



UNIVERSITY OF NAIROBI  
School of Computing & Informatics

*A Multi-Agent System Approach for Performance Monitoring of  
Electronic Channels in Banks*

Submitted by  
Bernard Omoi Ondara

Supervisor: Dr. Lawrence Muchemi

A project report submitted in partial fulfillment of the requirement for the award of  
the degree of Master of Science in Computer Science

August 2014

# DECLARATION

## Student

I declare that this report is my original work and has not been presented to any other university for an academic award.

Sign: .....

Date:.....

Bernard Omoi Ondara

P58/75904/2012

## Supervisor

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

Sign: .....

Date:.....

Dr. Lawrence Muchemi

SCHOOL OF COMPUTING & INFORMATICS

## **ACKNOWLEDGEMENTS**

To begin with, I would like to thank the Almighty God, who made the completion of this project possible through his Sustaining Grace and Providence. I am grateful to my supervisor, Dr. Lawrence Muchemi, for helping me gain a better comprehension and scoping of the research topic besides guidance on writing the paper end to end.

I am truly indebted to my dear wife Brigid Kerubo Mogita, for giving me moral support and encouragement that saw me complete my coursework and write this thesis report. Finally, I want to thank the University of Nairobi, for the ample study environment and learning resources that helped me undertake my Master's degree programme and subsequently write this report.

## ABSTRACT

Modern banking is conducted within an environment of multiple electronic channels to meet the various needs of customers. These banking channels include Automated Teller Machines (ATMs), Point of Sale (POS) devices, Internet banking, and Mobile banking among others and banks are keen to monitor the performance of these channels. However, conventional techniques such as database queries, application monitoring, network links monitoring, data mining, and rule-based technologies used in monitoring the performance of electronic channels are mainly manual or semi-automated and lack an integrated intelligent model for this task. The literature that was reviewed did not show the application of multi-agent systems technology in monitoring the performance of banking electronic channels although this technology has been shown to be successful in several areas of systems performance monitoring.

This study therefore set out to research on the use of a multi-agent system based approach to performance monitoring for banking electronic channels. The model used simulated data and it was evaluated based on key functionalities in terms of the ability to generate simulated transactions, monitoring channel activity and generate reports advising on e-channels performance and its realism. Four evaluators tested this model and confirmed it had the functionalities of a useful performance monitoring and reporting tool in banking and that it addressed key inadequacies current techniques face. The research found out that multi-agent system technology, through its use of intelligent and autonomous agents, could continuously monitor and report performance of the different e-channels with limited or no human intervention thus avoiding one of the main challenges conventional techniques face.

This study is significant to the academic community interested in artificial intelligence or the specific area of using multi-agent systems in monitoring the performance of an electronic system. The findings will also prove useful to software developers and company procurement departments in identifying essential features of an intelligent performance monitoring and reporting solution.

**Keywords:** Agent, Multi-Agent Technology, Multi-Agent System, Performance Monitoring, Electronic Channels, E-Channels, ECPM, JADE, POS, ATM, Artificial Intelligence, E-Banking

# TABLE OF CONTENTS

DECLARATION .....	i
ACKNOWLEDGEMENTS .....	ii
ABSTRACT .....	iii
TABLE OF CONTENTS .....	iv
LIST OF FIGURES .....	vii
LIST OF TABLES .....	viii
ACRONYMS AND KEY TERMINOLOGIES.....	ix
CHAPTER 1: INTRODUCTION .....	1
1.0 Introduction .....	1
1.1 Background to the study.....	1
1.2 Problem statement and Purpose of study .....	2
1.3 Research objectives .....	2
1.4 Justification and significance of the study .....	3
1.5 Assumptions and limitations of the research.....	4
1.6 Chapter Summary.....	4
CHAPTER 2: LITERATURE REVIEW .....	5
2.0 Introduction .....	5
2.1 Infrastructure of banking electronic channels .....	5
2.1.1 Network Infrastructure .....	5
2.1.2 Software Infrastructure .....	6
2.2 The need for integrated e-channels performance monitoring .....	6
2.3 Common Electronic Banking services delivery channels .....	7
2.1.1 Automated Teller Machines (ATMs) .....	7
2.1.2 Point of Sale (POS) devices.....	8
2.1.3 Internet banking.....	8
2.1.4 Mobile banking.....	9
2.4 Integrated performance monitoring.....	9
2.4.1 Current model for banking e-channels performance monitoring and reporting.....	9
2.4.2 Current methods for banking e-channels performance monitoring.....	10
2.4.3 Cases of successful system performance monitoring using multi-agents technology...11	

2.5 Agents, Multi-agents, and Agent Architectures .....	13
2.5.1 Agents and Multi-Agents.....	13
2.5.2 Agent Architectures (Classes of agents).....	13
2.6 Applications of Multi-Agent Systems technology .....	14
2.7 Multi-agent methodologies and Frameworks.....	15
2.7.1 Methodologies and frameworks for multi-agent systems.....	15
2.7.2 Java Agent Development Environment (JADE) .....	16
2.8 The proposed Model.....	17
2.8.1 Context Diagram.....	17
2.8.2 Conceptual model of the proposed ECPM Model.....	17
2.9 Chapter Summary.....	19
CHAPTER 3: METHODOLOGY .....	21
3.0 Introduction .....	21
3.1 Research Design and its justification .....	21
3.2 Data Collection.....	23
3.2.1 Sources of data and their relevance to the problem.....	23
3.2.2 Tools, procedures and methods for data collection and their justification.....	23
3.2.3 Data Analysis.....	24
3.2.4 Purpose of Data Collected .....	27
3.2.5 Limitations of the methodology and how they are overcome .....	28
3.3 Database and Model Design.....	28
3.3.2 Database Design .....	29
3.3.3 Model Design .....	30
3.4 Prototype Development.....	40
3.4.1 Development Tools and Algorithms.....	40
3.4.2 Database Implementation .....	41
3.4.3 Logic implementation of the prototype .....	41
3.5 Testing of the prototype .....	42
3.5.1 Unit Testing .....	43
3.5.2 Integration Testing.....	46
3.5.3 System Testing .....	47

3.5.4 User Acceptance Testing .....	48
3.5.5 Prototype Deployment Model .....	50
3.6 Chapter summary .....	50
CHAPTER 4: EVALUATION .....	51
4.0 Introduction .....	51
4.1 Simulation of transactions .....	51
4.2 Monitoring channel activity and reporting channel performance .....	52
4.3 Agents Interaction .....	54
4.4 Realism of the prototype .....	56
4.5 User Feedback .....	58
4.6 Chapter summary .....	60
CHAPTER 5: CONCLUSIONS & RECOMMENDATIONS .....	61
6.0 Introduction .....	61
6.1 Recap of this research .....	61
6.2 Discussions .....	61
6.3 Conclusions .....	63
6.4 Recommendations for further research .....	65
REFERENCES .....	66
APPENDICES .....	69
Appendix A – Data Collection Questionnaire.....	69
Appendix B – Sample Code .....	73
Appendix C – UAT Questionnaire.....	79

## LIST OF FIGURES

Fig 2.1 – Infrastructure of the electronic channels of a bank.....	5
Fig 2.2 - How ATMs connect to the bank .....	7
Fig 2.3 – How POS devices connect to the bank.....	8
Fig 2.4 – How Internet banking channel devices connect to the bank .....	8
Fig 2.5 – How Mobile banking channel devices connect to the bank .....	9
Fig 2.6 – Current scenario of e-channels performance monitoring and reporting.....	10
Fig 2.7 - A model for a multi-agent system.....	15
Fig 2.8 - JADE Architecture .....	16
Fig 2.9 - The relationship between JADE’s main architectural elements.....	16
Fig 2.10 – Context diagram for the proposed ECPM model .....	17
Fig 2.11 - MASST Framework for decision making in stock trading .....	18
Fig 2.12 – Conceptual model for the proposed ECPM Model .....	18
Fig 3.1: The conceptual ECPM model.....	28
Fig 3.1- Actor-diagram for the E-Channels Monitoring Stakeholders .....	31
Fig 3.2 - The System-to-be operating in its environment as an actor .....	32
Fig 3.3 – The proposed ECPM Model.....	33
Fig 3.4 - Actor diagram for the e-channels performance monitoring system.....	34
Fig 3.5 - Capability diagram of Master Agent.....	36
Fig 3.6 - Capability diagram of POS Reporting Agent.....	37
Fig 3.7 - The get_data_from_data_sources plan diagram.....	38
Fig 3.8 - Plan diagram for POS Reporting Agent.....	39
Fig 3.9 - Agent Interaction Diagram.....	39
Fig 3.10 - Prototype’s Graphical User Interface .....	42
Fig 3.11 - Deployment Model of the E-Channels Performance Monitoring Prototype.....	50
Fig 4.1 - Sample notification of simulated transactions and network status .....	52
Fig 4.2 - Sample Performance Report for All Channels with Summary .....	53
Fig 4.3 - Agents Interaction .....	55
Fig 4.4 - Sample performance report for POS, ATMs, and E-Channels .....	57



## LIST OF TABLES

Table 3.1 - An analysis of data gathered from completed questionnaires .....	25
Table 3.2 - The different solutions banks use to monitor the performance of e-channels.....	26
Table 3.3 - The preferred e-channels by banks.....	26
Table 3.4 - Percentages of media used for sending performance reports in banks.....	27
Table 3.5 - Actors capabilities .....	35
Table 3.6 - Types of agents and their assigned capabilities.....	35
Table 3.7 - The tasks in Unit Testing:.....	43
Table 3.8 - Starting JADE and system agents (UT-1) .....	44
Table 3.9 - Generating new transactions into MySQL Database Server (UT-2).....	44
Table 3.10 - Sniffing new transactions, report generation, and reporting .....	45
Table 3.11 - Agents Interactions Testing .....	47
Table 3.12 - User Acceptance Testing (UAT) Tasks.....	48
Table 3.13 - UAT Test Cases.....	49
Table 4.1 – Simulation of transactions .....	51
Table 4.2 – Monitoring channel activity.....	53
Table 4.3 – Agents Interaction.....	54
Table 4.4 - Realism of prototype .....	56
Table 4.5 - User Comments discussion.....	59

## ACRONYMS AND KEY TERMINOLOGIES

**ACL – Agent Communication Language:** a language that outlines the protocol agents use to communicate with each other in a multi-agent system

**Agent:** This is an autonomous software component capable of meeting certain objectives by performing particular tasks in a given environment through perceptors and actuators

**ATM - Automated Teller Machine:** An electronic device, usually located outside bank branches for customers to perform transactions like cash withdrawal and deposits on their own

**E-channels, Channels, Electronic Channels or Service Delivery Channels:** The various traditional and direct channels for the delivery of services to a bank's customers, which include ATMs, POS terminals/devices, Internet banking, and Mobile banking.

**ECPM – Electronic Channels Performance Monitoring:** This is the model developed in this research to intelligently monitoring and reporting the performance of banking e-channels

**FIPA – Foundation for Intelligent Physical Agents:** a standard for software agent technology

**JADE - Java Agent Development Environment:** A framework for developing multi-agent systems using FIPA specifications. It provides the environment for agents' interaction.

**Multi-Agent System (MAS):** a system consisting of multiple agents that interact based on set communication protocols in a given environment to achieve specific objectives

**Performance:** A task or operation seen in terms of how successfully it is performed

**POS - Point of Sale Terminal:** an electronic device used by banks or merchants to acquire card transactions from remote sites. The devices are connected to banks for processing.

**RMA - Remote Monitoring Agent:** an agent within JADE used to monitor other agents

# CHAPTER 1: INTRODUCTION

## 1.0 Introduction

This chapter introduces the research work in this report entitled “*A Multi-agent Systems Approach for Monitoring the Performance of Electronic Channels in Banks*”. The first section of this chapter will present the background to the study, which will lead to the problem statement and purpose of the study. After this, the next section will cover the research objectives followed by the justification and significance of this study. The chapter will then highlight key assumptions and limitations of the study before closing with a summary of this chapter.

## 1.1 Background to the study

Up until the 1980s, banks provided only one channel for delivering banking services to their clients: the branch network (Uppal, 2011). Later, due to e-banking needs, banks acquired direct banking e-channels including ATMs, POS devices, Mobile banking, and Internet banking (Daniela, Simona & Dragos, 2010) to improve e-banking performance through anywhere, anytime and multiple banking services (Kumbhar, 2011).

However, banks now support direct channels that include mobile banking, internet banking, and social media in rendering better customer services. This development has given customers several channels to choose from based on the unique benefits of each channel and what channels complement each other in service provision (Ostergaard, 2010). Even though banks recognize the significance of multi-channel customer service delivery that aims at improving customer experience, one of their key challenges is monitoring these disparate platforms and systems to ensure optimal uptime and rapid response to incidents reported that affect the provision of services to customers through these channels. However, providing 24/7 service delivery is more practicable on certain channels such as ATMs but not bank branches. Therefore, there is a focus shift from manned channels to self-service channels (Uppal, 2011).

Manual ad hoc monitoring and reporting of the performance of banking channels coupled with the absence of an intelligent integrated performance-monitoring tool are often the key challenges banks face in managing e-channels. This is compounded further by human errors, fatigue, and delayed detection of channel inactivity leading to delayed service restoration and loss of revenue.

These conventional methods fail to offer timely and comprehensive performance reports and restoration of services, leaving top management unaware of channels performance challenges. Such manual methods include data mining and rule-based (Estevez, Held & Perez, 2006) techniques.

## **1.2 Problem statement and Purpose of study**

There is a lack of an intelligent model for monitoring the performance of banking e-channels since current methods and techniques are largely manual or lack the necessary intelligence for handling the complex task of actively monitoring the performance of the channels in real time. Channel performance should be within perceived tolerance limits among customers (Kumbhar, 2011), that is, not below the expected performance levels.

The purpose of this project, therefore, is to research on the use of multi-agent based systems in monitoring the performance of e-channels in banking and develop a model and prototype to demonstrate the suitability of autonomous intelligent agents in providing channel performance monitoring facility. This study was motivated by the finding that there is no published intelligent model developed specifically for purposes of monitoring the performance of banking electronic channels. However, there are publications showing that multi-agent system technology is the best for solving this problem by addressing the inadequacies current techniques have (Gao *et al.*, 2006). The developed solution will aim at ensuring relevant bank employees conveniently and economically access a consolidated channel performance report to support appropriate decision making. This decision-making will focus on two key performance areas: uptime restoration for channels and high-level reporting on channels performance. This monitoring tool will provide a solution driven by autonomous agents that collaborate with each other (Worku, 2007).

## **1.3 Research objectives**

This study proposes a multi-agent system approach for the analysis of transactional data in a commercial bank to determine how to automate monitoring and reporting of the performance of its e-channels. To achieve this, the analysis will involve using intelligent software agents to recognize events and behavior patterns that help them decide if a certain e-channel is experiencing downtimes or underperforming in terms of transaction volumes at a given time or

delayed inflow of transactions. The resulting solution will run with minimal human intervention after its deployment. All the decisions these agents will make largely depend on data available in a bank's transaction server. Since transactional data is highly confidential, it will be unavailable for use in this project thus prompting the use of simulated data.

This paper intends to achieve the following four research objectives:

- 1) To carry out a research on banking e-channels and identify techniques banks currently use to monitor and report e-channels performance
- 2) To create a conceptual model of a multi-agent based performance monitoring system for banking e-channels
- 3) To develop and test a Multi-agent based prototype to support e-channels performance monitoring and reporting for banks
- 4) To conduct an analysis of the results and findings from using the prototype

#### **1.4 Justification and significance of the study**

Monitoring the uptimes of a bank's service delivery channels is a complex global concern in banking. It therefore requires a complex approach to solving it. Since conventional methods such as rule-based and data mining techniques do not solve the problem adequately, there is a need to explore the use of multi-agent based technology, which has been demonstrated as suitable in handling complex performance monitoring needs. When delivery channels experience downtimes, banks lose revenue, inconvenience their customers, and sometimes face litigation due to inconveniences and embarrassments caused to their customers. This research aims at using a multi-agent system based approach to monitoring the performance of different e-channels of a bank to ensure prompt reporting that aims at triggering appropriate response to incidents detected and eventual optimal service delivery to customers. The research includes a listing of examples of where agents have been used to monitor systems successfully. A prototype system of how intelligent agents would work in solving this global problem for banks will be developed.

This study will benefit several audiences. First, it will provide useful contributions to the field of Computer Science particularly where artificial intelligence is used in the context of distributed systems. The prototype is developed following Tropos methodology, which is one of the leading

methodologies for designing multi-agent solutions (Bellifemine, Caire & Greenwood, 2007). This will benefit scholars interested in studying this problem space and implement their solutions using this methodology. Second, the solution yielded in this study will benefit banks experiencing challenges in monitoring the performance of their electronic channels if used.

The framework can be used by any bank with a few modifications to accommodate more channels than will be demonstrated in this research. Third, it will benefit a bank's internal software developers in understanding how the solution to performance monitoring of channels is built and works. A bank's procurement department will benefit from the system features listed. Fourth, Retail companies that want to monitor the performance of their service delivery channels will benefit too as the concept is applicable in their line of business. Finally, governments, through their Central Banks may recommend optimal uptimes for e-channels, which may then compel banks to adopt this kind of technology because of its demonstrated success.

### **1.5 Assumptions and limitations of the research**

While banks ordinarily have many service delivery channels, this study's scope is limited to the following channels: ATMs and POS devices. The study assumes the existence of an integrated database of transactions from all e-channels. Since there will be no real-world e-channels to be used in testing the developed system prototype, a simulation will be used to demonstrate the flow of transactions from the selected two e-channels. This is because a bank's transactional records are legally kept secure due to their privacy and confidentiality.

### **1.6 Chapter Summary**

This chapter provided a background to the research by stating the research problem, highlighting four research objectives, stating the research outcomes, and the assumptions and limitations of this study. The research problem is that there is no published intelligent monitoring model or framework for banking channels. The purpose of this research is to develop an intelligent model for monitoring the performance of banking electronic channels. The next chapter, Literature Review, will focus on what is published about two areas: (1) banking electronic channels and their performance monitoring and (2) multi-agent system technology, culminating in the development of a performance-monitoring model for banking e-channels.

## CHAPTER 2: LITERATURE REVIEW

### 2.0 Introduction

Banks offer services to their customers through various e-channels. These channels include but are not limited to ATMs, POS devices, Internet banking, and Mobile Banking (Daniela, Simona & Dragos, 2010). They offer complementary services based on their different features (Ostergaard, 2010; Birgelen *et al.*, 2006). This chapter focuses on monitoring the performance of these channels, with specific attention to ATMs and POS devices. The chapter is divided into three sections. The first section reviews published literature on performance monitoring for banking channels by covering e-channels infrastructure, the need for integrated e-channels performance monitoring, and current techniques used in monitoring their performance. The second section is about agents and multi-agents system technology, which includes an introduction to agents and agent architectures, applications of multi-agent systems, and multi-agents methodologies and frameworks. The last section introduces the proposed model for performance monitoring for banking e-channels and its benefits.

### 2.1 Infrastructure of banking electronic channels

#### 2.1.1 Network Infrastructure

The following is a high-level diagram depicting the infrastructure of the various e-channels of a bank.

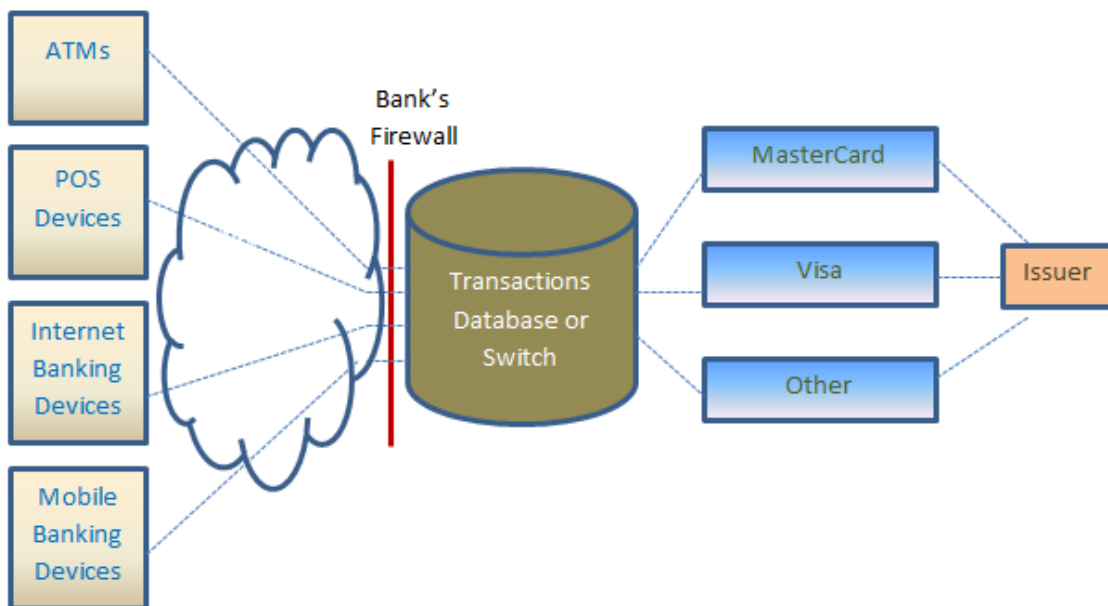


Fig 2.1 – Infrastructure of the electronic channels of a bank

In the figure 2.8 above depending on the channel used in doing a bank transaction, the implemented network link (GPRS, Internet, or Dial-up) will be used to carry the transaction message to the bank's transaction database. This is the case for transactions the bank approves as is the case for card transactions for cards issued by the bank to its customers. In case a card issued by a different bank is used, the message proceeds to the issuer of the card via the MasterCard, Visa, or other Global Payment Switches for approval. Once the transaction is processed, the message flows back to the device that initiated the transaction to output a receipt or display the response on the screen.

### **2.1.2 Software Infrastructure**

The above infrastructure is supported by different software applications including the following:

- Operating systems: Windows, Linux, and Unix
- Transaction Switch (Card Switch) applications: vary with banks
- Database applications: vary with banks. They include Oracle, SQL Server, MySQL, DB2, Informix, Sybase ASE, Sybase IQ, and Teradata (Market Share, 2016).
- Query supporting applications: SQL Developer, Oracle SQL Developer, and others
- Others including Office applications and graphical tools

### **2.2 The need for integrated e-channels performance monitoring**

Increasing competition among banks has created a need for banks to have success or failure measures on their electronic service delivery channels dictated by customer experience and service delivery. This has resulted in attempts to monitor applications that process transactions from each channel. The performance of each channel directly impacts the performance of the bank (Ostergaard, 2010) since customers can only use active channels for banking services. Therefore, it is very critical that all channels are monitored and system malfunctions that are affecting the delivery of customer services are addressed in the shortest time possible. More importantly, banks always want key channels up all the time.



