UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

A MULTI- AGENT MODEL FOR SYSTEM USER ACCESS RIGHTS AUDIT

OKOTH FREDRICK

P58/61495/2010

SUPERVISOR

CHRISTOPHER MOTURI

JULY, 2012.

A Research Project submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science of the University of Nairobi.

Declaration

The Research Project as presented in this report is my original work and has not been presented for any other University Award. Materials of work done by other researchers are mentioned by clear reference.

Signature: _____ **Date:** 29TH AUGUST 2012

Okoth Fredrick

The Research Project has been submitted in partial fulfillment of the Requirements for the Degree of Master of Science in Computer Science at the University of Nairobi with my approval as the University supervisor.

Signature: _____ **Date:** 29 August 2012

Mr. Moturi Christopher

Deputy Director

School of Computing and Informatics

Dedication

To my late mum Mrs. Eunice Bitta who taught me the three great virtues in life; hard work, honesty and integrity.

To my son Jude and wonderful wife Kezia for the moral support, patience and understanding during the course of this study.

Acknowledgements

I wish to acknowledge the guidance and support offered by the panelists (Mr.Orwa, Prof. Waiganjo, Mr. Muchemi & Mr. Ogutu) in the course of writing this project. This would not have been possible without your constant reviews and feedback.

Am greatly indebted to my project supervisor Mr. Christopher Moturi for the critical input and guidance in the course of carrying out this research.

Abstract

Insider threats are alive with us today and so access to the Information systems has become so critical that organizations have incorporated periodic user access rights audit in their Information security policy's to be carried out by System auditors. System auditors need to consistently audit user's access to applications while cross referencing the same with related user roles and responsibilities as captured in the Job description to ensure compliance. Appropriate segregation of duties is key in this review as mismatch is reported and investigated in a timely manner.

This study proposes a multi-agent model where autonomous agents represent the various aspects of access controls captured in the Job description, active users log and the organizational policy on system access. These agents communicate to establish scenarios where conflicts exist. The conflicts are defined as either applications accessed by system users not captured in their Job descriptions, users accessing the same application as both user and super user and access policy violations. These conflicts are reported in a risk matrix format as either low, medium or high. The tropos methodology was adopted to model this multi-agent system.

The study looked at a sampled number of system users from which a total of 11 system users reported violations representing 23% of the sample size. The proposed model provides a platform for auditing what system users' access, their role and responsibilities within the organization as well as the policy requirements governing system access and usage.

Table of Contents

List of figures

List of Tables

Definition of Terms

1. BDI: Belief, Desire and Intention
2. CCMCM : Continuous Control Monitoring Certification Manager
3. CERT : Computer Emergency Readiness Team
4. CFO : Chief Finance Officer
5. CIO : Chief Information Officer
6. CRM: Customer Relationship management
7. ERP App : Enterprise Resource Planning Applications
8. Ex. Audit : External Audit
9. Fin Sys : Financial Systems
10. I.T : Information Technology
11. IdA : Identity Audit Application
12. IdM : Identity Management Software
13. IS: Information System
14. ISACA : Information Systems Audit and Control Association
15. JD: Job Description
16. MAS: Multi-Agent Systems
17. Purc Sys : Purchasing Systems
18. RBAC : Role Based Access Controls
19. RM : Risk Management

CHAPTER 1 INTRODUCTION

1.1 Background

Organizations continue to suffer from fraud committed by disgruntled employees who for personal gain or gratification swindle their employers. Most of these crimes are as a result of abuse of systems access rights assigned to them. Today, computer security implementations put a lot of emphasis on external attacks while ignoring the threat from within the organization. Insider threats are alive with us and so access to the Information Systems has become so critical that organizations have incorporated periodic user access rights audit in their Information Security Policy to be carried out by the systems auditors. A systems auditor gives an independent opinion on the controls implemented with regards to access within an Information system environment. There exists potential ground for computer fraud if system users have excess access rights to Information system resources which are not appropriately segregated in line with the specific user's daily roles and responsibilities.

Criminals especially I.T savvy ones have become expert at recognizing weaknesses in system access and are knowledgeable about the tools necessary to successfully exploit weak systems. Statistics from Computer Emergency Readiness Team (CERT) and industry security analysts show that about 80% of all malicious activities come from current or former employees. Thus more than ever, one of the prime concerns in any audit and for management is the logical access to computer systems and data. (ISACA, 2010).

The need to consistently audit user's access to Information system while cross referencing the same with their roles and responsibilities while ensuring appropriate segregation of duties is a policy requirement as mismatch is reported and investigated in timely manner. This helps in averting the potential grounds for fraud and limits the level of damage in case the same had been initiated by the perpetuators. The model provides a platform for auditing what system users access, their role and responsibilities within the organization an well as the policy requirements for the system users. It forms the basis upon which further computer forensic investigation can be carried out incases where violations are reported.

1.2 Problem Definition

The need to continuously review what authorized system users' access in the Information System by an independent party is a key undertaking not only in risk management but also aims at reducing fraud as well as ensuring compliance with the organization's Information Security policy requirements. To cross reference what users actually access in the system with their roles and responsibilities in the organization is a manual process which consumes considerable time and resources with regards to manpower and efficiency.

The assurance that new system users are correctly defined and appropriate roles assigned to them continues to be a major challenge. Ensuring that segregation of duties in user access to the Information System as defined in the user definition forms is implemented without any omissions by the system administrators is a challenge as sampling is done for manual comparison with the active user responsibilities log. Cases where system users with excessive access rights have abused these privileges to commit computer fraud therefore become very difficult to detect because there exist no indicators.

Network and database administrators put a lot of focus on logged network and database violations information and do not usually see the threat within the organization relating to data being accessed by authorized users.

In this study, we proposed to build multi-agent model used for consistent user access rights audit with the ability to cross reference what system users have accessed in the application database against their defined roles and responsibilities within the organization. The model was to incorporate at the highest level access control policies and related procedures or business rules as defined by management, defined user roles and responsibilities, application database logs of active users and their responsibilities. An audit report was to be generated based on the analysis of conflicts between these parameters. The conflicts would be categorized for purposes of isolating fraud indicators for further computer forensics investigation.

1.3 Research Objectives

The following were the specific objectives of this study:

1. Identify and model user access policies, procedures and business rules of an organization using multi-agents.
2. Identify and model user roles and responsibilities for a sampled number of system users within an organization.
3. Identify and model a user access log extracted from the Database application.
4. Develop a multi-agent user access rights audit model with key risk indicators (Potential areas of fraud)
5. Test the practical application of this model in the real world scenario especially in the user access rights audit of a Kenyan oil marketer.

1.4 Research Questions

The following research questions arose based on the objectives of the study;

1. How can one model access policies, procedures and business rules using multi-agents?
2. How can one model user roles and responsibilities within an organization using multi-agents?
3. How can one model user activity logged in the database application?
4. What constitutes a conflict scenario and how can this be modeled using multi-agents?
5. Which conflict scenarios are most likely fraud indicators?
6. What constitutes segregation of duties and how can this be modeled?
7. How can one map access policies, user roles and user activity logs?
8. How can the model be tested to verify its workability?
9. What type of input is required and how will the output appear?

1.5 Justification

The purpose of this research was to bring together the concept of multi-agents through a model for purposes of auditing users' access to the database application in a fast effective and efficient manner. This would eliminate the manual process of comparison that existed. The model would be able to appropriately identify users accessing applications which do not relate to their roles and responsibilities. This research would also ensure compliance with the Information security policy on user access to Information systems. The model therefore provided an assurance to the business that users' responsibilities were appropriately segregated and so Information system risks related to application access were identified and controlled.

1.6 Scope and Limitation

The research problem was based on a Kenyan oil marketer. The scope was limited to what current active users were accessing in the system which was extracted from the application log and a comparison made with the related users' roles and responsibilities. Applications allocated to a specific user was also evaluated against the organizations Information Security policy in order to establish support for segregation of duties among the various system users. The major limitation was that the agents would not be moving on a live environment but on the application log extracted from the Oracle database. The interface with Teammate audit reporting software was not implemented.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

User access review is a process that an organization implements to actively monitor and verify the appropriateness of a users' access to systems and applications based on an understanding of the minimum necessary requirements for users to perform or support business activities or functions. The responsibility for granting access and performing periodic verification of the appropriateness of that access rests with the business owner of the system or application. User access reviews help implement the principles of 'least privileges' based on business need and segregation of incompatible roles and functions. User access reviews serve to verify and validate that user access to systems and applications is appropriate given users' roles and responsibilities within the organization. (ControlCase, 2011).

In the economic downsizing that organizations are faced with today, insider threat ranks highest and is the highest concern for corporate I.T and corporate investigative divisions. Those who are inside the security perimeter and are about to be let go due to the reduction in force have access to intellectual properties, research and development documentation and ongoing business deals that could easily affect the organizations bottom line. Another facet of downsizing threat is the motivation to exact revenge on the organization. (ISACA, 2010).

In general, a Role represents a set of responsibilities needed to conduct business operations or transactions, Access represents the privileges and resources used by someone within a role and Identity represents someone with a given role at a certain point in time. (ISACA, 2011)

2.2 Objectives of access controls

The objectives of an access control system are often described in terms of protecting system resources against inappropriate or undesired user access. From a business perspective, this objective could just as well be described in terms of the optimal sharing of information. After all, the greater objective of IT is to make information available to users and applications. A greater degree of sharing gives rise to increased productivity. Although on the surface, access control appears to gets in the way of this objective, in reality, a well-managed and effective access

control system actually facilitates sharing. A sufficiently fine-grained access control mechanism can enable selective sharing of information where in its absence, sharing may be considered too risky altogether. (Ferraiolo et al 2007).

When considering any access control system one considers three abstractions of control: access control policies, access control models, and access control mechanisms. Policies are high-level requirements that specify how access is managed and who, under what circumstances, may access what information. While access control policies may be application-specific and thus taken into consideration by the application vendor, policies are just as likely to pertain to user actions within the context of an organizational unit or across organizational boundaries (Kuhn, 2007).

For instance, specific policies may pertain to the resources that can be accessed by consultancies or other business partners. Such policies may span multiple computing platforms and applications. Policies may pertain to resource usage within or across organizational units or may be based on need-to-know, competence, authority, obligation, or conflict-of-interest factors. Although there are several well-known access control policies, generating such a list is of limited value, since business objectives, tolerance for risk, corporate culture, and the regulatory responsibilities that influence policy differ from enterprise to enterprise, and even from organizational unit to organizational unit. In determining the user's ability to perform operations on resources, access control mechanisms compare the user's security attributes to those of the resource. Access control checks can be evaluated based on a previously determined set of rules.

2.3 Role based access controls (RBAC)

A role is chiefly a semantic construct forming the basis of access control policy. With RBAC, system administrators create roles according to the job functions performed in a company or organization, grant permissions or access authorization to those roles, and then assign users to the roles on the basis of their specific job responsibilities and qualifications. (InterNational Committee for Information Technology Standards, 2004)

In the view of Coyne et al (2008), with RBAC, permissions are assigned to roles instead of users. This creates a layer of abstraction where users can be assigned to roles instead of permissions being assigned directly to these users. Because the number of roles in an organization is usually much smaller than the number of permissions in that organization's IT systems and networks, the layer of abstraction provided by RBAC can simplify the authorization of permissions to the users. Also, roles and permissions typically change much more slowly over time than do personnel, which is one reason why RBAC is effective in reducing administration costs. Users are assigned to the roles and they automatically received all the permissions associated with the assigned roles.

It becomes a relatively simple matter for a person to be assigned the roles they need without the necessity of the assigner even understanding what the permissions are on the various networks or systems. That work will have been done once by software engineers. Once roles with their permissions have been defined, an administrative rather than a technical person can perform the assignment of users to roles. This is a major benefit of RBAC. The advantages that RBAC can provide include; reduced administrative costs, support for finer-grained access control policies, and support for auditing and reporting on authorizations of users to access corporate resources. (Coyne et al 2008).

2.4 Identity audit applications (IdA)

The purpose of IdA applications is to help organizations identify differences between user permissions and user access activity. IdA applications generally operate by loading lists of user rights from repositories such as Windows Server Active Directory, importing and aggregating user access data from systems and application activity logs into a centralized data store, and using pattern-matching algorithms to correlate user identities across various logs and compare user access activity to user rights. The application then presents access policy exceptions on a dashboard.

IdA tools represent a distinct category from identity management (IdM) software, which automates controls over provisioning of user access privileges. Although IdM tools are effective for the problem they are designed to solve, they are poorly suited to the needs of auditors and

other reviewers because they do not report on actual user access activity and policy exceptions. In fact, IdM applications typically do not track user activity at all and therefore lack the critical information needed to perform IdA. By contrast, IdA applications ensure the verification of policy remains separate and distinct from the enforcement of policy. Although the IdA application may be able to send remediation information to the IdM provisioning system and linking the systems creates its own development and deployment challenges. The IdA solution should not provide remediation directly on its own, as this functionality could breach separation of control and auditing. Instead, the IdA application should pinpoint access exceptions in a way that quickly highlights access compliance weaknesses and tracks managers who are inattentive to remediation. (Glithero et al 2010).

2.4.1 Permissions Analyzer for Active Directory

This is a tool that is used by system administrators to get a hierarchal view of the effective permissions, access rights for files or shared folders from a single dashboard. It enables the systems administrator to get a complete hierarchical view of the effective permissions and access rights for a specific file folder (Network file system) or shared drive and easily see what permissions a user has for an object. The systems administrator is able to see all permissions from a desktop dashboard and browse permissions by group or individual user as well as analyze user permissions based on group membership combined with specific permissions. (www.solarwinds.com/products/freetools permissions analyzer for active directory)

2.4.2 Novell Identity Audit Application

This tool provides a simple yet powerful framework for searching, reporting and alerting on security, system and application events. It aggregates event data from a variety of Novell Identity and Access Management solutions and provides predefined reports that help demonstrate compliance, identify potential security issues and ensure the system is working as designed. Real-time alerts allow detection of critical events as soon as they occur, providing administrators with needed insight into user activity. By including an embedded database and reporting system in an intuitive Web 2.0 interface, Novell Identity Audit provides the necessary tools for

monitoring your identity infrastructure without requiring major investments in hardware, database licenses or personnel. (www.novell.com/products/audit/)

2.4.3 HP Select Audit software

The tool helps to manage and automate identity audit lifecycle processes across the entire identity and access management infrastructure. Using a visual control model, HP Select Audit software provides an auditor's perspective on adherence to regulatory requirements. It aggregates identity audit information in a tamper-aware store, with real-time alert handling and workflow-based attestation of reports, to provide insight into identity and security controls and how they align with the desired state of the business. It's key features and benefits include; Visualizing adherence to compliance guidelines and enterprise audit policies using a dashboard providing control modeling of identity audit, Use of workflow-based attestation of reports to provide the enterprise with documented proof that reports were read and approved and adhere to the desired state of the business, enabling of automated consolidation of identity-based audit covering the entire identity and access management infrastructure and enabling segregation of duty and data privacy through audit access management and tamper-aware data. This product by HP was discontinued in October 2010 due to obsolescence. (www.spsnet.com/documents/slctaud_ds.pdf)

2.4.4 Quest Access Manager

It controls user and group access to resources throughout the Windows enterprise and Network attached storage devices in order to meet security and compliance requirements, control operational costs and optimize infrastructure performance. It intelligently suggests who should own which data resources, bringing accountability and visibility from a single console into resources that are actively used. (www.quest.com/access-manager).

2.5 Multi-Agent concept and approach

Multi-Agent is an organization of coordinated autonomous agents which interact in order to achieve common goals. An agent is a component of software or hardware, which are capable of acting exactingly in order to accomplish tasks on behalf of their users. Agents exhibit the

following characteristics: autonomy, reactiveness, proactiveness, social ability, veracity, benevolence, rationality, learning/adaptation and have distinct personality, behavior, name and role. (Georgini et al, 2001).

Multi-Agents being open source are therefore able to operate in multiple platforms continuously monitoring what users' access in the system and comparing with related roles of the same user as defined in the job description. The agents also make comparisons to establish whether there is appropriate segregation of duties within a specific user's access in the system. They are guided in decision making by the three abstractions of access control systems which include; access control policies, access control models and access control mechanisms as shown below.

The three abstractions of access control systems.

