MODELLING OF RELIABLE SERVICE BASED OPERATIONS SUPPORT SYSTEM (MORSBOSS)

by

Okuthe Paul Kogeda

A dissertation submitted in fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in the Department of Computer Science, University of the Western Cape

Promoter: Prof. Johnson I. Agbinya

June 2008

Declaration

I declare that *Modeling of Reliable Service Based Operations Support Systems (MORSBOSS)* is my own work, that it has not been submitted before for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Full Name: Okuthe Paul Kogeda Date: June 2008

Signature: ……………………………….

Abstract

The ever increasing number of subscribers, size, competition, heterogeneity and advancement of technology in the telecommunications industry has put a lot of pressure on the network service providers to not only provide high quality services but also to ensure that services are available whenever they are required. The 3G and 4G network services require robust network system with a predictable operation level and reliability. Network faults may degrade or paralyze the network performance and its operations. Network failures are unavoidable but early detection, identification and prediction of such faults is very crucial to the robustness of the network.

Different artificial intelligent techniques are used in design and implementation of the system for the reason that network faults are uncertain and dynamic in their behaviour. Network faults models are derived using Bayesian networks, which forms the engine of mobile intelligent agents (MIA). MIA operates like honeybee within a network environment and is used because of being platform independent, dynamic adaptation, network traffic reduction, mobile and robust.

The specific design and implementation is done using Java Agent DEvelopment framework (JADE). The MIA designed in JADE can learn, decipher and exploit the information that is logged into the database about network faults behaviour using Bayesian network models. A clustering of network faults logged into database by a certain cellular network service provider through its operations support system is used as the basis for constructing a wireless LAN. Seven different network faults were injected to test the quality of MIA developed. The experimental results showed accurate mapping of network services likely to be affected by the foreseen network faults. The MIA gives 98% detection rate of unknown and known network faults. The network faults (i.e., power) could be reported as early as 13 minutes before they occurred. The network faults prediction rate of 79% was achieved.

KEY WORDS - **OSS, Network faults modeling, Network service modeling, Network service reliability, Service Level Agreement (SLA), Network service dependency, Mobile Intelligent Agents (MIA), Network faults prediction, Bayesian Network, Honeybees, PSO, JADE-LEAP.**

I dedicate this work to my son

Acknowledgements

First and foremost I would like to acknowledge my Lord and heavenly father God, for the health, ability, opportunity, love and grace he provided to me and which has enabled me to complete this academic feat.

Prof. Johnson Ihyeh Agbinya for the support, guidance, inspiration, encouragement, challenge and belief in me. It was great that we could reason together about Computer Science and other aspects of life. Prof. Vladimir B. Bajic for the support, guidance and advice, which made this work a success. Prof. Christian W. Omlin for support, guidance and advice during the early part of this work, which made this work a success.

I would like to thank and sincerely acknowledge the National Research Foundation (NRF), South Africa for their financial support. I will not forget to acknowledge the support I received from Telkom South Africa.

My Son, Master Samson Ogeda Okuthe, for the time I was away from your life-time due to this noble course. My parents, sister, Gorety Ogeda and the rest of my family for all the sacrifices they made for me and their unshakable support during the tough period of this research project. Without you it would have been an uphill battle.

For taking his time in showing me probabilistic calculations Mr. Clement Bula Basuayi is duly acknowledged. I also appreciate the support and help I received from the Computer Science department. Indeed it made my stay and research in South Africa possible. Lastly, I would like to acknowledge my colleagues Ms Marylyne Harlott, Ms Anita Gong, Mr. Yaw Nkansah-Gyekye, Mr. Raguram Maladi, Mr. Pavan Kumar, Mr. Ananth Samudrala, Mr. Addmore Machanja, Mr. Camilius Sanga and all my friends who contributed in one way or another towards this goal.

Publications

Parts of this work have already been published as indicated below:

Okuthe P. Kogeda and Johnson I. Agbinya, “Cellular Network Fault Prediction Using Mobile Intelligent Agent Technology”, In the Proceedings of Southern African Telecommunication Networks and Applications Conference (SATNAC), Sugar Beach Resort, Mauritius, 9-13 September 2007. ISBN: 978-0-620-39351-5.

Okuthe P. Kogeda and Johnson I. Agbinya, “Proactive Cellular Network Faults Prediction Through Mobile Intelligent Agent Technology”, In the Proceedings of the 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007), Sydney, Australia, 27-30 August 2007. ISBN: 978-0-7695-2846-5.

Okuthe P. Kogeda and Johnson I. Agbinya, “Automating Cellular Network Fault Prediction Through Intelligent Agent Technology”, In the Proceedings of Southern African Telecommunication Networks and Applications Conference (SATNAC), Cape Town, South Africa, 4-6th Sept. 2006, ISBN: 0-620-37043-2.

Okuthe P. Kogeda, Johnson I Agbinya, and Christian W. Omlin, "A Probabilistic Approach To Faults Prediction in Cellular Networks", In the Proceedings of the 5th International Conference on Networking (ICN2006), Mauritius, April 23 - 28, 2006, ISBN: 0-7695-2570-9.

Okuthe P. Kogeda and Johnson I. Agbinya, “Prediction of Faults in Cellular Networks Using Bayesian Network Model”, In the Proceedings of First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2006), Sydney, Australia, March 13–16, 2006, ISBN: 0-9775200-0-5.

Okuthe P. Kogeda, Johnson I. Agbinya and Christian W. Omlin, “Probabilistic Faults Prediction in Cellular Networks”, Proceedings of Southern African Telecommunication Networks and Applications Conference (SATNAC), Kwazulu-Natal, South Africa, 11th–14th September 2005, Vol. 2, pp.31-32, ISBN: 0-620-34907-7.

Okuthe P. Kogeda, Johnson I Agbinya and Christian W. Omlin, "Impacts and Cost of faults on Services in Cellular Networks", Proceedings IEEE International conference on Mobile Business, Sydney, Australia, 11-13 July 2005, pp.551–555, ISSN 0-7695-2367-6.

Okuthe P. Kogeda, Johnson I. Agbinya and Christian W. Omlin, “Faults and Service Modeling for Cellular Networks”, Proceedings of Southern African Telecommunication Networks and Applications Conference (SATNAC), Cape Town, South Africa, 6th -8th September 2004, pp. 369-370, ISBN: 0-620-32632-8.

Oral Presentations

Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Sugar Beach Resort, Mauritius, 9-13 September 2007; Oral presentation by Okuthe P. Kogeda: “Cellular Network Fault Prediction Using Mobile Intelligent Agent Technology”.

IEEE 5th International Conference on Networking (ICN'06), Mauritius, April 23 - 28, 2006; Oral presentation by Okuthe P. Kogeda: “A Probabilistic Approach to Faults Prediction in Cellular Networks”.

Southern Africa Telecommunications Network Conference (SATNAC), Drakensberg, September 2005; Oral presentation by Okuthe P. Kogeda: “Probabilistic Faults Prediction in Cellular Networks”.

Southern Africa Telecommunications Network Conference (SATNAC), Cape Town, September 2004; Oral presentation by Okuthe P. Kogeda: “Faults and Service Modeling for Cellular Networks”.

Prof. Johnson I. Agbinya has also presented parts of this work at:

- Sydney Australia, 2007: “Proactive Cellular Network Faults Prediction Through Mobile Intelligent Agent Technology”, IEEE Proceedings.
- Sydney Australia, 2006: “Prediction of Faults in Cellular Networks Using Bayesian Network Model”, IEEE Proceedings.
- Sydney Australia, 2005: “Impacts and Cost of faults on Services in Cellular Networks”, IEEE Proceedings.

Poster Presentation

Okuthe P. Kogeda and Johnson I. Agbinya, “Automating Cellular Network Fault Prediction Through Intelligent Agent Technology”, Proceedings of Southern African Telecommunication Networks and Applications Conference (SATNAC), South Western Cape, South Africa, 4th -6th September 2006, ISBN: 0-620-37043-2.