There are various reasons for monitoring the performance of banking e-channels:

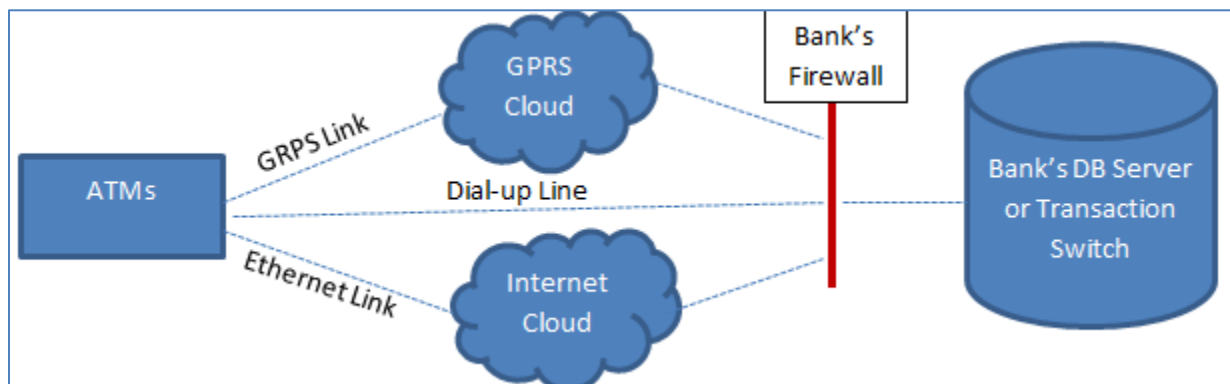
- To achieve end-to-end visibility into the performance of mission-critical systems
- To accelerate problem resolution by reducing the Mean-Time-To-Resolution (MTTR)
- To monitor production servers for stability and realize quicker system service restoration
- To improve teamwork by using an integrated performance monitoring tool

Monitoring the performance of e- channels offers several benefits to banks and customers. Banks are able to better manage customer expectations by offering seamless services, better understand their customers through improved communication, and provide customers with targeted products/services for optimal return on investment. Customers can choose their preferred channels for transactions, combine channels during a transaction, and enjoy seamless services

## 2.3 Common Electronic Banking services delivery channels

### 2.1.1 Automated Teller Machines (ATMs)

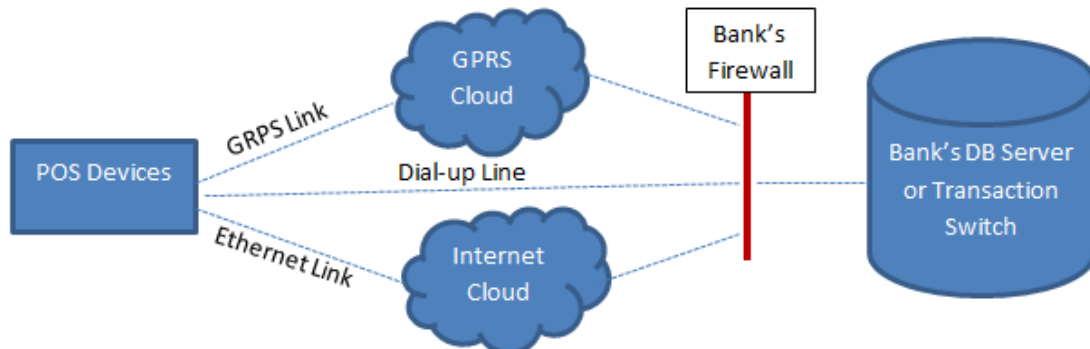
This channel is attractive to the young generation. ATMs are normally located outside and/or inside a bank's branches. They can also be located away from the branch network. Young customers make frequent withdrawals and deposits thereby finding ATMs suitable to them (Ostergaard, 2010). ATMs are connected to banks through physical IP-based data links but some operate on GPRS data links. This channel is within the scope of this study. This channel's performance remains high because of its ability to offer cash to customers unlike other remote channels such as Internet banking. However, its performance by volume is now relatively lower due to the existence of other remote channels such as mobile banking (Catalan, 2004).



**Fig 2.2 - How ATMs connect to the bank**

### 2.1.2 Point of Sale (POS) devices

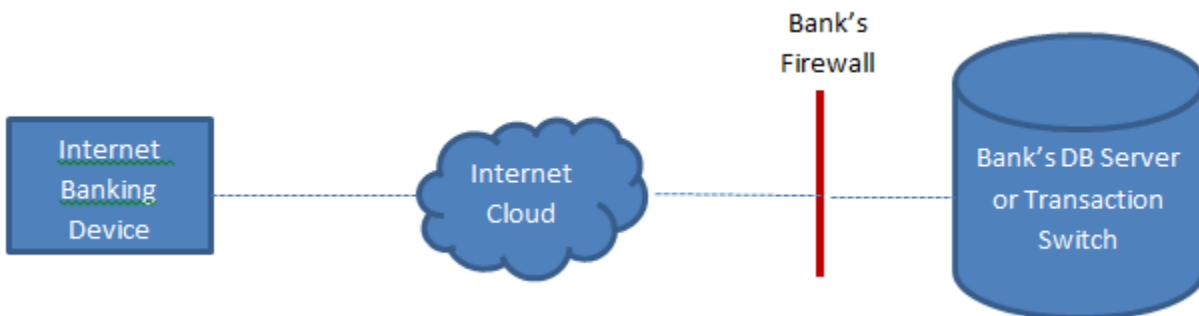
These are technology-aided products (Uppal, 2011) deployed by banks at remote sites for transactions such as purchase of goods. It is the preferred channel where carrying cash would be risky like when shopping at supermarkets or buying items on the Internet. POS devices are connected to banks through Ethernet links, Dial-up telephone links, or GPRS data lines. The increasing usage of this channel is attributed to the rise in the usage of debit/credit card technology. Its performance is affected by technology overloads (Kumbhar, 2011).



**Fig 2.3 – How POS devices connect to the bank**

### 2.1.3 Internet banking

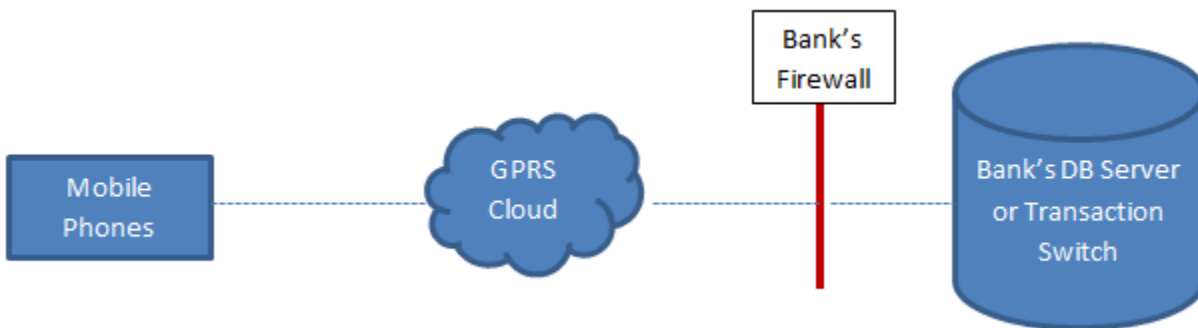
Hitt and Frei (2002) established through their study that Internet banking is much more convenient to consumers of banking services. The success of Internet banking is dependent on two key factors: consumer satisfaction and channel performance (Birgelen *et al.*, 2006). This channel rides on the Internet cloud to access banking services so customers with Internet access through computers and other devices can use this channel. Technology overload and Internet availability are known to cause loss of performance of this channel among users and banks (Daniela, Simona & Dragos, 2010). The figure below shows the infrastructure of this channel



**Fig 2.4 – How Internet banking channel devices connect to the bank**

### 2.1.4 Mobile banking

This channel may be as important as Internet banking channel among the youths who have smartphones or ordinary mobile phones and want to access various banking services conveniently (Ostergaard, 2010). However, research conducted in 2004 by Curry and Penman contradicts this finding. Users of this channel access banking services through their mobile phones, which connect to the bank through existing mobile telephone infrastructure by various telephone service providers. Technology overloads affect the performance of this channel (Daniela, Simona & Dragos, 2010). The figure below shows the infrastructure of this channel.



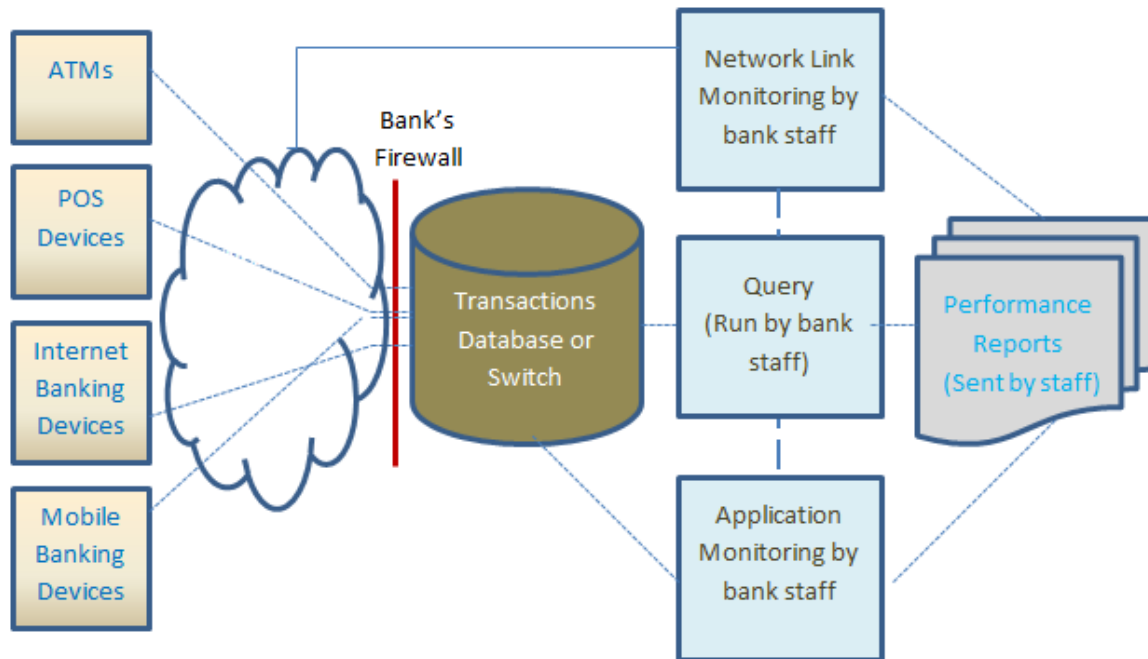
**Fig 2.5 – How Mobile banking channel devices connect to the bank**

## 2.4 Integrated performance monitoring

### 2.4.1 Current model for banking e-channels performance monitoring and reporting

Currently, banks use various techniques to monitor the performance of their electronic channels as E-banking services expand. Popular among these are database queries, application monitoring, network link monitoring are popular. The problems in this model include the following:

- Executing database queries or monitoring applications requires human intervention. Sharing performance reports is done via phone calls or emails sent by monitoring staffs.
- Notifications are not prompt because until workers identify a performance hitch, management may remain unaware of channels underperformance --leakage of revenue.
- When employees that are monitoring and reporting the performance of e-channels are out of office relevant parties may not know when e-channels experience underperformance.
- Staff members responsible for monitoring e-channels often have other activities. Being human, they might forget to monitor or report channel performance, creating a room for unnoticed channel underperformance, until, for instance, a customer complains.



**Fig 2.6 – Current scenario of e-channels performance monitoring and reporting**

#### **2.4.2 Current methods for banking e-channels performance monitoring**

There are various conventional or traditional tools and methods of performance monitoring in respect of various delivery channels of a bank. The most common tools and methods are:

##### **Server applications monitoring**

The use of dedicated personnel for monitoring channel activity using graphical tools is one of the commonest techniques banks employ currently. Monitoring the performance of server applications and services (Banks, 2012) such as POS device drivers for uptime/downtime is common among banks. However, this method is manual, meaning, when employees charged with the responsibility of monitoring channels performance fail to notice system failures.

##### **Rule-based performance monitoring**

Performance monitoring rules can be set on the transaction server applications. When these rules are applied, a performance decision is made based on selected performance parameters. The challenges with this approach include inadequacy in processing huge online datasets, absence of pattern recognition, and prevalence of high false positives (Estevez, Held & Perez, 2006). This makes the approach largely unsuitable.

## **Data mining techniques**

Queries can output transactional records that a performance analyst can assess based on historical performance to decide whether the channel is doing well. However, in practical data mining, patterns must be extracted that help the analyst predict if performance is as expected or otherwise there is a downtime for some reason(s). In cases where the volume is lower than expected, then troubleshooting mechanisms are launched. Data mining is time consuming, concentrates on recognizing certain performance patterns, and poses human challenges in analyzing its results. For this reason, this method is inadequate (Verma & Kusiak, 2012).

## **Multi-Agent system based monitoring**

Monitoring the performance of multiple e-channels of a busy bank is a complex problem, which requires a complex solution that Multi-Agent based technologies provides. In practice, intelligent collaborating agents are able to monitor different aspects of a delivery channel and report on the status of each channel autonomously. When well implemented, a Multi-Agent System is the perfect solution for this problem space (Gao *et al.*, 2006) hence why this study proposes its use.

### **2.4.3 Cases of successful system performance monitoring using multi-agents technology**

Intelligent agents have been used to monitor the performance of different systems:

- 1) Gao, Xu, Wang, and Green (2009) studied the possible use of software agents in the control and prevention of money laundering because this is a complex problem due to ever-changing risks that face banks. The software agents developed were found to have a better ability to monitor and diagnose different money laundering schemes and report suspicious activities. The research established that agent technology yielded a more flexible, knowledge-based, and adaptive solution to anti-money laundering in banks.
- 2) In stock trading, a multi-agent system is suitable for various operations including portfolio monitoring essentially, because the agent technology is able to handle the ever-changing nature of news and data in a distributed environment using a computational model other technologies are incapable of comprehensively handling. In their study, Luo, Liu, and Davis (2002) established that multi-agent technology suits the decision support needs for a stock trading business.

- 3) A research by Chan, Ray, and Parameswaran (2008) demonstrated that an agent-based health approach works very well in monitoring patients' health using a mobile e-health monitoring system. Multi-agents were made available on mobile devices thus making tracking of patients' details and results available remotely to patients and doctors. This reduced the dependence on Internet availability that other monitoring systems often use.
- 4) Another successful example involving multi-agents is that of ATM networks monitoring tool designed to monitor congestion of traffic on the network. The results of this study, which was conducted by Sreenivasulu, Prasad, and Raju (2011), showed that the use of multi-agent system bore greater benefits compared to using conventional methods.
- 5) In a study by Lau and Li (2005), it was established that multi-agent systems are a powerful tool and approach to support collaboration within a supply chain. In this case, one of the agents is a monitoring agent that obtains the performance monitoring criteria from other agents regarding disturbance events about the speed of processing. Upon receiving this information, the monitoring agent notifies another agent (coordination agent) every time such events occur. This approach yielded better performance monitoring results than conventional methods in use.
- 6) In another study by Godo *et al.* (2002), a multi-agent system was used to monitor the prescription of certain antibiotics under restriction of usage. The multi-agent system developed was able to assist the medics in revising the antibiotic prescriptions. In this case, agents such as pharmacy and patient agents were involved in checking various medical aspects relating to the prescribed therapy.
- 7) The Multi-Agent technology was used in conducting online diagnosis of partial discharge (PD) for apparatus with high voltage. In the paper "Studies on Multi-Agent Based Partial Discharge Online Monitoring System for High Voltage Apparatus", the authors applied this technology to achieve improved results in preprocessing, fingerprint analysis, features extraction, and trend analysis -all critical for partial discharge diagnosis, the completed system performed significantly better upon its evaluation (Yao *et al.*, 2009).

## **2.5 Agents, Multi-agents, and Agent Architectures**

### **2.5.1 Agents and Multi-Agents**

An agent is a term that has been used in various technologies such as databases, computer networks, operating systems, and artificial intelligence thereby making it difficult to find a common definition (Russel & Norvig, 2003 as cited in Bellifemine, Caire & Greenwood, 2007). Generally, agents are special autonomous software components, which avail an interoperable interface to other systems, and have behavior akin to human agents, whose goal is to perform certain tasks. A system composed of many agents may have the agents have mutual or varying goals in what is called a multi-agent system (Bellifemine, Caire & Greenwood, 2007).

### **2.5.2 Agent Architectures (Classes of agents)**

Agent architectures or classes are the basic means that determine the effective behavior of agents in a real world, open and dynamic environments (Bellifemine, Caire & Greenwood, 2007). Each architecture begins with an abstracted view of what the agents would be and ending with specific internal structures of agents and their operations. Each of these architectures has its benefits and limitations hence the need to choose a preferred architecture based on the problem at hand.

#### **Logic-based (symbolic) architecture**

These agents make their decisions based on logical deduction. This architecture is based on making a symbolic/logical representation of a given environment, desired behavior, and a manipulation of the environment using syntax. Their successful development requires using logical formulae for symbolic representation, and theorem proving (logical deduction) using syntactic manipulation (Mangina, 2002; Bellifemine, Caire & Greenwood, 2007)

#### **Deliberative architecture / Belief-desire-intention (BDI) architecture**

This architecture enables agents to make decisions based on how they manipulate data structures developed to represent different inbuilt agent beliefs, desires, and intentions. This is based on practical reasoning (Mangina, 2002). Bellifemine, Caire and Greenwood (2007) argue that BDI architectures are deliberative architectures based on reason-driven action. The key attributes of practical reasoning are (1) deciding achievable goals and (2) deciding how to attain those goals. The former is called *deliberation* while the latter is called *means-end* reasoning.

Available choices in deciding the action to take to achieve goals become *intentions* that subsequently determine action an agent would perform. It is these intentions, through a feedback mechanism, that influence an agent's practical reasoning in the future. This is because the intentions lead to action. Moreover, intentions relate very closely with beliefs (Mangina, 2002).

### **Reactive / behavioral architecture**

These are agents in which decision-making is achieved through some kind of direct mapping involving changing a situation into an action. According to Bellifemine, Caire and Greenwood (2007, p.4), this architecture is a stimulus-response type of architecture. In fact, reactive agents have behaviors aimed at accomplishing tasks through performing specific tasks achieved through mapping of perceptual input to desired action (Mangina, 2002).

### **Layered (hybrid) architecture**

In this agent architecture, agents make their decisions through implementations on different software layers. Each of the layers has some level of explicit reasoning given the existing environment but at various abstraction levels. The architecture is a blend of both deliberative and reactive architectures (Bellifemine, Caire & Greenwood, 2007).

## **2.6 Applications of Multi-Agent Systems technology**

Because of the ever-advancing technologies in the AI and general computing, MAS are increasingly getting a widespread application globally. MAS have been applied in various areas for personal assistance and industrial use for mission-critical systems. It is within the industrial environment that MAS were initially developed to work. MAS systems are used for among other mission-critical applications including the following:

1. Control of industrial processes
2. Business process modeling
3. Systems diagnostics
4. Transportation logistics
5. Network management
6. Manufacturing
7. Information management



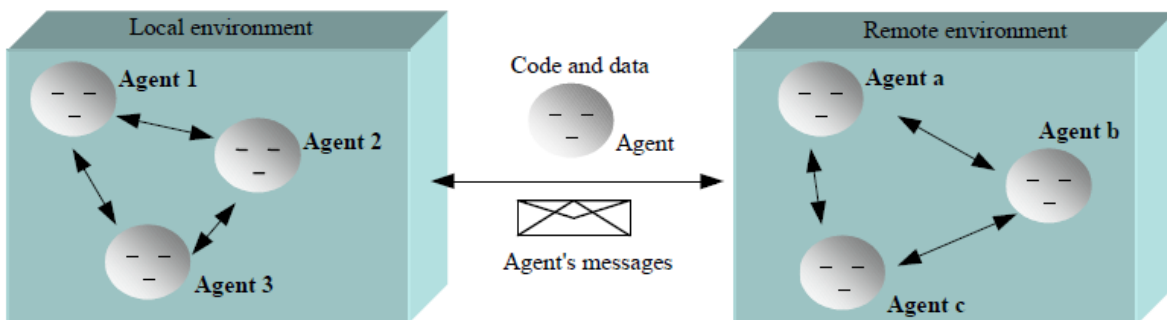
MAS application in information management is especially important because of the distributed nature of Internet resources. Examples of MAS application in the Internet environment is searching for information and internet banking, which is an aspect this paper tackles but from the perspective of performance monitoring (Bellifemine, Caire & Greenwood, 2007).

## 2.7 Multi-agent methodologies and Frameworks

### 2.7.1 Methodologies and frameworks for multi-agent systems

MAS software development methodologies encompass what software engineers need to analyze and design MAS software systems and such requirements include concepts, processes, models and notations, techniques, and tool support. Of the many methodologies available for engineering Agent Oriented Software, Prometheus, MaSE, Tropos, and Gaia are the most widely used because of their greater maturity (Bordini, Dastani and Winikoff, 2006). The development of the system prototype used in this study will apply the Tropos methodology.

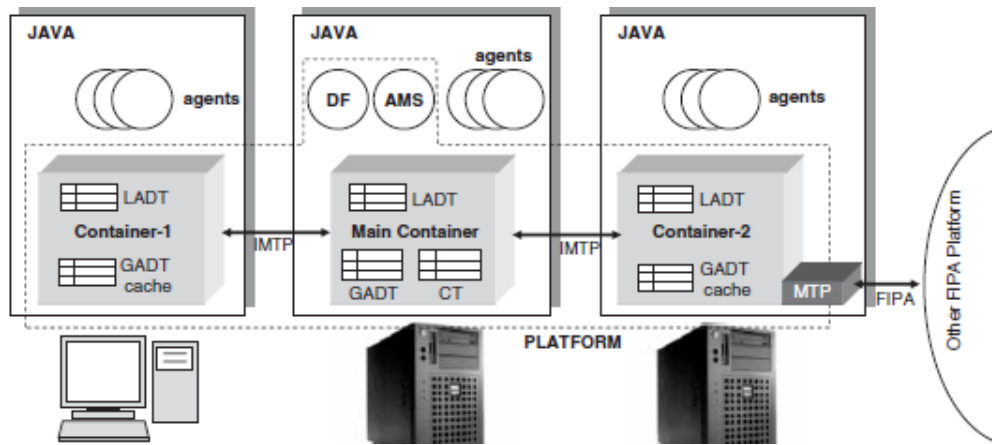
A Multi-agent framework is a technique applied to make it possible to design and code Multi-agent systems. This is made possible through high-level abstraction from within a given development domain. Frameworks make it possible to design agent-based systems without re-inventing the wheel. In using a framework, every agent has inbuilt abilities to perceive its environment; engage in communication with other agents, applications and users; and make decisions on the next actions. This ensures that users have limited information and work overload. Examples of MAS that can be built using frameworks are meeting schedulers, interface agents, intelligent tutors and performance monitoring and reporting agents (Avancini, 2000). For instance, the figure below shows a MAS model depicting agents operating in a local and remote environment.