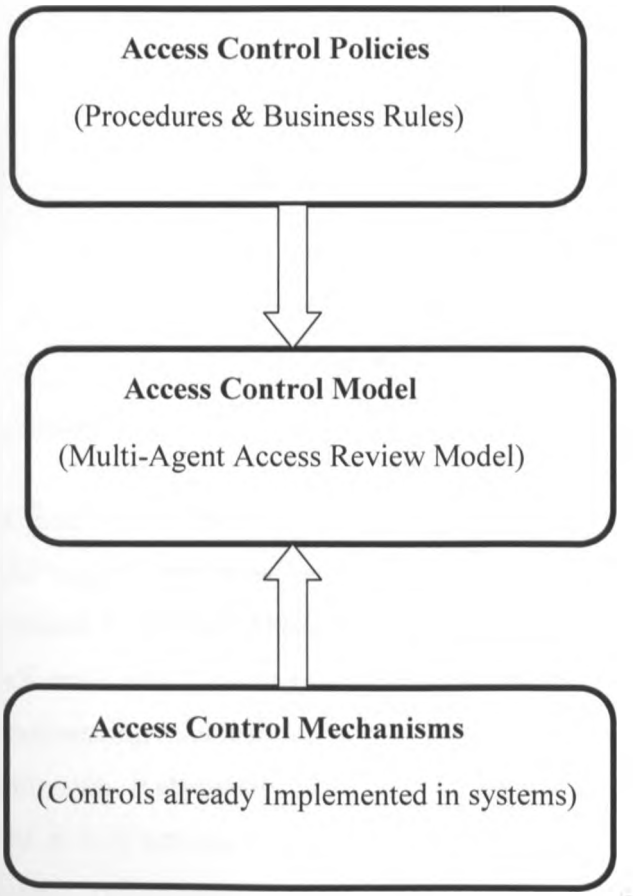


Figure 2.1: The three abstractions of access controls

2.6 Continuous Controls Monitoring Certification Manager (Conceptual Model)

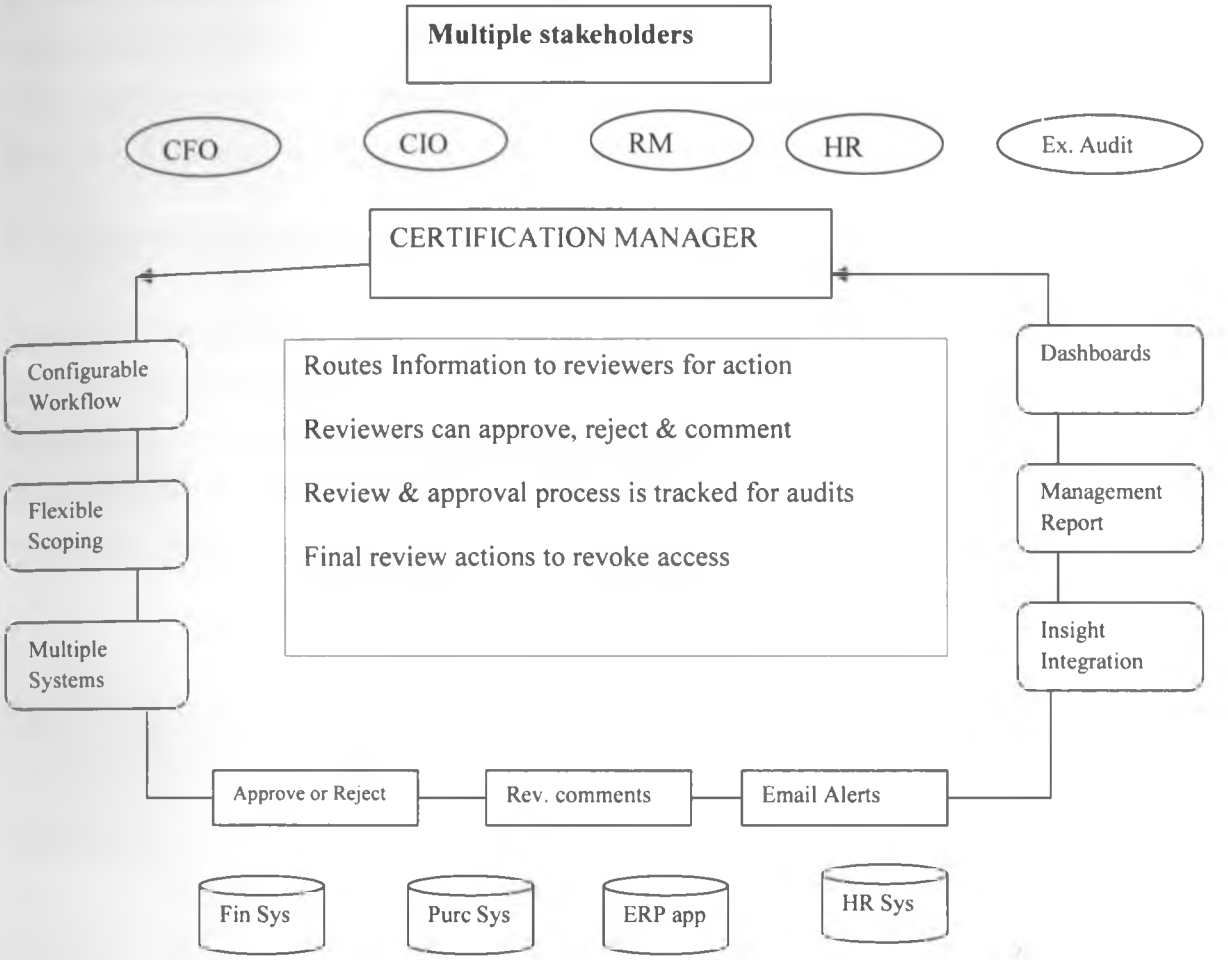


Figure 2.2: Continuous Controls Monitoring Certification Manager

Approva's Certification Manager automates the end-to-end process for reviewing user access rights across ERP systems and other business applications. Comprehensive, easy-to-understand summaries are routed to approving reviewers so they can accept or revoke access rights for their employees. Audit trails provide evidence for external audits. The framework however does not carryout cross referencing of actual user access with their related duties and responsibilities within the organization. It also does not capture the aspect of IS policy with regards to system access. However, it does automate the aspect of reviewing user rights before granting them access to the system. (www.approva.net/products/certificationmanager)

Agents sit 'over' applications, watching, learning, and eventually doing things without being told by taking the initiative. Multi-agent systems are highly scalable and robust and so are able to support interoperability. They are also reusable and are applicable in distributed environments. They exhibit tremendous speed and efficiency while maintaining flexibility in multiple platforms. These are the benefits that will be realized from the proposed model.

2.7 Evaluation of existing access rights auditing tools

The table illustrates the comparison and review of available access rights audit tools and methods with the proposed model. It is of significance to note that all the four existing Identity audit applications do not report on actual user access activity and policy exceptions as they do not track user activity at all and so they lack that critical information. IdA applications ensure the verification of policy remains separate and distinct from the enforcement of policy.

Tool	**Permissions Analyzer**	**Novell Identity Audit Application**	**HP-Select Audit Software**	**Quest Access Manager**	**Continuous Controls Monitoring Certification Manager (Conceptual model).**
Properties	Hierarchical view of all permission on NTFS or shared drive. Specific to Networks & not applications.	Platform specific Real-time alerts Searching & reporting on security, system & application events	Platform specific Focuses on regulatory compliance. Real-time alerts	Windows based Platform specific Suggests data ownership	Not based on actual user activity Designed for assigning rights to new system users Support for segregation of duties. Both application & network based.

Table 2.1: Evaluation of existing access rights auditing tools

CHAPTER 3 METHODOLOGY

The study was conducted in phases which each phase being evaluated before the next. The approach supports both iteration and incremental models.

3.1 Data Analysis

3.1.1 Data collection

Sources of data

An understanding of the available access rights review tools was a key source of data for the study. Related work done based on the literature review formed a major source of data. Books, academic papers, journals and the internet were very significant data source in the study.

Sampling methodology and sample size

Systematic sampling which is a statistical method involving the selection of elements from an ordered sampling frame was used. The most common form of systematic sampling is an equal-probability method, in which every k*th* element in the frame is selected, where k is the sampling interval sometimes known as the skip and is calculated as:

$k = N/n$: where n is the sample size, and N is the population size. The sample size will be defined by the alphabetical ordering of the sample population who are the active system users.

With this technique, every element in the population has a known and equal probability of selection. With this technique, I arrived at a sample size of 48 based on the population of 147.

Experiments & Interviews

Data from a Kenyan oil marketer was used in the experiment. This data was a log of active users and their responsibilities extracted from the Oracle application and contained the following information; User name, Security group, Application, Responsibility within the application and User access start and end dates. Interview of a sampled number of system users for purposes of getting information on their specific roles and responsibilities within the organization was carried out. The organization's IS policy document was also obtained and a study done to gain

knowledge on its guide with regards to access to Information Systems. Simple interview questions were used in order to elicit feedback from system users on their daily activities within the organization. This was to enable implementation of the stipulated objectives.

Instrument/ Tool design

The questionnaire was designed and piloted amongst 10 system users. The test and retest for reliability was done. The objective was to establish the reliability of this tool for the purposes of the research work. The feedback satisfied the research questions behind the design.

3.2 Data analysis method

The following was the basis of data analysis for this project:

Requirements	Data Source	Analysis Plan
Log of Active Users & Their Responsibilities	Oracle Database Application	Based on systematic sampling, generate a sample alphabetically as per system usernames through quantitative analysis. Model the applications accessed by the sampled users by qualitative analysis. Merge this model component with others to establish consistency or conflict
System Users Roles & Responsibilities	Human Resources Department & System Users	Based on the sampled active system users, get the related user Job descriptions through quantitative analysis Targeted questionnaires to the sampled system users for purposes of getting feedback with regards to their role and responsibilities in the system (Quantitative analysis) Model this Job descriptions and feedback from questionnaires based on what users are supposed to access through qualitative analysis Merge this model component with others to establish consistency or conflict
The ICT Policy	ICT Manager	Identify Information Security policies on system access through quantitative analysis With regards to user applications, define what constitutes a conflict on segregation of duties within the sampled

		system users through qualitative analysis. Model these conflicts and merge this component with others to establish consistency or conflict
Key Risk Indicators	Internal Audit Manager (PWC Risk Framework)	Identify and model access conflicts as High, Medium or low through qualitative analysis. Merge this model component with other for reporting.

Table 3.1: Data analysis method

3.2.1 Access Rights Data Analysis

Systematic sampling which is an equal probability sampling technique was used with a sampling interval (skip) of three. Every third system user from the active user log was sampled giving a total sample size of 48 system users. From this sample, the following analysis was carried out in relation to the feedback from the questionnaires, Users Job descriptions and the IS access policy of the organization. The analysis was conducted in phases as follows;

The first phase involved the mapping of the Job descriptions with the feedback from the questionnaires. Where the two were consistent then a Yes was reported and when there was inconsistency then a No was reported. This was denoted by a Y and N respectively. The outcome of this analysis was then mapped with the active users' access log as described later in the detailed analysis. The feedback was also reported as a Yes for consistency and a No for violations or inconsistency. The active users log was further mapped into the IS policy access parameters modeled and violations reported accordingly. The following tabulation captures the actual results from this analysis;

1.Sampled User IDs'	**2**. Access Log (AL) based on usernames	**3**. Job Description sampled	**4**. Quest. Feedback Received	**5**. Quest. Feedback compliance with JD	**6**. JD & Quest. feedback compliance with AL	**7**. IS Policy Compliance with AL & 5	**8**. Risk Matrix on violations
Total	**48**	**48**	**39**	**37**	**37**	**37**	**11**

Table 3.2: Access rights data analysis results

3.2.2 Detailed Analysis

From the analysis of 48 samples, a total of 11 violations were reported. These violations were reported in risk matrix as either low, medium or high and the assumption made was that a single star represents a single violation. The results are as tabulated below;

Violations	Risk Matrix
6	Low
2	Medium
3	High
11	**Total**

Table 3.3: Access violations risk matrix

From the results, it was noted that whenever a violation was reported in the mapping of the Job description and the active users' log, the access policy was also violated. This was because the application access policy number 6.5.2.3-b defined authorization to modify data or execute commands, transactions or programs, or other access to production data on a need-to-know basis by job function. This policy was entirely reliant on access being granted based on the job description of the system user.

The detailed analysis proceeded as follows;

Analysis of the feedback from questionnaires.

The questionnaire was designed purposefully to elicit the users' feedback with regards to their duties and responsibilities with the organization. The objective of the questionnaire was to establish whether there was consistency between what the users say they do and the documented job description signed between the same users and their respective supervisors.

Initially, piloting was done with 10 questionnaires which represented 20% of the sample. The reason behind the piloting was to certain whether the feedback would satisfy the research

objectives. The piloting was successful and so the entire sample was dispatched to the various respondents. The response rate was 81%. That is out of a sample of 48 system users, response was received from 39 respondents which was scientifically viable. From the analysis we noted two cases in which the feedback from the questionnaire was completely different from the documented users' job descriptions. On further inquiry we established that one of the users had been promoted to a new department but his JD had not been updated accordingly while the second user had since left the organization although his access rights to the system had not been deactivated.

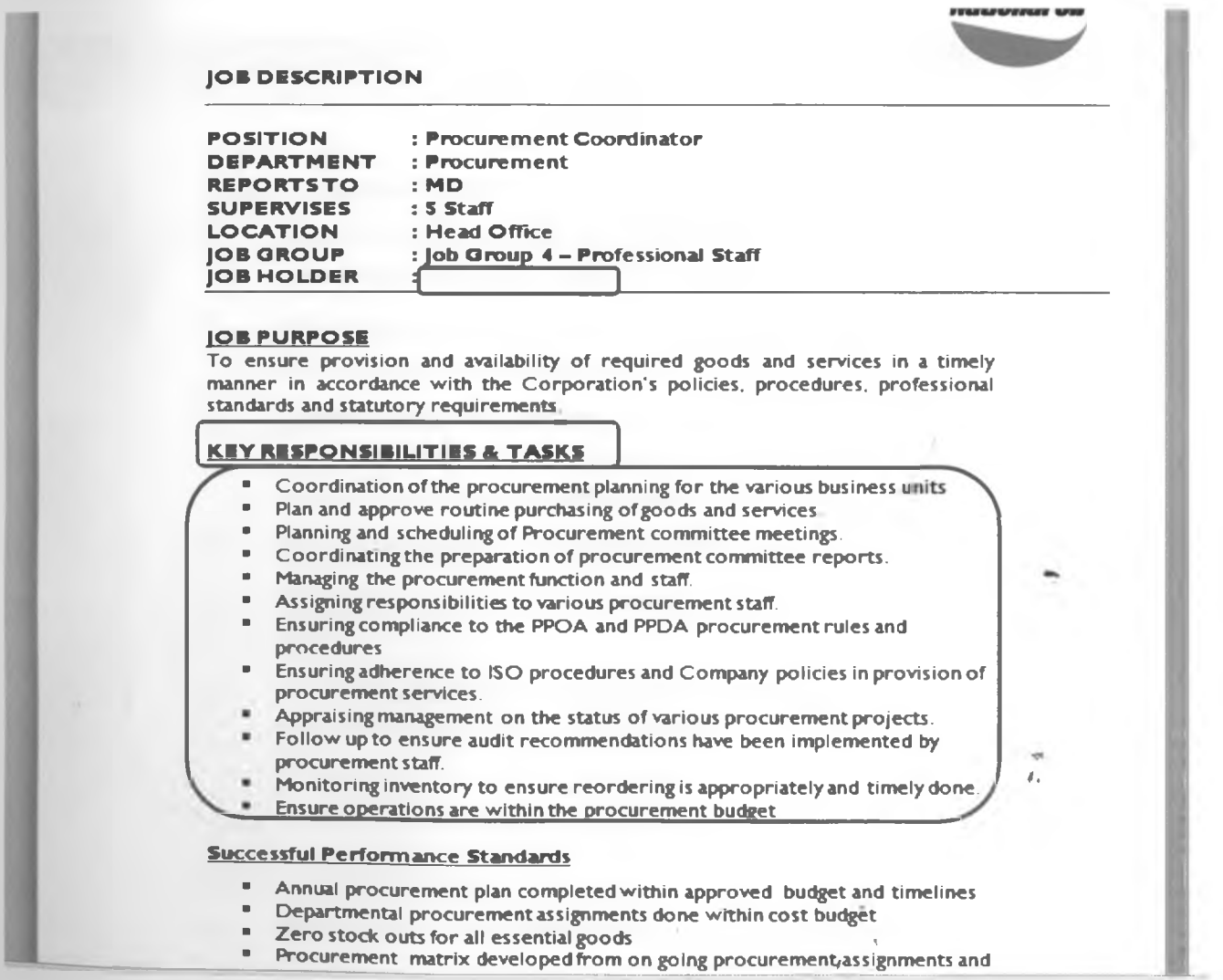
The feedback from 37 respondents showed that there was consistency between the job descriptions and their day to day duties and responsibilities. Based on this therefore the modeling was done for the Job descriptions because they represented the users opinion on their daily chores within the organization. For the two samples where there was no consistency between the feedback from the questionnaires and their job descriptions, modeling for the same was done separately and from the risk matrix they formed part of the three high risk areas which was to be escalated.

Questionnaires sent	Responses received	Percentage response rate
48	39	81%

Table 3.4: Questionnaires response rate analysis

Analysis of the Job descriptions.

This was one of the key sources of data for the research. From the sampled 48 system users, related job descriptions were obtained from Human resources department of the organization. This job descriptions were then compared with the feedback from the questionnaires sent to all the sampled system users. The objective was to establish the consistency between the two documents. Consistency was established for 37 respondents with the related job descriptions while 2 respondents were inconsistent with their job descriptions. For the 9 non responsive samples, the job descriptions were assumed to be consistent with their current duties and responsibilities. Below is a sample of the job description of a system user.



JOB DESCRIPTION

POSITION : Procurement Coordinator
DEPARTMENT : Procurement
REPORTS TO : MD
SUPERVISES : 5 Staff
LOCATION : Head Office
JOB GROUP : Job Group 4 – Professional Staff
JOB HOLDER :

JOB PURPOSE

To ensure provision and availability of required goods and services in a timely manner in accordance with the Corporation's policies, procedures, professional standards and statutory requirements.

KEY RESPONSIBILITIES & TASKS

- Coordination of the procurement planning for the various business units
- Plan and approve routine purchasing of goods and services
- Planning and scheduling of Procurement committee meetings.
- Coordinating the preparation of procurement committee reports.
- Managing the procurement function and staff.
- Assigning responsibilities to various procurement staff.
- Ensuring compliance to the PPOA and PPDA procurement rules and procedures
- Ensuring adherence to ISO procedures and Company policies in provision of procurement services.
- Appraising management on the status of various procurement projects.
- Follow up to ensure audit recommendations have been implemented by procurement staff.
- Monitoring inventory to ensure reordering is appropriately and timely done.
- Ensure operations are within the procurement budget

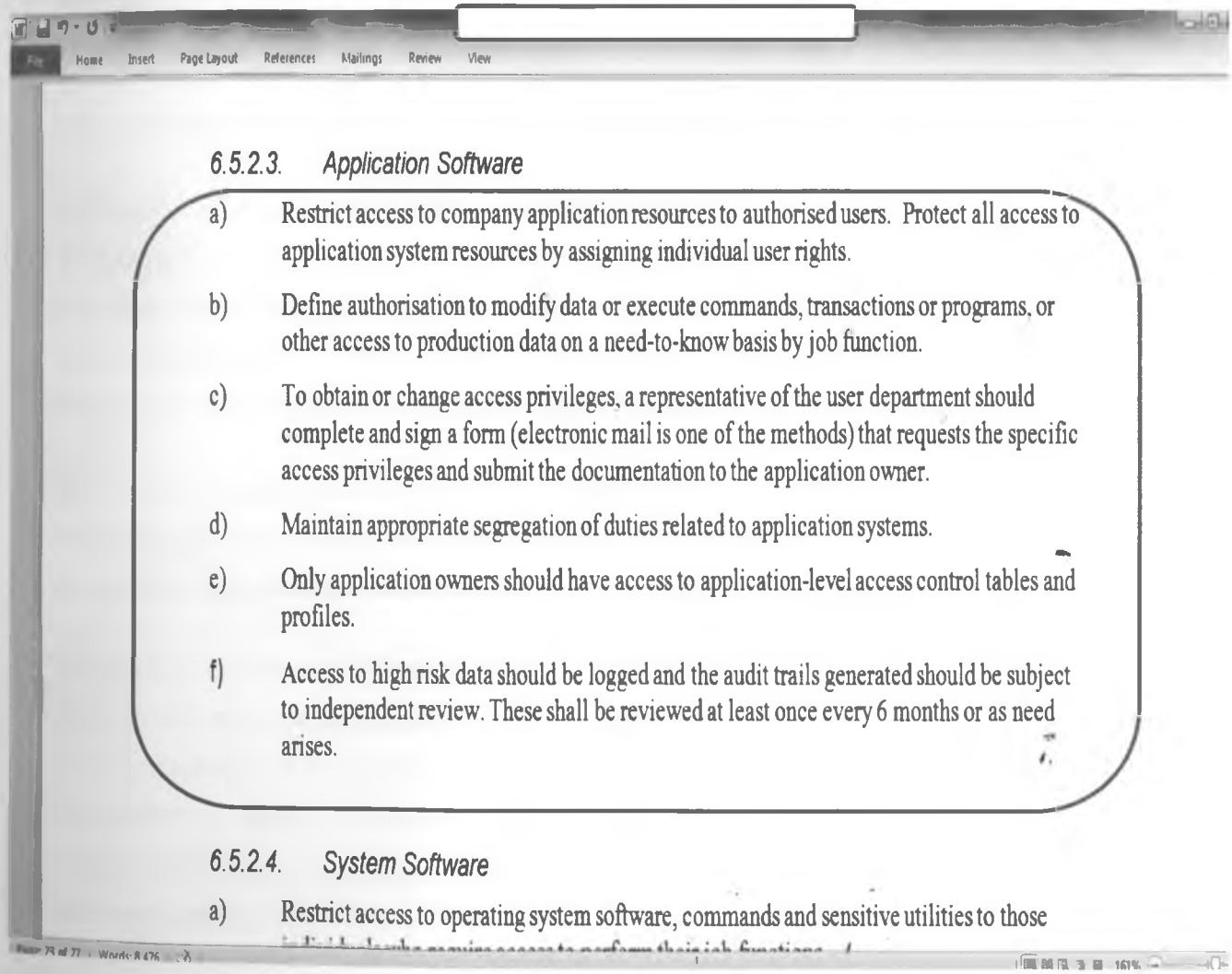
Successful Performance Standards

- Annual procurement plan completed within approved budget and timelines
- Departmental procurement assignments done within cost budget
- Zero stock outs for all essential goods
- Procurement matrix developed from on going procurement assignments and

Since the analysis at this stage showed that the job descriptions were consistent with users' daily roles and responsibilities with 95% to the affirmative, the same was therefore used for purposes of modeling and mapping into the various user responsibilities within the database application log.