Abbreviations

3GPP – Third Generation Partnership Project

ACL – Agent Communication Language

ACTP – Agents Communication and Transfer Protocol

ATM – Asynchronous Transfer Mode

BAN – Broadband Access Network

BSC – Base Station Controller

BTS – Base Transceiver Station

CAMEL – Customized Applications for Mobile network Enhanced Logic

CCITT – Consultative Committee for International Telegraphy and Telephony

CMIP – Common Management Information Protocol

CORBA – Common Object Request Broker Architecture

DECT – Digital Enhanced Cordless Telecommunications

EDGE – Enhanced Data Rates for GSM Evolution

FDMA – Frequency Division Multiple Access

FIPA – Foundation for Intelligent Physical Agents

FMC – Fixed Mobile Convergence

GPRS – General Packet Radio Service

HOSS – Hybrid Operations Support System

HSCSD – High-Speed Circuit-Switched Data

IIOP – Internet Inter-Orb Protocol

IPN - Internet Protocol Networks

ITU-T – International Telecommunication Union – Telecommunication standardization sector.

LLA – Logical Layered Architecture

MAE – Mobile Agent Environment

MASIF – Mobile Agent System Interoperability Facility

MSC – Mobile Switching Centre

MVNO – Mobile Virtual Network Operator

NEML – Network Element Management Layer

NGOSS – Next-Generation Operations Support System

NML – Network Management Layer

OSA – Open Services Access

OSFs – Operations Systems Functions

PSTN – Public Switched Telephone Network

RMON – Remote MONitoring

SCADA – Supervisory Control and Data Acquisition

SCP – Service Control Point

SML – Service Management Layer

SNMP – Simple Network Management Protocol

TINA – Telecommunications Information Networking Architecture

TMN – Telecommunication Management Network

TNMS – Telecommunication Network Management System

UMTS – Universal Mobile Telecommunications System

UPT – Universal Personal Telecommunications

VHE – Virtual Home Environment

VPN – Virtual Private Network

WCDMA – Wideband Code Division Multiple Access

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

In this Chapter, a brief overview of Operations Support System (OSS) in wireless communications networks is given and the motivation for this study is explained. The Chapter also explains the problem statements, the research approaches and methods and contributions of the thesis respectively. Finally, the thesis layout is provided.

1.1 Overview of Operations Support System

Cellular network service providers face a challenging landscape. Their success hinges on providing high quality and flexible range of services, in a timely manner and at competitive rates. These success factors can be achieved, by among other factors, ensuring a robust and reliable network OSS infrastructure.

Today, network service providers around the globe are focusing on maximizing operational efficiencies. To be profitable, they are taking advantage of revenue-generating network resources and services.

The move towards Broadband Access Network (BAN) is an evidence of the changing face of telecommunications industry around the world. The major factors promoting this change are:

- The ending of monopolies in the provision of telecommunications services;
- The revolution in network capacity precipitated by the introduction of fibre to the network;
- The explosive growth of the Internet;
- The increasing levels of market penetration for IT products with data communication capabilities and the convergence of the telephone and data networks. In particular record numbers of new Internet users and new Internet devices, explosion in data traffic and increasing demand for media-rich content, demonstrate the phenomenal growth of the Internet and the need for increased bandwidth and Quality of Service (QoS) to the customers.

Operations Support System (OSS) generally refers to system(s) that perform(s) management, inventory, engineering, planning, and repair functions for cellular network service providers and their networks. They include diagnostic, maintenance, and network management capabilities that are of use by Enhanced Service Providers (ESPs) in controlling their telecommunications services efficiently.

An OSS can also be defined as a combination of software and hardware to process information related to telecommunications management, with the goal of monitoring, coordinating, analyzing and controlling the telecommunications functions. It is mostly characterized by implementing management functions named OSFs (Operations Systems Functions) [1] [2].

Traditional Service Assurance (TSA) and OSS systems are element based, concentrating on addressing faults at the element level even when such faults may not have significant effects on system performance. A better approach in QoS based networks demands a clear view of the effects of a problem on services and the level of Service Level Agreements (SLA) between customers and ESPs.

Originally, OSS is used in the telecommunications to describe the processes and teams that monitor the underlying networks, which predominantly look at functional and non-functional requirements of such systems. Monitoring, End-to-End design, Error Handling tend to be the main core areas of work.

Usually, wireless communication carriers deploy an array of support systems specific to several tasks. In some cases tasks and functions are duplicated across the various pieces of OSS. Not only is this approach uneconomic, it is also prone to duplication of efforts and staff to man various systems. Integrating of OSS into a single platform or software saves costs and provides focus.

Intelligent networks including Third Generation (3G) and in future Fourth Generation (popularly known as Next Generation Network (NGN)) networks will utilize to a great extent Open Services Access (OSA). Although a common platform will and are provided through which interfaces to the individual services are created, there is no parallel in terms of Hybrid Operations Support System (HOSS).

BAN uses existing fixed network lines to transmit high-speed data, voice and video services over the last miles of a communications network to consumers and/or business end users. Services include E-mail, file transfers, video and audio downloading and viewing, e-shopping, direct advertising, Internet surfing, personalized greetings and interactive, multi-person gaming. The applications are truly limitless.

The telecommunications network should be reliable and available at all times to customers. This requires that all factors that affect reliability to be monitored carefully. Data that is critical for the maintenance of SLA needs to be defined and captured to ensure reliability. Faults that commonly occur during network operation need to be known and rectified immediately. This will ensure that services are available, which indicates network reliability.

For a cellular network service provider to meet its obligations of service provision it must be reliable. It is therefore; in this context that Model of a Reliable Service Based Operations Support System (MORSBOSS) is proposed and seeks to model data, faults and services that are critical for ESPs and its customers.

1.2 Motivation of the Research

OSS forms part of Network Management system that supervises and controls the functioning of the network so that it meets the requirements of both its users and its proprietors [2]. However, this objective remains elusive due to complexity, heterogeneity and highly distributed nature of telecommunication networks.

The existing TSA and OSS systems, which are elements based may not support the critical and new services on offer to customers according to the SLA between customers and ESPs. It is therefore necessary to build a tool that can be used to ensure that the cellular network service providers make available these critical services. Although implementing an OSS is not new, looking at OSS from the point of view of network services and implementing and/or designing a system in terms of services is new.

With reliable network services being provided over the telecommunications network, both ESPs and customers are set to exploit their full potential. MORSBOSS is proposed to bring this into a reality. Some of the vital points that will be realized include:

- Customer retention through improved service quality in telecommunications networks and services. This study proposes innovative, customer-specific solutions for the effective and proactive management of telecommunications networks and services. These help customers, network operators and service providers optimize the quality of their services. The result is retention of high-valued customers.
- Service Management. This project seeks to develop integrated solutions for monitoring and improving service quality of telecommunications enterprises, e.g.:
  - Service level management solutions, backed by systems monitoring service performance and by trouble ticket systems supporting an effective problem management process.
  - Field service support systems for control and supervision of network maintenance (staff, network technology, spare parts) based on the proposed MORSBOSS-component platform.
  - Service integration and automatic service provision, enables for flow-through enhancement, automating the operations business processes of a telecommunication service provider.
- Network Management: The proposed study is to implement up-to-date network-management solutions for both the manager and agent thereby linking diverse management models through mediation techniques. Policy and QoS management, customer management, and end-to-end umbrella management applications can help secure a carrier's business. Network planning and documentation and configuration management systems support day-to-day network operations.
- Technology and Migration: This study proposes the solution that combines various technologies which enables us to implement time-critical services with solid quality-bias.
- Customer Relations Management: The work proposes data warehouse and customer-relations management applications that help maintain a consistent, integrated view on customer data through business portals.

1.3 Statement of the problem

The main research problem is to address various challenges that cellular network service providers face to ensure that service provision to customers is reliable and is of high quality. Thus the study is tailored to address the following objectives:

1. To determine the best faults reporting methodology and how can such faults be segmented and implemented to provide quick response to restore services.
2. To determine the types of network faults that affect critical network elements (resources) responsible for network services provided.
3. To determine the important data to be captured in the database.