**Fig 2.7 - A model for a multi-agent system**

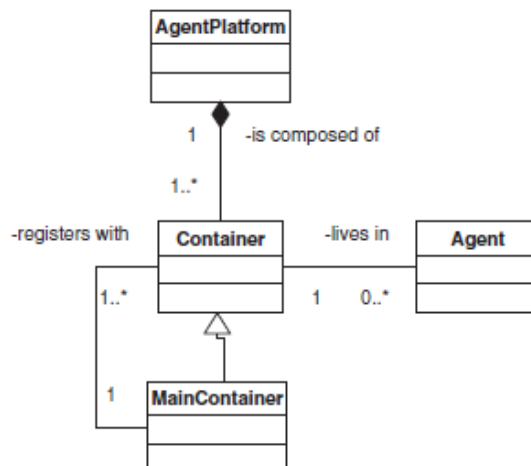
### 2.7.2 Java Agent Development Environment (JADE)

JADE is arguably, it is the most widespread agent-oriented technology platform in use today. It is a fully distributed and flexible middleware system, permitting easy customization and inclusion of add-on modules to extend its use. As a development framework, JADE supports complete agent-based software development through a run-time environment that puts into effect features that support an agent during its life cycle, graphical tools and the agents' core logic. Being an implementation made in Java, JADE benefits from Java's rich library and programming abstractions that enable development of MAS with relatively limited skills in agent theory (Bellifemine, Caire & Greenwood, 2007). Figure below shows a JADE architecture.



**Fig 2.8 - JADE Architecture**

In a JADE's runtime, agents live in containers, which in turn exist in platforms.



**Fig 2.9 - The relationship between JADE's main architectural elements**

## 2.8 The proposed Model

### 2.8.1 Context Diagram

The following diagram depicts the context in which the model this research will develop will operate. All the bank's channels will connect to the bank through their normal network links. For performance monitoring purposes, intelligent agents will interact with the database to monitor performance by mining intelligence information that will show the performance of the various channels leading to the production of reports showing the performance of each channel and a consolidated report giving a summary of how all the channels are performing.

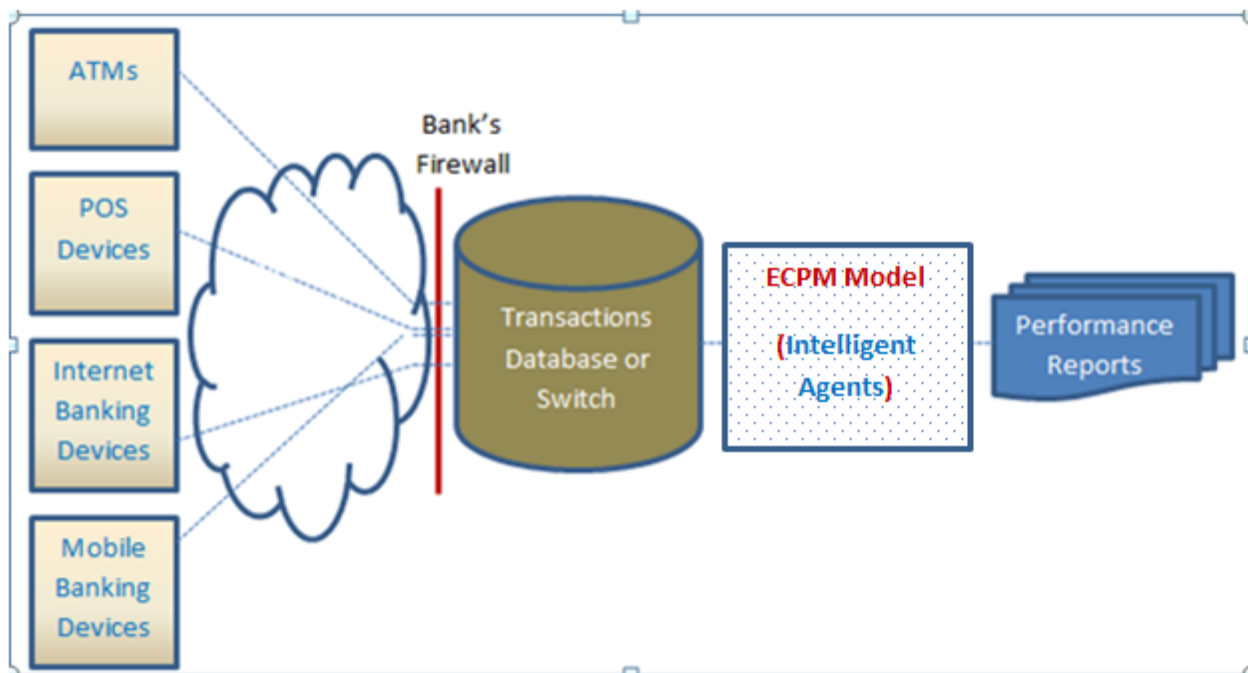
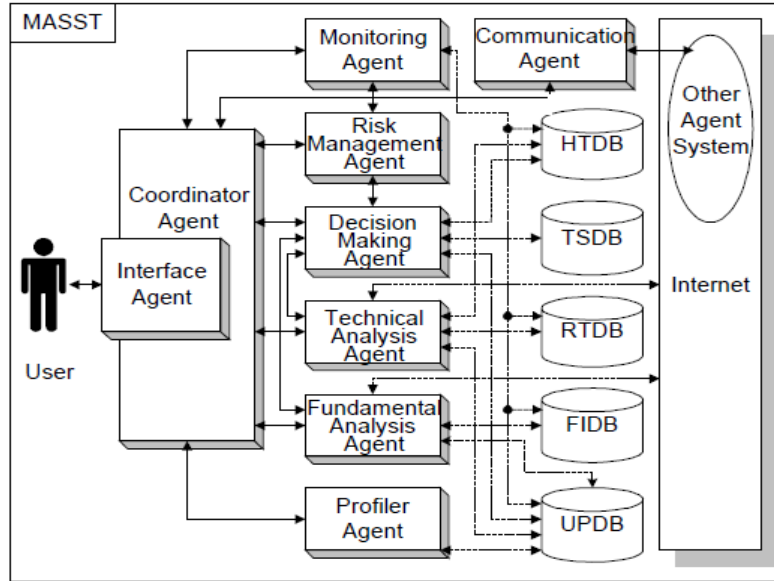


Fig 2.10 – Context diagram for the proposed ECPM model

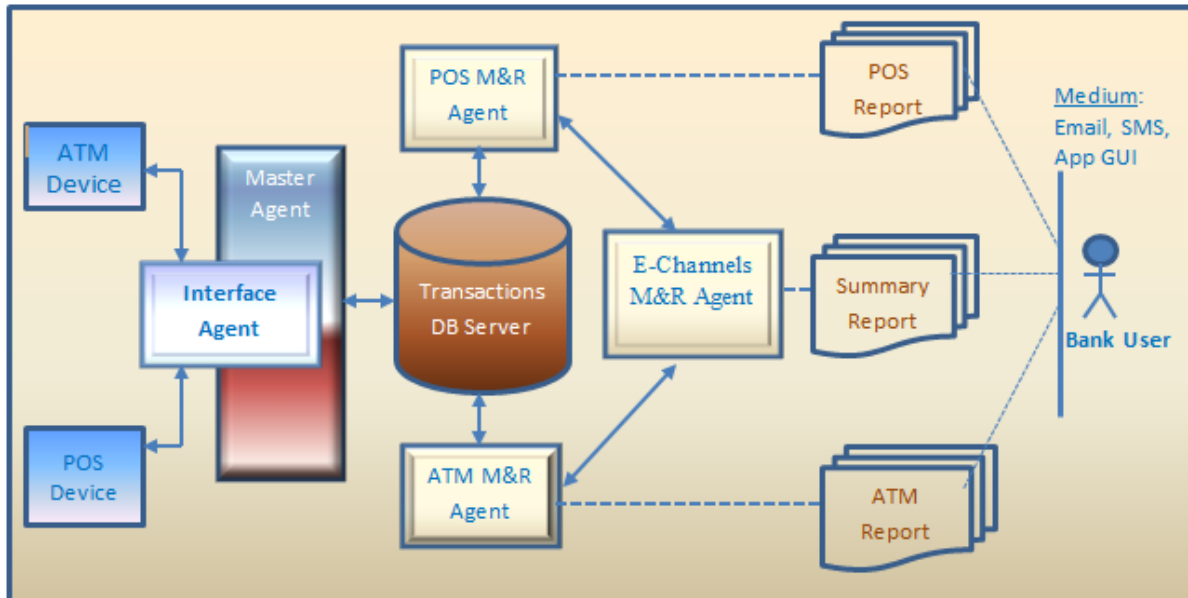
### 2.8.2 Conceptual model of the proposed ECPM Model

Luo, Liu, and Davis (2002) developed the framework shown in figure 2.11 in their research entitled “A multi-agent system framework for decision support in Stock Trading”. This framework acts as the intermediary between investors operating in a stock market and the supplier side. The relevance of the model to this research is that the model uses multi-agents doing different roles to achieve the goal of informing decisions on stock trading. It forms foundation on which the conceptual model for this research was developed.



**Fig 2.11 - MASST Framework for decision making in stock trading**

In the conceptual model below, this research proposes using one agent per channel for monitoring and reporting the performance of those channels. Then, to generate a consolidated performance report, an e-channels agent is required to combine reports from individual channel agents and produce a summary report. In this model, the Interface Agent sends transactions to the database server for monitoring. This model is derived from the MASST framework above.



**Fig 2.12 – Conceptual model for the proposed ECPM Model**

In the above diagram, only two channels are depicted because of the limitation imposed by the scope of this research. The Master Agent coordinates the activities of the other agents while the Transactions Agent interfaces with the various channels to monitor transactions from the various channels going into the Transactions DB (database) Server. The ATM M&R Agent monitors and reports the performance of the ATM channel while the POS M&R Agent does the same but for ATM channel. The E-Channels M&R Agents picks high-level details from the reports generated by the ATM M&R Agent and POS M&R Agent and produces a Summary Report. Thus, where other channels are integrated into the model, additional agents will be required for monitoring and reporting their performance while the E-Channels M&R Agent will produce a bigger summary report because of additional details. This makes this model scalable.

#### **Benefits of the proposed MAS as an e-channels performance monitoring technique**

- ✓ The system runs with little or no human intervention
- ✓ Reports are auto-generated and auto-emailed, requiring no human intervention
- ✓ Scalable – more channels can be added into the solution with limited modification
- ✓ Email notifications will ensure relevant mobile employees remain informed of when systems are underperforming so they can take necessary steps to restore services

#### **2.9 Chapter Summary**

This chapter covered a review of published literature regarding the research topic. In the initial subtopics (2.1 through 2.4), the study reviewed literature on electronic channel banking, culminating in the use of multi-agent systems in monitoring the performance of electronic channels for banks. The study identified and briefly explained four key electronic service delivery channels that most banks use. These include Internet banking, Mobile banking, Automated Teller Machines and Point of Sale (POS) devices. The drive behind monitoring how these channels perform is to ensure optimal uptime on these channels and consequently minimal service interruptions in a bid to foster higher levels of customer expectation management through improved, convenient and seamless service provision.

The study established that banks and customers benefit from a multi-channel service delivery paradigm. The paper revealed that many banks do not have an integrated electronic channels performance-monitoring tool. For this reason, they use various approaches including employees who periodically keep checking volumes of transactions on host systems to check for transactions flow as a way of determining uptimes and downtimes. However, the study assures that the use of multi-agent systems in performance monitoring for electronic channels will not only make the process automated but will offer many more benefits. This part culminated in presenting examples of cases where multi-agent systems have been successful in system performance monitoring.

The subsequent subtopics (2.5 through 2.8) reviewed literature on the multi-agent systems technology. This section began with presenting working definitions or descriptions of agents (software agents) and multi-agent systems. This was followed by a section describing the agent architectures: logic-based, deliberative, reactive and layered architectures. The next section was the applications of multi-agent systems that listed key mission-critical application areas for MAS. Multi-agent methodologies were described in brief. This literature review ended with a brief description of the frameworks for MAS and JADE that will be used in the development of a prototype for this study. Tropos methodology was identified as what will guide the design of the model. The next chapter, methodology, will dwell on the design, development and testing of the model based on data collected and the conceptual model developed in this chapter.

## **CHAPTER 3: METHODOLOGY**

### **3.0 Introduction**

This chapter covers aspects involved in the design, development and testing of banking e-channels performance monitoring and reporting model using MAS technology. The first section is the research design and is followed by data collection and analysis to inform model and database design. The next section is the actual database and model design. The final section is prototype development and testing. In the bid to design a model that would adequately address the research problem, the following four main tasks were carried out:

1. Channel activity detection
2. Channel performance reporting
3. Implementation of the proposed system prototype
4. Testing how the implemented system prototype works

### **3.1 Research Design and its justification**

Because of the nature of the research problem, both quantitative and qualitative research methods were employed. Primary data was obtained from local banks employees. The sampled banks support ATM and POS channels since these two are in the scope of this study. The study used questionnaires that were given to employees who handle performance monitoring or system administration for their respective e-channels. The combination of these two categories of employees helped obtain data about aspects one should look at when monitoring the performance of an electronic channel on one hand and the kind of database table fields useful in monitoring and reporting channel performance. This data was significant in the design of the prototype.

An in-depth literature review of scholarly publications was useful as well. This was an exploratory research methodology that was suitable for this study's problem, which sought to identify application areas where multi-agent systems have been employed to monitor performance of an electronic system preferably in retail banking channels, systems performance monitoring, and multi-agent systems. The literature materials included journal articles, academic textbooks, academic papers, and analysis reports. This review also covered recent knowledge on the design, development, and implementation of different multi-agent system technologies in mission-critical system performance monitoring was conducted.

Finally, a prototype study methodology was used in formulating the proposed model for monitoring the performance of the two e-channels used banking. By using the Design Science methodology (Vaishnavi & Kuechler, 2004), the study sought to develop a prototype that would express and demonstrate the novelty of the solution suggested. This proof-of-concept prototype research methodology adopted in this study is frequently used in clinical studies (Lawrence, 2005; Fardon *et al.*, 2007). The prototype development aspect benefitted from using Tropos methodology, which entailed the following four phases:

### **Early Requirements Analysis**

This involves identifying and conducting an analysis of the stakeholders of the e-channels performance monitoring and reporting problem in a banking setup. Thus, the intentions of different stakeholders were identified and consequently analyzed. This was followed with modeling the stakeholders as social actors. In this modeling, their goals, plans, needed resources, and dependencies were modeled. This was necessary to support decision-making.

### **Late Requirements Analysis**

At this phase, the research dived into analyzing the electronic channels performance monitoring system within its environment of operation. In this case, the system was seen as an actor that depends on several other actors operating within and without a banking setup. Such external actors include bank clients.

### **Architectural Design of the model**

In this third phase of design, the overall architecture of the system was seen as consisting of different subsystems that were also represented in the form of actors with interconnections and dependencies modeled using control flows and data. Further different plans and capabilities that these actors need to realize their goals were identified and agents to carry out these plans and realize the goals were defined.

### **Detailed Design of the model**

In this stage, a detailed description of all agents was carried out. Agents' beliefs, capabilities, and goals were specified accordingly. A description of agents' communication was defined.



## **Development of the prototype**

Being the last phase in designing the MAS-based e-channels performance monitoring and reporting solution following Tropos methodology, JADE framework was used.

### **3.2 Data Collection**

#### **3.2.1 Sources of data and their relevance to the problem**

The purpose for carrying out data collection in this project was to facilitate the understanding of performance monitoring of e-channels in a real-world bank. More specifically, the exercise aimed at establishing the structure of transactional data held by various retail banks. However, due to challenges of such transactional data being confidential hence protected by consumer privacy and confidentiality laws, the data gathered from the selected respondents working in the local banks was useful in simulating financial data need to meet the objectives of this study.

#### **3.2.2 Tools, procedures and methods for data collection and their justification**

The study conducted data collection by means of completing two questionnaires (see Appendixes A and C of this project report). Six employees from three different retail banks in Kenya, with each bank providing two respondents, completed the first questionnaire. Thus, the researcher sent six questionnaires to six employees of three retail banks in Kenya that support ATMs and POS channels besides other electronic channels. This was a 100% sample because only three banks in Kenya support POS channel and the project scope was limited to the two channels. These banks enable their customers to perform transactions on ATMs and POS devices in addition to other electronic channels such as mobile and internet banking. Four questionnaires were delivered by hand while two questionnaires were emailed. Responses to the questionnaires delivered by hand were collected at the reception desks of the banks' departments/divisions. Those questionnaires that were emailed had their responses similarly emailed back as scanned copies. The respondents returned the questionnaires in a week's time as requested.

These questionnaires were responded to by two categories of respondents: system administrators and performance monitoring staff members responsible for different independent/integrated e-channels. Of the six respondents, 50% held the positions of Systems Administrators while the other 50% of respondents held positions of Technical Services or Channel Managers with a

mandate to monitor performance of specific e-channels of the banks they work for. All the six respondents worked at management levels. The questionnaire in Appendix C was used during User Acceptance Testing to obtain user comments about the solution developed in this work.

Important details like the structures of bank records or database tables holding transactions from different channels and key pointers that a channel is experiencing downtimes. These pieces of data were important in designing agents that make decisions on whether certain channels are up and running (performing normally) or down (performing below normal or expected targets or not working at all) and therefore be able to inform relevant staff about the status while management benefits from knowing the overall performance of channels for the decision making.

Now, because transactional data is standardized, the structure of database tables obtained from the three sampled banks was found to be more or less the same and was assumed representative of transactional table structures in any retail bank. Moreover, the structures follow a standard provided by PCI/DSS ISO 8583 Transaction Data Elements Format given to banks for compliance. This data was used to design dummy database tables and feed them with dummy data using an active software agent, using the dummy samples provided by respondents. This was necessary before running the prototype.

### **3.2.3 Data Analysis**

Data obtained from the questionnaires about the database table structure was analyzed for validity and reliability. Responses were a mixture of open-ended and closed-ended answers with appropriate restrictions with both qualitative and quantitative responses required. A basic analysis using Microsoft Office Excel 2010 application sufficed for tabulation of the results. This analysis was critical to the simulation of data used in this study by providing room to ensure simulated data was as close as possible to the real world data held in banks.

To remain consistent with the study's scope, only data relating to transactions from ATMs and POS devices was collected and simulated. The analysis presented in this chapter further provided help in the designing of classification algorithms employed in determining the performance of different electronic channels in banks. Finally, it proved significantly useful in the design and development of the E-channels Performance Monitoring System Prototype.

### Analysis of the Data from the Questionnaires

Six (100%) responses were obtained between June 2, 2014 and June 6, 2014. This yielded a 100% response rate. The following are the deductions obtained from the responses. Table 3.1 presents an analysis of the data obtained with respect to data structures of financial or transactional records held by banks.

**Table 3.1 - An analysis of data gathered from completed questionnaires**

<i>Transactional Data</i>	<i>% of respondents that indicated the usefulness of the transactional data item in electronic channel performance monitoring and reporting</i>
Date/Time	100%
Terminal ID	100%
Terminal Type	100%
Terminal Owner	50%
Transaction Amount	100%
Currency Type	75%
Transaction Type	25%
Completion Status	100%

A more detailed analysis of the feedback obtained from the questionnaires is highlighted below:

- 1) All respondents agreed that their banks monitor the performance of their e- channels.
- 2) All respondents indicated their banks depend on database queries to generate reports. 75% indicated they use application monitoring and database queries while 25% indicated they depend on graphical tools used for monitoring systems. No respondent (0%) noted that they use agent-based solutions in monitoring the performance of their e-channels
- 3) All respondents stated that the most frequent challenge in performance monitoring of e-channels is network outages and system service or server instabilities (75%).
- 4) All respondents noted that different customers have varying preferences of e-channels they use. The two commonest electronic channels preferred by different customers are ATMs and POS devices. This is why banks focus more on these two. The banks have a rising interest in Mobile and Internet banking channels as technology advances.

- 5) All respondents (100%) indicated that there are different teams employed by banks to monitor the performance of e-channels. 50% of the banks have an integrated database server but none (0%) have an integrated monitoring solution for their e-channels.
- 6) All respondents (100%) indicated that banks have varying preferences regarding each channel based on its revenue generation per unit time. In this case, the ranking was done as follows: ATMs, POS devices, and Mobile Banking.
- 7) All respondents indicated that transaction completion status was vital in determining performance of an e-channel. This status includes “Approved, Declined, Error, etc.
- 8) Terminal ID, Transaction Amount, Date & Time, Type of Transaction, Currency Type, and Completion Status were the key data fields extracted for performance reporting.

**Table 3.2 - The different solutions banks use to monitor the performance of e-channels**

Tool	Users	% Users
Application Monitoring	6	100.00
Data Mining Tools	0	0.00
Database Queries	6	100.00
Graphical Tools	1	16.67
Multi-Agent Systems Technology	0	0.00
Network Link Monitoring	2	33.33
Rule-based Technology	0	0.00

The above table shows that all banks sampled for this study use application monitoring and database queries in monitoring the performance of their e-channels. 33.33% of the responses were in favor of using network link monitoring while 16.67% responses indicated the use of graphical tools. Data mining tools, multi-agent technology, and rule-based technology received a 0% response showing no banks currently use these technologies.

**Table 3.3 - The preferred e-channels by banks**

Channel	Use it	Ranking	%Preference
ATMs	6	1	100.00
POS Devices	6	2	100.00
Internet Banking	6	3	100.00
Mobile Banking	6	4	100.00

Table 3.3 above shows bank’s preferences to ATM, POS, Internet and Mobile banking channels

**Table 3.4 - Percentages of media used for sending performance reports in banks**

Medium	Users	%Users
System Interface	6	100.00
Email	3	50.00
SMS	0	0.00
Phone Calls	2	33.33

The above table shows that all the sampled banks depend on the screen output showing the performance of electronic channels. Only 50% of the banks email performance reports to the consumers of the reports while 33.33% of the banks use phone calls to alert intended users of the reports or channel performance. No bank uses SMS to send performance reports though.