Information Systems access policy analysis

A study was carried out of the organization's Information Security policy with regards to application access. The objective was to establish if there existed any rules, procedures or policies defining how system users should perform their duties. The following rules were obtained from the IS policy document page 23 and was relevant for purposes of this research;



Home Insert Page Layout References Mailings Review View

6.5.2.3. *Application Software*

a) Restrict access to company application resources to authorised users. Protect all access to application system resources by assigning individual user rights.

b) Define authorisation to modify data or execute commands, transactions or programs, or other access to production data on a need-to-know basis by job function.

c) To obtain or change access privileges, a representative of the user department should complete and sign a form (electronic mail is one of the methods) that requests the specific access privileges and submit the documentation to the application owner.

d) Maintain appropriate segregation of duties related to application systems.

e) Only application owners should have access to application-level access control tables and profiles.

f) Access to high risk data should be logged and the audit trails generated should be subject to independent review. These shall be reviewed at least once every 6 months or as need arises.

6.5.2.4. *System Software*

a) Restrict access to operating system software, commands and sensitive utilities to those

In analyzing application access with regards to the policy statements 6.5.2.3 a-f for purposes of modeling. the Ernest & Young SOD matrix was used following conclusions arrived at with respect to each policy requirement;

a. Restrict access to company application resources to authorized users. Protect all access to application system resources by assigning individual user rights:

   This policy requirement was already implemented by the organization as captured in the active users' log. Access to system resources was assigned to individual users as per policy requirements. For purposes of this research, the audit was based on what authorized users access in the application and this policy was treated as an overall benchmark for defining access to systems within the organization and was not therefore modeled.

b. Define authorization to modify data or execute commands, transactions or programs, or other access to production data on a need-to-know basis by job function.

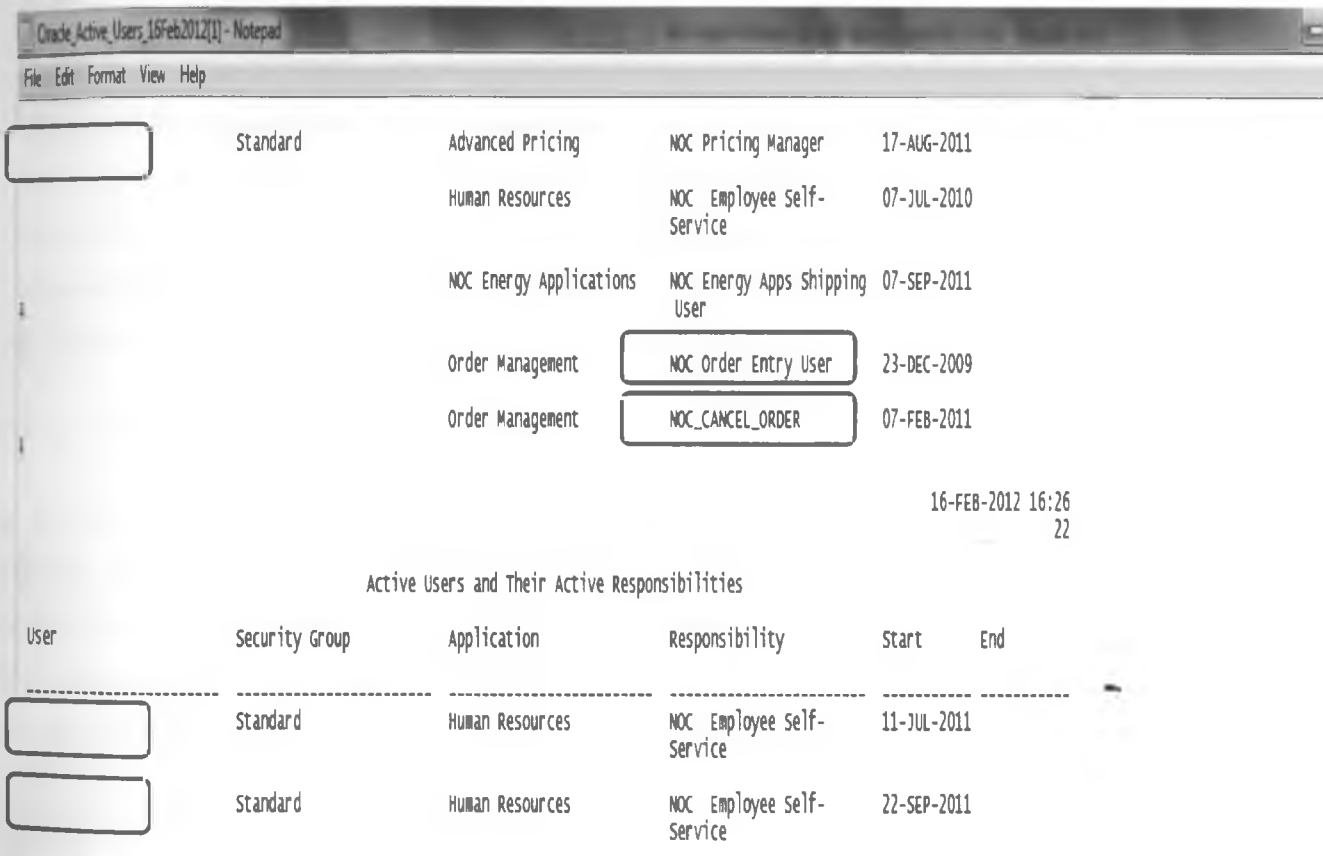
   This policy requirement was an integral part of the research as it captured the aspect of users being granted access based on their job function and need-to-know basis. It exclusively represents the results of analysis done earlier on the job descriptions and questionnaires. This therefore implies that in cases where the actual access is not in compliance with the JD then this policy is also violated. It was therefore modeled by the management agent.

c. To obtain or change access privileges, a representative of the user department should complete and sign a form (electronic mail is one of the methods) that requests the specific access privileges and submits the documentation to the application owner.

   This policy requirement defined the approval process before granting access to new system users as well as changing access privileges for existing users. It is what was entirely covered by the Approva model earlier discussed in the literature review and did not include a comparison of actual user access against the respective job functions. This was a pre-implementation and so was not modeled for purposes of this research. It was on this basisalso that some improvements were proposed on the initial conceptual model (Approva model).

d. Maintain appropriate segregation of duties related to application systems.

This policy requirement was independent of the outcome from analysis of the JD and feedback from questionnaires. It could more specifically isolated when auditing individual user logs from the application database. As illustrated below, the highlighted user was granted access to the same application twice both as an Order Entry User and NOC-CANCEL-ORDER. With this access, the user was able to enter an order, make price adjustments to the same, process and later cancel the same. This could be an avenue for fraud and so was captured as inappropriate segregation of duties.



Oracle_Active_Users_16Feb2012[1] - Notepad

File Edit Format View Help

[redacted]	Standard	Advanced Pricing	NOC Pricing Manager	17-AUG-2011
		Human Resources	NOC Employee Self-Service	07-JUL-2010
		NOC Energy Applications	NOC Energy Apps Shipping User	07-SEP-2011
		Order Management	NOC Order Entry User	23-DEC-2009
		Order Management	NOC_CANCEL_ORDER	07-FEB-2011

16-FEB-2012 16:26
22

Active Users and Their Active Responsibilities

User	Security Group	Application	Responsibility	Start	End
[redacted]	Standard	Human Resources	NOC Employee Self-Service	11-JUL-2011	
[redacted]	Standard	Human Resources	NOC Employee Self-Service	22-SEP-2011	

In modeling this therefore applications that aid a process were defined and grouped. Allocation of responsibilities within these applications was then monitored where any user with two or more responsibilities within an application was reported as a conflict in segregation of duties.

e. Only application owners should have access to application-level access control tables and profiles.

This policy requirement could be mapped to the JD's of the various application users and so with the analysis of the JD's and feedback from questionnaires this requirement was captured. Application level access control tables and profiles rested with IT database administrators and application developers. This policy was captured in the respective JD's of the DBA's

f. Access to high risk data should be logged and the audit trails generated should be subject to independent review. These shall be reviewed at least once every 6 months or as need arises.

The log of active users and their responsibilities is part of the audit trails run on the database application query. The whole purpose of this study was to ensure that this trail is complemented with compliance to policy requirements and users JD's. With this implementation the review could be done more regularly in a fast and efficient manner. This policy requirement therefore sums up the whole idea behind this research.

Analysis of Oracle database active users' log

The log of active users and their responsibilities was extracted from the Oracle E-business application as at 16th February 2012 for purposes of this research. This log represents the active user applications and responsibilities allocated to the various system users. Other information contained in the logs which was also significant includes the username, security group and the date responsibility was assigned to the user as well as termination date.

The detailed job description after justification of its viability through questionnaires was mapped in to the various user responsibilities within the application log. Where there existed a responsibility within an application which could not be mapped into any job description then a violation was reported.

The initial sample from the analysis reported a violation and screenshots below shows how the mapping was done:

Oracle_Active_Users_16Feb2012[1] - Notepad

File Edit Format View Help

16-FEB-2012 16:26
1

Active Users and Their Active Responsibilities

User	Security Group	Application	Responsibility	Start	End
[redacted]	Standard	CRM Foundation	CRM Administrator	15-NOV-2011	
		Financial Intelligence	Daily Receivables Intelligence	07-JUN-2011	
		General Ledger	NOC General Ledger User	14-OCT-2010	
		Human Resources	NOC Employee Self-Service	14-OCT-2010	
		NOC Energy Applications	NOC Energy Apps Shipping User	14-OCT-2010	
		Order Management	NOC Order Management Super User	14-OCT-2010	
		Receivables	NOC CREATE CUSTOMERS	14-OCT-2010	
		Receivables	NOC Receivables Manager	14-OCT-2010	
[redacted]	Standard	Human Resources	Global HRMS Manager	08-DEC-2011	
		Human Resources	NOC Employee Self-	19-JAN-2012	
[redacted]	Standard	Application Object Library	Application Diagnostics	05-JUL-2011	
		Human Resources	NOC Employee Self-Service	31-MAY-2011	
		Inventory	NOC Inventory User	31-MAY-2011	
		Purchasing	Purchasing Super User	05-JUL-2011	

Violation

le assigned
loyees
pply leave
payslips in

JOB DESCRIPTION

POSITION : **Procurement Coordinator**
DEPARTMENT : **Procurement**
REPORTSTO : **MD**
SUPERVISES : **5 Staff**
LOCATION : **Head Office**
JOB GROUP : **Job Group 4 – Professional Staff**
JOB HOLDER : [redacted]

JOB PURPOSE

To ensure provision and availability of required goods and services in a timely manner in accordance with the Corporation's policies, procedures, professional standards and statutory requirements.

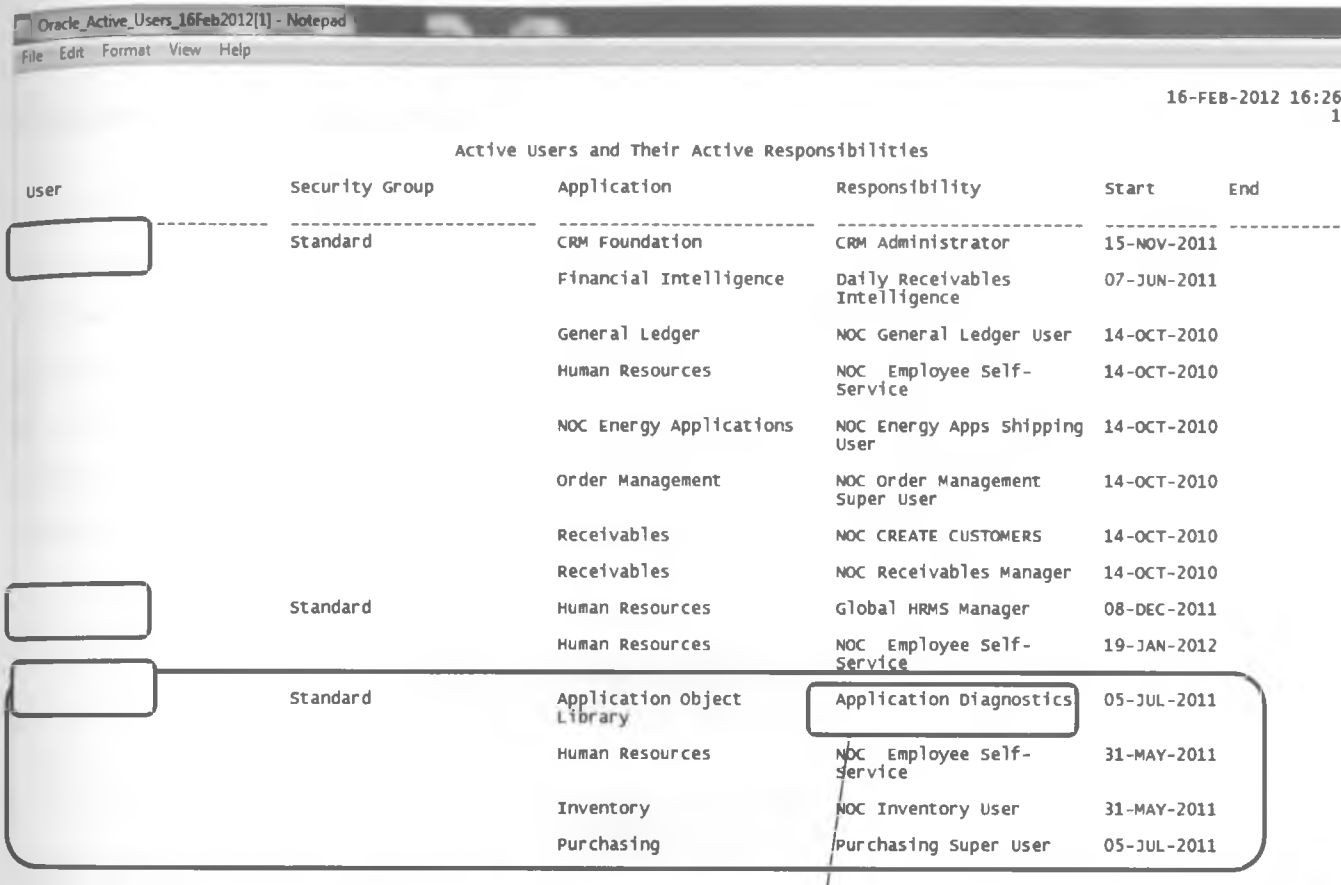
KEY RESPONSIBILITIES & TASKS

- Coordination of the procurement planning for the various business units
- Plan and approve routine purchasing of goods and services.
- Planning and scheduling of Procurement committee meetings
- Coordinating the preparation of procurement committee reports.
- Managing the procurement function and staff.
- Assigning responsibilities to various procurement staff.
- Ensuring compliance to the PPOA and PPDA procurement rules and procedures
- Ensuring adherence to ISO procedures and Company policies in provision of procurement services.
- Appraising management on the status of various procurement projects.
- Follow up to ensure audit recommendations have been implemented by process owners.
- Monitoring inventory to ensure reordering is appropriately and timely done.
- Ensure operations are within the procurement budget

Successful Performance Standards

- Annual procurement plan completed within approved budget and timelines
- Departmental procurement assignments done within cost budget
- Zero stock outs for all essential goods
- Procurement matrix developed from on going procurement assignments and

The log was also mapped with the IS policy on access to applications and the following shows how the mapping occurred.



Oracle_Active_Users_16Feb2012[1] - Notepad

File Edit Format View Help

16-FEB-2012 16:26
1

Active Users and Their Active Responsibilities

User	Security Group	Application	Responsibility	Start	End
	Standard	CRM Foundation	CRM Administrator	15-NOV-2011	
		Financial Intelligence	Daily Receivables Intelligence	07-JUN-2011	
		General Ledger	NOC General Ledger User	14-OCT-2010	
		Human Resources	NOC Employee Self-Service	14-OCT-2010	
		NOC Energy Applications	NOC Energy Apps Shipping User	14-OCT-2010	
		Order Management	NOC Order Management Super User	14-OCT-2010	
		Receivables	NOC CREATE CUSTOMERS	14-OCT-2010	
		Receivables	NOC Receivables Manager	14-OCT-2010	
	Standard	Human Resources	Global HRMS Manager	08-DEC-2011	
		Human Resources	NOC Employee Self-Service	19-JAN-2012	
	Standard	Application Object Library	Application Diagnostics	05-JUL-2011	
		Human Resources	NOC Employee Self-Service	31-MAY-2011	
		Inventory	NOC Inventory User	31-MAY-2011	
		Purchasing	Purchasing Super User	05-JUL-2011	

6.5.2.3. Application Software

a) Restrict access to company application resources to authorised users. Protect all access to application system resources by assigning individual user rights.

b) Define authorisation to modify data or execute commands, transactions or programs, or other access to production data on a need-to-know basis by job function.

c) To obtain or change access privileges, a representative of the user department should complete and sign a form (electronic mail is one of the methods) that requests the specific access privileges and submit the documentation to the application owner.

d) Maintain appropriate segregation of duties related to application systems.

e) Only application owners should have access to application-level access control tables and profiles.

f) Access to high risk data should be logged and the audit trails generated should be subject to independent review. These shall be reviewed at least once every 6 months or as need arises.

From this mapping outcome it was realized that whenever there was a conflict between the JD and the application log for a system user then the same would be reflected in the IS policy on application software 6.5.2.3 parts b & d.

Risk matrix (Reporting Framework)

The risk matrix was used as the reporting framework for this model. The matrix was grouped into three categories; high risk, medium risk and low risk. This was represented by a single star for low risk, double star for medium risk and triple star for high risk. The high risk represented potential areas for fraud which needed immediate attention and further forensic audit to be done to establish whether the applications had been abused. Medium risk represented areas with both application and policy violations which need to aligned for compliance while low risk represented areas which also needed to be corrected to seal future loop holes.

Risk	Color	Star
Low		×
Medium		××
High		×××

Table 3.5: Reporting framework

3.3 Design

3.3.1Multi-Agent design

The research was guided by the Tropos methodology. This methodology was chosen because it supports an incremental development through iteration which was needed for this model to support the various systems, applications and networks. Actors, goals, tasks resources and social dependency between actors was established.The design was based on two key ideas: First, the notion of agent and all related mentalistic notions for instance goals and plans which were used in all phases of software development, from early analysis down to the actual implementation. Second, it also covered the very early phases of requirements analysis, thus allowing for a deeper understanding of the environment where the software was to operate and the kind of interactions that was to occur between software and human agents.

