TILAB

[23] R. Luis, L. Loredana, B. Pilar and R. I. Jose, " *A Review of Wireless Sensor Technologies and Applications in Agriculture and Food Industry: State of the Art and Current Trends* ". Sensors 2009, 9, 4728-4750; doi:10.3390/s90604728, 2009, pp 4730 - 4732

[24] T. M. Jan, "*Comparison of the IEEE 802.11, 802.15.1, 802.15.4 and 802.15.6 wireless standards*", 2007
Url: - http://janmagnet.files.wordpress.com/2008/07/comparison-ieee-802-standards.pdf - Last visited Feb 1st, 2012

[25] Frank H.P. Fitzek, R. Frank, "*Mobile Phone Programming and its Applications to Wireless Networking*", Springer, 2007, pp 372 – 395

[26] http://www.java-samples.com/showtutorial.php?tutorialid=440 - Last visited Mar 1st, 2012

[27] http://www.thecattlesite.com/articles/1053/heat-stress-in-dairy-cows-implications-and-v nutritional-management - Last visited Feb 23rd, 2012

[28] F. Fummi, D. Quaglia, F. Ricciato, M. Turolla, "*Modeling and Simulation of Mobile Gateways Interacting with Wireless Sensor Networks*"
Url: http://www.sci2ools.net/read/modeling-and-simulation-of-mobile-gateways-interacting-with-wireless-sensor-networks-101013 - Last visited Feb 1st , 2012

[29] C. Patrick, M. Vallipuram, "*Performance Analysis of a Mobile Phone*

Serving as a Gateway in a Body Sensor Network"
Url: http://www.issnip.unimelb.edu.au/__data/assets/pdf_file/0009/187650/Abstract17.pdf
- Last visited Feb 1st, 2012

[31] L. E. Chase, "*Climate Change Impacts on Dairy Cattle*". Climate Change and Agriculture: Promoting Practical and Profitable Responses
Url: http://climateandfarming.org/pdfs/FactSheets/III.3Cattle.pdf - Last visited Feb 1st, 2012

[32] http://java.com/en/download/faq/whatis_j2me.xml - Last visited Mar 1st, 2012.

[33] C.R. Kothari, "*Research Methodology Methods and Techniques*". New Age International (P) Limited, 2004, pp 31- 32.

[34] http://code.google.com/p/iwsn/ - Last visited May 11th, 2012

[35] G. Bart, "Modern dairy farming in warm climate zones. Agromisa Foundation, Wageningen, 2005, pp 6
Url: http://www.cd3wd.com/cd3wd_40/LSTOCK/001/agrodoks/EM-23-e-2005-Digital.pdf - Last visited Aug, 1st ,2012

[37] http://en.allexperts.com/q/Java-1046/Math-random-method.htm - Last visited Aug 1st, 2012

[38] H. Perros, "*Computer Simulation Techniques: The definitive introduction!*". NC State University Raleigh, NC, 2009, pp 4- 29

[39] M. Anu, "*Introduction to modeling and Simulation*", Proceedings of the 1997 Winter Simulation Conference, 1997.
Url: http://www.inf.utfsm.cl/~hallende/download/Simul-2-2002/Introduction_to_Modeling_and_Simulation.pdf - Last visited Aug 3rd, 2012