### **3.2.4 Purpose of Data Collected**

#### **Use of data collected in the Design and testing of the model**

The findings helped the research design the prototype for electronic channel performance monitoring based on the key determinant factor: system/server uptime, which is achieved through database queries for real-time transactions and time gaps between neighboring transactions. This was based on the finding that 100% of the responses indicated the use of database queries to show uptime and downtime of a channel in monitoring and reporting the performance of each channel. The second purpose of the data collected was useful in the design of the database. This was made possible through the responses received showing the key data fields that constitute a transaction. These data fields included:

1. Date/Time
2. Terminal ID
3. Terminal Type
4. Terminal Owner
5. Transaction Amount
6. Currency Type
7. Transaction Type
8. Completion Status

#### **Use of data collection in Model evaluation**

In this regard, data collected included key parameters that banks consider vital in determining if a channel is performing normally or below normal targets. These parameters included:

1. Total number of transactions per unit time
2. Total failure rate (i.e. number of failed /successful transactions per unit time)
3. Total amount (monetary value) of transactions processed in a given unit of time
4. The duration of network outages

### 3.2.5 Limitations of the methodology and how they are overcome

The greatest limitation in this study was lack of primary data sources for the prototype. This problem arose from the fact that, legally, banks cannot give unauthorized access to their transactional data for confidentiality and privacy concerns. This problem was overcome by means of simulating such data based on responses to questionnaires. Second, the concept of MAS in this study is relatively new to many banking institutions as most banks are focusing on having permanent staffs for performance monitoring using database queries. Third, there is very little published literature relating to performance monitoring for electronic channels but sufficient literature was available on multi-agent technology in general. This hindered an all-inclusive investigation of the research problem thus making it difficult to generalize the findings. Banks may have different performance monitoring tools. For this reason, coupled with the benefits mentioned earlier about using MAS, the findings are beneficial to banks that do not already have an integrated electronic channels performance-monitoring tool.

### 3.3 Database and Model Design

The following conceptual ECPM model was designed to meet the requirements of a monitoring and reporting tool for banking e-channels. The model features five agents that will be designed with key agent plans and capabilities. Central to their functioning in channel monitoring and performance reporting is a transactions database.

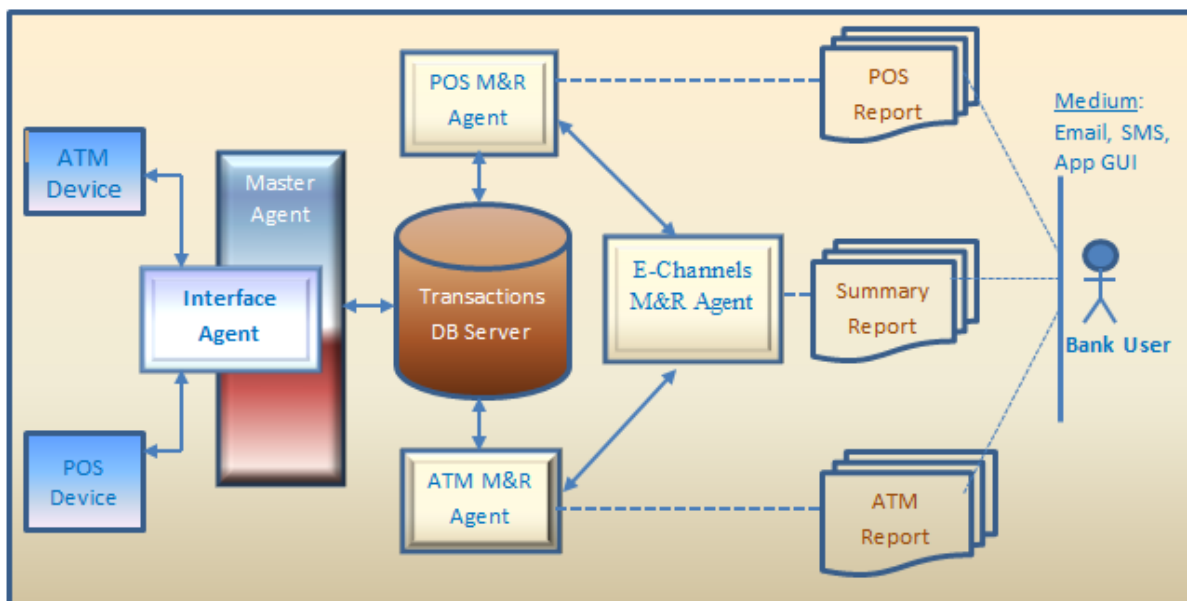


Fig 3.1: The conceptual ECPM model

## **Model Requirements**

To accomplish these tasks, the prototype had to meet the following functional requirements:

1. Generate simulated dummy transactions for ATM and POS channels
2. Monitor and report performance of each channel based on:
  - a) The number of successful transactions
  - b) Number of failed transactions
  - c) Total amount for successful transactions
3. Generate a Consolidated Report summarizing the performance of the two channels
4. Send Email notifications when the following events occur:
  - a) All channels are currently experiencing service or network outage
  - b) Either channel is experiencing an outage or services are not running
  - c) The number of failed transactions per unit time exceeds the allowed maximum.

To achieve the above functions, the ECPM model must therefore contain two components: (1) a database, and (2), the application using the database. The design of the database will be based on the data structures indicated in the responses from questionnaire respondents. The complete model will need to have a database containing simulated transactions and five intelligent agents responsible for both simulation of these transactions in addition to monitoring and reporting the performance of the various banking e-channels.

### **3.3.2 Database Design**

The ideal scenario for this project would have been the use of real-world transactional data from a bank and run the prototype in a banking environment. However, because transaction data is private and confidential, banks would not allow access to that data. For this reason, the project depended on simulated data that generally remained as close as possible to the real world data, fit for performance monitoring and reporting purposes. The simulated data was created out of the responses from bank employees who completed questionnaires sent to them to collect such information. All respondents were from two local banks in Kenya.

### **The objective of data simulation**

This work used simulation with the objective of generating the following kinds of data:

- Date and time of transaction
- Terminal/Device Identity
- Transaction amount
- Type of transaction
- Completion status of the transaction

Essential data structures from the above include:

1. Date and time
2. Terminal ID
3. Transaction Amount
4. Type of transaction
5. Transaction Completion Status

### **3.3.3 Model Design**

#### **Stage 1: Early requirements analysis**

At this stage, different stakeholders were identified and analyzed to help establish their intentions. The modeling of these stakeholders as social actors that rely on one another in a bid to attain their goals was done, which encompassed environmental analysis.

#### **Stakeholder identification**

- 1) Bank: uses a transaction processing system (or Card Switch) to process transactions originating with customers at various e-channels. The institution is also responsible for monitoring these transactions to help management make appropriate decisions based on performance of different channels at any given time or when seeking channel expansion
- 2) Card Switch (Transaction Processing System): platform for approving or switching card transactions from different electronic channels.

#### **Stakeholders' goals identification**

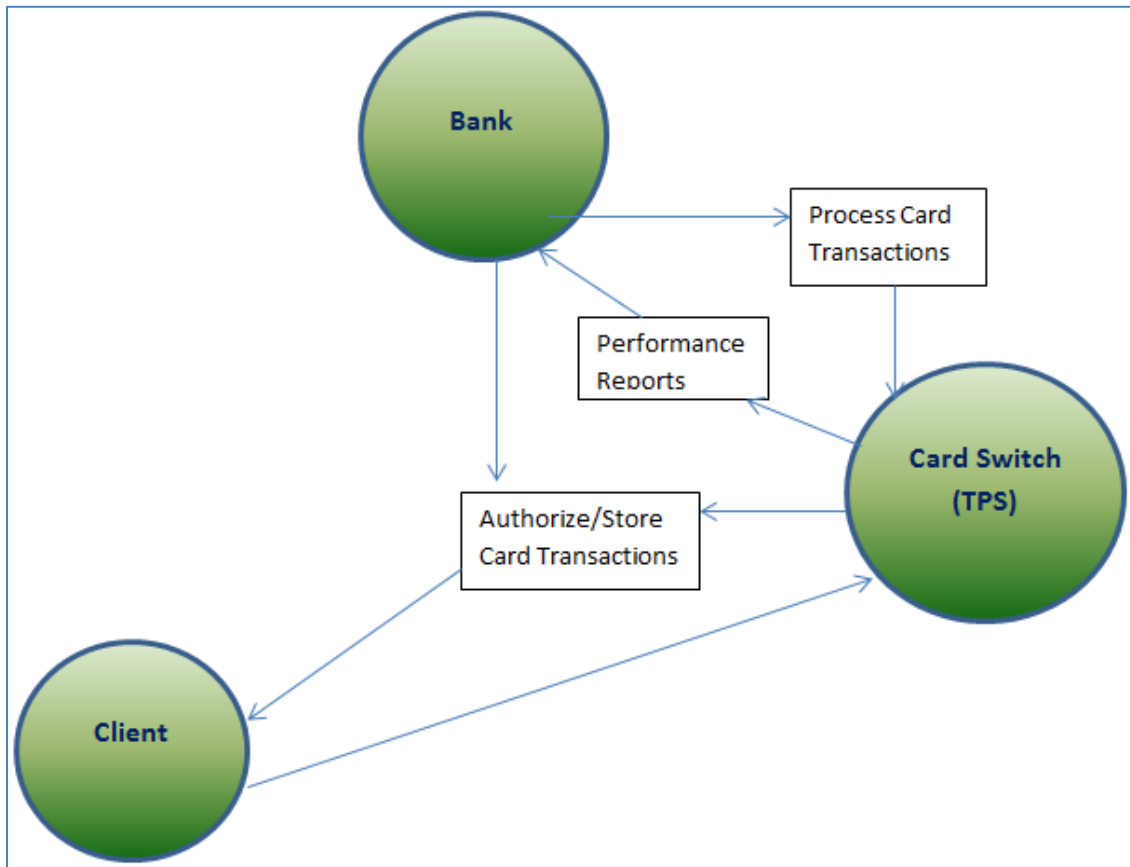
*Bank (Financial Institution):*

- Process bank transactions by customers
- Monitor the performance of various electronic channels
- Ensure service delivery to customers via different e-channels



*Card Switch (Transaction Processing System Server):*

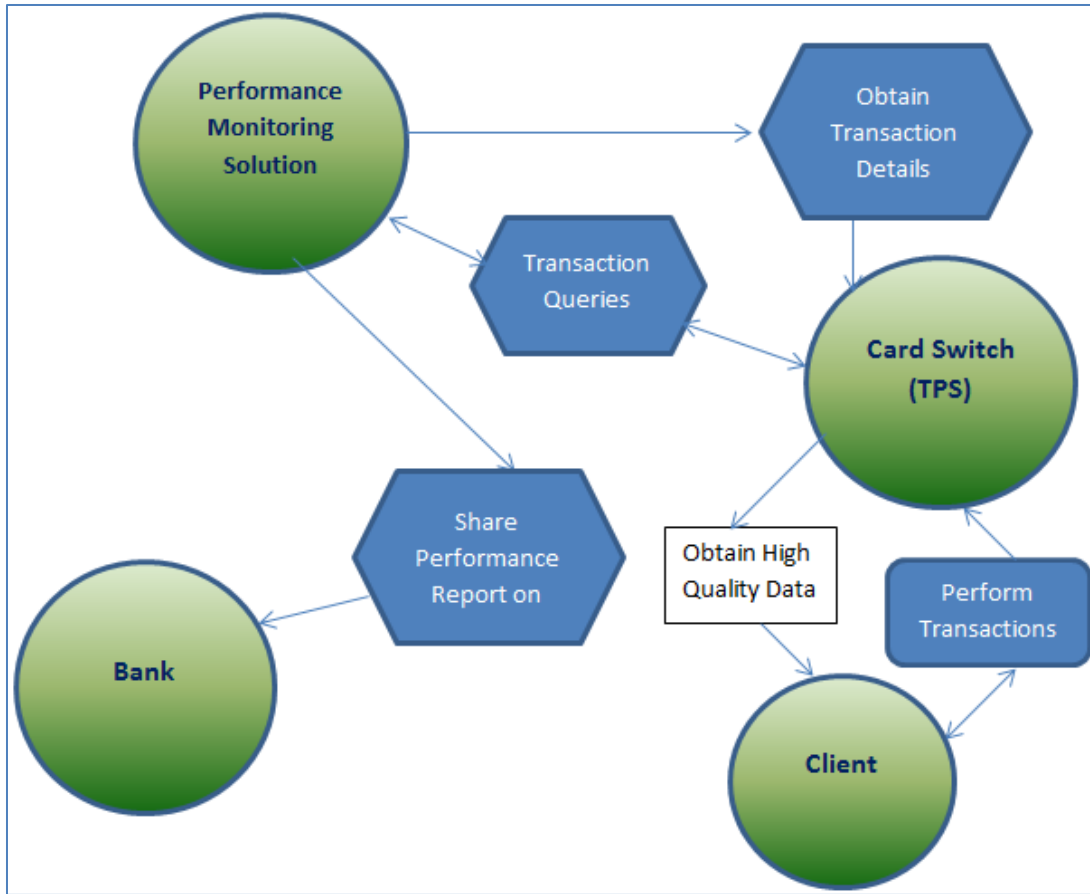
- Process transactions from different electronic channels of a bank
- Provide tools for basic transaction monitoring and reporting



**Fig 3.1- Actor-diagram for the E-Channels Monitoring Stakeholders**

### **Stage 2: Late Requirements Analysis**

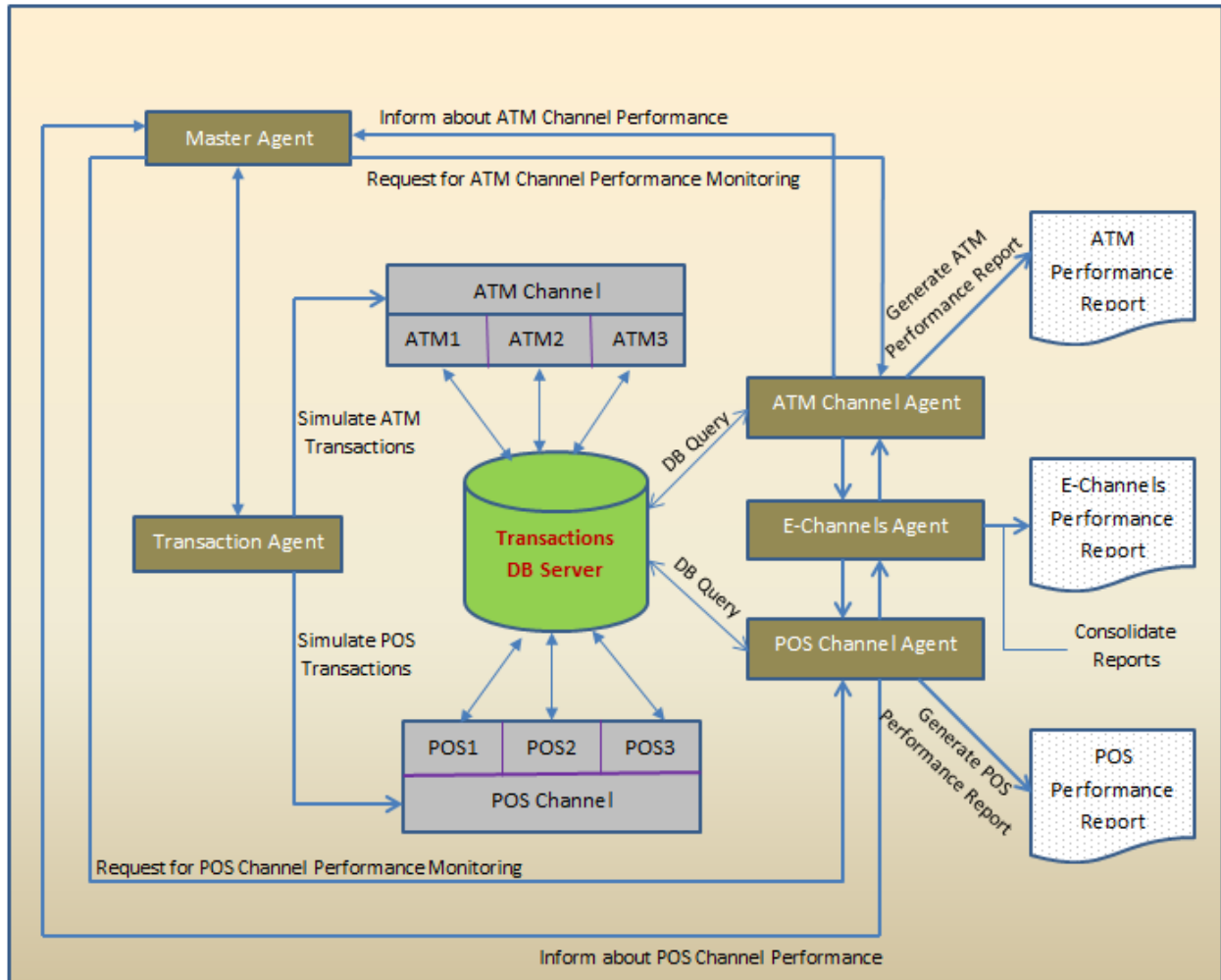
At this stage, the actor will be the developed performance monitoring solution. The key purpose of this solution will be the identification and interpretation of both functional and non-functional requirements and establishing the dependencies between the system-to-be and other bank actors.



**Fig 3.2 - The System-to-be operating in its environment as an actor**

### **Stage 3: Architectural Design**

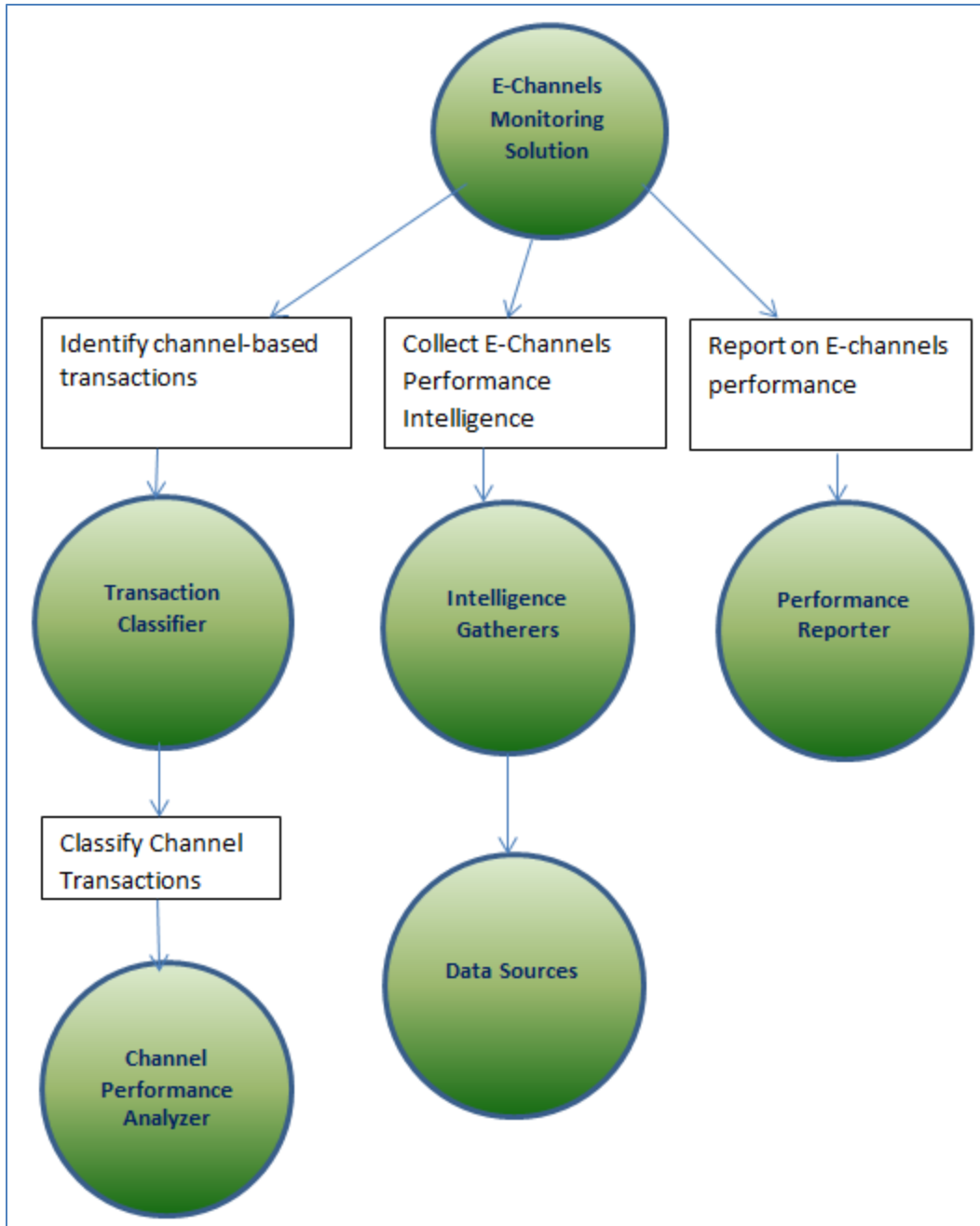
This section presents the overall architectural design of the e-channels model basing it on the various subsystems that will be interconnected by means of control/data flows. The independent subsystems are modeled as actors. Dependencies are modeled using data flows and/or control flows. The figure below is the detailed model for e-channels performance monitoring in banks. The model below shows the architecture of the model this research intends to develop.



**Fig 3.3 – The proposed ECPM Model**

**Step 1: The overall architectural design of the e-channels performance monitoring system**

This stage introduces some new actors whose roles include delegating sub-goals and the achievement of the solution’s non-functional requirements.



**Fig 3.4 - Actor diagram for the e-channels performance monitoring system**

**Step 2: The identification of actors' capabilities required to achieve their goals**

By interpreting the actor diagram above, different capabilities of agents emerged.