1.4 Approach and Methodology

Several approaches and methods were adopted for the success of this research. The first was the review of relevant literature. Several books were reviewed in order to establish what other researchers in the same field have done as well as to enrich the study with varied and latest information in the field of investigation. Journals, white papers, magazines (i.e., IEEE), news letters, and conference publications were also reviewed and they provided useful source for vital information.

A conceptual development and interpretive approaches were used. A conceptual approach developed after thorough study of literature, outlining the main concepts that were needed for the success of this study was used. Interpretations of the results were made based on data and behaviour of the network under study.

Several models were designed and adopted in this study. Some of the models developed include network service dependency models, network service models and network faults prediction models. With the help of data from a certain cellular network service provider, a database was designed and developed depicting the actual captured data. A simulation of these models using MATLAB (Simulink toolbox) was developed and applied with the data from a certain cellular network service provider being used as input parameters. Although simulation had the advantage of allowing for repetition of the experiments; it must be noted that the certainty of its accuracy is not entirely full proof.

Therefore, a small wireless local area network (WLAN) was setup to carry out physical simulation of the system under study. It consisted of eight devices: 2 PCs, 2 iPAQs, 1 Wireless router, 2 Laptops, and 1 Access point. Mobile intelligent agent software developed was tested under this environment. Common network faults were injected using various faults injection methods. The aim of this was to detect and predict network faults before they occurred under real network environment.

1.5 Contribution of the Thesis

Network faults prediction has generated a lot of research and there are many authors who have done some substantial work in this area. But the relationship between the cellular network services and network faults is not spelt out at all. Strength of this thesis is in the fact that it offers network service dependency modeling through which network faults are mapped onto network services that they are most likely to affect (cf. Chapter 3).

The model is generic and may be utilized to facilitate the understanding of any cellular network service provider. Besides that, the model is also reliable and robust, and is easily adaptable to architectural or technical alterations in the modeled networks.

From a general model of a cellular network service provider, developed in Chapter 3, a general cellular network services model is proposed, which constitute another contribution of this thesis (cf. Chapter 3). The proposed model adopts, as a strategy to approach complexity and dynamic nature of the problem of services modeling.

Another contribution of this work is the comparative study of the several approaches for services and faults modeling in cellular networks. The subsequent development of biologically inspired mobile intelligent agent software operating like honeybee is also a major contribution of this work (cf. Chapter 5).

This thesis also contributes a faster, accurate and reliable cellular network faults prediction algorithm. The mobile intelligent agent software will contribute to the knowledge in this industry.

1.6 Thesis Organization

The thesis consists of seven chapters. Chapters are arranged and connected to each other with a flow line as shown in Figure 1.1, which looks like a network consisting of nodes grouped into two layers. The thesis roadmap, objectives and literature review together with network service dependency modeling forms the upper tier. The lower tier with connections from the upper deals with design, implementation and testing results of the models developed.

In Chapter 1, an overview of OSS is presented. The motivation behind the research is also presented with the research problem, stating the research questions. The approach and methodology used in the research are also presented.

A review on modeling of OSS is presented in Chapter 2. It provides the evolution of cellular network management with specific view on cellular network faults management system. Cellular network modeling approaches and software intelligent agents are explored. A review of methods and algorithms of faults modeling highlighting strengths and weaknesses is also presented in this chapter.

In Chapter 3, network services are explored. The classification and management of network services with a discussion of issues affecting cellular network services are presented. Relationship and network service dependency modeling is also explored. Network services dependency types, benefits and models are provided in this chapter.

Cellular network faults modeling using Bayesian network is presented in Chapter 4. The classification of cellular network faults is presented. Basics of Bayesian network and its use in cellular network faults prediction as well as reasons for the choice (cf. Section 4.2.2) is presented. Model description and construction of the structure of the Bayesian network is also presented in this chapter.

In Chapter 5, cellular network faults prediction using mobile intelligent agents is presented. The basic concepts of biologically inspired modeling and mobile intelligent agent modeling as well as reasons for the choice (cf. Section 5.1.2) are presented. Application of particle swarm optimization algorithm is presented. The components of the system architecture are also presented in this chapter.

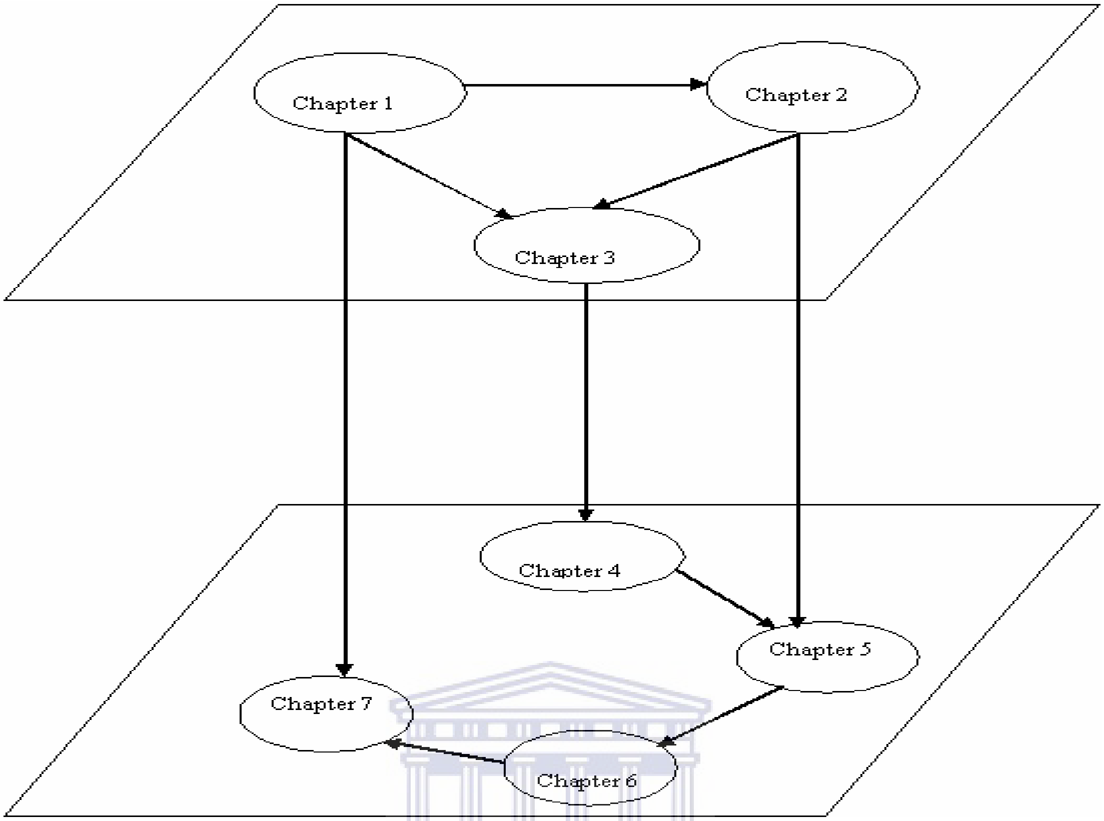


Figure 1.1: Thesis Organization

Implementation and experimental results are presented in Chapter 6. The software and technologies used in implementation are presented. Scenarios of experiments and experimental setup are presented. The results and discussions are presented in this chapter. The conclusion and future work are presented in Chapter 7.

1.7 Summary

The OSS section of faults management system of telecommunications network service providers takes the central focus with the main objectives of providing high quality services, retaining customers and making profits. The objectives can be achieved by among other factors an efficient OSS in place, which is able to operate effectively with minimal resources.

The Chapter mainly presented the overview of OSS, motivation of the research, problem statement, approach and methodology used in the research, as well as the contributions made by this thesis. A review of OSS and cellular network faults modeling algorithms and methods are explored into details in the next chapter.