Advantages of the Tropos methodology.

- Pays attention to activities that precede the specification of the prescriptive requirements like understanding how and why the intended system would meet the organizational goals.
- Deals with all phases of system requirements analysis and all phases of system design and implementation in a uniform and homogenous way based on common mentalistic notions as those of actors, goals, soft goals, plans, resources and intentional dependencies.
- This methodology rests on the idea of building a model of the system-to-be that is incrementally refined and extended from a conceptual level to executable artifacts by means of a sequence of progressive transformational steps.
- It supports the Belief, Desire and Intention (BDI) agent nature for programming intelligent agents thus ensuring reduced time in agent decision making.
- The methodology supports verification and validation of the model implemented and so is scientifically verifiable.
- It supports a top down development approach which comes with its relative advantages.

Drawbacks of the Tropos methodology.

- The iteration if not controlled can lead to a never ending incremental loop.
- Requires specialized knowledge and expertise.

3.3.2 Early Requirements (Organizational model): Here the organizational setting was analyzed. The interest was not in describing the system-to-be, but just the most relevant actors and their relationships in the domain where the system will operate. This stage was concerned with the understanding of the problem by studying an organizational setting and the output was an organizational model which included the relevant actors, their goals and dependencies.

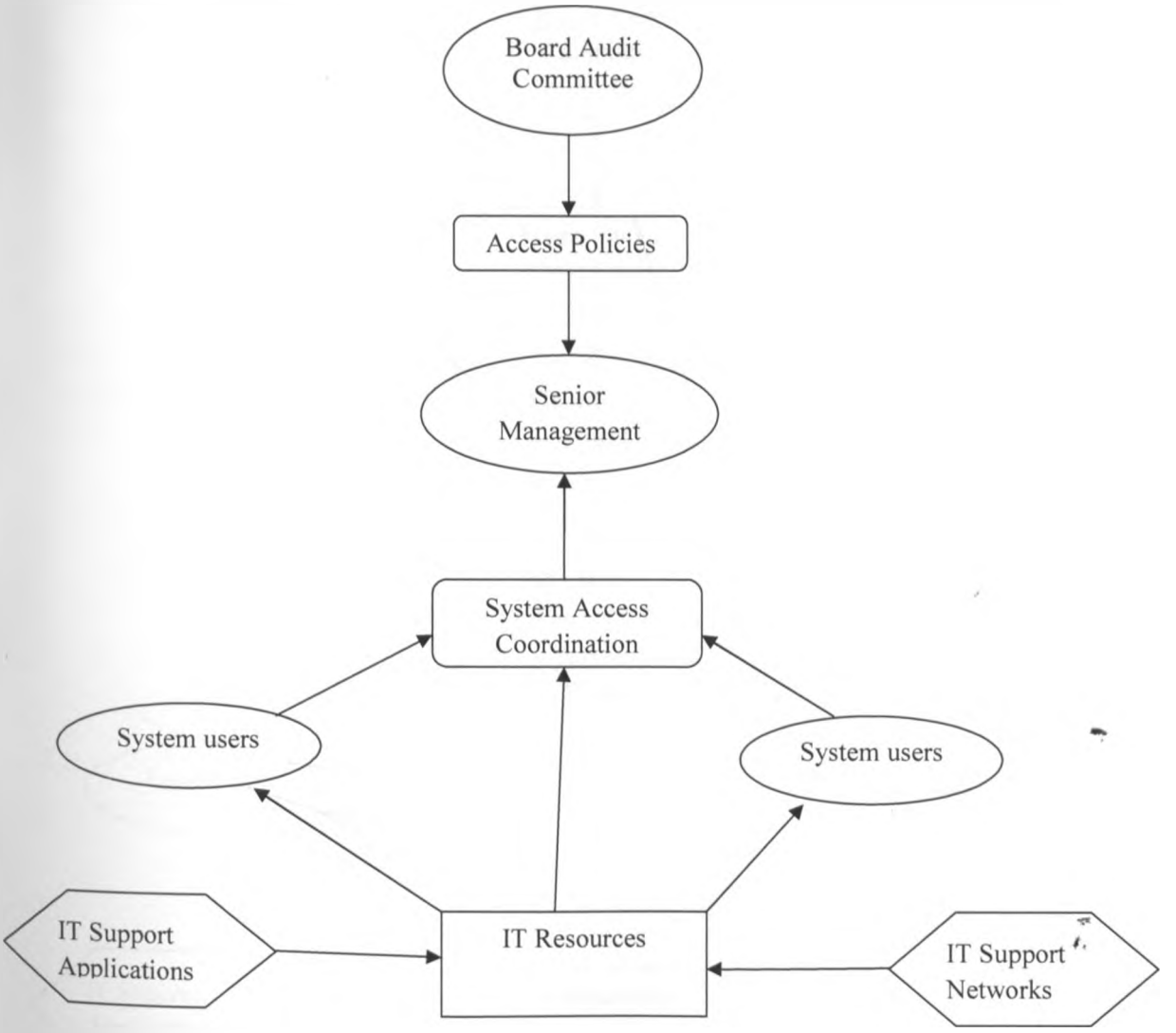


Figure 3.1: The organizational model

3.3.3 Late Requirements

At this stage, the focus was on the system-to- be within its operating environment. The system-to-be was introduced as another actor related to stakeholder actors in terms of actor dependencies; these indicated the obligations of the system towards its environment and what the system could expect from actors in its environment. It was described within its operational environment, along with relevant functions and qualities.

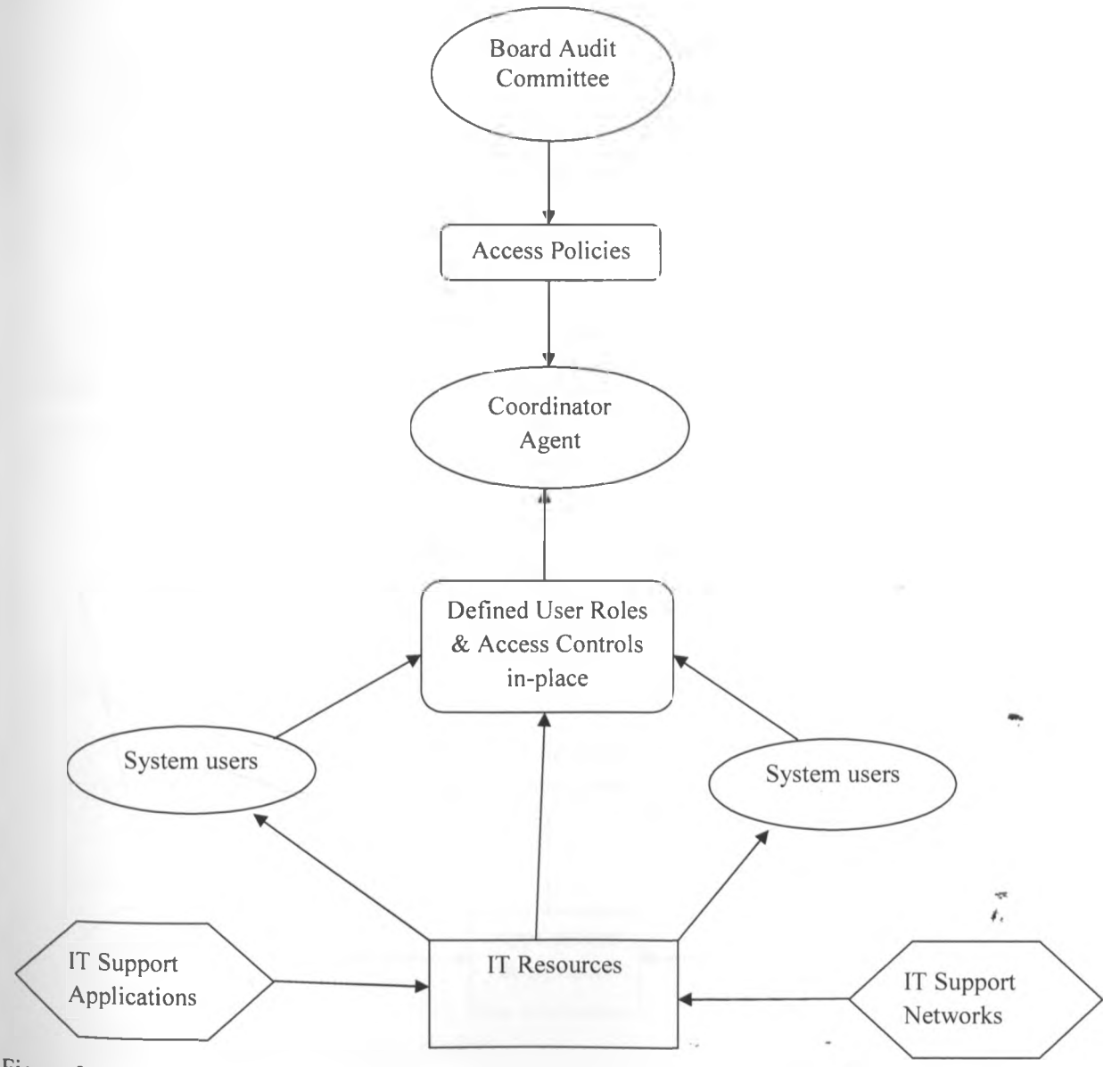


Figure 3.2: The operational environment

3.3.4 Architectural Design

A total of four actors were introduced and assigned goals or subtasks of the goals.

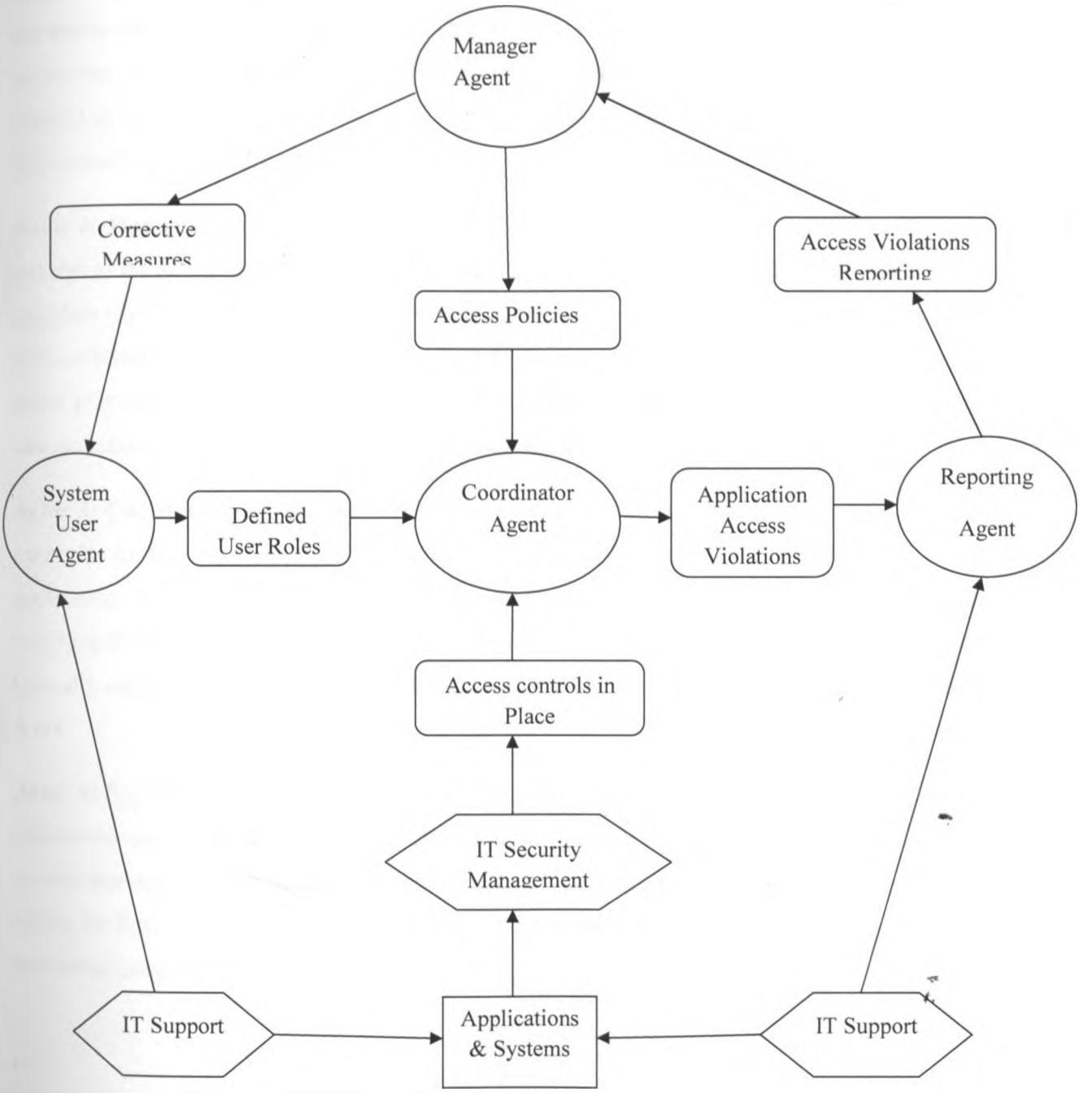


Figure 3.3: The Architectural design

The four actors were namely: system users, coordinator agent, reporting agent and management/ Board audit committee.

Actor 1: System User Agent The main goal of this agent was to define user roles as mapped in the active users log from the application database. This agent would submit to the coordinator agent what it deems to be the actual responsibilities of the system user within the application as defined in the Job description. This would then enable the coordinator agent to make comparison with actual log to establish whether there is a conflict.

Actor 2: Management Agent The main goal of this agent was to map application access policies to the active users log from the database through the coordinator agent in order to establish existence of conflicts within the same. Other sub-goals included performing real time updates based on access violation reports from the reporting agent through corrective actions either granting or deletion of some access rights to enable compliance. This agent also defines new policies on access to be monitored by the model.

Actor 3: Coordinator Agent The main goal of this agent is to facilitate the mapping of defined user roles from the system user's agent and the active users log from the Oracle database application for queried users to establish consistency and report conflicts. It also facilitates mapping of the same log with the application access policy from management agent. This agent basically defines conflicts where there is a mismatch in the mapping and consistency when there is not.

Actor 4: Reporting Agent The main goal of this agent is to report violations captured by the coordinator agent. This violations are reported to the management agent who in turn instructs the system user agent to correct the violations reported. It accomplishes its task by continuously asking for feedback based on submitted query to the coordinator agent. It reports both violations and compliance to the management agent.

3.3.5 Database Design

MySQL database was used to capture the various parameters such job descriptions, user access profiles as well as user logs from the Oracle application.

MySQL Database

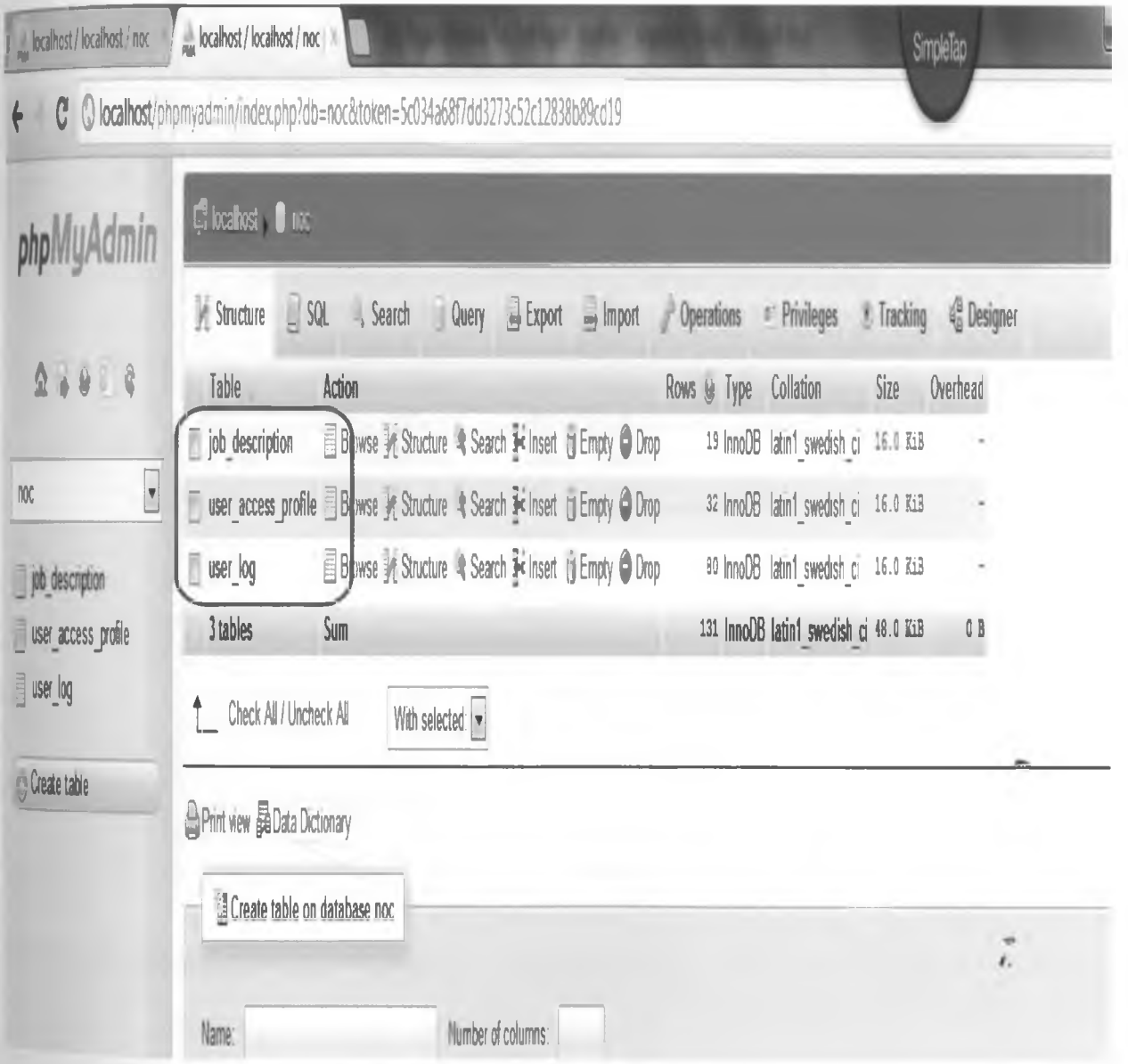


Figure 3.4: The database design

Database shot: Job Description

This defined the access rights as captured from the analysis of the job descriptions as per system user ID's.