**Table 3.5 - Actors capabilities**

Name of Actor	No.	Actor's Capability
Transaction Generator	1	Generate Transactions for both ATM & POS channels
	2	Post the transactions to the database and browser's window
	3	Send email notifications on system's status
ATM Reporter	5	Monitor or Check ATM transactions
	6	Generate ATM performance reports
	7	Advice Master Reporter of the report generated
POS Reporter	8	Monitor or Check POS transactions
	9	Generate POS performance reports
	10	Advice Channels Reporter and Master Reporter of the report
Channels Reporter	11	Receive report messages from POS and ATM channel reporters
	12	Compile Consolidated Report
	13	Share Report via different media
Master Reporter	14	Request other agents to execute their tasks
	15	Report the execution of tasks by other agents

**Step 3: Assignment of agents**

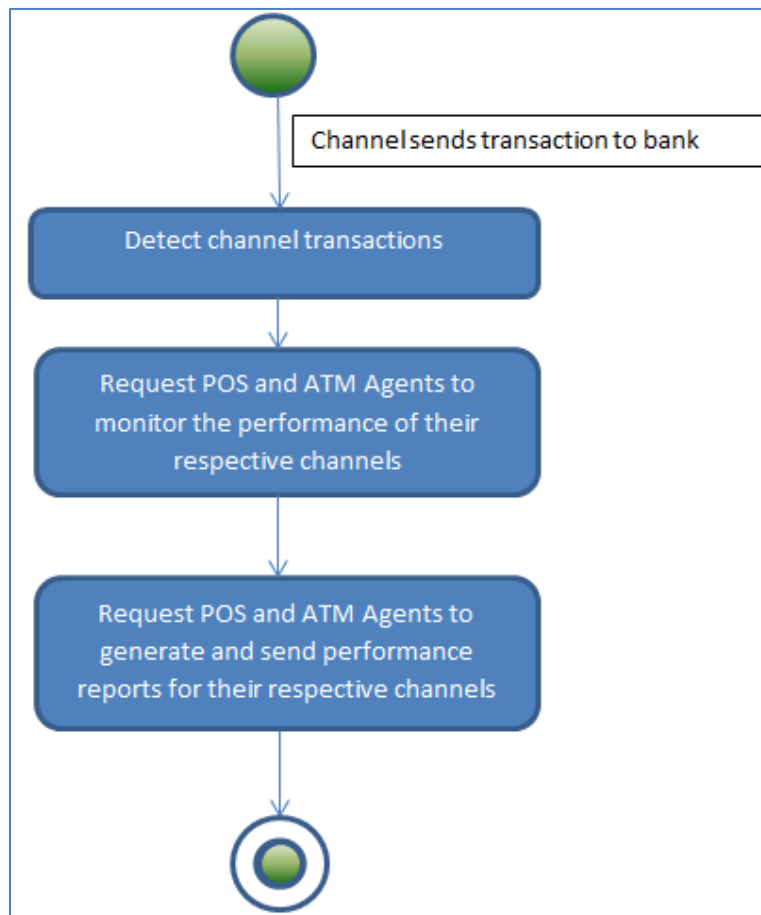
This phase terminates the design phase. Its key purpose is to identify and define the different agents required in the system and assigning them necessary capabilities aimed at achieving the overall goal of the system. The Table 3.6 below shows the various agents and their assigned capabilities. For agent capabilities assignment, see Table 4 in the previous section.

**Table 3.6 - Types of agents and their assigned capabilities**

Type of Agent	Assigned Capabilities
Transaction Generator Agent	1-4
ATM Reporting Agent	5-7
POS Reporting Agent	8-10
Channels Reporting Agent	11-13
Master Agent	14-15

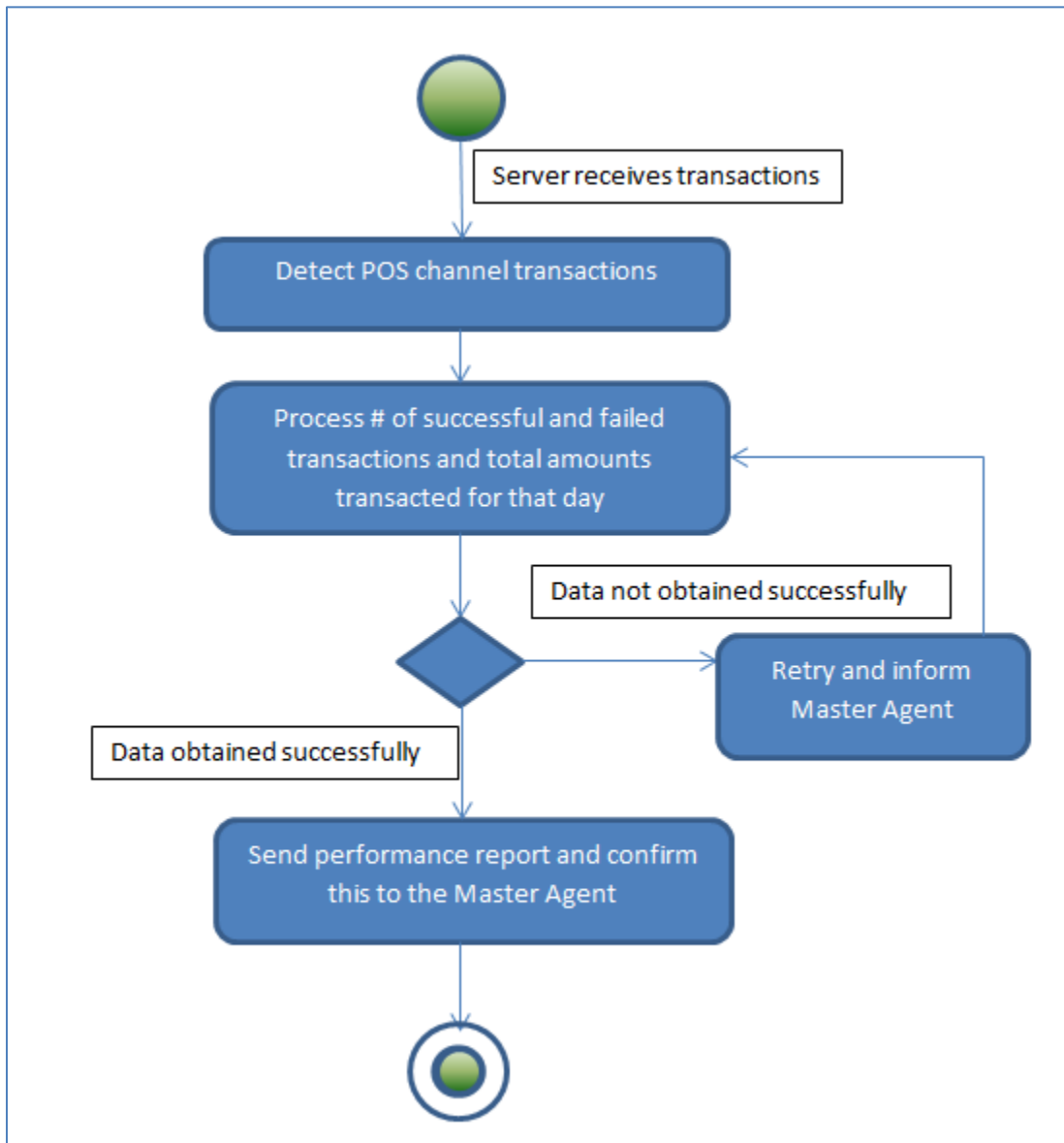
#### Stage 4: Detailed Design

In this phase, the objective is to establish detailed agents specifications. This involves detailed specification of the beliefs, capabilities, goals, and communication details among all involved agents. In this report, two levels of activity diagrams to represent the plans and capabilities of agents. These are diagrams are called plan diagrams and capability diagrams. Capability diagrams consist of plan nodes that essentially permit the modeling of agent capabilities. The plan diagram then depicts all plan nodes on the capability diagram. The figure below shows the capability diagram for the Master Agent. When the Transaction Agent is running, it keeps populating the database with simulated transactions that PosReporting and AtmReporting agents monitor. For this to happen, the Master agent, upon detecting new transactions, sends request to PosReporting and AtmReporting agents to prepare and submit their reports on the performance status of POS and ATM channels. The two agents then interrogate their channels and submit status reports and informs the Master agent that they have completed posting their reports.



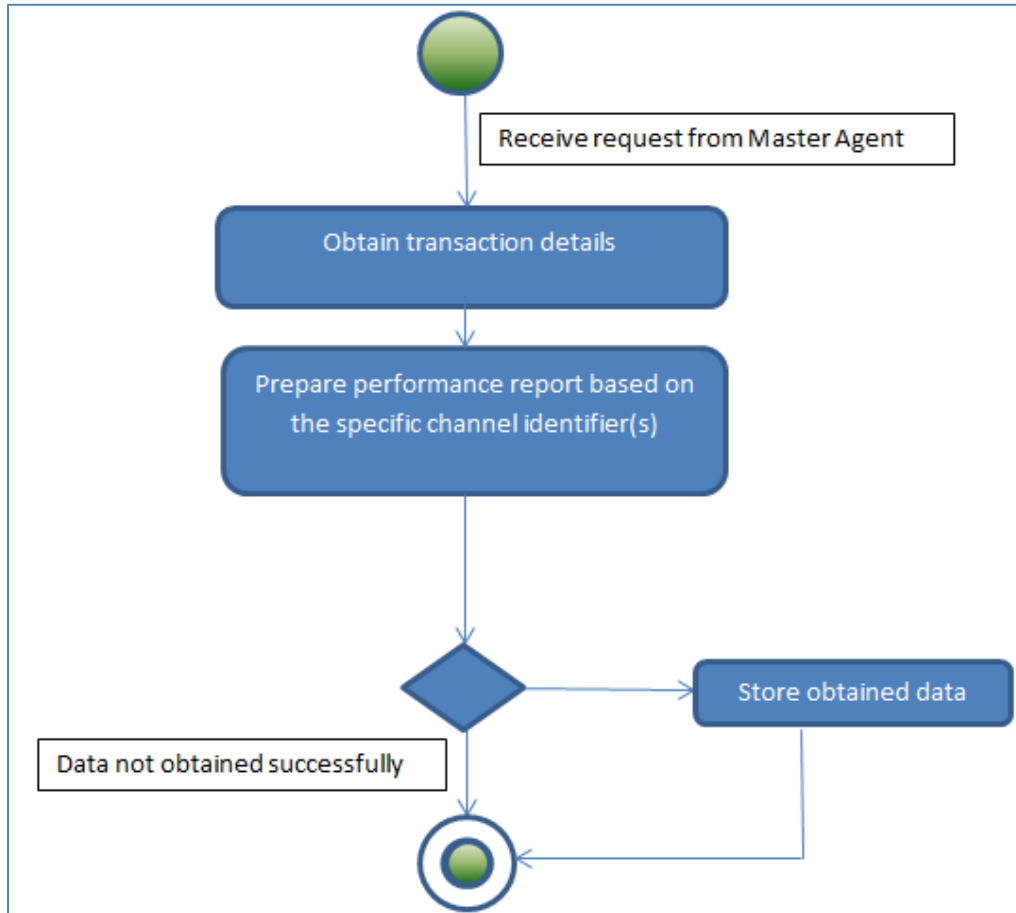
**Fig 3.5 - Capability diagram of Master Agent**

The figure 3.6 below is the capability diagram of a POS Reporting Agent. This is the same diagram for ATM Reporting Agent, and ideally, if implemented for other channels. If implemented for other channels, the correct channel identifier is the distinction among the channels. In this diagram, the POS Reporting has a few capabilities. This include detecting new POS channel transactions, executing queries for performance intelligence on POS channel, reporting performance results, and informing the Master Agent of its completion of tasks it has been requested to perform. The agent communicates with the Master Agent.



**Fig 3.6 - Capability diagram of POS Reporting Agent**

The figure 3.7 below is the plan diagram depicting how a POS Reporting Agent and ATM Reporting Agent plans to get data from the data sources. In this diagram, the agent has a few plans including waiting for a request from Master Agent, obtaining transaction details for the database server, preparing performance reports based on the obtained data, and storing the data.

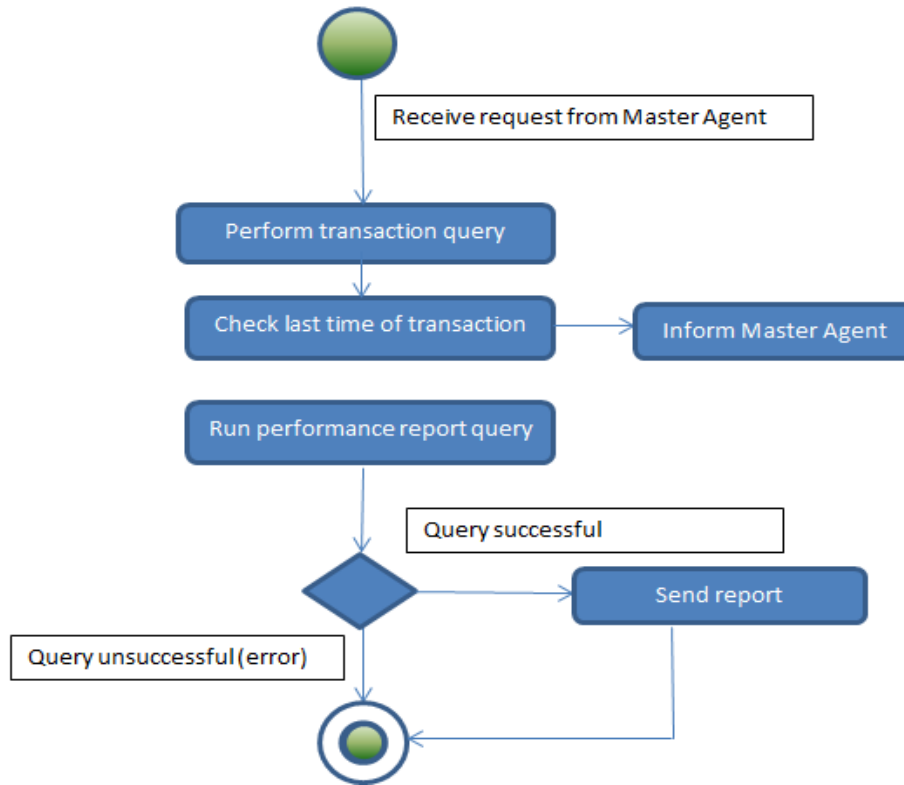


**Fig 3.7 - The get\_data\_from\_data\_sources plan diagram**

The figure 3.8 below is the plan diagram of a POS Reporting Agent on how to conduct its activities to achieve system goal. First, the agent listens for request from the Master Agent to report on the performance of the POS Channel. The agent’s plans are as indicated on the diagram/plan. This plan works for the ATM Reporting agent as well. To perform query transaction for POS channel, the plan requires among others, the following pieces of data:

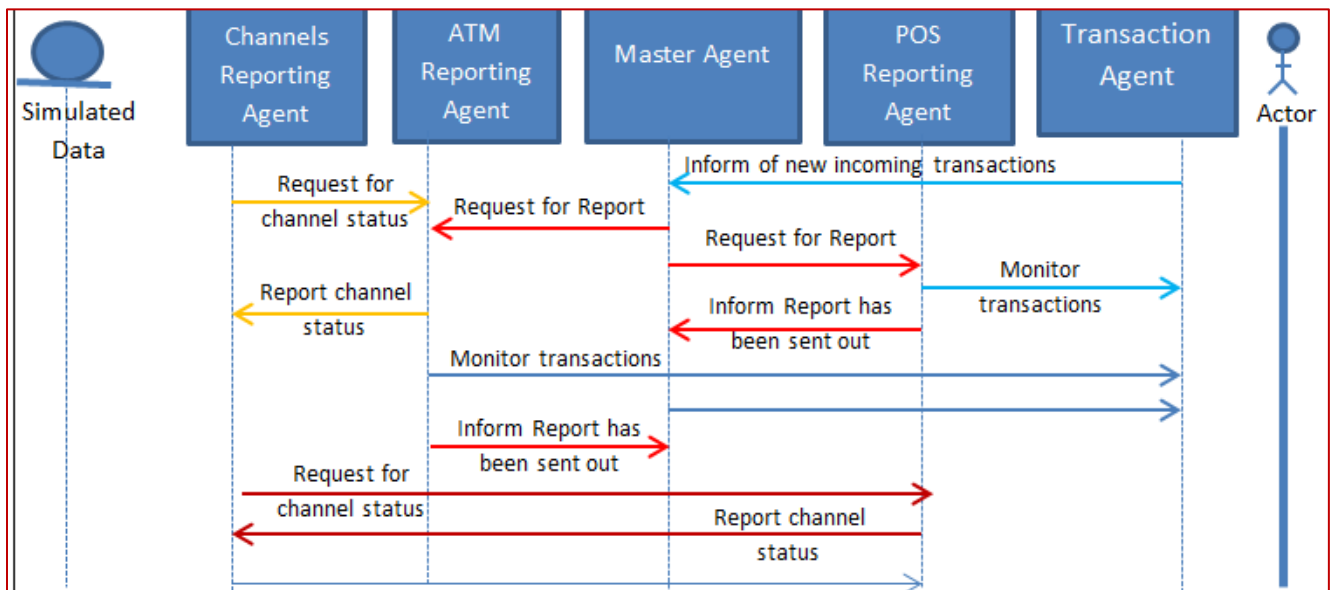
- (i) Transaction Reference Number
- (ii) Terminal ID
- (iii) Transaction Amount
- (iv) Transaction Type
- (v) Transaction Status





**Fig 3.8 - Plan diagram for POS Reporting Agent**

Figure 3.9 below shows the agents' interaction diagram for the E-channels performance-monitoring model, following the Foundation for Intelligent Physical Agents (FIPA) Agent Communication Language (ACL) that is recommended for agents' communication.



**Fig 3.9 - Agent Interaction Diagram**

## 3.4 Prototype Development

### 3.4.1 Development Tools and Algorithms

#### Development Tools

For purposes of developing the multi-agent based model for performance monitoring of banking e-channels, the following development tools were used:

1. **Java Agent Development Environment (JADE):** This was the framework or development environment used to implement the Multi-Agent System. This middleware is Java-based. It provides the development framework for developing MAS that comply with FIPA specifications.
2. **Java Development Kit (JDK) 7:** This tool offers the development environment for using Java language in designing and developing MAS agents and JADE library classes.
3. **XAMPP (Cross-platform, Apache, MySQL, PHP, and Pearl):** This tool allows one to develop web applications using Apache Server, MySQL Database Server, and PHP Scripting language. XAMPP is open source. The X in XAMPP means the tool can be used across different operating systems like Microsoft Windows, Linux, and Mac OS.

#### Algorithms Used

This solution implemented a few algorithms to realize the desired functionalities. These algorithms include the following (see code in Appendix C):

1. *Data simulation algorithm:* used for generating simulated data for this system prototype. The data that is input into this algorithm is time and the output is a transaction record
2. *Classification algorithm:* used for classifying transactions into various channels. The data that is input into the algorithm is a complete transaction and the output is transactions classified according to different electronic channels
3. *Sniffing algorithm:* used for detecting new transactions. The key data input for this algorithm is a time difference between the last two transaction records
4. *Reporting algorithm:* used to detect events and send reports via email and on the GUI. The data input for this algorithm is all records that meet a certain query criteria and the output is an advice showing the performance of a channel.

### 3.4.2 Database Implementation

Several database tables were created in MySQL Database Management System. This database has several tables including the following:

- 1) Atmtransaction (has atmtransaction\_id, atmtransaction\_name, terminal\_id, transaction\_amount, transactiontype\_id, transactionstatus\_id, and transaction\_time).
- 2) terminal (has terminal\_id, terminal\_name, and description fields)
- 3) transactionstatus (has transactionstatus\_id, transactionstatus\_name, and description)
- 4) transactiontype (has transactiontype\_id, transactiontype\_name, and description)
- 5) transhistory (has transhistory\_id, atmtransaction\_id, period, and transaction\_time)

The first table, transactions, holds all transactions from all supported electronic channels. It has the minimal number of fields or columns needed for prototyping. The second table, terminal, is used to identify the category of e-channel through terminal\_id or terminal\_name field. The transactionstatus table holds details appertaining to the various processsign status of each transaction, such as “Approved”, “Declined”, and “Failed”. The transactiontype table holds details about various types of transactions permitted. This includes “Withdrawal”, “Deposit”, and “Purchase” among others. Finally, the transhistory table holds details about the history of transactions. This is important for monitoring and reporting purposes.

The following key fields are the most essential in monitoring the performance of any channel. Through referential integrity, several tables are usually queried to obtain needed results.

- |                        |                         |
|------------------------|-------------------------|
| 1) Atmtransaction_id   | 5) Transactiontype_id   |
| 2) Atmtransaction_name | 6) Transactionstatus_id |
| 3) Terminal_id         | 7) Transaction_time     |
| 4) Transaction_amount  |                         |

### 3.4.3 Logic implementation of the prototype

The Multi-Agent System Prototype of E-channels performance monitoring solution was implemented using JADE framework. Simulation of transactions together with the functionalities of the agents was implemented using Java programming language. The web application, used for browser-based monitoring and reporting was implemented using PHP together with web

development scripting languages. HTML, CSS, and jQuery were additional tools for making the application work in a web environment, have a good GUI, and be real-time mode. The following screenshot of the main window of the prototype in run-mode.

**XYZ Bank: E-Channels Monitoring & Reporting Tool**

Home | ATM Reports | POS Reports | E-Channels Reports | **ATM and POS Channels.**

**3: 'No Failure Reported.'**

**Live Transactions from ATM and POS Channels**

Transaction Ref No.	TID	Amount	Txn Type	Txn Date & Time
POCT/00000193/2014	ATM 003	396000	Purchase	2014-07-29 14:50:36
POCT/00000192/2014	ATM 003	761000	Deposit	2014-07-29 14:50:31
POCT/00000191/2014	ATM 002	843000	Withdrawal	2014-07-29 14:50:22
POCT/00000190/2014	POS 003	174000	Purchase	2014-07-29 14:50:16
POCT/00000189/2014	ATM 002	801000	Withdrawal	2014-07-29 14:50:06
POCT/00000188/2014	POS 003	667000	Purchase	2014-07-29 14:50:01
POCT/00000187/2014	POS 003	856000	Withdrawal	2014-07-29 14:49:51
POCT/00000186/2014	ATM 001	684000	Deposit	2014-07-29 14:49:46
POCT/00000185/2014	POS 001	580000	Purchase	2014-07-29 14:49:31
POCT/00000184/2014	ATM 001	783000	Purchase	2014-07-29 14:49:21
POCT/00000183/2014	POS 001	659000	Purchase	2014-07-29 14:49:16
POCT/00000182/2014	POS 001	451000	Deposit	2014-07-29 14:49:06
POCT/00000181/2014	POS 003	203000	Purchase	2014-07-29 14:48:51
POCT/00000180/2014	POS 003	844000	Deposit	2014-07-29 14:48:46

**Today's Performance**

ATM: Successful Transactions = 12 || Failed Txns = 4 || Amount = KES 3,794,000.00

Transaction Ref#	Transaction Time	Period Elapsed (Min)
POCT/00000192/2014	2014-07-29 14:50:36	0.1
POCT/00000191/2014	2014-07-29 14:50:31	0.2
POCT/00000189/2014	2014-07-29 14:50:16	0.2
POCT/00000186/2014	2014-07-29 14:49:51	0.1

**Status Advice:**  
 [1] ATM Channel Performance: Below Normal.  
 [2] POS Channel Performance: Normal.

POS: Successful Transactions = 13 || Failed Txns = 2 || Amount = KES 5,228,000.00

Transaction Ref#	Transaction Time	Period Elapsed (Min)
POCT/00000190/2014	2014-07-29 14:50:22	0.1
POCT/00000188/2014	2014-07-29 14:50:06	0.1
POCT/00000187/2014	2014-07-29 14:50:01	0.2
POCT/00000185/2014	2014-07-29 14:49:46	0.3

Copyright © 2014: XYZ Bank: E-Channels Monitoring & Reporting Tool.