Chapter 2

A REVIEW ON OPERATIONS SUPPORT SYSTEMS

Overview of OSS is provided in the form of the evolution of Cellular network management (Section 2.1) with a prospective study on cellular network OSS being explored in Section 2.2. Section 2.3 offers a panoramic view of cellular network faults management system in the literature. In Section 2.4, telecommunication services architectures is provided. Software intelligent agents are explored for network modeling in Section 2.5. Cellular network service dependency is presented in Section 2.6. Overview of network faults modeling is presented in Section 2.7. In Section 2.8, related work is presented with the summary being drawn in subsequent section.

2.1 The Evolution of Cellular Network Management

The adoption of Network Management System (NMS) standards by Consultative Committee for International Telegraphy and Telephony (CCITT) study group IV in 1985 addressed the problems of the increasingly complex public telephone networks [3][4]. Then deregulation and shifting competitive landscape later forced the former monopolistic telephone companies to revamp their inefficient processes in an effort to retain their customer base and maintain their profitability. As such, the concept of network management was quickly adopted in the interest of achieving improved fault management, equipment configuration, usage accounting, performance verification, and network security.

2.1.1 Network Management Framework

The ever-increasing demand for NMS has resulted in a sub-industry of hundreds of NMS providers. This in turn led to a need for interoperability between systems, and the telecommunications industry decided that a framework for network management was needed, which would serve as guide for future development of networks and NMS.

In order to implement the framework, the specification of Telecommunication Management Network (TMN) involved the establishing of reference points, which represent boundaries among functional blocks, whose purpose is to identify the information exchanged among these blocks. Generally, each reference point represents an interface between two functional blocks, for implementation purposes. For information exchange to take place among these interfaces, the International Telecommunication Union – Telecommunication standardization sector (ITU-T) and International Organization for Standardization (ISO) [5][6][7][8] defined standards both for the protocols and for the information models to be adopted in the communication industry.

The ITU-T defined TMN [9][10][11] recommendations, which consist of a series of standards that collectively defined the framework. The recommendations were presented on general vision of the ITU-T [12], principles [14][15], architectures [11][13][16][17], definitions [18-21] and specifications [22-32] needed for the implementation of a TMN.

TMN essentially provides functional capabilities of various *network elements*, a guide for layered network architectures, and a building block approach to defining network investments. The interconnection between different OSS and the telecommunications network, which is essentially constituted of heterogeneous network elements, is the main objective of TMN.

2.1.2 Management Functional Areas

Network management functions can be grouped into five distinct functional areas (Figure 2.1) as follows [9][33][34]:

1. Fault Management – this is a set of functions that

(a) detect, isolate, and correct malfunctions in a telecommunications network,

(b) compensate for environmental changes, and

(c) include maintaining and examining error logs, accepting and acting on error detection notifications, tracing and identifying faults, carrying out sequences of diagnostic tests, correcting faults, reporting error conditions, and localizing and tracing faults by examining and manipulating database information**.**

2. Configuration Management – this aims at preparing, initializing, starting, keeping operating and terminating interconnection services, through the exchange of information with managed objects and by acting upon these objects. It also includes the creation or deletion of managed objects.

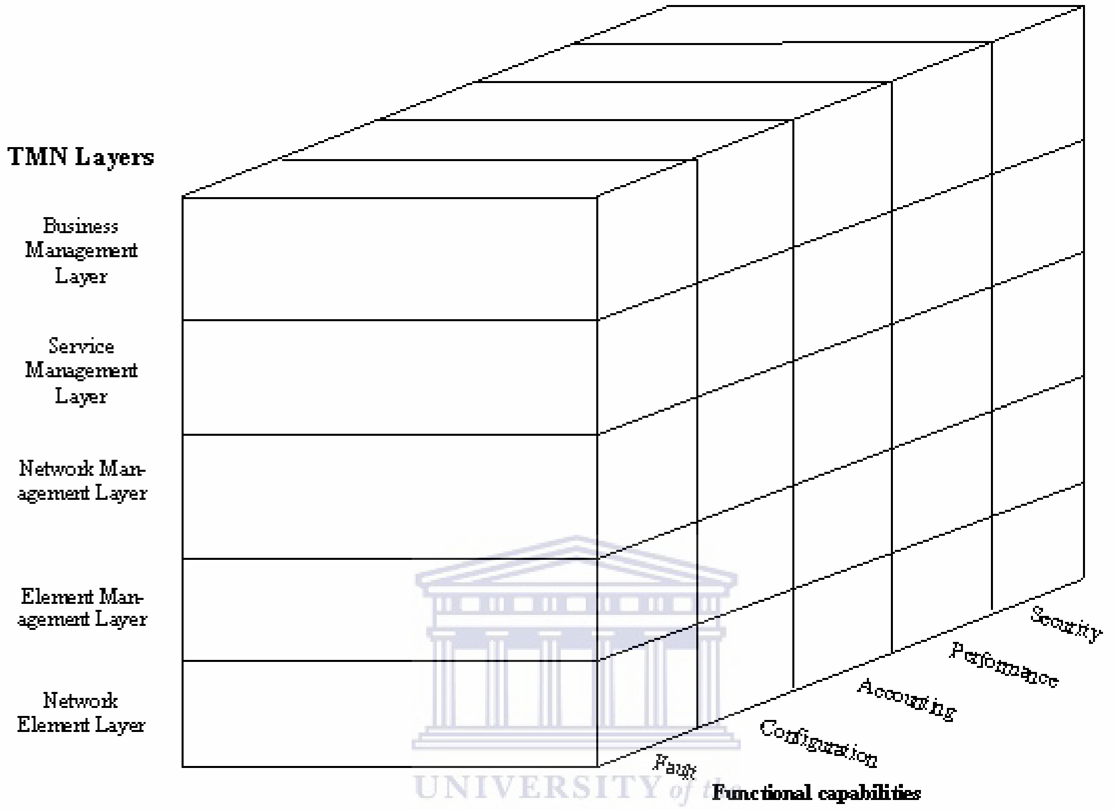


Figure 2.1: Telecommunication Management Network Functional Architecture

3. Accounting Management – this aims at identifying the costs of the network features utilized in order to meet the needs of a given communication objective, so that these costs may be communicated to the users and the corresponding tariffs be applied.

4. Performance Management – contains a set of functions that evaluate and report the behavior of telecommunications equipment and the effectiveness of the network. The sub-functions may include gathering statistical information, maintaining and examining historical logs, determining system performance under natural and artificial conditions, and altering system modes of operation.

5. Security Management – contains a set of functions that aims at protecting telecommunication networks and systems from unauthorized access by persons, acts, or influences. The sub-functions include creating, deleting, and controlling security services and mechanisms; distributing security-relevant information; reporting security-relevant events; controlling the distribution of cryptographic keying material; and authorizing subscriber access, rights, and privileges.

Adequate management and integration of functional areas will ensure effective utilization of telecommunications network. The information generated in one area may be useful in the other, i.e., the occurrence of a fault may cause a reduction of the network performance and may also compromise its security, leading to the need to act on its configuration.

2.1.3 Telecommunication Management Network Layers

This is the architecture that describes the hierarchy of management responsibilities, also known as Logical Layered Architecture (LLA). It is defined as a way of facilitating the understanding of the functionalities of telecommunications management systems by ITU-T [3][11][13][34]. Functionalities of OSS are grouped in OSF into four logical management layers as shown in Figure 2.1:

- Network Element Management Layer (NEML) - OSF manages functions of individual Network Elements (NEs) and deals with vendor specific management functions and hides these functions from the layer above. Functions performed by this layer include detection of equipment errors, measuring power consumption, logging of statistical data, measuring the temperature of equipment, updating firmware, measuring the resources that are being used (i.e., buffer space, CPU-time, queue length, etc), etc.

- Network Management Layer (NML) – with the support from NEML, this layer is able to know, monitor and control the utilization of the network resources, thus guaranteeing its functioning according to adequate performance standards and service quality. Functions performed in this layer include detection of faults, monitoring of link utilization, creation of the complete network view, optimizing network performance, modification of routing tables, and creation of dedicated paths through the network to support QoS demands of end users, etc.