APPENDICES

Appendix I: Sample code for the three agents

a. Sensor Agent

```
package agents;

/**

* @author Joseph.ochieng

*/

import jade.core.AID;

import jade.core.Agent;

import jade.core.behaviours.CyclicBehaviour;

import jade.core.behaviours.TickerBehaviour;

import jade.lang.acl.ACLMessage;

import java.io.BufferedReader;

import java.io.FileInputStream;

import java.io.InputStreamReader;

import java.util.List;

import java.util.Scanner;

import java.util.StringTokenizer;

import java.util.logging.Level;

import java.util.logging.Logger;


public class SensorAgent extends Agent {

    protected List<String> data;

    @Override

    protected void setup() {

        super.setup();

        final Object[] args = getArguments();

        if (args[0] != null) {data = (List<String>) args[0]; }
```

```
            else
                    { System.out.println("Error assigning parameters");}
   addBehaviour(new TickerBehaviour(this, 100) {
    protected void onTick() {   addBehaviour(new SensorAgent.SendMessage(myAgent));
    }
  });
  System.out.println("the sender agent " + this.getLocalName() + " is started");
}
@Override
protected void takeDown() {
}
public void delimitTheString(String finalOutputLine) {
   StringTokenizer st = new StringTokenizer(finalOutputLine, ":");
   String finalToken = "";
   String sensorID = "";
   String sensorData = "";

    while (st.hasMoreTokens()) {
      finalToken = st.nextToken();
      sensorID = finalToken.substring(finalToken.indexOf("senso"), finalToken.indexOf("-"));
      sensorID = sensorID.replaceAll("sensor_id", "");
      // System.out.println("Sensor ID: " + sensorID);
      sensorData = finalToken.substring(finalToken.indexOf("-") + 1, finalToken.length());
      sensorData = sensorData.replaceAll("temp_reading", "");
    }
  }
  public class SendMessage extends CyclicBehaviour {
    private static final long serialVersionUID = 9088209402507795289L;
private boolean finished = false;
public SendMessage(final Agent myagent) {
  super(myagent);
}

@Override
public void action() {
  try {
     Scanner sc = null;
     Scanner delimitScanner;
     SensorAgent obj = new SensorAgent();
     String result;
     try {
        //Read the texFile last line input
        String inputLine = "";
        String finalOutputLine = "";
        // Set this location to the simulator folder
        FileInputStream in = new FileInputStream("F:\\Sensor_data.txt");
        BufferedReader br = new BufferedReader(new InputStreamReader(in));
        String strLine = null, tmp;
        while ((tmp = br.readLine()) != null) {
           strLine = tmp;
        }
        String lastLine = strLine;
        finalOutputLine = lastLine.substring(lastLine.indexOf("senso"), lastLine.length());
        obj.delimitTheString(finalOutputLine);
        StringTokenizer st = new StringTokenizer(finalOutputLine, ":");
        String finalToken = "";
  String sensorID = "";
  String sensorData = "";

  while (st.hasMoreTokens()) {
      finalToken = st.nextToken();
      sensorID = finalToken.substring(finalToken.indexOf("senso"), finalToken.indexOf("-"));
      sensorID = sensorID.replaceAll("sensor_id", "");
      // System.out.println("Sensor ID: " + sensorID);
      sensorData = finalToken.substring(finalToken.indexOf("-") + 1, finalToken.length());
      sensorData = sensorData.replaceAll("temp_reading", "");
      // System.out.println("Sensor Data: " + sensorData);
      sensorData = sensorID.concat(sensorData);
      //System.out.println("Sensor Data: " + sensorData);
      sensorData = sensorData.replaceAll("  ", "                              ");
    }
    // Send contents in a message
    ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
    msg.setContent(sensorData.toString());
    msg.setSender(this.myAgent.getAID());
    msg.addReceiver(new AID("AnalysisAgent", AID.ISLOCALNAME));
    this.myAgent.send(msg);
  } catch (Exception ex) {
    System.out.println("Exception " + ex);
  }
  Thread.sleep(50000);
} catch (InterruptedException ex) {
  Logger.getLogger(SensorAgent.class.getName()).log(Level.SEVERE, null, ex);
}
```

```
    }
  }
}
```

b. **Analysis Agent**

```
package agents;
/**
 * @author Joseph.ochieng
 */
import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

public class AnalysisAgent extends Agent {

    protected List<String> data;
    @Override
    protected void setup() {
        super.setup();
        //get the parameters given into the object[]
        final Object[] args = getArguments();
```

```
  if (args.length != 0) {
    System.out.println("Error!");
  }
  //Add the behaviours
  addBehaviour(new AnalysisAgent.ReceiveMessage(this));
}
@Override
protected void takeDown() {
}

public class ReceiveMessage extends CyclicBehaviour {
  private static final long serialVersionUID = 9088209402507795289L;
  private boolean finished = false;
  private int step = 0;
  public ReceiveMessage(final Agent myagent) {
    super(myagent);
  }
  @Override
  public void action() {
    // receive the message
    ACLMessage msg = this.myAgent.receive();
    if (msg != null) {
      this.finished = true;
      try {
        msg.setSender(this.myAgent.getAID());
        msg.addReceiver(new AID("ViewerAgent", AID.ISLOCALNAME));
        // get thresholds from file
        FileReader r = null;
  StringBuffer contents = new StringBuffer();
  try {
      String path = "Threshold.txt"; // The file having temperature limits
      r = new FileReader(path);
      int i;
      try {
          while ((i = r.read()) != -1) {
              contents.append((char) i);
          }
       } catch (IOException ex) {
          Logger.getLogger(AnalysisAgent.class.getName()).log(Level.SEVERE, null, ex);
       }
     } catch (FileNotFoundException ex) {
        System.out.println("File Not Found");
     }
     // Send contents in a message
     ACLMessage msg_t = new ACLMessage(ACLMessage.INFORM);
     msg_t.setContent(contents.toString());
     // classify the reading
     if ((Float.parseFloat(msg.getContent().substring(3)) > Float.parseFloat(msg_t.getContent()))) {
         msg.setContent(msg.getContent() + "                    " + "ABNORMAL");
      } else {
         msg.setContent(msg.getContent() + "                    " + "NORMAL");
      }
      this.myAgent.send(msg);
    } catch (NullPointerException | NumberFormatException n) {
    }
} else {
        block();
      }
    }
  }
}
```