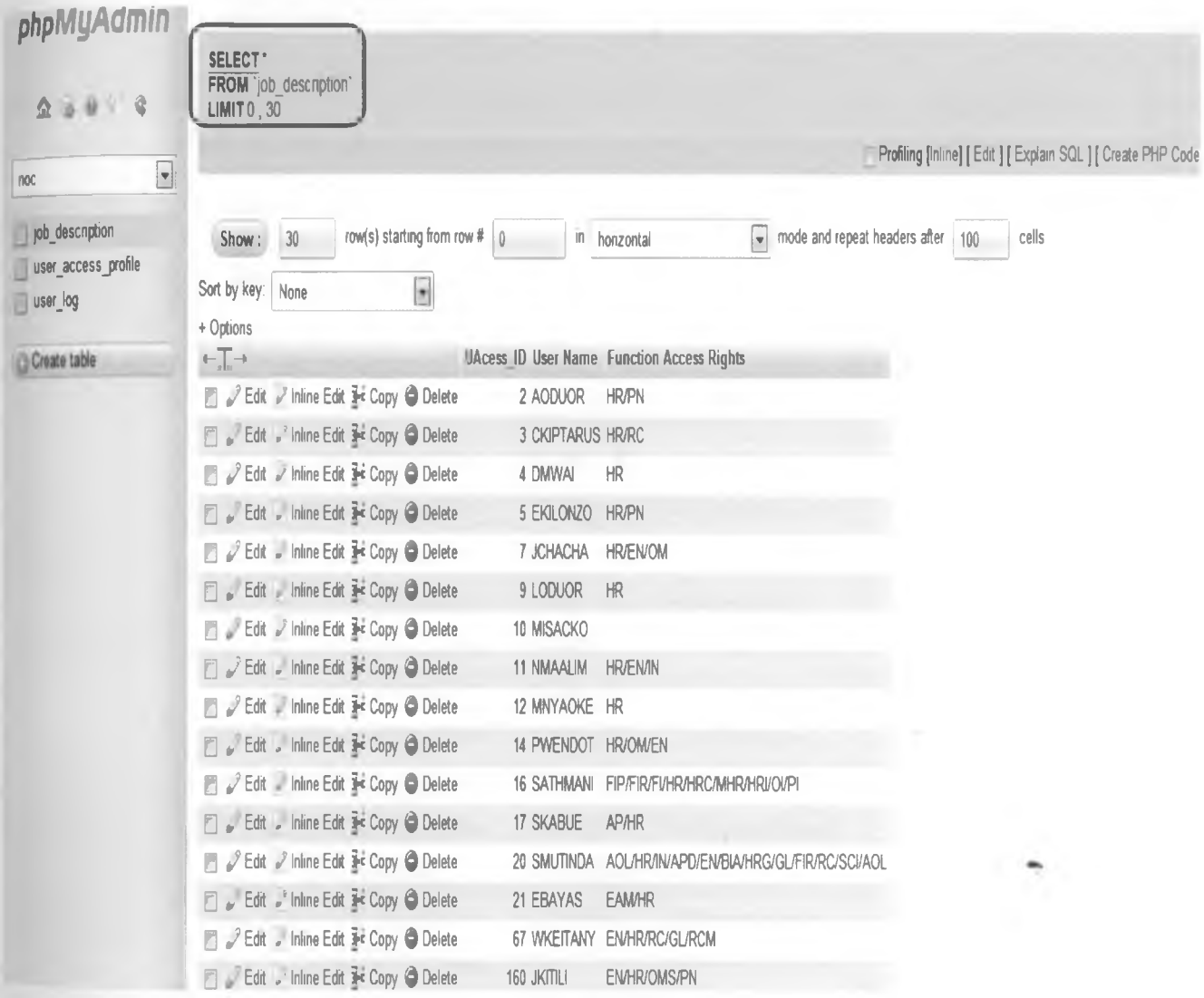
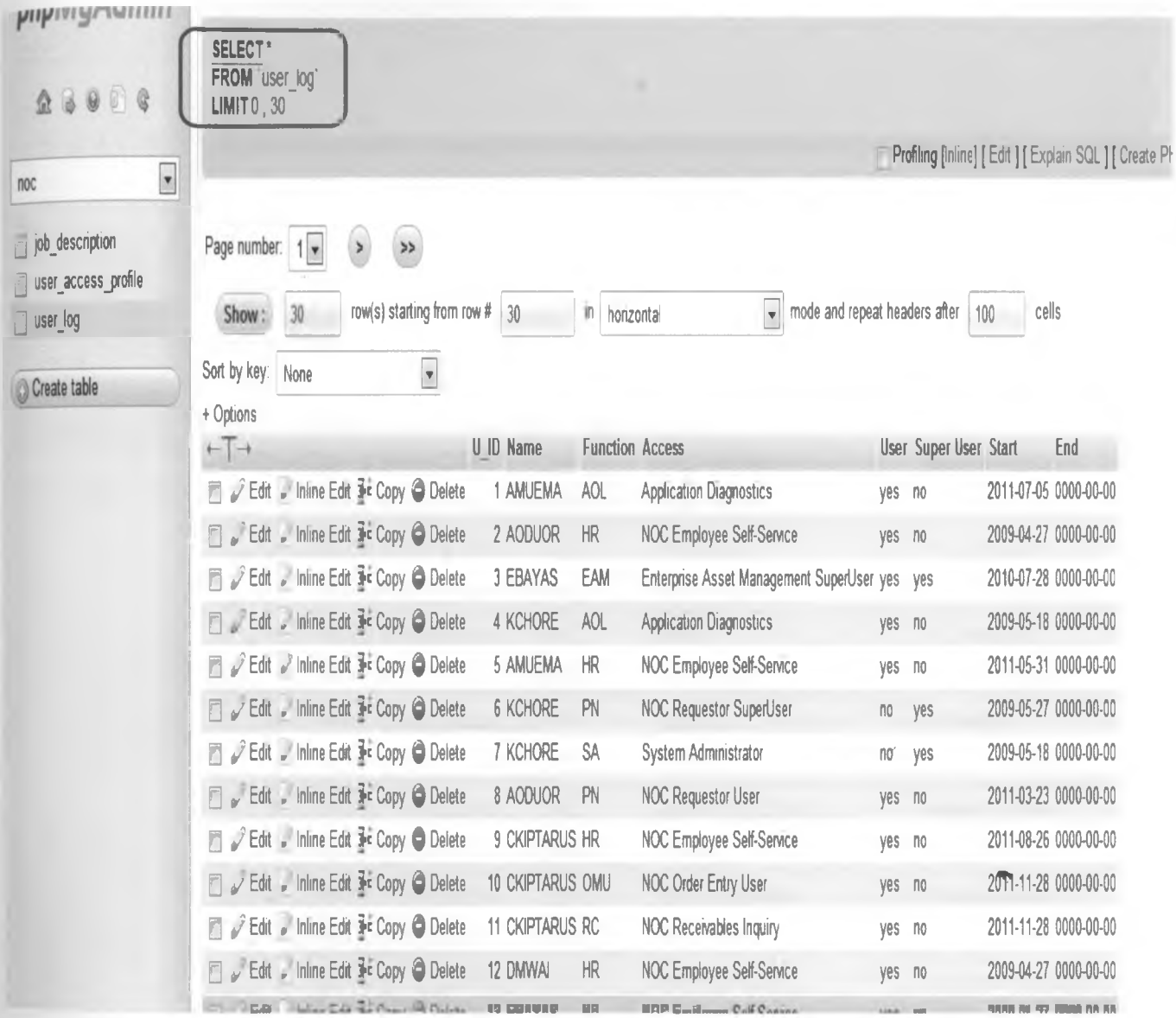


Figure 3.5: Job description parameters

Database shot: User_Log

This defined the system users' actual access as captured from the active users' responsibility log in the Oracle application.



	U_ID	Name	Function	Access	User	Super User	Start	End
Edit Inline Edit Copy Delete	1	AMUEMA	AOL	Application Diagnostics	yes	no	2011-07-05	0000-00-00
Edit Inline Edit Copy Delete	2	AODUOR	HR	NOC Employee Self-Service	yes	no	2009-04-27	0000-00-00
Edit Inline Edit Copy Delete	3	EBAYAS	EAM	Enterprise Asset Management SuperUser	yes	yes	2010-07-28	0000-00-00
Edit Inline Edit Copy Delete	4	KCHORE	AOL	Application Diagnostics	yes	no	2009-05-18	0000-00-00
Edit Inline Edit Copy Delete	5	AMUEMA	HR	NOC Employee Self-Service	yes	no	2011-05-31	0000-00-00
Edit Inline Edit Copy Delete	6	KCHORE	PN	NOC Requestor SuperUser	no	yes	2009-05-27	0000-00-00
Edit Inline Edit Copy Delete	7	KCHORE	SA	System Administrator	no	yes	2009-05-18	0000-00-00
Edit Inline Edit Copy Delete	8	AODUOR	PN	NOC Requestor User	yes	no	2011-03-23	0000-00-00
Edit Inline Edit Copy Delete	9	CKIPTARUS	HR	NOC Employee Self-Service	yes	no	2011-08-26	0000-00-00
Edit Inline Edit Copy Delete	10	CKIPTARUS	OMU	NOC Order Entry User	yes	no	2011-11-28	0000-00-00
Edit Inline Edit Copy Delete	11	CKIPTARUS	RC	NOC Receivables Inquiry	yes	no	2011-11-28	0000-00-00
Edit Inline Edit Copy Delete	12	DMWAI	HR	NOC Employee Self-Service	yes	no	2009-04-27	0000-00-00

Figure 3.6: System user access log parameters

Database shot: User Access Profile

This defined the various user profiles for the sampled users and their respective function codes.

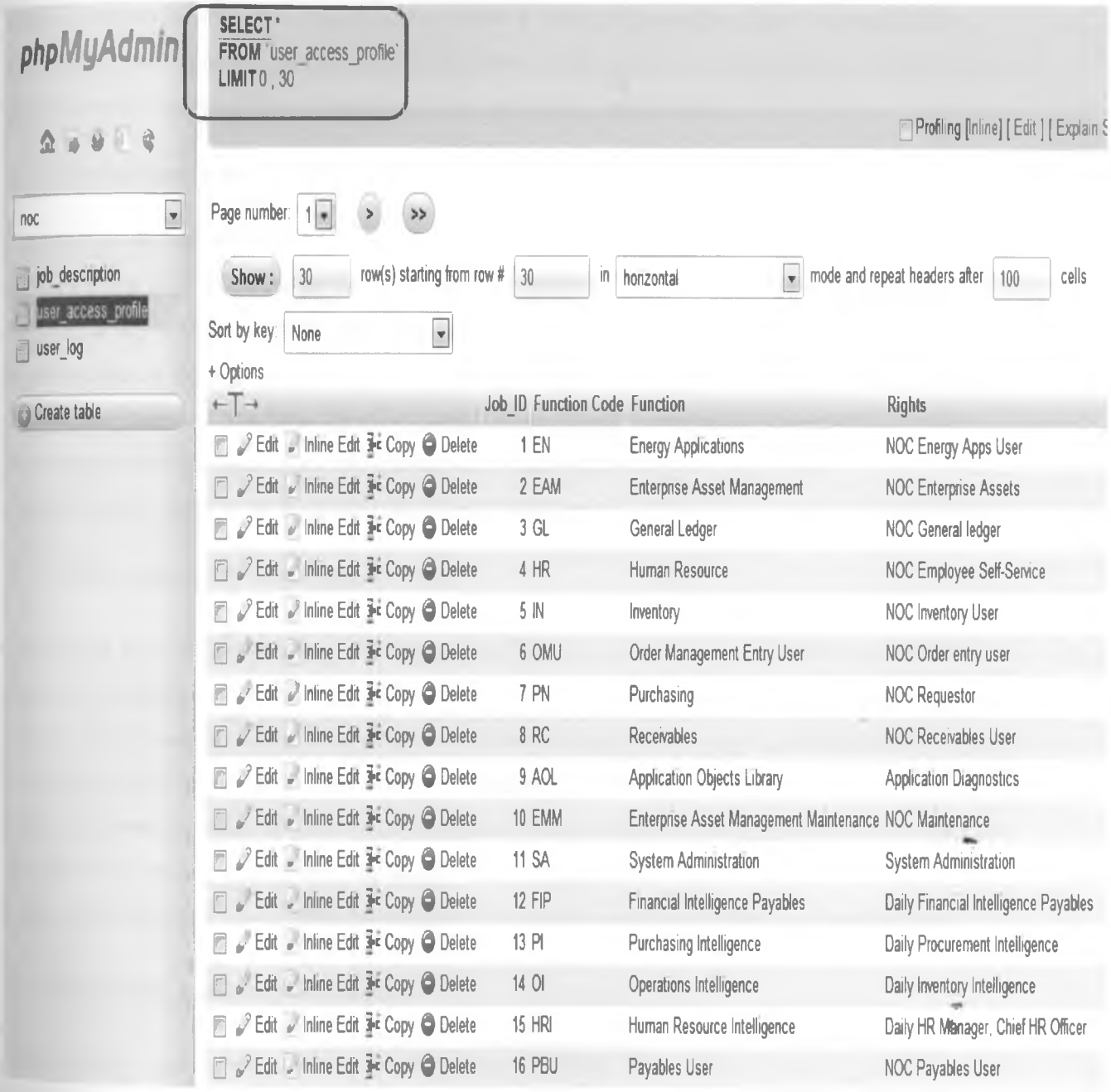


Figure 3.7: Use access profile parameters

CHAPTER 4 IMPLEMENTATION AND RESULTS

4.1 Agent implementation

Prior to implementation, the system was tested as a whole. Agent interaction, communication mechanism and reporting was tested and since the system was not to be deployed in a production environment, it was subjected to scenario data and results evaluated

4.1.1 System user agent implementation

This agent was implemented as an initiator of several processes including the actual access as defined in the SQL database as well as the defined user roles and responsibilities as captured from the analyis of users job descriptions and feedback from questionaires. For purposes of implementation this agent did a comparision of the actual user access to the application against the defined roles and responsiilities for a specific user when prompted by the reporting agent. The output from this agent was also submitted to management agent for purposes of establishing compliance or violation of the access policies as modeled.

4.1.2 Management agent implementation

This agent was implemented for two purposes; first to ensure the applications accessed by the users are compliant with the organizations IS policy on application access and report violations incase of non compliance , secondly to facilitate updates to the users job descriptions when required to ensure reported violations are addressed. Through this agent therefore, whenever violations were reported, the user was prompted if they wished to make updates to the users job description to ensure compliance. If the users accepted this then they were allowed to incoporate some of the applications reported as violations to the users job description so at to complay.

4.1.3 Reporting agent implementation

The reporter agent provided the graphic user interface from where several prompts were addressed. The main duty of this agent was to capture input and generate an output of either violation or non violation. The GUI interface for the reporter was used to capture the username whose access rights was to be audited and to generate a report on violation based on the very

user. This agent also generated a GUI for updating the users job description incases of reported violations which were to be corrected to ensure compliance. The agent could generate a sequence of access violation reports based on the data input.

4.1.4 Coordinator agent implementation

This agent was the coordinator of all operations within the system. It defined the sequence in which the other three agents were to be called into action. All activities were executed at the call of the coordinator agent. The application violations established by the system user agent were reported by the reporter agent at the call by the coordinator agent. In determining violations, the comparisons made between the actual access and the defined user roles and responsibilities in the SQL database was through the coordinator agent.

The coordinator agent ensured proper inter-agent communication and competition as well as coordination for purposes of realizing effective audit and reporting of user access violations. The coordinator agent prompted the initialization of the reporter agent and ensured consistency in reporting based on first in first out basis. This agent also initiated the management agent's actions with regards to compliance to access policies as well as updates to user job descriptions for compliance on reported application violations.

The coordinator was at the center of all agents operation; issuing instructions to the other agents on what to execute and in which order so as to avoid collusion. It was also responsible for the termination of other agents' actions after successful execution of assigned tasks. When calling the agents into action, it defined the sequence of execution of the tasks. It was the center of all agents' activities.

4.2 Results from the prototype

4.2.1 Calling the agents into action

This is the initialization process where all agents and agent libraries are called into action in jade.

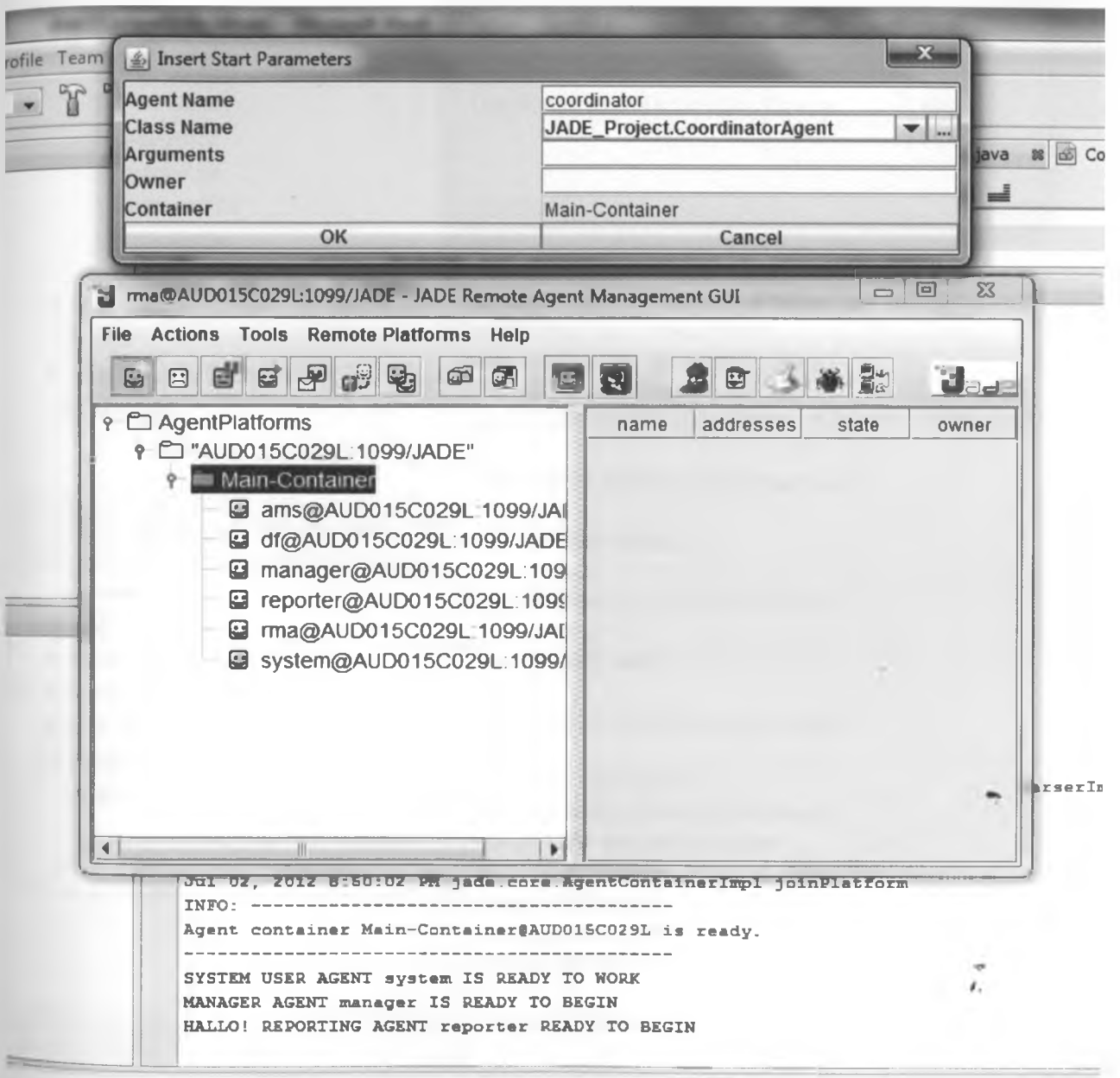


Figure 4.1: Calling agents into action

4.2.2 Agents input

This shows the prompt generated by the reporter agent when called into action.