- Service Management Layer (SML) - provides the main point of contact to clients with the service provider. Therefore it must have updated and precise information on the activation and deactivation of services, the quality of these services and the occurrence of faults in rendering of these services. Accounting, address assignment, QoS management, addition and removal of users, maintenance of group addresses, are some of the functions performed at this layer.

- Business Management Layer – this layer is responsible for the management of the whole enterprise with the aim of obtaining a better utilization of telecommunication resources, under the business point of view, which consists of searching the best return on investment. Functions include the support to the decision processes related to the realization of new investments and to the allocation of resources administration and maintenance of telecommunications resources.

2.2 Cellular Network Operations Support System

The worldwide rapid growth of subscribers depending on cellular network service providers [35] requires a reliable cellular network management system in place to manage the ever-increasing subscriber base. The term *reliable* is used in this thesis to mean a system which is dependable, consistent in performance and predictable over time and use. The degrees to which random errors occur indicate the relative reliability of the measuring instrument and data produced. The less random errors are, the higher the reliability.

As one of the key network management solutions, OSS gives cellular network service providers the ability to create, deploy, manage, maintain, and bill for network services [37]. OSS also performs ‘network facing’ functions for the cellular network service providers. Carriers are looking for solutions that will inter-work wireless, IP, and traditional PSTN elements within their enterprise and extend a comparable level of management and control to dissimilar OSSs as part of a wholesale strategy. Figure 2.2 shows an OSS model.

Telecommunications Network Management System (TNMS) is the assignment and control of proper network resources to address service performance needs and the network objectives. It has become impossible to carry out these functions without the support of automated tools due to ever-increasing size and complexity of underlying networks and services.

The Next Generation Networks (NGNs) will manage their services from the smallest component of the network, *network element*, which communicates with the TMN, according to ITU-T defined standards, with the purpose of being monitored and / or controlled [9] [13].

As cellular networks increasingly become automated, elements within the network will be managed more efficiently, as network management systems become more streamlined, cost-efficient and easier to use in NGN. This should support both the legacy and next generation network services, which would allow for the continuity of the network and retention of customers.

Customer centric way of management improves the relationship between customers and cellular network service providers. This relationship is characterized by attracting more customers and retaining the current ones, service provision at reasonable fees, accurate billing, provision of services according to customers' taste, etc. Customers will always use measurable indicators to evaluate the performance of the network, i.e., service quality, meantime between failures, mean time to repair, etc.

Availability is a critical need for advanced OSS. The mechanisms that can improve availability include [36]: interface redundancy; application backup and recovery; management system / application redundancy; and data and event replication.

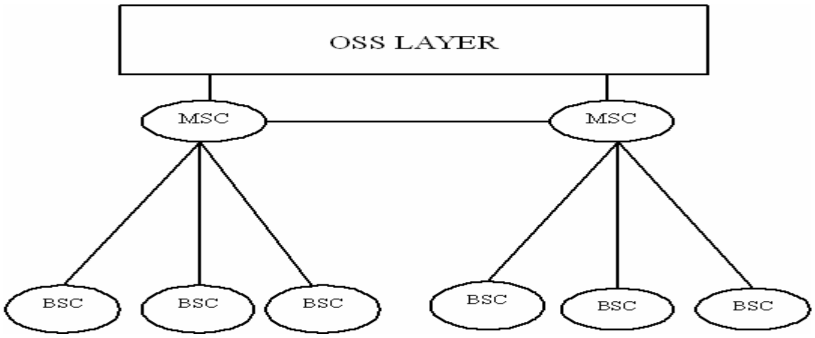


Figure 2.2: Cellular Operations Support System Model

2.3 Cellular Network Faults Management

Network faults management consists of monitoring, detection, diagnosis and correction of malfunctions and faults that occur in the cellular network [9]. Currently used systems mainly deal with monitoring the events (alarms and state changes) generated by the underlying equipment. However, maintenance staffs still broadly perform troubleshooting and repair work manually.

In fault management different error rates and measures are commonly stored in Management Information Base (MIB) for each managed object. MIB variables are scalar valued counter, which are often redundant partly measuring similar behaviour from slightly different perspectives. Not all variables need to be tested in order to detect abnormalities. An observation in a test may be based on the nominal values of the MIB variables or a function may be applied to the variable, like taking the time difference in order to gain the rate of change over time of the counter.

2.3.1 Requirements of an Effective Fault Management System

An effective fault management system should [38][39][33][40]: (1) scale up well to growing networks (2) perform non-intrusively (3) the management activity should not interfere with normal operations of the network, must only intervene when necessary. Excessive polling wastes bandwidth that could be used for other important services (4) robust, (5) perform even when the network is not fully operational, as management is mostly needed in abnormal situations, i.e., when connections are broken.

2.3.2 Faults Estimation

Faults estimation is a calculated approximation of quantity of the occurring fault using physical system parameters [41]. However, faults with zero (or almost zero) thresholds are very difficult to estimate.

2.3.3 Faults Reporting

Cellular network faults are non-deterministic in nature and therefore very difficult to predict [42]. A series of triggered events are usually reported to the element controller when a fault occurs in a cellular network. These events are also known as objects in which they reside, then store this information in MIB, and finally provide it (proactively or reactively) to management entities within NMS via network management protocol for example, Simple Network Management Protocol (SNMP) and Common Management Information Protocol (CMIP).

Network faults detection is complete when the devices send SNMP trap messages, SNMP polling, Remote MONitoring (RMON) thresholds, and syslog messages. The end user is alerted by the management system when a network fault is reported and corrective actions can be taken.

One of the methods of reporting faults is by using RMON alarm and events, which are defined in RMON specification. A network element can be configured to monitor itself with RMON. By incorporating RMON as part of management solution, a network can proactively be monitored. However, this can only work for minor faults.

2.3.4 Network Faults Modeling Approaches

Network faults modeling has been the subject of intense research over the last decade, with two main approaches being used: on the one hand, mathematical models have been derived for solving fault management problems; on the other hand, a WLAN can be built imitating the real network under study. Mathematical modeling is very cost effective, takes less time, and is not technically demanding as compared to a WLAN model built to imitate the real network in trying to find a real and applicable solution to network problems (cf. Chapter 6).

Scientists use mathematical models to analyze a system to be controlled or optimized. It usually describes a system by a set of variables and equations that establish relationships between the variables. The values of the variables can be practically anything (i.e., real, integer, Boolean or String values). The variables represent some properties of the system, for example, measured system outputs often in the form of signals, timing data, counters, event occurrence, (yes/no). The actual model is the set of functions that describe the relations between the different variables (cf. Chapter 3).

2.4 Telecommunications Services Architectures

Telecommunications services architectures integrate existing systems, applications and users into a flexible architecture that can easily accommodate changing needs. Integrated design, reuse of existing telecommunications networks investments and, above all, industry standards are the elements needed to create services architecture.

As part of the third generation (3G) mobile telecommunications network or Universal Mobile Telecommunications System (UMTS), Open Services Architecture (OSA) describes how services are architected in a UMTS network. The standards for OSA are being developed as part of the third Generation Partnership Project (3GPP) [43] and are published by ETSI [44] and 3GPP. The API for OSA is called Parley, and is developed jointly in collaboration by 3GPP, ETSI, and the Parley Group [43][44][45].

Services Architecture is a design concept for telecommunications networks with a strong emphasis on standards. Telecommunications networks vendors have abandoned costly product development strategies that delivered closed or proprietary innovation, and have adopted a more standards-based product base. Services Architecture will enable companies to take advantage of this. The future telecommunications systems will be organized not as monolithic structures deployed by a single business entity, but rather as a dynamic confederation of multiple-sometimes cooperating and sometimes-competing service providers.