**Fig 3.10 - Prototype's Graphical User Interface**

### 3.5 Testing of the prototype

The testing of this MAS model encompassed testing at four different levels. These were:

1. Unit testing
2. Integration testing
3. System testing
4. User Acceptance Testing

### Testing Environment

The testing of the prototype developed in this research was conducted using the following hardware and software tools as a testing environment:

- A laptop with 4GB RAM
- Windows 7 x64 bits
- Java JDK version 1.8

- JADE Framework version 4.3.2
- XAMPP Version 1.7.1 (Apache Server 2.2.11 MySQL Server 5.1.33, PHP 5.2.9)

### 3.5.1 Unit Testing

This level of testing aimed at verifying the functionalities of different java classes developed.

These classes include the following:

1. Util class (provides common interface for all classes)
2. CommunicationAgent class (for establishing connection with data sources)
3. TransactionAgent class (for generating simulated transactions)
4. MasterAgent class (for coordinating reporting of performance)
5. PosReportingAgent class (for monitoring and reporting the performance of POS channel)
6. AtmReportingAgent class (for monitoring and reporting the performance of ATM channel)

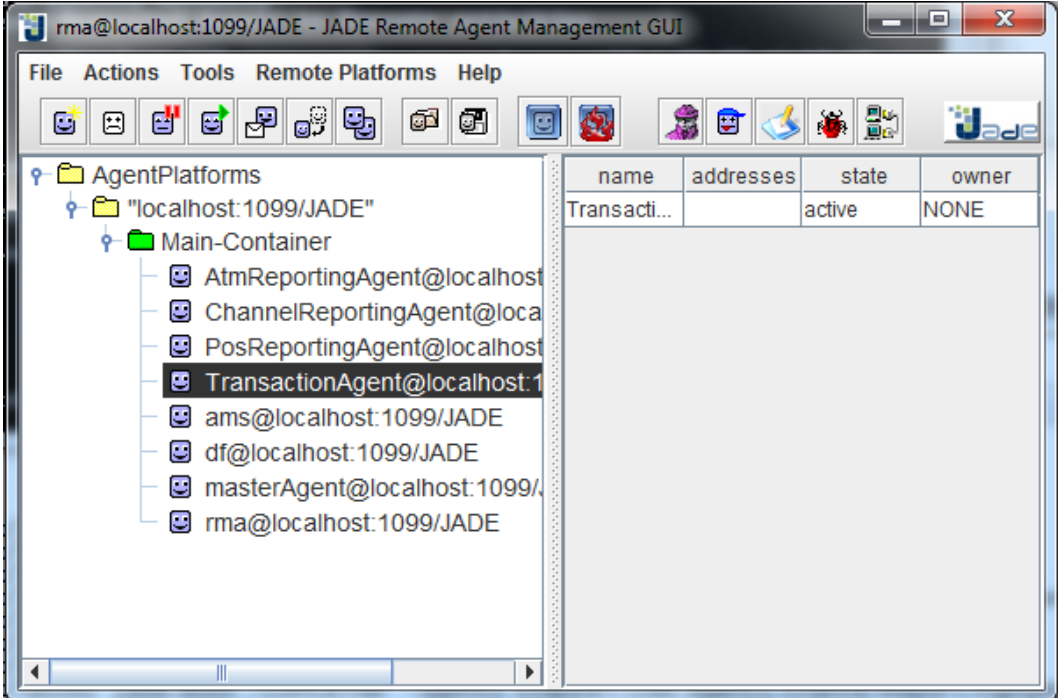
Both PosReporting and AtmReporting agents are Classifier agents because they must obtain transactions associated with their dedicated channels that they monitor and report.

**Table 3.7 - The tasks in Unit Testing:**

Task #	Task Description
UT-1	Run JADE and developed agents
UT-2	Connection to MySQL Database Server
UT-3	Generation of transactions (POS and ATM)
UT-4	Sniffing new transaction into the database server
UT-5	Sending / Displaying notifications of new transactions detected
UT-6	Compiling performance reports
UT-7	Sending performance reports
UT-8	Sending notifications of system failures

The following tables (3.8 to 3.10) show the test plans for Unit Testing.

**Table 3.8 - Starting JADE and system agents (UT-1)**

<b>Input</b>	<b>JADE and Solution Agents</b>								
Test Items	The TransactionAgent								
Method	Start JADE and its agents. JADE and the agents will start and remain running								
Valid	<p>All agents are registered with the JADE platform on the Main Container</p>  <p>The screenshot shows the JADE Remote Agent Management GUI. On the left, a tree view shows the hierarchy: AgentPlatforms &gt; "localhost:1099/JADE" &gt; Main-Container. Under Main-Container, several agents are listed, including TransactionAgent@localhost:1099/JADE, which is highlighted. On the right, a table displays the status of the selected agent:</p> <table border="1"> <thead> <tr> <th>name</th> <th>addresses</th> <th>state</th> <th>owner</th> </tr> </thead> <tbody> <tr> <td>Transacti...</td> <td></td> <td>active</td> <td>NONE</td> </tr> </tbody> </table>	name	addresses	state	owner	Transacti...		active	NONE
name	addresses	state	owner						
Transacti...		active	NONE						
Invalid	<ul style="list-style-type: none"> <li>JADE framework does not start or the RMA fails to display any outcomes</li> <li>The JVM (Java Virtual Machine) may render any related errors</li> </ul>								

**Table 3.9 - Generating new transactions into MySQL Database Server (UT-2)**

<b>Input</b>	<b>Data subsets</b>
Test Items	Data subsets
Method	<p>i) Log into your email and check email sent confirming status of the channels</p> <p>ii) Log into the GUI (web browser) and notice new transactions coming in automatically in the Live Transactions section of the window</p> <p>iii) Notice the MasterAgent on the Windows Command Line reporting new transactions from ATMs and POS channels</p>
Valid	Email sent successfully confirming connectivity to database

	<p><b>System Status Notification</b></p> <p>Channel Performance Monitoring System (multiagentmonitoringsystem@gmail.com)</p> <hr/> <p>System Status Notification</p> <hr/> <p>There was a network problem at 2014-07-26 15:31:32</p> <p>New transactions visible on the browser window</p> <table border="1" data-bbox="358 579 1250 816"> <thead> <tr> <th colspan="5">Live Transactions from ATM and POS Channels</th> </tr> <tr> <th>Transaction Ref No.</th> <th>TID</th> <th>Amount</th> <th>Txn Type</th> <th>Txn Date &amp; Time</th> </tr> </thead> <tbody> <tr> <td>POCT/00000146/2014</td> <td>ATM 001</td> <td>220000</td> <td>Deposit</td> <td>2014-07-27 13:52:44</td> </tr> <tr> <td>POCT/00000145/2014</td> <td>ATM 002</td> <td>62000</td> <td>Deposit</td> <td>2014-07-27 13:52:39</td> </tr> <tr> <td>POCT/00000144/2014</td> <td>POS 003</td> <td>369000</td> <td>Purchase</td> <td>2014-07-27 13:52:29</td> </tr> </tbody> </table> <p>The Master Agent windows (Command Line Window) displays new transactions</p> <pre data-bbox="358 890 1393 972"> Network Running Normally Transaction posted at 2014-07-27 13:52:39 The last POS transaction was made as at 2014-07-27 13:52:29 </pre>	Live Transactions from ATM and POS Channels					Transaction Ref No.	TID	Amount	Txn Type	Txn Date & Time	POCT/00000146/2014	ATM 001	220000	Deposit	2014-07-27 13:52:44	POCT/00000145/2014	ATM 002	62000	Deposit	2014-07-27 13:52:39	POCT/00000144/2014	POS 003	369000	Purchase	2014-07-27 13:52:29
Live Transactions from ATM and POS Channels																										
Transaction Ref No.	TID	Amount	Txn Type	Txn Date & Time																						
POCT/00000146/2014	ATM 001	220000	Deposit	2014-07-27 13:52:44																						
POCT/00000145/2014	ATM 002	62000	Deposit	2014-07-27 13:52:39																						
POCT/00000144/2014	POS 003	369000	Purchase	2014-07-27 13:52:29																						
Invalid	<ul style="list-style-type: none"> <li>i) Email not sent</li> <li>ii) Browser windows does not show new transactions</li> <li>iii) The MasterAgent windows reports an error</li> </ul>																									

**Table 3.10 - Sniffing new transactions, report generation, and reporting**

Input	Sniffing transactions, Monitoring channels, and Reporting performance
Test Items	MasterAgent, PosReportingAgent, AtmReportingAgent, ChannelsReportingAgent
Method	<ul style="list-style-type: none"> <li>i) Start JADE and its agents. JADE and the agents will start and remain running</li> <li>ii) Open the MasteAgent Command Window to view newly generated transactions</li> <li>iii) Open the browser window to view new transactions</li> <li>iv) On the browser window’s right section, check for performance reports</li> <li>v) Check email for notifications of event failures (if any)</li> </ul>
Valid	<ul style="list-style-type: none"> <li>i) JADE and its agents started. All agents running</li> <li>ii) MasterAgent Command Line Window running</li> <li>iii) Browser displays new transactions</li> <li>iv) Browser displays changing performance reports</li> </ul>

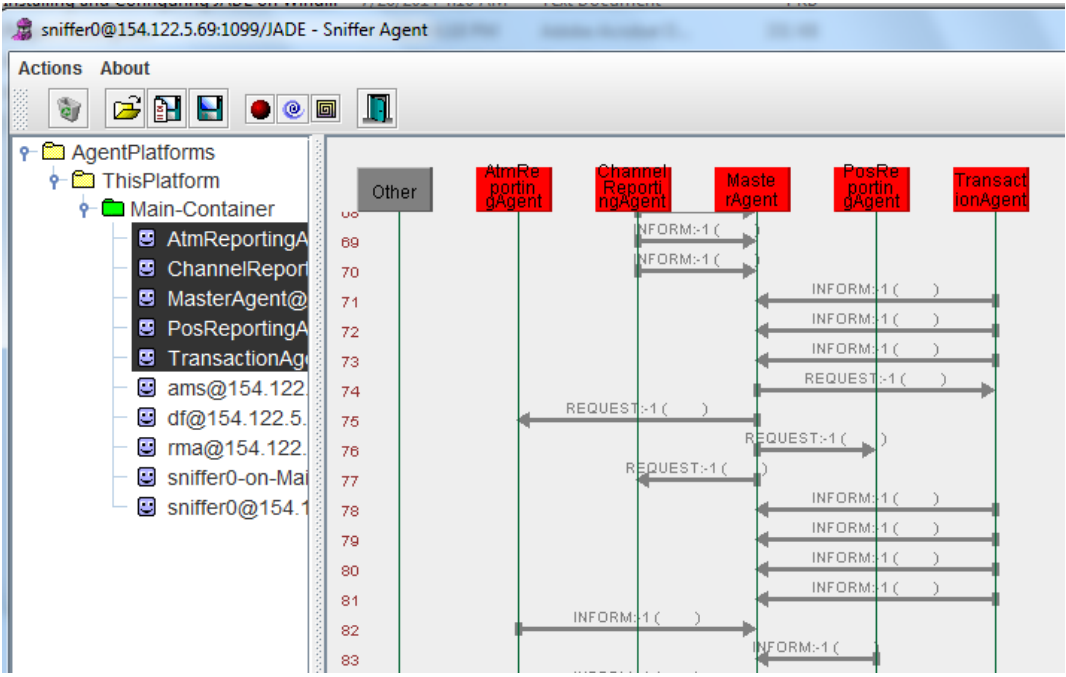
	<p style="text-align: center;"><b>Today's Performance</b></p> <p>ATM: Successful Transactions = 30    Failed Txns = 6    Amount = KES 1,921,000.00           &gt;&gt; Performance is below Normal Target!</p> <table border="1" data-bbox="370 346 1312 630"> <thead> <tr> <th>Transaction Ref#</th> <th>Transaction Time</th> <th>Period Elapsed (Min)</th> </tr> </thead> <tbody> <tr><td>POCT/00000064/2014</td><td>2014-07-28 16:59:42</td><td>0.3</td></tr> <tr><td>POCT/00000063/2014</td><td>2014-07-28 16:59:22</td><td>0.2</td></tr> <tr><td>POCT/00000062/2014</td><td>2014-07-28 16:59:12</td><td>0.3</td></tr> <tr><td>POCT/00000061/2014</td><td>2014-07-28 16:58:52</td><td>0.5</td></tr> <tr><td>POCT/00000060/2014</td><td>2014-07-28 16:58:22</td><td>0.2</td></tr> <tr><td>POCT/00000059/2014</td><td>2014-07-28 16:58:12</td><td>0.3</td></tr> </tbody> </table> <p><b>Status Advice:</b>  <b>[1] ATM Channel Performance: Below Normal.</b>  <b>[2] POS Channel Performance: Normal.</b></p> <p>POS: Successful Transactions = 35    Failed Txns = 8    Amount = KES 3,190,000.00           &gt;&gt; Performance is below Normal Target!</p> <table border="1" data-bbox="370 898 1312 1182"> <thead> <tr> <th>Transaction Ref#</th> <th>Transaction Time</th> <th>Period Elapsed (Min)</th> </tr> </thead> <tbody> <tr><td>POCT/00000055/2014</td><td>2014-07-28 16:57:12</td><td>0.3</td></tr> <tr><td>POCT/00000053/2014</td><td>2014-07-28 16:56:42</td><td>0.5</td></tr> <tr><td>POCT/00000052/2014</td><td>2014-07-28 16:56:12</td><td>0.3</td></tr> <tr><td>POCT/00000051/2014</td><td>2014-07-28 16:55:52</td><td>0.2</td></tr> <tr><td>POCT/00000050/2014</td><td>2014-07-28 16:55:42</td><td>0.3</td></tr> <tr><td>POCT/00000048/2014</td><td>2014-07-28 16:54:52</td><td>0.2</td></tr> </tbody> </table> <p>Email notification sent where failure was detected</p>	Transaction Ref#	Transaction Time	Period Elapsed (Min)	POCT/00000064/2014	2014-07-28 16:59:42	0.3	POCT/00000063/2014	2014-07-28 16:59:22	0.2	POCT/00000062/2014	2014-07-28 16:59:12	0.3	POCT/00000061/2014	2014-07-28 16:58:52	0.5	POCT/00000060/2014	2014-07-28 16:58:22	0.2	POCT/00000059/2014	2014-07-28 16:58:12	0.3	Transaction Ref#	Transaction Time	Period Elapsed (Min)	POCT/00000055/2014	2014-07-28 16:57:12	0.3	POCT/00000053/2014	2014-07-28 16:56:42	0.5	POCT/00000052/2014	2014-07-28 16:56:12	0.3	POCT/00000051/2014	2014-07-28 16:55:52	0.2	POCT/00000050/2014	2014-07-28 16:55:42	0.3	POCT/00000048/2014	2014-07-28 16:54:52	0.2
Transaction Ref#	Transaction Time	Period Elapsed (Min)																																									
POCT/00000064/2014	2014-07-28 16:59:42	0.3																																									
POCT/00000063/2014	2014-07-28 16:59:22	0.2																																									
POCT/00000062/2014	2014-07-28 16:59:12	0.3																																									
POCT/00000061/2014	2014-07-28 16:58:52	0.5																																									
POCT/00000060/2014	2014-07-28 16:58:22	0.2																																									
POCT/00000059/2014	2014-07-28 16:58:12	0.3																																									
Transaction Ref#	Transaction Time	Period Elapsed (Min)																																									
POCT/00000055/2014	2014-07-28 16:57:12	0.3																																									
POCT/00000053/2014	2014-07-28 16:56:42	0.5																																									
POCT/00000052/2014	2014-07-28 16:56:12	0.3																																									
POCT/00000051/2014	2014-07-28 16:55:52	0.2																																									
POCT/00000050/2014	2014-07-28 16:55:42	0.3																																									
POCT/00000048/2014	2014-07-28 16:54:52	0.2																																									
Invalid	<ul style="list-style-type: none"> <li>▪ JADE and agents failed to start</li> <li>▪ MasterAgent Command Line Window displayed error(s)</li> <li>▪ Browser windows does not show new transactions</li> <li>▪ Email not received yet there was a system failures</li> </ul>																																										

**3.5.2 Integration Testing**

At this stage of testing, the interest shifted to confirming that the agents that make up the prototype were active on JADE’s Remote Agent Monitoring GUI interacted with one another. To achieve this, JADE is packaged with a Sniffer agent, which enables one to see agents interactions on the GUI. The test plan used for integration testing is shown in Table 11 below.



**Table 3.11 - Agents Interactions Testing**

Input	JADE environment, Prototype's Agents, and Simulated Transactions
Test Items	MasterAgent, PosReportingAgent, AtmReportingAgent, ChannelsReportingAgent
Method	Start JADE and its agents. JADE and the agents will start and remain running Use the RMA's GUI in JADE to start the Sniffer Agent Select all agents and use Sniffer Agent to view agents interactions
Valid	JADE and its agents started. All agents running Agents interacting via REQUEST and INFORM messages 
Invalid	JADE and agents failed to start No INFORM and REQUEST messages on the Sniffer Agent's GUI

### 3.5.3 System Testing

In this phase, the objective was to verify that the whole system had the functionalities required for monitoring and reporting the performance of e-channels of a bank. One key element in this testing was the veracious scrutiny of reports that the system generated or displayed when running. Appendix E includes sample screenshots of these reports. These reports include:

1. Generated Transactions Report – this shows simulated transactions (for all channels)
2. ATM Transactions Report – this shows transactions for ATM channel only
3. POS Transactions Report – this shows transactions for POS channel only

4. Suspicious Events Report – this shows failed transactions prompting investigations
5. Summary Report – this shows summary performance of both POS and ATM channels
6. Transaction Classification Report – this shows the intelligent classification of transactions to reveal different performance parameters for reporting. Classification is based on the following parameters:
  - a) Number of successful transactions per channel per day
  - b) Number of failed transactions per channel per day
  - c) Total amount from successful transactions per channel per day

### 3.5.4 User Acceptance Testing

This formed the final stage of testing the developed MAS prototype. At this stage, employees working in banks that support the two selected channels were given access to use the solution and provide some comments about the solution. The primary objective of this testing was to confirm if all agents functioned as required thereby meeting user expectations of a solution that can monitor and report the performance of different electronic channels of a bank.

**Table 3.12 - User Acceptance Testing (UAT) Tasks**

<b>Task #</b>	<b>Task Description</b>
UAT-1	Generation of simulated transactions
UAT-2	Detection and display of new transactions
UAT-3	Classification of channel performances
UAT-4	Notifications of performance failure events
UAT-5	Viewing summary reports

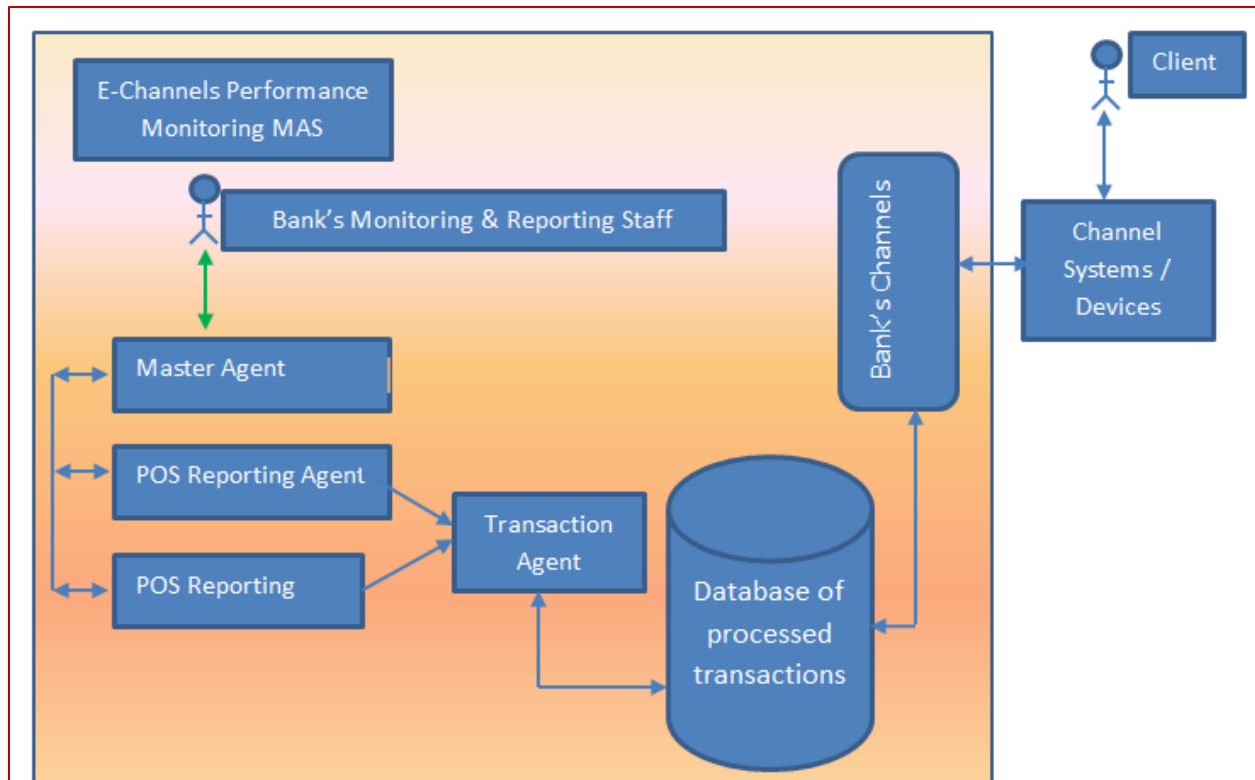
In order to ensure proper testing of these tasks, the users were given test cases (see Appendix C – UAT Questionnaire) in the form of a questionnaire and allowed to comment on each task. See Appendix X for this questionnaire. .