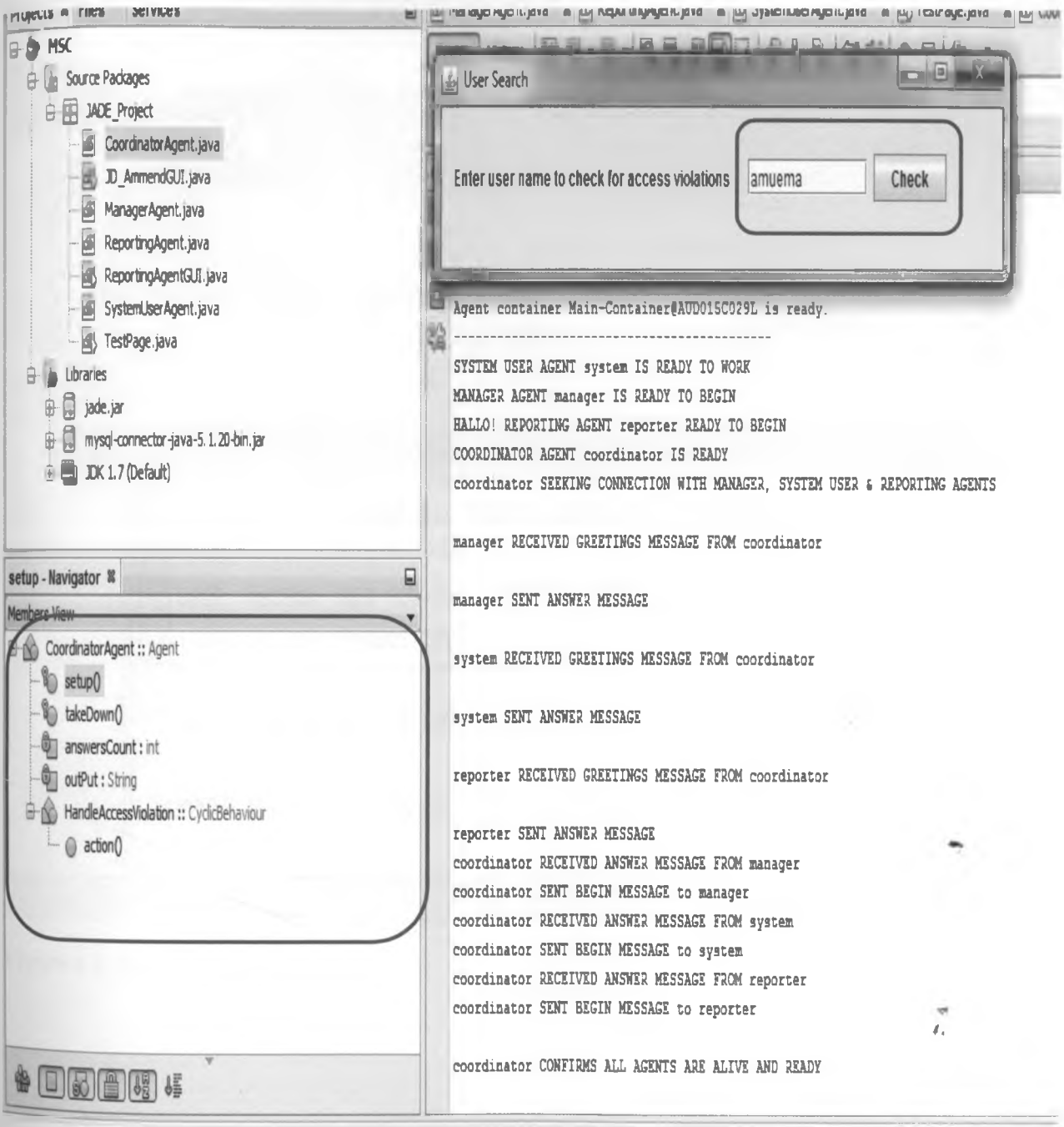


Figure 4.2: Prompts generated when agents are called

4.2.3 Agents output on access violation reporting

The output was generated in a specific format with the username coming first followed by the risk level, number of violations, policies violated in case of any and the particular application violated. Sample access violation report shown below:

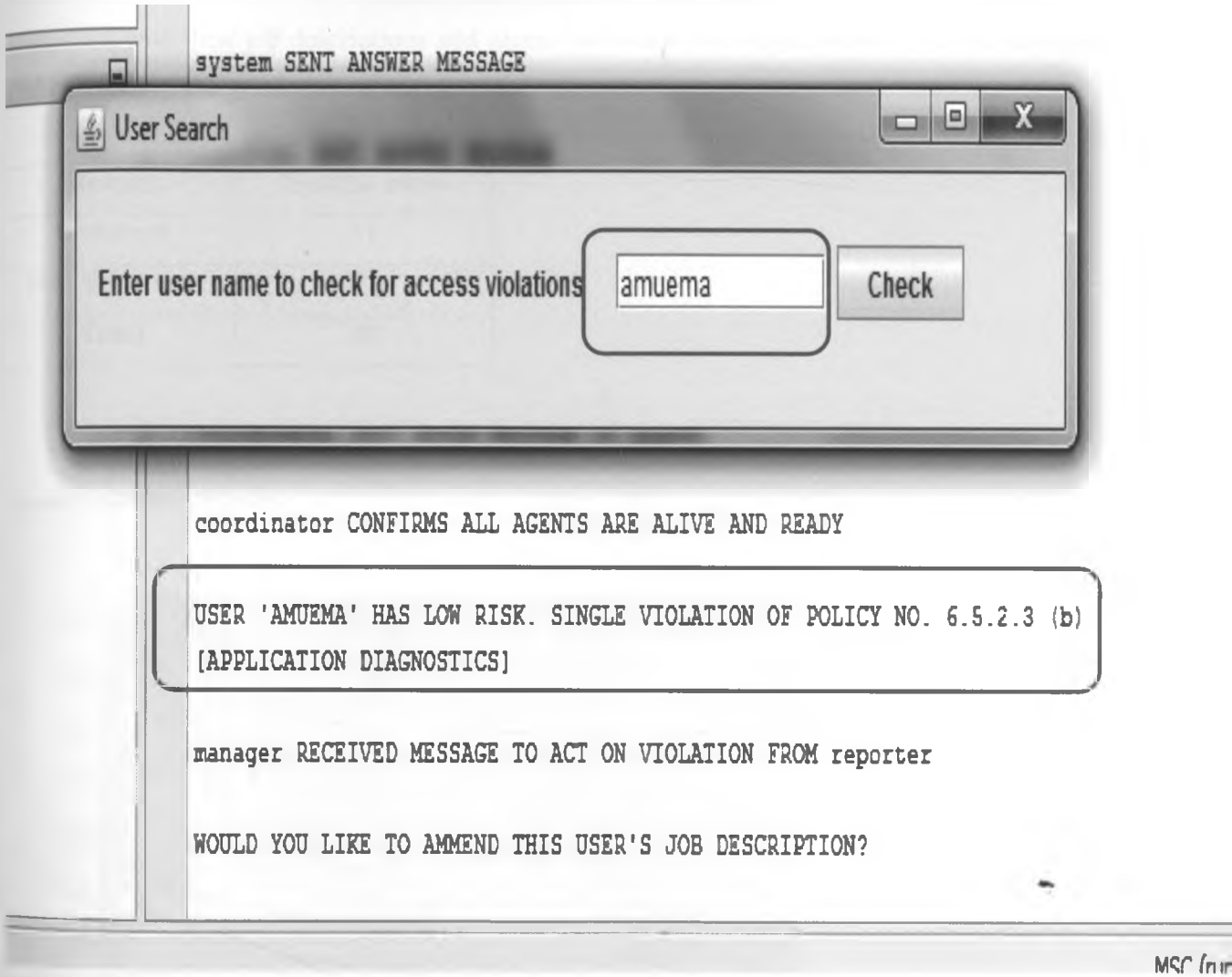


Figure 4.3: Violation reporting

4.3 Model results

4.3.1 Results based on sampled system users

From the total number of 48 sampled system users, 11 reported access violations while 37 were complaint with their job descriptions and access policies of the organization. This was analyzed and captured in the graph shown below.

Results	System users
Violations	11
Non Violations	37
Total	**48**

Access violation results

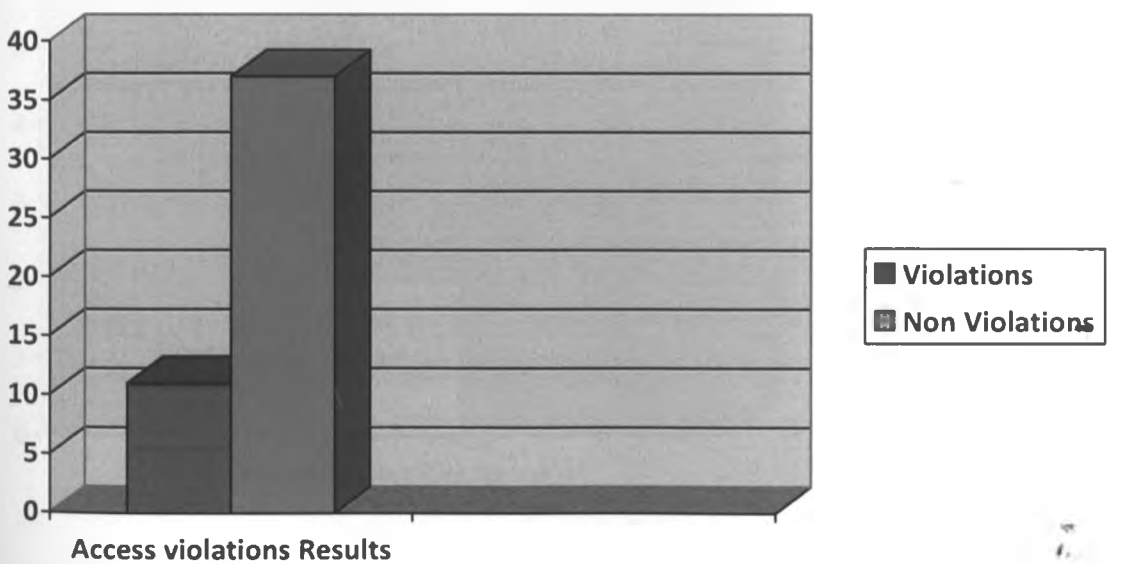


Figure 4.4: Graphical representation of violation results

4.3.2 Results of access violations based on risk matrix

On further analysis of the 11 reported violations, it was established based on the risk matrix that six cases were low risk; two cases were medium risk while three cases were high risk. The same was tabulated and results represented graphically as shown below.

Violations	Risk Matrix
6	Low
2	Medium
3	High
11	**Total**

Access violations based on the risk matrix

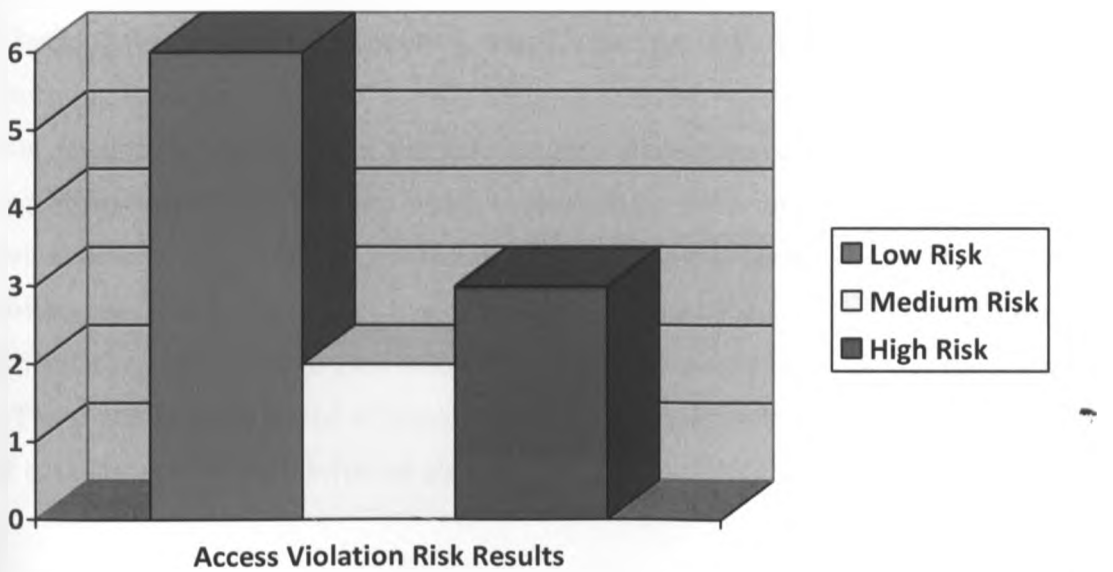


Figure 4.5: Graphical representation of violation results based on the risk matrix

4.4 Model Testing

4.4.1 Agent Level Testing

Agent testing was done incrementally during software development. During development, each agent's functionality was tested. Much of this testing required another agent to trigger an event inside the agent to be tested, such as message from another agent. An important issue related to the bounds of testing MAS is that it is not possible (or very expensive) to predict the agent behavior. Therefore, when developing a single agent for inclusion into the community of four agents, it was necessary to make sure that it responded correctly to given inputs from other agents.

Some of the errors types which occurred while testing an agent included:

a) Incorrectly addressing a message to another agent
b) Putting an incorrect request in a message so the receiving agent does not recognize the message
c) Incorrectly parsing incoming messages
d) Checking for the wrong performative, which is the type of the communicative act, in an incoming message
e) Not developing code to accept all the messages it was supposed to accept.

The agent was tested as a black-box which focused on the behavior of the agents. Testing its behavior corresponds to begin an interaction and evaluating its result. The agent was also tested as a white-box, which corresponds to test the agent internal behaviors, which is the same of how behaviors are related and their flow of control, where each single behavior was seen as a black-box. The result therefore a kind of black-box testing of a subsystem (the agent composed of the agent base class and several behavior classes)

Communicative Agent ACLMessage Failure: agent sending message to another agent with the wrong receiver aid name expressed in code and Jade screenshot below: mssg.addReceiver (new AID("ADMIN",AID.ISLOCALNAME));

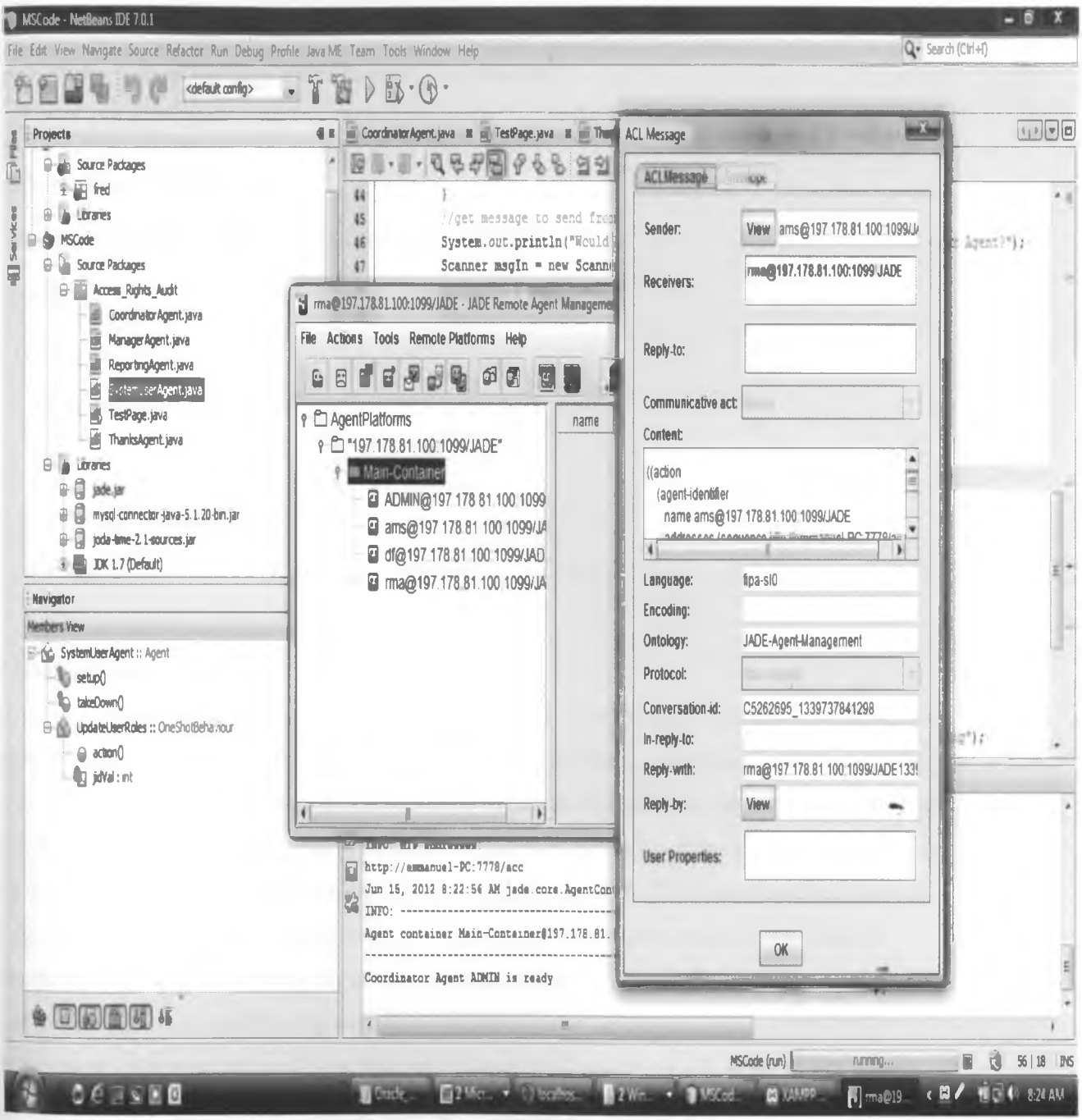


Figure 4.6: Inter agent communication failure

JADE ACLMessage communication error message shown below:

```
((action
   (agent-identifier
    :name ams@197.178.81.100:1099/JADE
    :addresses (sequence http://emmanuel-PC:7778/acc))
   (create-agent
    :agent-name ADMIN
    :class-name Access_Rights_Audit.SystemUserAgent
    :container
     (container-ID
      :name Main-Container
      :protocol JADE-IMTP
      :address "<Unknown Host>"
      :protocol JADE-IMTP))) already-registered)
```

4.4.2 Society Level Testing

Testing the community of four agents involved two issues. The first one was how to ensure that the agents in the community work together as designed previously, which again was related to the problem defined. And the second one was how to ensure that the resultant work was the one expected. During the society test, the validation of the overall results of the different agents was done and the successful integration of the different agents verified. This involved checking that each agent in the society received the correct messages from the correct agent, provided the correct responses, and interacted with environment correctly. It also involved checking that the goal of the community where the agents were interacting was being achieved.