2.4.1 Criteria Services Architecture Should Meet

Services architecture will be required to meet the following criteria [46]:

- *Diversity rather than homogeneity*: future telecommunications systems will be characterized by many kinds of end devices, access networks, services, applications, service providers and content providers. This would require that services architecture embrace diversity and exploit software mediation to achieve interoperability.
- *No overarching access network*: a variety of access networks and data services will co-exist for some time to come. For example, in the wireless access network, we will have EDGE, HSCSD, GPRS, 3G packet, etc. The architecture must achieve application transparency by providing the necessary services to integrate these.
- *Enable emerging business models*: the service architecture must support technologies that enable new business models, such as Mobile Virtual Network Operator (MVNO) to achieve enhanced efficiency of resource usage.
- *It's all about services*: the service architecture must enable business entities to provide enhanced services, as a primary means of differentiating one provider from

another. Support for sophisticated capability negotiation and service level peering, and concepts like enhanced preferences management in support of the Virtual Home Environment (VHE), are essential pieces of any such service architecture.

- *It's all about confederated services*: the service architecture must support overlapping service provider regions, with subscribers able to roam among them for service provisioning, even without actually "moving". The architecture must support a relationship that is, sometimes cooperating and sometimes competing among service providers, forming dynamic syndicates for the purpose of service provision.
- *Network-application awareness*: to better support subscribers in their tasks, the service architecture must provide mechanisms that make the network more aware of applications (e.g., near-term future indications of needed bandwidth), while the applications become more aware of the availability of system parameters and resources (e.g., user location, proximity of system resources to the user's current location, etc.).

2.4.2 Benefits of Services Architecture

The tangible benefits of services architecture can be summarized as follows [46]: (1) telecommunications cost reduction, which is achieved by standardizing the way services talk to each other and improved quality through leveraging of new voice over IP technologies. (2) Secure and reliable information access across wired and wireless infrastructure, with reduced risk and exposure to fines and litigation; (3) optimized performance and lower operating cost for essential applications through application-optimized network infrastructure; (4) increased systems and business availability and resiliency; (5) increased revenue and improved customer service through "anywhere, anytime" customer access.

2.4.3 Intelligent Network

The Intelligent Network (IN) is the main telecommunication architecture for service creation and control. It is a design, development, and deployment framework that allows rapid and efficient introduction of new tele-services. IN design is based on separation of service-specific software from basic call processing principle.

Prior to the advent of the IN, each switch manufacturer had their own recipe for incorporating services into their switches. This means when introducing new services in the network, the software in each switch requires modification. Such a process could take years to complete. Network operators were heavily dependent on their equipment suppliers.

The IN eliminated much of this dependency by moving service-specific software into a specialized node called the Service Control Point (SCP). Basic call processing is performed in the switches, now called Service-Switching Points (SSP) in IN parlance. The communication between SSPs and SCPs is done through a channel called Common Channel Signaling No. 7 (CCS7). IN has also reduced the time needed for introducing a new service from years to just a few months. The IN has gained acceptance due to the multitude of services that it supports and its application to cellular networks. Therefore, it is an inspiring concept for the provision of future services.

The IN service control provides the intelligence of a network/system. IN services (i.e., free-phone, Virtual Private Network (VPN), tele-voting, Universal Personal Telecommunications (UPT), Customized Applications for Mobile network Enhanced Logic (CAMEL), entertainment services, etc) are triggered from the network to give more advice to the call routing or to provide other facilities to users. The IN services are provided between the network and the end-user or within the network itself.

2.4.3.1 Internet Protocol (IP) Networks

These are characterized by devolved service architecture, i.e., there is no global and standardized framework for the provision and creation of services. Any user that can afford a server can create new services. Creating new services implies developing a distributed application that must be installed and executed in terminals and servers. IP-based applications take advantage of intelligent terminals and powerful user interfaces. However, some services (and even basic mechanisms) of the IP technology require specialized servers; the most important of such servers is the Domain Name Server (DNS).

2.5 Software Intelligent Agents for Faults Modeling

This section provides the definition, features, benefits, drawbacks and application areas of software intelligent agents.

2.5.1 Definition and Features

Software agents are defined as an entity with goals, actions and domain knowledge situated in an environment and its way of acting is called behaviour [47]. Software agents are classified into *weak* and *strong* notion of agency [48]. The agents with weak notion of agency, exhibit features like autonomy, sociability, reactivity and pro-activity. While agents with strong notion of agency, exhibit features of the agents with weak notion of agency plus mobility, veracity, benevolence and rationality as additional features.

Intelligent software agents are programs in themselves but differ from conventional programs on the following properties [49][50]: (1) they can act reactively and proactively as required, (2) are autonomous, (3) are adaptive to the environment, (4) are sociable, (5) are cooperative and (6) mobile, can migrate from one node to another. It is very important to note that; intelligent agents typically depend on collaboration with other agents often in a generic manner through an Agent Communication Language (ACL).

Software agents that are able to move independently from one node in a network to another node to complete the task assigned to them by the user are commonly referred to as Mobile Agents (MA). Mobile Intelligent agent architectures differ, but almost all of them contain a Mobile Agent (MA) and a Mobile Agent Environment (MAE). Agents in different MAEs but in the same host can communicate with each other. Services being provided by a host operate in MAE and can communicate and migrate to another host as shown in Figure 2.3 below.

Mobile agents allow for the easy programmability of remote nodes by migrating and transferring functionality where required as an advantage. However, static intelligent agents can collaborate to devise or negotiate flexible solutions for complex application scenarios. A disadvantage of mobile agents as compared to static intelligent agents is that they can often incur significant performance overheads in the network during migration.

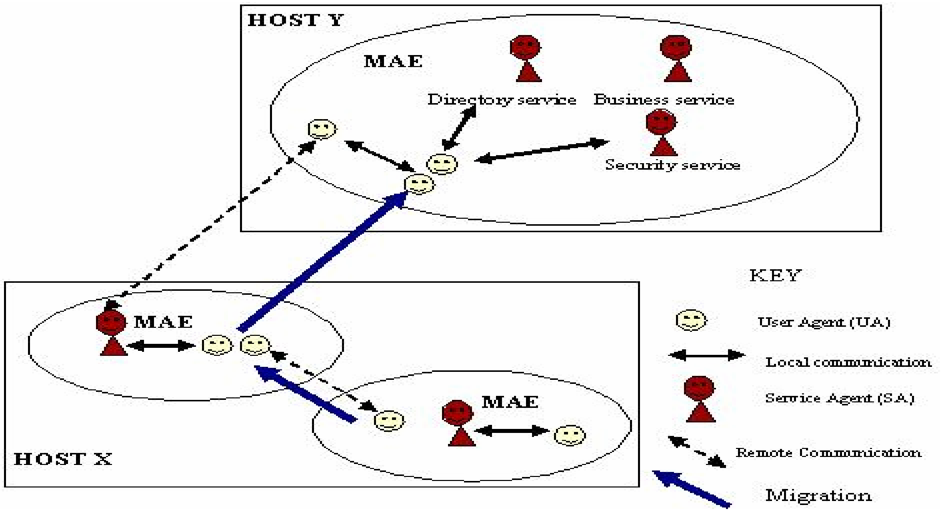


Figure 2.3: General Mobile Agent architecture

Though software agents are more clearly understood through their attributes and behaviours, not all of them exhibit all the commonly agreed characteristics by researchers. These features include [51] [72] [73]:

- *Autonomy*: agents should have a degree of control over their own actions and own internal state performing tasks without direct intervention of humans or other agents.
- *Social ability*: agents should be able to interact with other agents and humans in order to complete their own tasks and to help others with their activities where appropriate.
- *Responsiveness*: agents should perceive their environment (physical world, a user, a collection of agents, the INTERNET, etc.) and respond to occurring changes in time.
- *Pro-activeness*: agents should not only respond to their environment but exhibit opportunistic, goal-directed behaviour and take the initiative where appropriate.
- *Persistent* – keep running a process continually until the desired result is achieved.
- *Reasoning* – able to reason out their actions
- *Adaptable* – able to function on multiple platforms and solve technical difficulties on their own learning from its experience in dealing with its environment.
- *Veracity* – an agent will not lie knowingly.
- *Benevolence* – an agent will carry out its tasks faithfully.

- *Rationality* – an agent will act in order to achieve its goals.