**Table 3.13 - UAT Test Cases**

<b>Task #</b>	<b>Test Case Description</b>	<b>Expected Results</b>
UAT-1	<p>Title: Generation of simulated transactions</p> <p>Requirements: User has access to the system</p> <p>Steps:</p> <ul style="list-style-type: none"> <li>(i) Log into the web application (system) by means of supplied credentials</li> <li>(ii) Check the details of the last transaction</li> </ul>	<ul style="list-style-type: none"> <li>▪ Login is successful</li> <li>▪ New transactions appear (new amounts and current date and time)</li> </ul>
UAT-2	<p>Title: Detection and display of new transactions</p> <p>Requirements: continuation of UAT-1 above</p> <p>Steps:</p> <ul style="list-style-type: none"> <li>(i) View the browser window on the “Live ATM and POS Transactions section”</li> <li>(ii) Open the MasterAgent’s Window</li> </ul>	<ul style="list-style-type: none"> <li>▪ New transactions keep appearing on the browser window</li> <li>▪ Command Line Window displays list of generated transactions</li> </ul>
UAT-3	<p>Title: Classification of channels performance</p> <p>Requirements: continue from UAT-2 above</p> <p>Steps:</p> <ul style="list-style-type: none"> <li>(i) On the right hand side of the window, look at the 2 tables and on top of each</li> <li>(ii) Take note of the details and check them again after 30 seconds of system uptime</li> </ul>	<ul style="list-style-type: none"> <li>▪ ATM and POS performance tables are displayed</li> <li>▪ Summary details are displayed</li> <li>▪ The figures change with time</li> </ul>
UAT-4	<p>Title: Notifications of performance failure events</p> <p>Requirements: continue from UAT-3 above</p> <p>Steps:</p> <ul style="list-style-type: none"> <li>(i) Terminate agents by closing JADE’s GUI</li> </ul>	<ul style="list-style-type: none"> <li>▪ JADE closes down</li> <li>▪ Email is sent to notify of system malfunctioning or failure</li> </ul>
UAT-5	<p>Title: Viewing summary reports</p> <p>Requirements: continued from UAT-4 above</p> <p>Steps:</p> <ul style="list-style-type: none"> <li>(ii) Click on summary reports link</li> <li>(iii) Click on POS Report, ATM Report or Channels Report</li> </ul>	<ul style="list-style-type: none"> <li>▪ The Summary Reports link opens a new window with other links</li> <li>▪ Each of these links displays summary report</li> </ul>

### 3.5.5 Prototype Deployment Model

If it were possible, this prototype would have been deployed in a banking institution that would offer live transactions from different production e-channels. To achieve this, however, the agents would be installed on the same or different servers from where they will monitor and report the performance of e-channels. However, for study, the prototype was deployed on a single laptop.



**Fig 3.11 - Deployment Model of the E-Channels Performance Monitoring Prototype**

### 3.6 Chapter summary

This chapter covered the key components of research methodology. It focused on how this research designed, data collected and analyzed. The responses from questionnaires provided useful information that helped in designing both the database and model. The chapter then dived into designing the database followed with model design basing on Tropos methodology for designing multi-agent systems. The prototype was then developed using JADE, Java, XAMPP and web development tools including HTML, CSS and jQuery. The prototype was then passed through the testing phases, which included unit testing, system testing, integration testing and user acceptance testing. The next chapter, Evaluation, will focus on evaluating the model developed in this chapter based on the prototype's functionality, realism, and user acceptance. The evaluation will link the finding with theory and related works in published literature.

# CHAPTER 4: EVALUATION

## 4.0 Introduction

This chapter focuses on the evaluation of the multi-agent based banking e-channels performance model that was designed, developed, and tested in the previous chapter. This evaluation will involve four main activities based on the functionality and realism of the prototype. After this section, an analysis of the findings from the evaluation will follow. These evaluation activities, based on functionality of the prototype include Simulation of transactions, Monitoring of channel activity and generating performance reports, Agents Interaction and User feedback. This part evaluates the results obtained from integration and unit testing conducted on the developed prototype. The purpose of this was to establish if the prototype met the functional requirements of the model.

## 4.1 Simulation of transactions

### Objective

To establish if the prototype was able to simulate transactions for POS and ATM channels.

### Procedure

1. Start JADE and model agents
2. Establish connection to the database server
3. Generate simulated transactions by means of agents

### Data input and desired outcome

- Data Input: the algorithm for generating simulated transactions used time as data input
- Desired outcome: complete transaction record

### Results

**Table 4.1 – Simulation of transactions**

Task# (Procedure #)	Test Status: Tests	Test Results/Observations
TS1	Valid: Step 1: Starting JADE and developed Agents	JADE Framework and defined Agents started running and were registered on the JADE platform on the host machine
TS2	Valid: Step 2: Establishing a connection	Connection to MySQL Server established and an email confirming this

	to MySQL Server and generation of dummy transactions	connectivity was sent. Agents posted new transactions to the database, which were fetched by queries for reporting.
TS3	Valid: Step 3: Populating the transaction table with new records	Agents generated new transactions periodically. This was seen evidenced on the browser window

```

Network Running Normally
Transaction posted at 2014-07-27 13:52:39
The last POS transaction was made as at 2014-07-27 13:52:29

```

**Fig 4.1 - Sample notification of simulated transactions and network status**

### **Analysis of the results**

All agents of the prototype executed successfully and were registered on the JADE platform. This is because JADE is the environment from which agents execute their assigned tasks through interaction with each other and access to system resources. The model interacted with the database server, confirming the success of linking JADE agents with a database. The dynamic population of new transactions on the browser window confirms the agents posted new transactions to the database and browser window. This was possible because the agents were designed to possess capabilities and action plans to follow in interacting with the data source. The fact that agents were able to run successfully, register on the JADE framework, and interact with the database leads this research to the conclusion that the model was designed and implemented according to FIPA Specifications (Bellifemine, Caire & Greenwood, 2007).

## **4.2 Monitoring channel activity and reporting channel performance**

### **Objective**

To establish if the prototype was able to detect channel activity based on new transactions and be able to report performance of channels based on both existing and new transactions

### **Procedure**

1. Start JADE and model agents
2. Establish connection to the database server
3. Generate simulated transactions by means of agents

4. Monitor performance based on existing and new transactions
5. Send performance incident notifications
6. Generate performance reports

**Data input and desired outcome**

- Data Input: database records
- Desired outcome: performance report/advice

**Results**

**Table 4.2 – Monitoring channel activity**

Task# (Procedure #)	Test Status: Tests	Test Results / Observations
TS4 <i>*** Continue from Task# 3 in table 4.1</i>	Valid: above Detecting normal and/or below normal performance based on existing and new transactions	The performance monitoring solution confirmed receiving new transactions, querying the transactions, deciding on performance status and reporting the performance status per channel and for all channels. Email notifications were also sent



**Fig 4.2 - Sample Performance Report for All Channels with Summary**

**Analysis of the results**

The observations showed that agents were able to monitor incoming transactions per channel. Each agent (POS Reporting Agent and ATM Reporting Agent) was able to detect transactions coming through its channel. The prototype made its events notification reporting based on

current transactions giving it a real-time nature suitable for this kind of complex monitoring problem. This confirms the prototype has met its primary goal of monitoring e-channels activity. This research concludes that this prototype solved key problems that current monitoring techniques face. This was through automatically monitoring channel activity and reporting system failures via the prototype’s window and Email. No human intervention was needed.

### 4.3 Agents Interaction

#### Objective

To establish if the prototype had interacting intelligent agents

#### Procedure

1. Start JADE and all model agents
2. Establish connection to the database server
3. Generate simulated transactions by means of agents
4. Open the Sniffer Agent on JADE RMA’s GUI
5. Select all agents and choose “Sniff All”

#### Data input and desired outcome

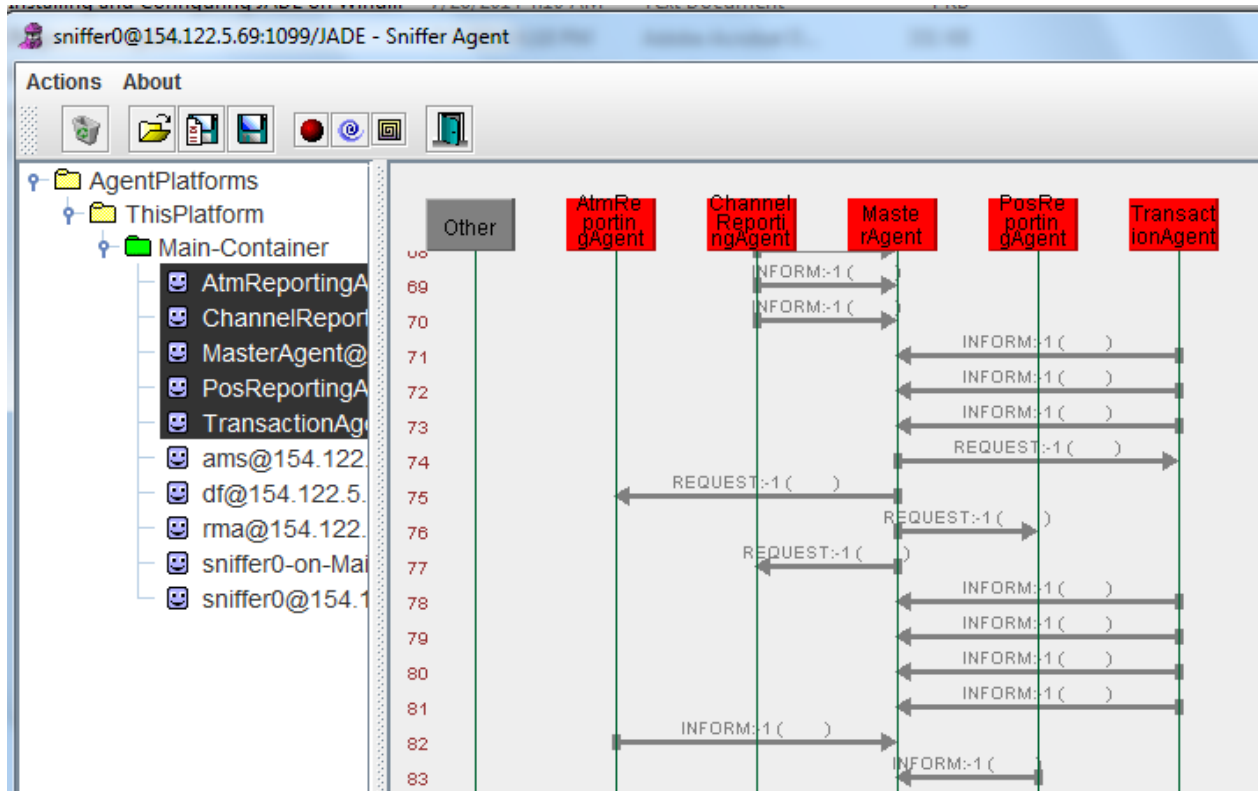
- Data Input: executing JADE agents
- Desired outcome: interactions between various agents on the Sniffer Agent window

#### Results

**Table 4.3 – Agents Interaction**

Task# (Procedure #)	Test Status: Tests	Test Results
Integration Testing <i>*** Continue from Task# 4 in table 4.2</i>	Valid: Agents Interaction	The Sniffer Agent in JADE shows that the system’s agents were interacting





**Fig 4.3 - Agents Interaction**

### Analysis of the results

The finding that agents were seen interacting with each other on the Sniffer Agent window provides evidence that this is a MAS solution consists of multiple agents working together to achieve the desired system goal in accordance with the definition of MAS. The automatic reporting of new performance reports and status confirms that agents are always interacting. The interaction and co-operation among agents with JADE demonstrate that the system’s agents comply with FIPA specifications on communication. The standard embraces principles such as the standardizations of the language and infrastructure for creating interoperable MAS without focusing on standardizing the core workings of agents (Bellifemine, Caire & Greenwood, 2007).

This research therefore concludes that the model developed in chapter 3 is indeed a multi-agent system built according to FIPA specifications and exhibiting true communication among agents.

#### 4.4 Realism of the prototype

##### Objective

To establish if the prototype exhibited features similar to a real MAS model or system

##### Procedure

1. Start JADE and all model agents
2. Establish connection to the database server
3. Generate simulated transactions by means of agents
4. Monitor performance through the browser window and performance reports

##### Data input and desired outcome

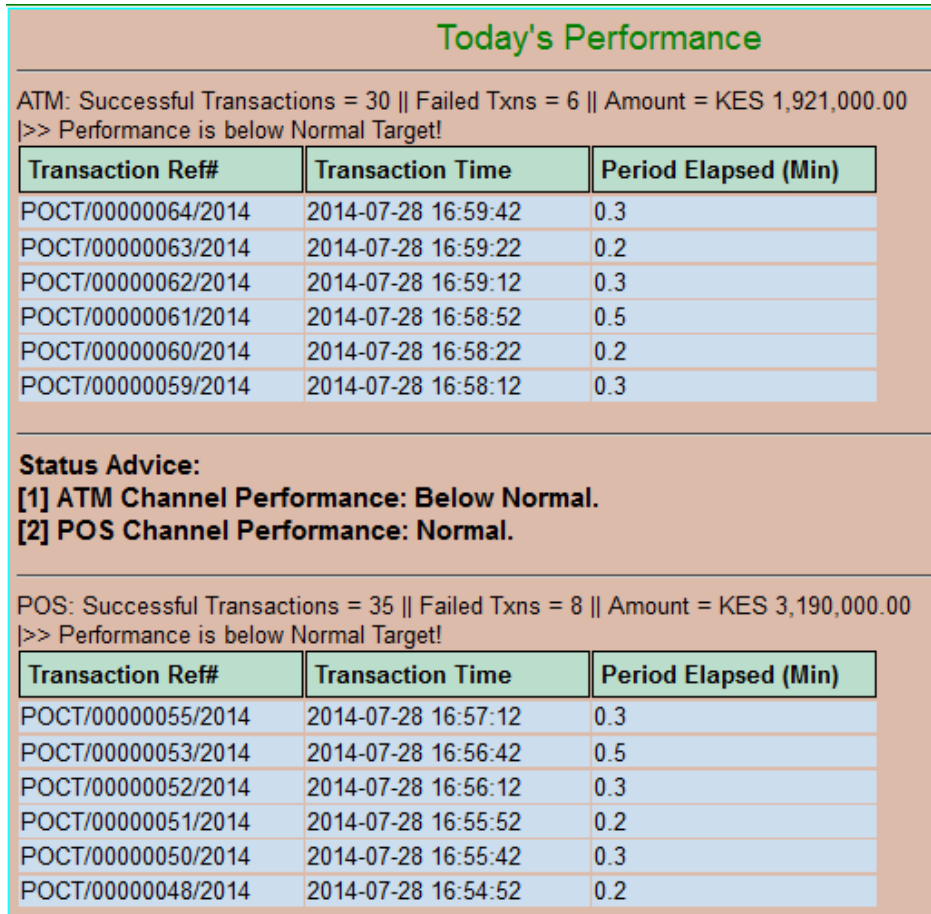
- Data Input: executing JADE agents
- Desired outcome: interactions between various agents on the Sniffer Agent window

##### Results

A comparison between the real world and the prototype based on its output reports was conducted. The Table 4.4 below provides this kind of analysis.

**Table 4.4 - Realism of prototype**

Name of Report	Description of the results of the report
Generated transactions	Displays generated transactions for different e-channels
E-channels Transactions	The report shows performance for ATM & POS channels
ATM Transactions	The report shows performance for ATM channel
POS Transactions	The report shows performance for POS channel
Performance Classification	Shows performance classification based on current transactions



**Fig 4.4 - Sample performance report for POS, ATMs, and E-Channels**

**Analysis of the results**

Section 3.2.3 gives an analysis of the responses obtained from questionnaires that were completed by employees from local banks, which led to the design of transaction data structure. The responses provided information that confirms agreement with the standardization of transaction data. The E-Channels report showed that ATM Channel was performing below expected normal performance while POS channel was performing within normal ranges. This is because in a real world, the performance of an e-channel is independent of other e-channels. The testing of ATM and POS channels performance was in accordance with Section 3.2.3, showing that banks are greatly interested in monitoring these two e-channels. Finally, Section 3.2.3 of the report confirms that to monitor the performance of any banking e-channels successfully, transaction records have to be considered and this is mostly achieved through executing database queries. This is why database queries were used in this prototype.

This research therefore concludes that the prototype significantly helped confirm that a multi-agent system approach is a better approach to solving the current challenge of monitoring the performance of banking channels through manual methods involving bank employees running database queries or monitoring application performance among other techniques that are not intelligent enough to solve this problem adequately.

## **4.5 User Feedback**

### **Objective**

To establish if the prototype met the expectations of intended users in the context of the quality of simulated data, quality of performance reports produced, and how transactions were classified.

### **Participants**

Four individuals participated in testing this prototype. All the testers came from banks, charged with the responsibility of performance monitoring and/or administrating electronic delivery channels of their respective banks. Two of them were responsible for managing e-channels, including the performance of such channels while the other two employees were in charge of systems administration for e-channels.

### **Procedure**

1. Start JADE and all model agents
2. Establish connection to the database server
3. Generate simulated transactions by means of agents
4. Generate performance reports
5. Notice event-driven system notifications on the model' GUI and Command Window

### **Data input and desired outcome**

- Data Input: executing JADE agents
- Desired outcome: user satisfaction with the model in solving the research problem

### **Results**

The Table 4.5 below presents observations and comments from the four users who tested and evaluated the prototype. Their comments were captured using a questionnaire found in Appendix C – UAT Questionnaire. This questionnaire captured four key user acceptance tests (UATs) that were aimed at establishing if the prototype offered a realistic solution for monitoring the performance of multiple electronic channels of a bank. The success of this project and the subsequent endorsement of this prototype are based on these users' comments.

**Table 4.5 - User Comments discussion**

UAT	Users Comments
UAT-1: Generation of transactions	<ul style="list-style-type: none"> <li>▪ Three users noted that transaction records lacked some details like card numbers and account numbers. However, the users indicated that such information was not critical in monitoring the performance of e-channels</li> <li>▪ One user noted that the types of transactions captured in the prototype were not adequate. However, for purposes of prototyping a performance model, the user noted that this had no impact on the model’s integrity.</li> <li>▪ Overall, the users were in agreement with the choice of fields used to represent transaction data in the context of prototyping</li> </ul>
UAT-2: Channel Performance Classification	<ul style="list-style-type: none"> <li>▪ All users noted that the prototype appropriately classified channel performance based on transactions with respect to each e-channel</li> </ul>
UAT-3: Performance reporting	<ul style="list-style-type: none"> <li>▪ One user indicated that it would be helpful to share the performance reports via Social Media and SMS especially for below performance reports that require immediate action. The prototype only implemented the reporting on web application window and Email, which all users were happy with in respect of how the prototype implemented them.</li> </ul>

**Analysis of the findings**

The fact that all the four users who tested the prototype were employees of retail banks that offer services to their customer via multiple channels, coupled with the fact that they worked in system administrative or performance monitoring roles, meant that their feedback and contribution to the testing of the prototype was very useful. Their comments indicate that the prototype actually works in a manner representing a real world system and its role in performance monitoring is so useful and would fit a real world situation. Therefore, the four users endorsed this prototype solution as a real solution to the real world challenge of monitoring the performance of different electronic channels of a bank.

#### **4.6 Chapter summary**

This chapter began with a highlight on the key evaluation activities, which included simulation of transactions, monitoring channel activity and generation of performance reports, prototype realism and user feedback. The key functionalities of the model were evaluated based on the observations and user feedback from testing it. The analysis linked the findings with similar published works --relating the success of this model to similar success stories in monitoring the performance of electronic systems as recorded in section 2.4.3. This leads us to the next chapter, Conclusions and Recommendations, which will focus on what this research concluded from the study based on literature review and the evaluation of the prototype before culminating in recommendations and future works.

## **CHAPTER 5: CONCLUSIONS & RECOMMENDATIONS**

### **6.0 Introduction**

This chapter presents the conclusions and recommendations that can be drawn from this research. It begins with a recap of the research problem, research objectives, methods used, what was designed, and how the designed prototype was tested. The next section is the discussions based on the finding from literature review and evaluation of the prototype. This is then followed by recommendations of this research before culminating in future works.

### **6.1 Recap of this research**

This research set out to develop a multi-agent based prototype for monitoring the performance of banking e-channels. This was motivated by the finding that current techniques face several challenges including overreliance on humans to execute performance monitoring and reporting tasks. However, it is the lack of an intelligent model for monitoring the performance of e-channels was even a greater motivation leading to the carrying out of this research. The main objectives of the research were to identify the current techniques used in monitoring systems performance. This research collected data from bank employees working in channel management or administration roles to provide relevant data that aided the design and development of the multi-agent model for monitoring the performance of banking e-channels. The model was developed following Tropos MAS development methodology and tested in four phases: unit testing, system testing, integration testing, and user acceptance testing. The evaluation of the model showed that the model was capable of monitoring and reporting the performance of e-channels of banks while overcoming the hurdles current techniques encounter.

### **6.2 Discussions**

This section discusses the findings from this study based on data collected from banks and the evaluation of the prototype. It covers the following four sub-sections: current methods banks use to monitor e-channels performance, the conceptual model design, development of the ECPM prototype, evaluation of the prototype, and the overall view based on these findings.

### **Current methods banks use to monitor and report e-channels performance**

The three banks that were sampled in this study confirmed using database queries for their monitoring and reporting of e-channels' performance; 75% of them also use application monitoring; 25% of use graphical tools; and none (0%) uses MAS technology (see section 3.2.3). The findings fulfill the first objective of this study, which aimed at identifying current techniques banks use to monitor the performance of their electronic channels. The reason most banks use these three techniques is perhaps due to their availability and simplicity.

### **Design of the conceptual design of the ECPM model**

A conceptual model of the ECPM model was developed (see section 2.8.2) based on the context model within which e-channels function. This model was used to build the architectural ECPM model (see section 3.4.3) as was informed by data collected from bank employees working in areas related to managing e- channels. This confirms this study met its second study objective.