1. Some types of errors that could be observed during the developing of a community were; miscommunicating the performative (the type of the communicative act)
2. Content on agent messages

3. Designing deadlocks into the messages exchanges
4. Another issue that had to be considered was the scalability. The larger the agent communities became, the harder it was to test them for proper functionality.

Using traditional tools for debugging agent societies was insufficient as it lacked efficiency and was inadequate especially because the multi-agent system was reactive. Generally, visualizing overall system behavior in systems with distributed control was a difficult task. Each agent in the system has only a local view of the organization, and the burden on the user to integrate into a coherent whole the large amounts of scope-limited information provided by individual agents.

The model below was adopted for purposes of both the functional and the system testing. Test scenarios listed in the agent and society level testing was adopted.

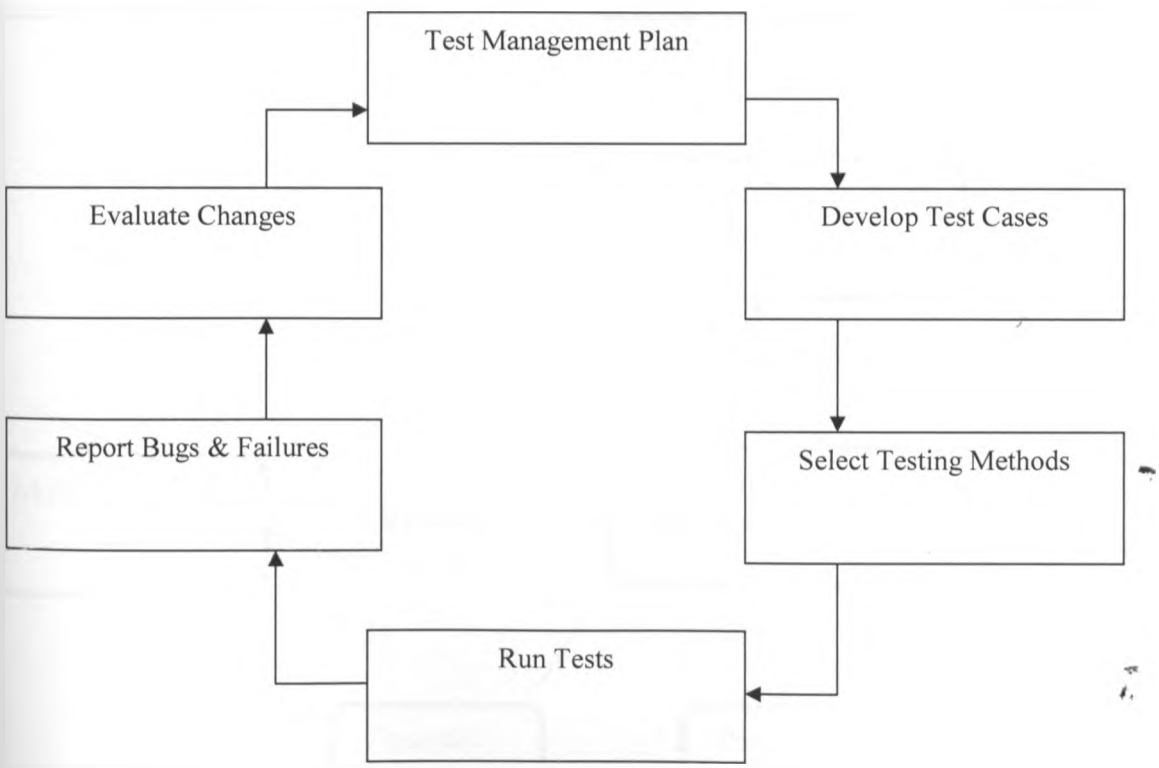


Figure 4.7: The testing model

4.5 Model for user access rights audit

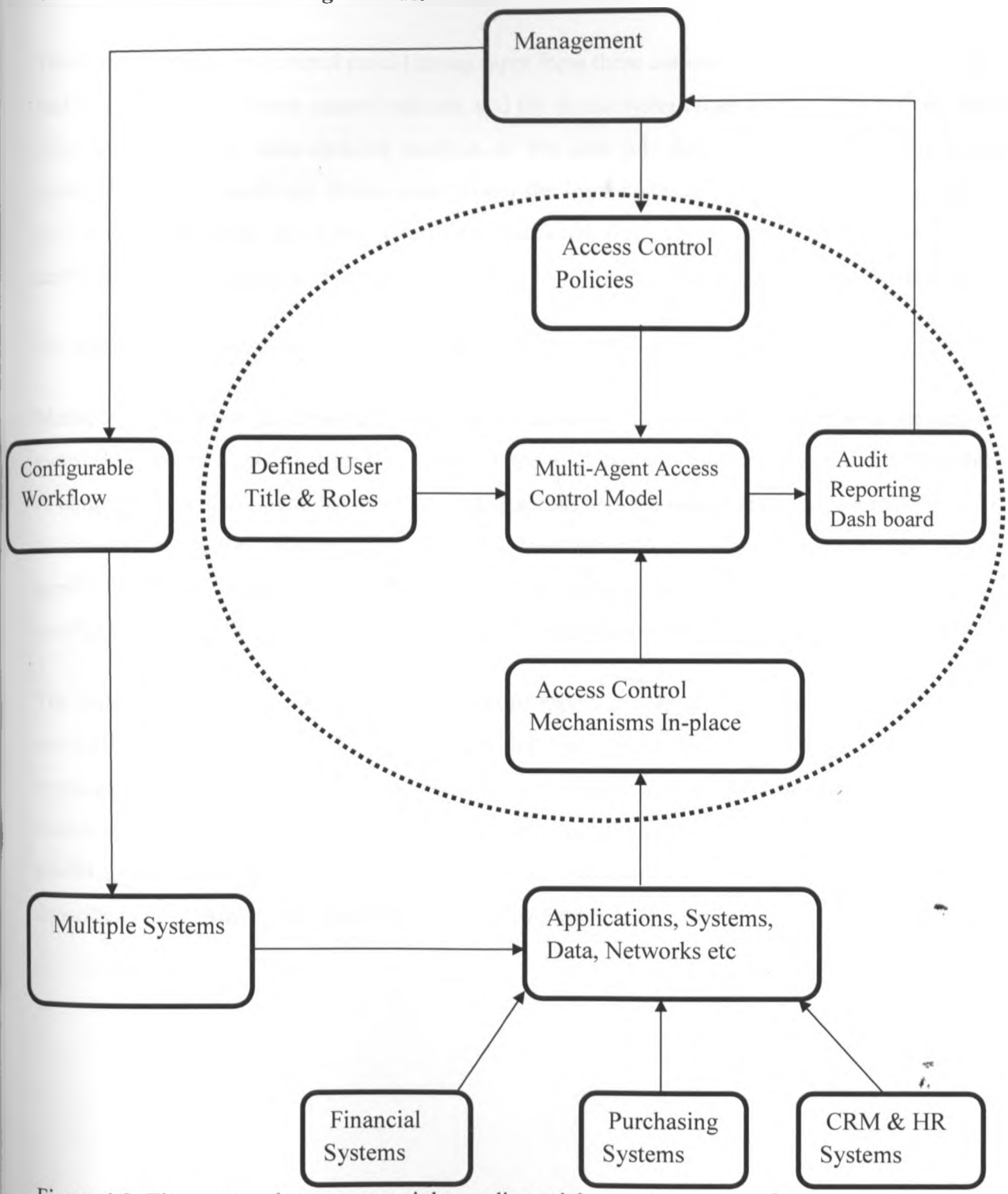


Figure 4.8: The proposed user access rights audit model

4.5.1 How the model results fit in this model

The Multi-agent access control model draws input from three components; the defined user roles and responsibilities, access control policies, and the actual system user access. The defined user roles was arrived at after detailed analysis of the user job descriptions and feedback from questionnaires on what user duties were within the organization. The access controls in place was representative of the active users' log extracted from the Oracle application database capturing actual system user access. Access policy with regards to application was also modeled.

The results of this analysis therefore fit in the model as follows;

Multiple applications accessed are presented to the multi-agent control model with respect to every user. When a specific user ID is queried, the application access is presented to the model. A comparison is then made between the actual access and the user roles and responsibilities as defined in the Job description. Violations are then reported in case of conflicts. The actual application access is then compared with the access policies to determine whether there exists conflicts in segregation of duties and the same reported accordingly.

The reporting dash board transmits the violations report to management who then determine if amendments need to be made to the users Job description so that they comply with the organizations requirements. When these amendments are made then based on the proposal a user becomes compliant or the rights are completely revoked. Then the circle continues. It was on this model structure that the 11 violations and 37 non violations were reported based on analysis described above. The results therefore fitted in this model.

CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Achievement of objectives

In light of work done relating to access rights audit, much emphasis when considering any access control system are the three abstractions of control: access control policies, access control models, and access control mechanisms. Policies are high-level requirements that specify how access is managed and who, under what circumstances, may access what information. While access control policies may be application-specific and thus taken into consideration by the application vendor, policies are just as likely to pertain to user actions within the context of an organizational unit or across organizational boundaries. This research set out to achieve certain specific objectives related to these. Below was how the objectives were realized:

1. Identification and modeling of user access policies, procedures and business rules using multi-agents. This was identified in the organizations IS policy and modeled based on the results of data analysis from the sampled system users
2. Identification and modeling of user roles and responsibilities for a sampled number of system users within an organization. This was arrived at after data analysis of both the job descriptions and feedback from questionnaires. And the same mapped to the user responsibilities within the application log extracted from the Oracle database application.
3. Identification and modeling of user access log extracted from the database application. A log of active users and their responsibilities was extracted from the Oracle database application and modeled after gaining an understanding of the various responsibilities assigned to the system users.
4. Development of a multi-agent user access rights audit model with key risk indicators. Based on data analysis, violations were reported and same modeled in the risk matrix format.
5. Testing the practical application of the model in the real world scenario. An SQL database was designed to capture data from the Oracle application database since the live environment could not be accessed and the same used to test applicability of the prototype.

5.1.1 Research Questions

The following research questions were also addressed;

1. How to model access policies, procedures and business rules using multi-agents. This was achieved through analysis and identification of policies and business rules relating to application access. The management agent was then designed purposefully to capture the policy requirements modeled.
2. How to model user roles and responsibilities within an organization using multi-agents. This was realized through data analysis of sampled users' job descriptions and feedback from questionnaires. The system user agent was then designed to capture the requirements modeled based on violations reported from the data analysis stage.
3. How to model user activity logged in the database application. This was achieved through identification of the active users log and understanding of the various responsibilities assigned to the sampled users within the application. The modeling involved an actual transfer of this log to an SQL database.
4. What constitutes a conflict scenario and how it can be modeled using multi-agents. Several conflicts were realized and defined. They included users accessing some applications that were not within their job descriptions, users who had left the organization but were still active in the system and users accessing the same application as a user, super user and manager. The modeling was based on coordination between the various agents to establish what was defined as conflict.
5. Conflict scenarios which are most likely fraud indicators. Here segregation of duties was a key issue. Where more than one violation was reported from a single user then the risk level was escalated as a potential area for fraud. Also former employees whose access rights had not been deactivated in the system were also considered potential areas for fraud.
6. What constitutes segregation of duties and how this can be modeled. Segregation of duties was defined where a single user having both user and super user rights to the same application.

7. How to map access policies, user roles and user activity logs. This was realized through data analysis where the active user log for specific users were identified and mapped accordingly with the job descriptions and the IS policy on application access of the organization.
8. How the model be testing can be done. This was achieved through both agent and society testing as detailed in the test plan. Generated errors were addressed during development.
9. The type of input required and expected output. The input was defined as the user name and output generated as no violation or the violation reported in terms of the number, applications accessed and the risk level.

5.2 Challenges

The following challenges came up in the course of working towards achievement of the above objectives;

The identification of which policies to model for this study with respect to user access to the application database was a key challenge. It was also not easy to define what constitutes segregation of duties amongst the various applications in the database.

The mapping of user roles and responsibilities as described in the job description into the application log was a key challenge since job descriptions had lengthy wordings. It was also significant to understand the role of every responsibility within an application in order to facilitate the mapping between the two. This exercise was very lengthy and time consuming. The mapping was also not possible for users who had moved from one department to another but their access rights in the system had not been updated to reflect their current roles and responsibilities.

The modeling of the active users log was another key challenge in the study as there existed some responsibilities which were globally assigned to all users such as NOC employee self-service under the Human Resources application. This responsibility could therefore not be mapped to any role within the job description as all employees had it for purposes of leave application, accessing the pay slips, air ticket requests and per diem application. Defining

segregation of duties within the access log was also a challenge because one had to understand the role of every responsibility within that application.

Defining conflict scenarios was also a key challenge as it involved the mapping of access policies with user roles and activity logs. The mapped job description was compared with the responsibilities in the access log and violations reported incase of mismatch. Policy violation was also reported as either a violation of the segregation of duties or job function. The output was in terms of the key risk indicators where a single violation was reported as low, two violations reported as medium and more than two violations reported as high.

5.3 Limitation

Because of the inability to access the production database due the organizational policy, the prototype was not connected for purposes of test to this database but instead a simulation of the same was done in SQL using the log extracted from the live environment. The SQL database was used for purposes of testing this prototype.

5.4 Research contribution

The research made a contribution to computer science by proposing a multi-agent model to be used for consistent system user access rights audit and thus will help in addressing the threat from within the organization relating to application access violations. It combined the three major concerns on access to information systems; the access policy, the users' job description and the actual user access to the system to come up with a tool which is of significance to the following stake holders:

System auditors: Use to consistently monitor what system users access. This therefore ensures compliance with regards to system access.

Database administrators: Much of the previous work done revolves around granting users access to information systems and external threats to the organizations infrastructure. Database administrators are able to be proactive in revoking access right to users who violate system access rules.

ICT Managers: With regards to adherence to the organizational policy on access to information systems, this tool enables the managers to be proactive in continuously monitoring users access and appropriately revoking the same incase of reported violations.

5.5 Recommendation and Further work

This model comprises of the key components required to consistently audit user access rights within an organization and is therefore recommended for use in modern organizations with enterprise resource planning systems running on Oracle E-business application.

Further work could be done on this research to make this tool real time by designing an interface between the tool and Oracle database application for continuous monitoring. The tool can in future be enhanced to have an interface with other auditing software such as Teammate and E-audit for purposes of real-time reporting and escalation of access violations to management for appropriate and timely action.

References

Approva Corporation., 2009. Continuous control monitoring. *(www.approva.net)*

Edward, J. C. and John, M. D., 2007. *Role engineering for enterprise security management.* London: Artech House.

Emerson, F. A. Lima, P. Machado, D. L. Fl´avio, R. Sampaio. J, Figueiredo, C. A., 2004. *An Approach to Modeling and Applying Mobile Agent Design Patterns.* ACM Software Engineering Notes, Vol. 29 number 4.

Glithero, B., 2010. Identity Audit applications streamline user access review. *IIA Internal Auditor Journal.*

InterNational Committee for Information Technology Standards., 2004 (INCITS), American National Standard for Information Technology *Role Based Access Control journal, 359.*

ISO/IEC 17799., 2005. Information technology security techniques code of practice for information security management. *International organization for standardization.*

ISO/IEC 27002 Code of practice., 2005. Information technology security techniques code of practice for information security management pp 5. *(www.iso27001security.com)*

Kannadiga, P., Zulkernine, M., 2005. A distributed intrusion detection system using mobile agents", Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, and First ACIS International Workshop on Self Assembling Wireless Networks. *Sixth International Conference,* pp (2325) 238 – 245.

Kuhn, D.R. Ferraiolo, F.D. and Chandramouli, R., 2007. *Role based access controls.* 2nd ed. London: Artech House

Schperberg, R., 2010. How businesses can protect themselves. *Governance risk and compliance ISACA Journal* vol.5 pp 11.

Shehory, O. and Sturm, A., 2001. Evaluation of modeling techniques for agent-based systems. *In Proceedings of the 5th International Conference on Autonomous Agents (Montreal, Ont., Canada, June)*. ACM, New York, pp. 624–631.

Stolfo, S. Prodromidis, A. Tselepis, S. Lee, W. Fan, D. and Chan. P., 1997. "JAM: Java Agents for Meta learning over Distributed Databases". *AAA197 Workshop on AI Methods in Fraud and Risk Management.*

Tommie, W. S., 2010. Mitigating I.T risks for logical access. *Governance risk and compliance ISACA Journal* vol.5 pp 7- 9

Wooldrige, M. Jennings, N. R., 1995. *Intelligent agents: Theory and practice*. Knowledge Engineering. Rev. 10, 2, 115–152.

Wooldrige, M., 2002. An Introduction to Multiagent Systems. *Wiley, New York.*

www.novell.com/products/audit.

Yang, K. Guo, X. Liu, D., 2000. *Security in mobile agent system: problems and approaches,* ACM SIGOPS Operating Systems Review.

Yogesh, S. Pradeep, K.B. and Omprakash, S., A review of studies on machine learning techniques. *International Journal of Computer Science and Security*, 1(1) pp. 71-84.

Appendix I: Access Rights Data Analysis Table

1.Sampled UserID	2. Access Log (AL) based on usernames	3. Job Description sampled	4. Quest. Feedback Received	5. Quest. Feedback compliance with JD	6. JD & Quest. feedback compliance with AL	7. IS Policy Compliance with AL & 5	8. Risk Matrix
AMUEMA	Y	Y	Y	Y	N	N	×
AODUOR	Y	Y	N	-	Y	Y	
BKIPRUTO	Y	Y	Y	Y	Y	Y	
CGENGA	Y	Y	Y	Y	Y	Y	
CKIPTARUS	Y	Y	Y	Y	N	N	×
CMUTHAMI	Y	Y	Y	Y	Y	Y	
CSYENGO	Y	Y	N	-	Y	Y	
DMWAI	Y	Y	Y	Y	Y	Y	
EBAYAS	Y	Y	Y	Y	N	N	×
EKILONZO	Y	Y	Y	Y	Y	Y	
EMACHARIA	Y	Y	Y	Y	Y	Y	
EMWANDOTO	Y	Y	N	-	Y	Y	
EOCHIENG	Y	Y	Y	Y	Y	Y	
FGATHOGO	Y	Y	Y	Y	Y	Y	
FGATURU	Y	Y	Y	Y	Y	Y	
FNJUGUNA	Y	Y	Y	Y	Y	Y	
GKOLETIT	Y	Y	Y	Y	Y	Y	
GOMARI	Y	Y	N	-	Y	Y	
HKIPKOECH	Y	Y	Y	Y	Y	Y	
JCHACHA	Y	Y	N	-	Y	Y	

JINGOLO	Y	Y	Y	Y	Y	Y	
JKITILI	Y	Y	Y	Y	N	N	×
JMUNIU	Y	Y	Y	Y	Y	Y	
JNJOROGE	Y	Y	Y	Y	Y	Y	
KCHORE	Y	Y	Y	N	N	N	×××
KMUGAMBI	Y	Y	Y	Y	Y	Y	
LCHIBOLE	Y	Y	N	-	Y	Y	
LMORAA	Y	Y	Y	Y	Y	Y	
LODUOR	Y	Y	Y	Y	Y	Y	
MISACKO	Y	Y	Y	N	N	N	×××
MKIRAGU	Y	Y	Y	Y	Y	Y	
MMUGAMBI	Y	Y	Y	Y	Y	Y	
MMUNGAI	Y	Y	Y	Y	Y	Y	
MNYAOKE	Y	Y	N	-	Y	Y	
MMAALIM	Y	Y	Y	Y	N	N	×
PCHERUIYOT	Y	Y	Y	Y	Y	Y	
PKWAMBAI	Y	Y	Y	Y	Y	Y	
PONDARI	Y	Y	N	-	Y	Y	
PWENDOT	Y	Y	Y	Y	Y	Y	
RMULANGE	Y	Y	Y	Y	N	N	××
RMWITHI	Y	Y	N	-	Y	Y	
SATHMANI	Y	Y	Y	Y	Y	Y	
SKABUE	Y	Y	Y	Y	N	N	××
SMUTINDA	Y	Y	Y	Y	N	N	×××
SORANYA	Y	Y	Y	Y	Y	Y	

TIRUNGU	Y	Y	Y	Y	Y	Y	
TNYAIRO	Y	Y	Y	Y	Y	Y	
WKEITANY	Y	Y	Y	Y	N	N	×
Total	**48**	**48**	**39**	**37**	**37**	**37**	

Appendix II: Sample Questionnaire

The following is an Academic Research Survey for a masters Degree in computer science project at the University of Nairobi. The project is titled "A multi-agent model for system user access rights audit". We would greatly appreciate your cooperation in filling out this questionnaire. Your responses will be treated as confidential and used for academic purposes only.