2.5.2 Benefits of Mobile Intelligent Agents

A number of benefits identified by researchers based on theoretical study of the agents properties and alternative paradigms, weakly supported by limited practical experience and assessment within specification contexts include [52][53][54]:

- *Mobile Agents reduce the network load* – moving service request agent to the target server will enable mobile agents to access the resources of the server and reduce its interaction with the initiator. This helps reduce the flow of data in the network, ease the bandwidth reliance, diminish communication delays and improve service quality.

- *Mobile agent moves autonomously and asynchronously* - the tasks can be packaged into a mobile agent that can be sent through the network. Then the connection between originator and destination machine can be disconnected. Afterward, the mobile agent is independent of the process that created it, and runs autonomously and asynchronously.

- *Mobile agents adapt dynamically* - mobile agents can sense their execution environment and react autonomously to changes. A mobile agent dynamically determines its next move based on the load of servers and network, and thus helps load balancing. Additionally the intelligent routing of mobile agents also lessens the judgments that need to be made when users are browsing and searching the network.

- *Mobile agent facilitates parallel processing* - mobile agents can create a cascade of clones in the network when administering parallel processing tasks, which will help increase efficiency and reduce processing time. Multiple mobile agents have the unique ability of distributing themselves among the hosts in the network to maintain the optimal configuration for solving a particular problem.

- *Mobile agents are naturally heterogeneous* - distributed network computing is fundamentally heterogeneous. Mobile agents provide optimal conditions for seamless system integration because they are dependent on only their execution environment.

- *Mobile agents are robust and fault-tolerant* - a mobile agents' ability to react dynamically to unfavorable situations and events makes it easier to build robust and fault-tolerant distributed systems. If a host is being shut down, all agents executing on that machine are warned and given time to dispatch and continue their operation on another host in the network.

2.5.3 Drawback of Agents

A number of drawbacks limiting agents from having an impact on telecommunication industry have been identified by researchers. These include [54] [55] [56] [57]:

- *Standardization and Interoperability*- inconsistency has greatly hindered the adoption of mobile agent technology with limited standardization so far. Even the few standardization efforts made are yet to be widely adopted. In the direction of standards, the Object Management Group (OMG) has produced the Mobile Agent System Interoperability Facility (MASIF) [71] that crucially addresses the issue of interoperability between mobile agent platforms. In addition, the Foundation for Intelligent Physical Agents (FIPA) has produced specifications for the "intelligent" communication between agents [58] [59].

- *Security and Safety*- although it is now possible to deploy a mobile agent system that adequately protects a node against malicious agents [60], numerous challenges remain. These involve the protection of nodes without artificially limiting agent access rights, protecting an agent from malicious nodes as well as protecting groups of nodes that are not under single administrative control.

- *Lack of killer application*- the current network environment, its usage and related management tasks have not revealed so far an application that can only be achieved through the use of mobile agents.

- *Limited practical experience*- while a theoretical base for mobile agents exists there is limited work on the application and practical assessment of agent technologies to specific contexts such as network management.

- *Performance Overheads*- mobile agent-based systems can help reduce network latency and bandwidth utilization, but this often comes at the expense of higher utilization of resources at network nodes. Furthermore, attention is needed regarding any agent migration overheads especially in scenarios involving multi-hop mobility.

- *Getting ahead of the evolutionary path*- it was unlikely that the current centralized client/server approach to management would move directly to mobile agent-based approach. The evolutionary path takes time and it will probably move gracefully from centralized protocols, to distributed object frameworks, followed by mobile code solutions and later by mobile agents.

2.5.4 Application Areas

The industry and researchers in different areas have used intelligent agent technology because of the inherent advantages i.e., agents provide bi-directional mode of information dissemination, reasoning abilities, mobility, standardized, and collaborative decomposition and resolution of problems. Intelligent agent technology is used in manufacturing systems [61], manufacturing planning and control [62], medical and health care systems [63][64] and inventory management [65], education [66], management of electricity transportation network for a utility company [67], air traffic management systems [68].

2.5.5 Mobile Intelligent Agent Architectures

A naming service and an execution environment comprise mobile agent architecture. A naming service for mobile agents caters for the naming, migrating agents and lookup of the agent execution environments hosting those agents. Reflection and introspection are mechanisms required for the binding and remote communication of mobile agents because of their dynamic nature. A proprietary Agents Communication and Transfer Protocol (ACTP) often support the communication. ACTPs are based on a number of protocols such as Java Remote Method Protocol (JRMP), the Internet Inter-ORB Protocol (IIOP) or sockets relying on TCP or UDP transport.

A large number of mobile architectures are available today, offering suitable runtime environment for mobile agents as well as high-level programming API supporting mobile agent capabilities. However, the work presented in this thesis was done using JADE-LEAP (mobile agent platform) distributed by Telecom Italia [69][70]. JADE was chosen for a combination of benefits offered such as: simplicity in usage and agent programming, good online community support and documentation, support for the FIPA standards, efficient and tolerant of faulty programming and it is an open source (without cost), etc.

2.6 Cellular Network Service Dependency

2.6.1 Introduction

Dependency provides a very interesting relationship between two or more components or services in cellular network. The consumer/provider relationship between different entities in a cellular network system is called dependency. When one component requires a service performed by another component in order for it to execute its function, this relationship between the two components is called a dependency. For example, a voice service depends on a cell where it is located for network signal and the database for billing purposes before a call is initiated. In turn these services depend on availability of power supply. In this case a voice service is the *dependent* and cell and database are the *antecedent*. Consequently, cell and database are the *dependents* and power is the *antecedent*. This relationship is shown in Figure 2.4.

Services cannot be considered isolated tasks. Services very much depend on other services or sub-services, lower level network elements, operating systems, physical components and communication infrastructure to be able to function.

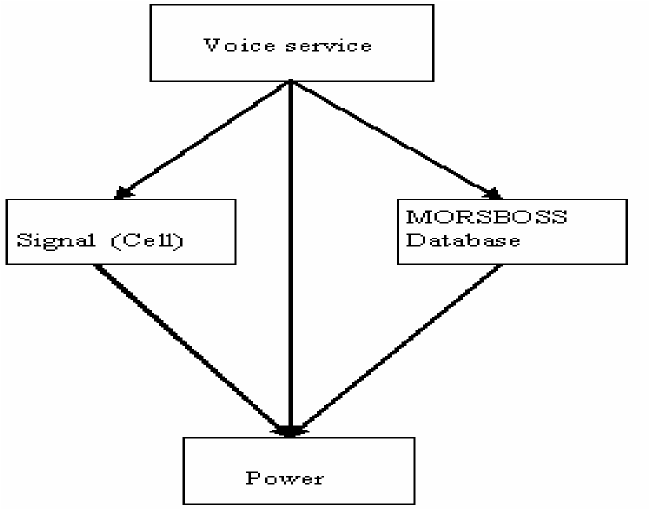


Figure 2.4: Example of (Voice) Service Dependency

This area has attracted several researchers with different viewpoints. Ensel et al [74] presents a scalable service dependency. They used neural networks for dependency detection modeling. Cervantes et al [116] presents a mechanism to automate service dependency management in a service-oriented component model. The mechanism they used eliminates complex and error-prone code from component-based applications dynamically.

Gruschke [113] proposed dependency graph for event correlation where dependency was described as a relationship between different entities. A generic approach proposed here could be used for different abstraction levels (i.e., system level, network level, service level). Caswell et al [114] describe dependencies for services with specific reference to Internet Service Providers. They went further by defining five types of dependencies (c.f. section 2.7.2). Gupta et al [115] presents analysis of temporal relationships of interactions to derive dependencies.

The approaches presented above fail to identify, detect and predict which network service would be affected by a network fault. In this work, an approach of dynamic dependency is used to pre-empt the likely services that a likely network fault would affect. While some of the approaches above cannot be implemented, the approach adopted in this work can be implemented backed with mathematical support.