c. Viewer Agent

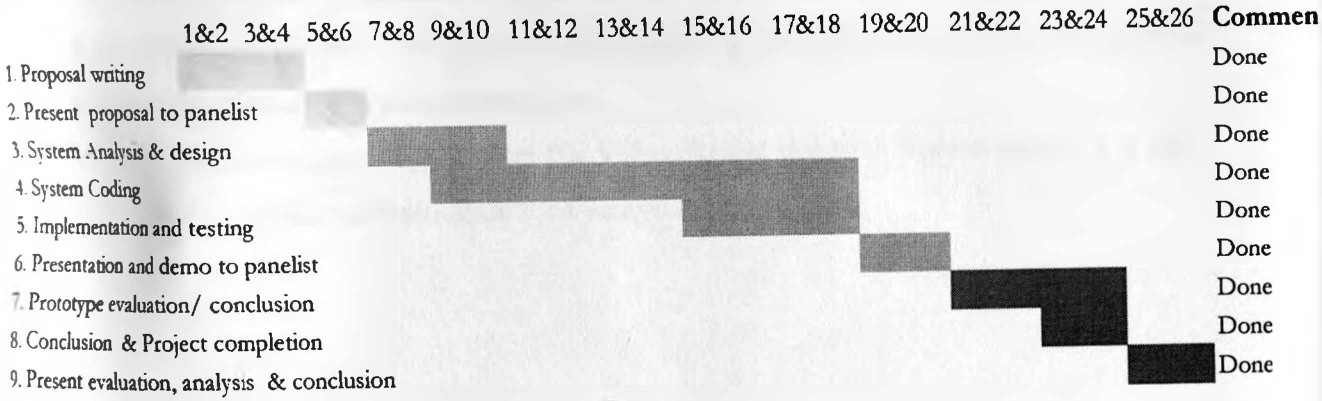
```
package agents;
/**
 * @author Joseph.ochieng
 */
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.util.List;
import javax.swing.*;

public class ViewerAgent extends Agent {

    JTextArea result = new JTextArea();
    JPanel panel = new JPanel();
    JButton clear = new JButton();
    JLabel label = new JLabel();
    JFrame frame = new JFrame();

    public ViewerAgent() {
```

```
  result = new JTextArea();
}
protected List<String> data;
@Override
protected void setup() {
  super.setup();
  //get the parameters given into the object[]
  final Object[] args = getArguments();
  if (args.length != 0) {
     System.out.println("Error!");
  }
  //Add the behaviours
  addBehaviour(new ViewerAgent.ReceiveMessage(this));
  frame = new JFrame("..::: Viewer Agent :::..");
  frame.addWindowListener(new WindowAdapter() {
     @Override
     public void windowClosing(WindowEvent we) {
        //unsubscribe();
        doDelete();
        System.exit(0);
     }
  });
  frame.add(label);
  result.setLineWrap(true);
  result.setVisible(true);
  result.setEditable(false);
  frame.add(clear);
  JScrollPane scp = new JScrollPane(result);
```

```
ırPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
llBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
(result);
e().add(scp);
0);
;

SOR ID" + "              " + " INPUT(C)" + "            " + " STATUS" + "\n");

ssage extends CyclicBehaviour {
ng serialVersionUID = 9088209402507795289L;
shed = false;

age(final Agent myagent) {

) {

 message
sg = this.myAgent.receive();
 {
 = true;

ender(this.myAgent.getAID());
ay messages
pend("    " + msg.getContent() + "\n");
```

```
        } catch (NullPointerException n) {
          n.printStackTrace();
        } catch (NumberFormatException ne) {
          ne.printStackTrace();
        }
      } else {
        block();
      }
    }
  }
}
```

Appendix II: Gant Chart



KEY

□ Milestone 1 (M1)

▨ Milestone 2 (M2)

■ Milestone 3 (M3)

Appendix III: How to run the simulation

To run this system the user has to follow the following steps:-

1. Install JAVA EE SDK 6u1 JDK and JAVA JDK 1.6.10 or later versions software on the computer to run the system (if not installed)
2. Install Netbeans IDE.
3. Copy and save the folder *Ouma* found on the CD which contains the system to a desirable part of your computer
4. Open Netbeans and select Open Project then locate the folder *Ouma* where the project is saved and select it.
5. Right click on the project *Ouma* in Netbeans and select run.
6. On Netbeans, press Enter to start the agents when prompted.
7. Open the *Ouma* folder above and double click the file SensorSimulator.jar. (This is the sensor network simulator).
8. Select step run of the sensor simulator and specify 10 as the number of steps.
9. Wait for considerable amount of time and the results will start displaying. When the WSN simulator progress bar is full, click pause on the viewer agent window and take count of the displayed results and tabulate.
10. To stop the system, use X to cancel the ViewerAgent window. Repeat steps 5 – 9 until you meet the number of simulation runs you would like to take