Background Questions:

1. Kindly provide the following details

Job Title_____

Department_____

Supervisor's Title_____

Date_____

2. Brief Job Description

_____

_____

Duties & Responsibilities_____

_____

_____

3. Have you worked with Oracle system? (e.g., Yes/No)

_____

Are some of your current duties done through Oracle system? (e.g., Yes/No)

_____

Kindly list them _____

_____

4. Are there colleagues with whom you share the same Job Title? (e.g., Payables Accountant)

_____

If Yes, do you share the same Oracle profile?_____

_____

5. Is there someone who relieves you?

_____

Are they in the same job title?_____

_____

What about the profile?_____

_____

6. Are there policies in place governing Oracle usage?

_____

_____

7. Any other important information you wish to state or clarify?

_____

_____

If you would like to get a copy of the results of analysis of this questionnaire please provide following details:

Name_____

Email_____

Tel. Contact_____

THANK YOU FOR PARTICIPATING IN OUR SURVEY

FREDRICK OKOTH

UNIVERSITY OF NAIROBI

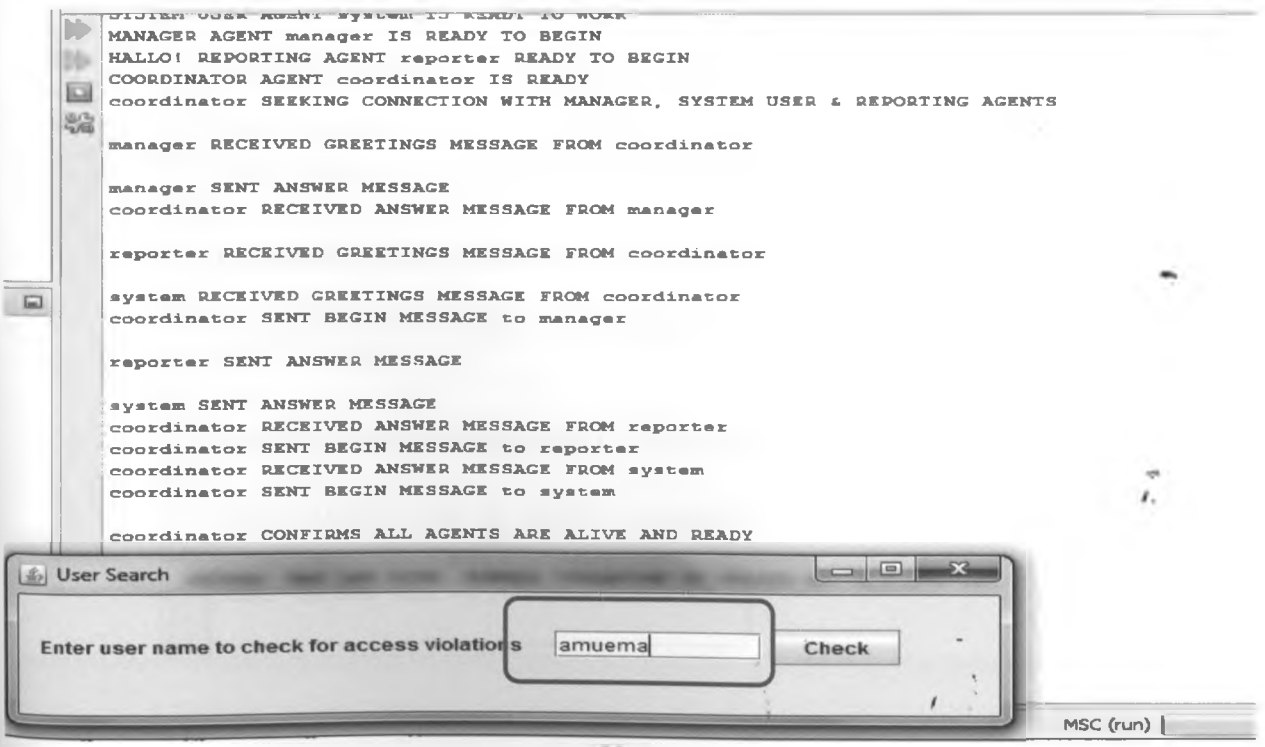
DEPARTMENT OF COMPUTING & INFORMATICS

TEL: +254 4447870 Email: fbitta@students.uonbi.ac.ke

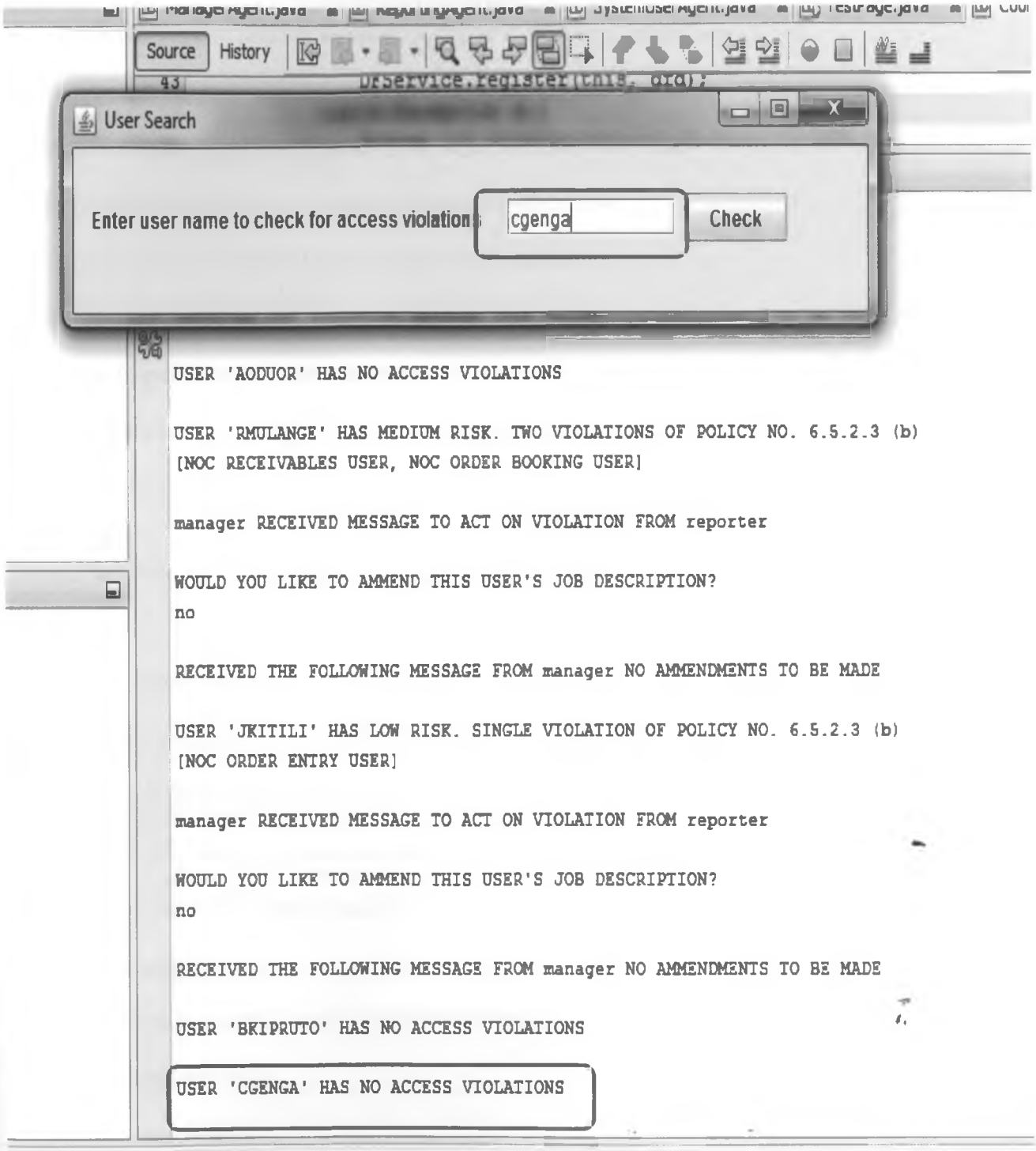
Appendix III: User Manual

1. Load MSc Jade project on NetBeans and one will be able to see all the agents in command window.
2. Click on Run button to get the agents platform and go to the main container in the agent platform
3. Start new agents by calling then in order from systems user agent, manager agent, reporter agent and finally coordinator agent
4. When one calls the reporter agent, a dialog box appears which request for system username to check for possible access violations
5. Also notice the alerts which appear on the command prompt as one calls the various agents into action
6. When prompted enter username to check for possible system violations.
7. When violations are reported, the manager agent is prompted for possible amendments to the Job description to ensure compliance and depending on the response, amendments can be implemented or not.

Sample input



Sample outputs



Appendix IV: Sample Source Code

```
package JADE_Project;

import java.sql.*; import java.util.ArrayList; import java.util.Arrays; import java.util.List;

import jade.core.Agent; import jade.core.behaviours.*; import jade.core.AID;

import jade.lang.acl.ACLMessage; import jade.lang.acl.MessageTemplate;

import jade.domain.DFService; import jade.domain.FIPAException;

import jade.domain.FIPAAgentManagement.DFAgentDescription;

import jade.domain.FIPAAgentManagement.ServiceDescription;

public class CoordinatorAgent extends Agent{

  private int answersCount = 0;

   private String outPut;

  //private                    MessageTemplate                    outputTemp                    =
MessageTemplate.MatchPerformative(ACLMessage.QUERY_REF);

  @Override

   protected void setup(){

     System.out.println("COORDINATOR AGENT "+getAID().getLocalName() + " IS READY");

     //enter user roles to policy matching service in Yellow Pages

     ServiceDescription sd = new ServiceDescription();

     sd.setType("Matching Users to Policies");

     sd.setName(getName());

     DFAgentDescription dfd = new DFAgentDescription();

     dfd.setName(getAID());

     dfd.addServices(sd);
```

```
                    replyT.setContent("Begin");

                    myAgent.send(replyT);

                    System.out.println(myAgent.getLocalName()+" SENT BEGIN MESSAGE to " +
ans.getSender().getLocalName());

            answersCount++;

            if(answersCount == 3){

              //breaks out of the loop after all agents respond with ANSWER

              try{

                Thread.sleep(1000);

              }catch(InterruptedException ie){}

              System.out.println("\n"+myAgent.getLocalName() + " CONFIRMS ALL AGENTS ARE ALIVE
AND READY");

              block();}}}else

          block();}}});

   addBehaviour(new HandleAccessViolation());

  }//end setup @Override

  protected void takeDown(){

    //de-register from Yellow Pages

    try{DFService.deregister(this);

    }catch(FIPAException fe){

      fe.printStackTrace();}

    System.out.println("COORDINATOR AGENT "+getAID().getLocalName() + " IS TERMINATING");

  }//end takeDown

  /**
```

```
 * Inner class HandleAccessViolation. This is the behaviour used by Reporting Agents to check for violations on request from GUI. */

private class HandleAccessViolation extends CyclicBehaviour{

    @Override

   public void action(){

     ACLMessage asmg = myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUEST));

     if (asmg != null){

       String myVal = asmg.getContent();

        ACLMessage reply = asmg.createReply();

        ACLMessage opt = new ACLMessage(ACLMessage.INFORM_REF);

       try{

          String Url = "jdbc:mysql://localhost:3306/noc?zeroDateTimeBehavior=convertToNull";

          String userName = "root";

          String password = "";

          String dbClass = "com.mysql.jdbc.Driver";

     Class.forName(dbClass).newInstance();

          Connection con = DriverManager.getConnection(Url, userName, password);

Statement s = con.createStatement();

          Statement b = con.createStatement();

           s.executeQuery("SELECT * FROM job_description");

          b.executeQuery("SELECT * FROM user_log");

         ResultSet rs = s.getResultSet();

          ResultSet bres = b.getResultSet();
```

```
    int counter = 0;
    int i=0;
    String[] far = new String[20];
    String[] storeFn = new String [20];
    int switchCase = 0;
    while(rs.next()){
      String nameVal = rs.getString("User Name");
      if(nameVal.equalsIgnoreCase(myVal))
        far = rs.getString("Function Access Rights").split("/");}
outside:
    while (bres.next()){
      String jina = bres.getString("Name");
      String fnVal = bres.getString("Function");
      String accessVal = bres.getString("Access");
      String userVal = bres.getString("User");
      String superVal = bres.getString("Super User");
      Date date = bres.getDate("End");
     if (jina.equalsIgnoreCase(myVal)){
       if(superVal.equalsIgnoreCase(userVal) && date != null || date!= null){
          switchCase = 1;
          storeFn[i] = fnVal;
          i++;
          break outside;
```

```
            }else if(superVal.equalsIgnoreCase(userVal)){
               switchCase = 2;
               storeFn[i] = fnVal;
               i++;
               break outside;
            }else{
               switchCase = 3;
               storeFn[i] = fnVal;
               i++; } }
         else{
            switchCase = 3;} }
         List <String> storeFnList = new ArrayList(Arrays.asList(storeFn));
         List <String> checkList = new ArrayList <String>();
      for(String string: storeFnList){
            if(string != null && string.length()>0)
               checkList.add(string);}
        List <String> farList = new ArrayList(Arrays.asList(far));
  //remove all elements in farList from checkList
         checkList.removeAll(farList);
         counter = checkList.size();
        List <String> viewList = new ArrayList<String>();
         String rightsVal = "";
Statement d = con.createStatement();
```

```
d.executeQuery("SELECT * FROM user_access_profile");
ResultSet ds = d.getResultSet();
while(ds.next()){
    String fnValT = ds.getString("Function Code");
    rightsVal = ds.getString("Rights");
    for(int z=0; z <checkList.size(); z++){
      if(fnValT.equalsIgnoreCase(checkList.get(z)))
        viewList.add(rightsVal);}}
outer:
switch(switchCase){
  case 1:
  {switch(counter){
      default:
        outPut = "\nUSER '"+ myVal+"' HAS HIGH RISK!! NON-EMPLOYEE" ;
        System.out.println(outPut);
        opt.setContent(outPut);
        opt.addReceiver(new AID("REPORTER",AID.ISLOCALNAME));
        send(opt);
        break outer;}}
  case 2:{
    switch(counter){
      case 0:
        outPut = "\nUSER '"+ myVal+"' HAS LOW RISK.SINGLE VIOLATION OF POLICY NO.
6.5.2.3 (d)\n";
```

```
                System.out.println(outPut);

                opt.setContent(outPut);

                opt.addReceiver(new AID("REPORTER",AID.ISLOCALNAME));

                send(opt);

                break outer;

            case 1:

                outPut = "\nUSER '"+myVal+"' HAS MEDIUM RISK. VIOLATION OF POLICY NO.
6.5.2.3 (b) AND POLICY NO. 6.5.2.3 (d)\n"+ viewList.toString().toUpperCase();

                System.out.println(outPut);

                opt.setContent(outPut);

                opt.addReceiver(new AID("REPORTER",AID.ISLOCALNAME));

                send(opt);

                break outer;

            default:

                outPut = "\nUSER '"+myVal+"' HAS HIGH RISK. VIOLATION OF MORE THAN TWO
POLICIES NOS. 6.5.2.3 (b) AND 6.5.2.3 (d)\n"+ viewList.toString().toUpperCase();

                System.out.println(outPut);

                opt.setContent(outPut);

                opt.addReceiver(new AID("REPORTER",AID.ISLOCALNAME));

                send(opt);

                break outer;}}

        case 3:

        { switch(counter){

            case 0:

                outPut = "\nUSER '"+myVal+"' HAS NO ACCESS VIOLATIONS";
```

```
                System.out.println(outPut);
                opt.setContent(outPut);
                opt.addReceiver(new AID("REPORTER",AID.ISLOCALNAME));
                send(opt);
                break;
            case 1:
                outPut = "\nUSER '"+myVal+"' HAS LOW RISK. SINGLE VIOLATION OF POLICY NO.
6.5.2.3 (b)\n"+ viewList.toString().toUpperCase();
                System.out.println(outPut);
                opt.setContent(outPut);
                opt.addReceiver(new AID("REPORTER",AID.ISLOCALNAME));
                send(opt);
                break;
            case 2:
                outPut = "\nUSER '"+myVal+"' HAS MEDIUM RISK. TWO VIOLATIONS OF POLICY
NO. 6.5.2.3 (b)\n"+ viewList.toString().toUpperCase();
                System.out.println(outPut);
                opt.setContent(outPut);
                opt.addReceiver(new AID("REPORTER",AID.ISLOCALNAME));
                send(opt);
                break;
            default:
                outPut = "\nUSER '"+myVal+"' HAS HIGH RISK. MORE THAN TWO VIOLATIONS OF
POLICY NO. 6.5.2.3 b) \n"+ viewList.toString().toUpperCase();
                System.out.println(outPut);
                opt.setContent(outPut);
```

```
                    opt.addReceiver(new AID("REPORTER",AID.ISLOCALNAME));
                    send(opt);} }}
          //closing connections
         ds.close(); rs.close(); bres.close(); d.close(); b.close(); s.close(); con.close();
         addBehaviour(new SimpleBehaviour(){
            @Override
            public void action(){
              if(outPut.endsWith("NO ACCESS VIOLATIONS")){
                 block();
              }else{
                 ACLMessage outMsg = new ACLMessage(ACLMessage.QUERY_REF);
                 outMsg.setContent(outPut);
                 outMsg.addReceiver(new AID("REPORTER",AID.ISLOCALNAME));
                 send(outMsg);}}
            @Override
            public boolean done(){
              return true; }});
}catch(Exception e){
          System.out.println("UNABLE TO ACCESS DATABASE!!");
          e.printStackTrace();
        }} else
        block(); //if no message arrives block behaviour
      }
 }// end of HandleAccessViolation
}//end CoordinatorAgent
```