### **Development of the ECPM Prototype**

The ECPM model was converted into a prototype by implementing two architectural components: the database and intelligent agents (see section 3.4). All agents were able to communicate with each other and access the database to perform their designated tasks. The model was thus developed in compliance with FIPA specifications. Positive test results were obtained for all functional tests conducted on the prototype. These tests included the successful generation of transactions, channel activity detection and performance reporting, agents' interaction, and the realism of the prototype and user feedback. More specifically, the agents ran successfully, registered on the JADE framework, interacted with the database and amongst themselves, generated useful performance reports, and sent event notifications (see sections 4.1 to 4.5), which leads to the conclusion that the prototype was designed as an integrated, intelligent, and autonomous MAS solution. Thus, this research met its third objective.

### **Evaluation of the ECPM Prototype**

To end with, the analyses obtained from data collected in questionnaires (see section 3.2.3) and the evaluation of the prototype agrees with the published works that this research reviewed in sections 2.4.3 and 2.6). For instance, the reviewed literature showed that MAS technology is



superior to existing technologies for problems that are complex and dynamic such as monitoring and reporting the performance of e-channels in banks. The evaluators of the prototype found it more powerful than existing technologies. The evaluators noted that it addressed monitoring and reporting challenges faced by current methods used by banks (see section 4.5). The evaluation on the prototype's realism (see section 4.4) revealed that transaction records matched with details provided in questionnaires, enforcing standardization of data structures for bank transactions. The performance parameters used in the prototype were derived from responses from bank employees. Thus, the prototype provided a working intelligent, integrated MAS based solution as stated in the problem statement, and confirming the research met its last objective.

Therefore, this study addressed the research problem by meeting its four objectives as identified in section 1.3. This was achieved through literature study to determine current methods banks use to monitor the performance of their e-channels, data collection to determine the data structure of bank transactions and performance parameters, the design of the ECPM model based on the conceptual model built the development of the prototype, and evaluation of the prototype. Through this, all the four study objectives were met and the resulting integrated intelligent ECPM model was confirmed as meeting the expectations of the evaluators.

### **6.3 Conclusions**

#### **i) There is a lack of an intelligent integrated e-channels performance-monitoring model**

The reviewed published literature revealed a lack of an integrated intelligent model for monitoring the performance of banking e-channels; a gap that motivated doing this research. The prototype developed featured active agents within JADE framework. This prototype met the requirements in section 3.3, which included generating/simulating transactions and posting them to the database, continuously monitoring and reporting the performance of all channels on the browser window of the web application, and sending email notifications of unexpected events such as network failures. This prototype achieved this through agents interaction. This was possible because the model was designed following FIPA specifications on agents' interactions.

## **ii) Collected data was useful in designing the intelligent ECPM prototype**

To develop this MAS prototype using the ECPM model in order to meet research objectives three and four, data was collected from bank employees who serve in roles related to monitoring the performance of e-channels. This data was important in the design of the database that became the data source, testing, and evaluation of the ECPM prototype. Two questionnaires were used: (1) data collection regarding the design of the database and model, and (2), user feedback collection for evaluators of the prototype. It was important to base database design on data provided by bank employees in channels departments since such data could not be availed directly for research due to legal constraints. The user feedback shows that the prototype met the requirements of an intelligent system capable of monitoring and reporting the performance of banking electronic channels. The evaluation of the prototype's realism revealed its close resemblance to existing tools but was intelligent and autonomous unlike current techniques.

## **iii) The ECPM prototype was autonomous, intelligent, and integrated**

The model developed had the capability of periodically querying the transactions database to produce an intelligent report showing channels performance. Thus, with this model, banks do not need employees whose primary role is to keep querying the database for anomalies and sending the reports since the MAS model is intelligent and autonomous enough to fulfill these tasks. The prototype was tested and the results showed that the system is accurate, reliable, effective, and efficient in solving the problem for multiple electronic channels of a bank. This agrees with similar studies documented in section 2.4.3 that highlights a few success stories in respect of using multi-agent systems in solving the problem of monitoring the performance of electronic systems. Theoretically, this can be explained from the angle that agents operate autonomously, are intelligent, and can cooperate to achieve a common goal in a shorter time than could be possible with contemporary software technologies as was explained in sections 2.5 and 2.6.

## **iv) ECPM prototype met the essential functional requirements**

The tests conducted on the prototype involved testing for prototype functionality and realism while taking into account user feedback from four bank employees who evaluated the prototype. The tests focused on detecting channel activity, which was achieved by monitoring new transactions into the transaction database server, performance reporting, interaction of agents,

and the quality of reports generated. The test results and user feedback point to the realization of an intelligent model for monitoring and reporting the performance of banking e-channels, something lacking presently. Even though simulated data was used, the model developed met the primary goal of this research: a MAS approach for monitoring the performance of e-channels.

#### **6.4 Recommendations for further research**

Because monitoring the performance of e-channels in banking environments is a complex and dynamic problem, the implementation of MAS provides the required solution but consumes relatively greater computing resources depending on the nature of the problem being solved. Therefore, it pays off for multi-agent system developers to consider optimizing database queries and employing efficient algorithms in developing intelligent performance monitoring systems.

This research developed a system prototype using simulated data because actual data could not be obtained from banks due to privacy and confidentiality reasons. While the simulated data as obtained from questionnaires was close to real world, it is important that further researches consider conducting using real transactional data from banks. This project did not have the prototype interact with an actual banking system. Thus, further research may consider developing a multi-agent system and have it tested within an actual banking environment for better results. However, this will require legal agreements between the researcher and bank to ensure that confidential information is not disclosed to unintended persons or otherwise abused

In addition, future research may also consider combining database queries, database server monitoring, and network status messages in monitoring and reporting performance of different channels. This might yield better results to this approach that relied on database queries and link status in reporting the performance of electronic channels. Finally, this work covered one cause of failure: network failure. The study therefore suggests that future work should focus on advancing towards identifying and reporting other causes of system failure causes such as server processes and database failures.

## REFERENCES

- Avancini, H. (2000). A Java Framework for Multi-agent Systems. *Journal of Informatics and Operations Research*, 3(1) 1-12.
- Banks, E. (2012). *Selecting the Right Server & Application Monitoring Tool*. White Paper. Retrieved April 05, 2014, from [http://web.swcdn.net/creative/pdf/Whitepapers/serverapptool\\_whitepapers.pdf](http://web.swcdn.net/creative/pdf/Whitepapers/serverapptool_whitepapers.pdf)
- Bellifemine, F., Caire, G. & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. USA: John Wiley & Sons, Ltd.
- Birgelen, V.M., Jong, d.A and de Ruyter, K. (2006). Multi-channel service retailing: the effects of channel performance satisfaction on behavioral intentions. *Journal of Retailing*, 82(4), 367-378.
- Bordini, R. H., Dastani, M., and Winikoff, M. (2006). *Current Issues in Multi-Agent Systems Development*. Invited Paper.
- Catalan, T. R. (2004). *Banking Channel Management - Global Trends and Strategies*. Massachusetts Institute of Technology. A Thesis.
- Chan, V., Ray, P. and Parameswaran, N. (2008). Mobile e-Health monitoring: an agent-based approach. *IET Communications*, 2(2), 223–230.
- Chapman *et al.* (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. Retrieved April 05, 2014, from <http://www.crispdm.org>, 2000.
- Curry, A. and Penman, S. (2004). The relative importance of technology in enhancing customer relationships in banking - a Scottish perspective. *Managing Service Quality*, 14(4), 331-341.
- Daniela, B., Simona, M. & Dragos, P. (2010). Electronic Banking - Advantages for Financial Services Delivery. *The Journal of the Faculty of Economics - Economic*, 1(2), 672-677
- Estevez, P., C. Held, and C. Perez (2006). Subscription fraud prevention in telecommunications using fuzzy rules and neural networks. *Expert Systems with Applications*, 31, 337–344.
- Gao, Xu, Wang and Green (2009). Knowledge-based anti-money laundering: a software agent bank application. *Journal of Knowledge Management*, 13(2), 63-75.
- Godo *et al.* (2002). A multi-agent system approach for monitoring the prescription of restricted use antibiotics. *Artificial Intelligence in Medicine*, 27, 259–282.

- Hitt, L. and Frei, F. (2002). Do better customers utilize electronic distribution channels?, The Case for PC banking. *Management Science*, 48(6), 732-48.
- Kumbhar, M. V. (2011). Alternative Banking Channels and Customers' Satisfaction: An Empirical Study of Public and Private Sector Banks. *International Journal of Business and Management Tomorrow*, 1(1), 1-24.
- Lau, S. K. & Li, X. (2005). A multi-agent approach towards collaborative supply chain management. In B. Liu (Eds.), *International Conference on Electronic Business* (pp. 929-935). Hong Kong: The Chinese University of Hong Kong.
- Laukkanen, T. (2007). Customer preferred channel attributes in multichannel electronic banking. *International Journal of Retail & Distribution Management*, 35(5), 393-403.
- Luo, Liu, and Davis (2002). Multi-Agent Decision Support System for Stock Trading. IEEE Network.
- Mangina, E. (2002). *Review of Software Products for Multi-Agent Systems*. UK: Applied Intelligence (UK) Ltd.
- Ostergaard, L. (2010). Optimizing retail banking channels for mass-market customers in Denmark. *PhD Thesis*.
- Padgham, L., Winikoff, M. (2004). *Developing Intelligent Agent Systems: A Practical Guide to Designing, Building, Implementing and Testing Agent Systems*. John Wiley & Sons.
- Sreenivasulu, M. Prasad, E.V., and Raju, G.S.S. (2011). Multi-Agent System Approach for Congestion Control in ATM Networks. *International Journal of Computer Trends and Technology*, 2(2).
- Uppal, R. K. (2011). E-delivery channels in banks-a fresh outlook. *Journal of Arts Science & Commerce*, 2(1), 180-191.
- Vaishnavi, V. and Kuechler, W. (2004). *Design research in information systems*. Retrieved October 2, 2004, from <http://www.isworld.org/Researchdesign/drisISworld.htm>, 2004.
- Verma, A. & Kusiak, A. (2012). Fault Monitoring of Wind Turbine Generator Brushes: A Data-Mining Approach. *Journal of Solar Energy Engineering*, 132(1), 1-9.
- Vlassis, N. (2003). *A Concise Introduction to Multiagent Systems and Distributed AI*. University of Amsterdam.
- Winikoff, M. (2007). Defining syntax and providing tool support for Agent UML using

a textual notation. *International Journal of Agent-Oriented Software Engineering* 1(2), 123–144.

Worku, Z.G. (2007). *A Framework for Multi-Agent Referral Decision Support*. Ethiopia: Addis Ababa University. A Thesis.

Yao et. al. (2009). Studies on Multi-Agent Based Partial Discharge Online Monitoring System for High Voltage. *WSEAS Transactions on Circuits and Systems Apparatus*, 7(8), 620-629.

Zhang, Z., Thangarajah, J., Padgham, L. (2007). Automated unit testing for agent systems. In: *Proceedings of the 2<sup>nd</sup> International Working Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2007)*, pp.10–18. INSTICC Press

Zhang, Z., Thangarajah, J., Padgham, L. (2008). Automated unit testing intelligent agents in PDT. In: *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pp.1673–1674.

## APPENDICES

### Appendix A – Data Collection Questionnaire

Questionnaire to gather data for purposes of completing a Master of Science degree in Computer Science at the University of Nairobi.

Full Name (Optional)	
Name of Bank (Optional)	
Position Held	
<b>Question</b>	
<b>Response</b>	
1) Does your bank conduct regular performance monitoring and reporting of its electronic channels (e-channels)?	<input type="checkbox"/> Yes <input type="checkbox"/> No
2) Which of the following channels is/are given priority in your bank in terms of performance monitoring and reporting? (Choose a maximum of two) A) ATMs (Automated Teller Machines) B) POS (Point of Sale Devices) C) Internet Banking D) Mobile Banking	
3) Which of the following factors may affect the customer's choice of his/her preferred electronic channel for doing business with your bank? A) Stability of the channel (in terms of how long it is running well) B) Types of transactions supported by the channel C) 24/7 accessibility to or availability of the channel for transactions D) Security of using a channel	
4) Which 2 e-channels do your bank customers frequently use? A) ATMs (Automated Teller Machines) B) POS (Point of Sale Devices) C) Internet Banking D) Mobile Banking	

<p>5) What are the key factors your bank is interested in during the performance monitoring of its e-channels? (List all that apply)</p> <p>A) Number of transactions per unit time</p> <p>B) Frequency of system uptime or downtime</p> <p>C) Time of the day, week, month, or year</p> <p>D) Amount of money involved in each transaction</p> <p>E) Type of transaction</p> <p>F) Type of channel</p> <p>G) Others</p>	
<p>6) If you chose G) in the above question, list 2 other factors you consider important in monitoring the performance of an electronic channel</p> <p>(i) .....</p> <p>(ii) .....</p>	
<p>7) For monitoring purposes of an e-channel, what does your bank rely on to determine if a channel performing normally or not? (Select all that apply)</p> <p>A) Number of transactions per unit time</p> <p>B) Duration between any two immediate transactions</p> <p>C) Network link status</p> <p>D) All the above</p>	
<p>8) What parameters or fields are necessary in each unique transaction record irrespective of the electronic channel in use?</p> <p>(i) .....</p> <p>(ii) .....</p> <p>(iii) .....</p> <p>(iv) .....</p> <p>(v) .....</p> <p>(vi) .....</p> <p>(vii) .....</p>	
<p>9) Does your bank have separate databases/servers for different channels?</p>	<p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>



<p>10) Which techniques do you use in your bank to monitor and report the performance of your electronic channels? (Choose all that apply)</p> <p>A) Database Queries</p> <p>B) Application Monitoring</p> <p>C) Data Mining tools</p> <p>D) Rule-based technology</p> <p>E) Multi-Agent Systems technology</p> <p>F) Other</p>	
<p>11) What challenges does your bank face in monitoring its electronic channels? (List at least 2 challenges).</p> <p>a. ....</p> <p>b. ....</p> <p>c. ....</p> <p>d. ....</p>	
<p>12) What consequences/risks arise from system downtimes with respect to electronic channels? (List at least 2)</p> <p>(i) .....</p> <p>(ii) .....</p> <p>(iii) .....</p> <p>(iv) .....</p>	
<p>13) Name the job roles responsible for monitoring e-channels in your bank.</p> <p>(i) .....</p> <p>(ii) .....</p> <p>(iii) .....</p> <p>(iv) .....</p>	
<p>14) What are the media through which performance reports are shared in your bank? (Choose all that apply).</p> <p>A) Email</p> <p>B) SMS</p> <p>C) Application Window</p> <p>D) Social Media</p>	

E) Other	
<p>15) What performance measures/parameters are used in your bank?</p> <p>A) Number of transactions per unit time</p> <p>B) Number of complaints from customers</p> <p>C) Duration that network links are up or down</p> <p>D) Amounts transacted per unit time</p> <p>E) Others (list them below)</p> <p>.....</p> <p>.....</p>	
<p>16) What is the frequency of sharing performance reports?</p> <p>A) Hours per day (specify number of hours)</p> <p>B) Daily</p> <p>C) Longer than Daily</p>	
<p>17) Have you experienced times when no one knew that a channel or several of the electronic channels were down (not working at all or performing below normal ranges) until a customer called?</p>	<input type="checkbox"/> Yes <input type="checkbox"/> No
<p>18) What do you attribute the above problem to?</p> <p>.....</p> <p>.....</p> <p>.....</p>	
<p>19) What key attributes would you recommend that a robust performance monitoring and reporting solution or system must have?</p> <p>.....</p> <p>.....</p>	
<p>20) Do you think your bank needs a different technology solution or an improvement to the existing solution to respond to performance monitoring needs in more efficient and effective manner?</p>	<input type="checkbox"/> Yes <input type="checkbox"/> No

## Appendix B – Sample Code

### A. Transactions Agent (Generating and posting simulated transactions)

```
package com.agents;
import java.io.BufferedReader;
import javax.swing.*;
import java.awt.event.*;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import jade.content.lang.Codec;
import jade.content.lang.sl.SLCodec;
import jade.content.onto.Ontology;
import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.OneShotBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
public class CommunicationAgent extends Agent {
private Codec language = new SLCodec();

private void register()//Registration
{
DFAgentDescription dfd = new DFAgentDescription();
dfd.setName(getAID());
ServiceDescription sd = new ServiceDescription();
sd.setType("Notification Agent");
sd.setName(getLocalName());
dfd.addServices(sd);

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

```

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

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

register();

//Add emailing function
postToURL("08");
}
public void postToURL(String receiverid){
    System.out.print("Checking the Central Server for Electronic Channel Performance Details " );
    try {
String stringToReverse = URLEncoder.encode("pi", "UTF-8");

URL url = new URL("http://localhost/atmis/home/sendatmemail.php");
URLConnection connection = url.openConnection();
connection.setDoOutput(true);

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

BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));

```

```

String decodedString;
while ((decodedString = in.readLine()) != null)
    {
        System.out.println(decodedString);
    }
in.close();
}
catch(IOException e)
    {
        e.printStackTrace();
    }
}
}

```

## **B. ATM Reporting Agent Code**

```

package com.agents;
import java.io.BufferedReader;
import javax.swing.*;
import java.awt.event.*;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import java.net.URLEncoder;
import jade.content.lang.Codec;
import jade.content.lang.sl.SLCodec;
import jade.lang.acl.ACLMessage;
import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.MessageTemplate;
import jade.content.*;
import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.OneShotBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAAgentManagement.DFAgentDescription;

```

```

import jade.domain.FIPAAgentManagement.ServiceDescription;

public class AtmReportingAgent extends Agent {
private String remoteServerResponse="";
private String strResponse="\n \t Checking ATM Transactions \n";
private String url="http://localhost/atmis/search.php";
private String transaction="ATM";
final ACLMessage reply = new ACLMessage(ACLMessage.INFORM);
protected void setup() {
    System.out.println("\n Agent "+getLocalName()+": Waiting for REQUEST message \n");
    ACLMessage msg =
blockingReceive(MessageTemplate.MatchPerformative(ACLMessage.REQUEST));

    //Create a repetitive behaviour to execute requests
addBehaviour(new CyclicBehaviour(this)
{
    public void action()
    {
        ACLMessage msg= receive();
        if (msg!=null){

            if(msg.getContent().equals("ExecuteActivity")){
                AmisUtil amisUtil=new AmisUtil();

                remoteServerResponse=amisUtil.accessUrl(url,"Check",transaction,strResponse);

                //if(remoteServerResponse!=null){

                    reply.setPerformative(ACLMessage.INFORM);
                    reply.addReceiver(msg.getSender());
                    reply.setContent("Checking ATM Transaction");
                    send(reply);
                // }

            }
        }
    }
}
}

```

```

        }
        block();
    }
});

}
protected void takeDown() {
    System.out.println("Agent "+getLocalName()+": terminating");
}
}}

```

### C. Master Agent Code

```

package com.agents;;
import java.util.Random;
import jade.core.Agent;
import jade.core.behaviours.WakerBehaviour;
import jade.core.behaviours.TickerBehaviour;
import java.io.BufferedReader;
import javax.swing.*.*;
import java.awt.event.*.*;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import jade.lang.acl.ACLMessage;
import jade.content.*.*;
import jade.core.AID;
import java.util.ArrayList;

/* Master Agent to initiate other agents to execute their tasks */
public class MasterAgent extends Agent {
    protected void setup() {
        System.out.println("Agent "+getLocalName()+" started.");
        // Add the Random Generation of data

```

```

        final Random rand = new Random();
        final int n = rand.nextInt(5000) + 1000;
        addBehaviour(new TickerBehaviour(this, 5000) {
            protected void onTick() {
                ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
                if((getTickCount()%7==0) || (getTickCount()%3==0)){
                    //do nothing
                } else{
                    msg.addReceiver(new AID("transactionAgent", AID.ISLOCALNAME));
                }
                msg.addReceiver(new AID("atmReportingAgent", AID.ISLOCALNAME));
                msg.addReceiver(new AID("posReportingAgent", AID.ISLOCALNAME));
                msg.addReceiver(new AID("channelReportingAgent", AID.ISLOCALNAME));
                msg.setLanguage("English");
                msg.setContent("ExecuteActivity");
                send(msg);
                System.out.println("Agent "+myAgent.getLocalName()+"Requested Agents to Execute activities\n");
            }
        });
        // Add the transact for a period of 2 minutes)
        addBehaviour(new WakerBehaviour(this, 240000) {
            protected void handleElapsedTimeout() {ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
                msg.addReceiver(new AID("channelReportingAgent", AID.ISLOCALNAME));
                msg.setLanguage("English");
                msg.setContent("Completed Posting Transactions For The Provided Period");
                send(msg);
                System.out.println("***** \n ");
                System.out.println("Agent "+myAgent.getLocalName()+": Completed Posting Transactions \n ");
                System.out.println("***** \n ");
                myAgent.doDelete();
            }
        });
    }
}

```



## Appendix C – UAT Questionnaire

### UAT QUESTIONNAIRE

A Multi-Agent System Approach to Performance Monitoring and Reporting of Electronic Channels of a Bank

Full Name (Optional)

Position held in the bank

Please give your comments on the reports you viewed in each User Acceptance Test (UAT). Example: “The transaction data fields are fewer than is the case in a real bank transaction...”

#### **UAT-1: Generation of simulated transactions from POS and ATM channels**

Log into the web application using the URL <http://localhost/atmis/> and login credentials provided. Take note of the last transaction’s details. Open Windows Command Prompt and change directory to C:\jade.

While at DOS Prompt, copy and paste the following command and press Enter:

```
java jade.Boot -gui
```

Copy and paste the following on the command prompt and press Enter key:

```
java -cp lib\jade.jar;classes jade.Boot -agents PostTransactions:com.agents.PostAgent
```

Observe messages displayed. Return to the browser and note any changes.

#### **UAT-2: Event detection & notifications**

While at the command prompt, press Ctrl + C. Notice there are no messages showing new transactions are being posted. Go to the browser and check if there are newer transactions or not. Login to your email and check if there is a notification / alert message sent reporting the event.

### **UAT-3: Reporting of performance**

Open a new Command Line window and run the following command from C:\jade directory:

```
java -cp lib\jade.jar;classes jade.Boot -agents AtmReport:com.agents.AtmReportingAgent
```

Open a new Command Line window and run the following command from C:\jade directory:

```
java -cp lib\jade.jar;classes jade.Boot -agents PosReport:com.agents.PosReportingAgent
```

Check the browser window and email for any alerts.



### **UAT-4: Agents communication**

Open the JADE window on the task bar. Go to Tools menu and select Sniffer submenu. View how different agents exchange messages on the *Sniffer agent* window as the program runs.



### **UAT-5: Summary reports**

On the browser window, click “POS Reports” or “ATM Reports” or “E-Channels Reports”. View the details of the report.



**Thank you!**

End