2.6.2 Types of Service Dependency

The main types of service dependency include [74] [75] [76]:

- *Execution dependency* – This dependency relates directly to an application server process being executed on a host machine. The performance of an application server process depends on the status of the host machine. The types of application servers that are executed on host machines include web, email, news, DNS, and NFS.
- *Link dependency* – performance of a service depends on the link status. For example, the communication between two nodes, *A* and *B*, may solely depend on the link between them *AB*.
- *Component dependency* – in case of a web service that is provided on different front-end servers, which are selected by a round-robin DNS scheduling the performance,

depends on the currently selected server. A component dependency occurs in order to ensure scalability and redundancy of a service. ISPs often replicate web, email, and News content across a number of servers. The round robin scheduling balances the load among the servers. The servers are grouped together and assigned a single domain name in the DNS database. When the DNS server receives a request for the domain name, the IP address of one of the servers is acquired in the round-robin scheme.

- *Inter-service dependency* – It occurs when one service accesses another service for its proper operation. This occurs between services, i.e., e-mail service depends on an authentication service and on an NFS service; a web service depends on DNS service to allow the subscriber to connect the web server host using its IP address, and an NFS service is used to access the web content.
- *Organizational dependency* – This dependency occurs when there are different ISP operations personnel (e.g., experts) who are responsible for different services and service components. For example, an ISP may have a first supervisor managing the web service, a second supervisor managing DNS, and a third supervisor managing NFS. Operational responsibilities may also be delegated based upon the geographical location of the service components.

The first three dependencies are grouped and referred to as *resource dependency*. In this case the service being offered depends on the resources (i.e., execution, link, component, and/or another service) available at the time. These resources in turn are affected by the cellular network faults, i.e., faults may degrade, reduce or totally take away the resources available to a service.

2.6.3 Benefits of Service Dependency

The main benefits of dependency modeling include [74]:

- *Root cause analysis* – it helps to find a common (root) cause of faults detected at different places within the cellular network environment. This can be used on network components reporting error conditions as well as to services, where end users detect problems. The faults that are normally reported to the management systems are descriptions of the symptoms. Therefore further knowledge about dependencies among the faults is necessary to derive their root cause [186].

- *Determination of availability requirements on services*. To minimize the time for resolving network fault.
- *Prediction of the impacts on other services due to management operations*. This is of particular interest when a resource goes down (for repairs) then it can be determined in real-time which services and customers are affected.
- It can be the basis of scheduling tasks and transactions. The service dependency provides a detailed task structures, which enables coordination of services better.
- It can be used to recognize service misuse and intrusion detection.

2.7 Overview of Network Faults Modeling

2.7.1 Definition of Faults

Cellular network fault can be defined as an abnormal operation or defect at the component, equipment, or sub-system level, which significantly degrades performance of an active entity in the network or disrupts communication. All errors are not faults as protocols can mostly handle them. Generally faults may be indicated by an abnormally high error rate.

A fault can be defined as an inability of an item to perform a required *function* (a set of processes defined for purpose of achieving a specified objective), excluding that inability due to preventive maintenance, lack of external resources, or planned actions [77].

2.7.2 Characteristics of Cellular Network Faults

There is lack of a generally accepted definition of what constitutes a behaviour of a normal network fault [78][79][80]. Therefore it is very difficult to characterize the cellular network faults accurately. However, there are estimations (based on statistics of the network traffic) as to what characterize a cellular network fault. Cellular network faults are characterized by transient performance degradation, high error rates, loss of service provision to the customers (i.e., loss of signal, loss of connection, etc), poor quality of service provision, delay in delivery of services and getting connectivity, etc.

2.7.3 Causes of Cellular Network Faults

The main causes of network faults differ from network to network. Managing complex hardware and software systems has always been a difficult task. The Internet and the proliferation of web-based services have increased the importance of this task, while aggravating the problem (faults) in at least four ways [34][105][111][112]:

- The speed of software development and release means less reliable and more frequently updated software.
- Multi-tier and distributed software architectures increase the complexity of the cellular network environment and obscure causes of both functional and performance problems.
- Internet style service construction implies more dynamic dependencies among the distributed software elements of the overall services making it difficult to construct and maintain accurate system models.
- Internet scale deployments increase the number of service elements under a particular administrator's responsibility.
- Many heterogeneous networks
- New innovations means interoperation of different networks must be kept to some level leading to faults.
- Overloading of power supply gadgets, natural disasters, etc.

2.7.4 Goals of Faults Modeling

A mathematical representation of cellular network faults at the highest level of abstraction possible is known as fault modeling. The goals of faults modeling can be summarized as follows: (1) to reduce the number of individual faults that have to be considered in network fault management , (2) to reduce the complexity of the system description that must be modeled, tested and analyzed, (3) allows for analysis, test generation and decision to be made on whether to implement a system under study or not, (4) simplicity-it makes the understanding of the system much more clear and simple, and (5) reduces cost of implementing the faults management system.

2.7.5 Methods and Algorithms for Faults Modeling

A detailed review of methods and algorithms for faults modeling is provided by Meira [34] and Holst [120] in their respective works. While Meira [34] gives the methods and algorithms for alarm correlation, a summary of some of these methods under machine learning are provided by Holst [120]. Most of these methods and algorithms can be utilized in the faults prediction process. Probabilistic approaches and the approaches in which the network entities are modeled as finite state machines are identified [34] for faults identification, detection and prediction in cellular network service providers. Others apply principles defined in non-conventional logics and others adopt *ad hoc* methods to deal with faults modeling.

Among the methods and algorithms include: Fuzzy Logic [81][82], Artificial Neural Networks [86][87], Decision tress [88][89], Model Based Reasoning [90][91], Case Based Reasoning [92][93][94], Rule Based Systems [95], Blackboard [96] and Bayesian networks [84] among others.

There is no single model or method in terms of complexity, cost, precision, time, and robustness, which can be regarded as best method to be used in network faults modeling. Recent approaches lean towards a combination of two or more methods together to solve complex problems (like cellular networks systems) [83][84][97]. The choice of a method would depend on a specific problem case. However, the following factors may be considered: (1) Implementation complexity, (2) Facility for construction of a theoretical model of the *object network,* (3) Performance, (4) Facility to adapt to change in the object network, and (5) Precision. Network faults prediction must take into account the object network characteristics, which can be looked at from technology viewpoint i.e., CDMA, FDMA, GSM, UMTS, WCDMA, CDMA2000 [97], GPRS, EDGE, DECT, etc.

The nature of application area will also dictate the choice of a method to adopt. The high cost of implementation and adaptation to changes in the object network make it difficult to apply rule-based approaches in large cellular networks and hence are better used in network elements whose configuration is rarely altered. The other approaches that are less sensitive to changes in the object network i.e., case-based approaches still lack a theoretical basis, which would allow their utilization in large, size commercial cellular service network systems.

The complexity of the problem to be solved sometimes brings the *exceptions*, which can be effectively treated by mechanisms that are used for the implementation of *non-monotonic* reasoning [98]. This means each identified exception requires the reformulation of the already established rules or the creation of new ones at the development phase. The complexity of the solution increases, which reduces the performance and robustness. Rule-based systems provide additional structuring which facilitates the development of applications but because of the tendency to reduce performance, they are not attractive for the implementation of more complex cellular network systems.

The complexity of cellular network faults prediction problem makes it extremely difficult to obtain exact solutions [99]. This brings uncertainty as a factor in fault prediction process that needs to be considered. Fuzzy logic, Bayesian networks (cf. Section 2.7.6), case-based reasoning and artificial neural networks are some of the approaches that can deal with uncertainty. Each of these alternatives have got advantages and disadvantages, for example, defenders of fuzzy logic based approaches argue that they simplify applications development and result in working products with excellent performance; on the other hand it is tuning (like membership functions) is very hard and it lacks a solid mathematical support which hinders its adoption in a larger number of applications.

Bayesian network based methods, which were first utilized in 1921 in the analysis of harvesting results [100] count on a solid mathematical support. They win more acceptances in the community of computing scientists as a suitable option to the solution of problems involving uncertainty [100]. These factors contributed to the adoption of Bayesian networks in this work.

2.7.6 Bayesian Networks

Bayesian networks model is named after *Thomas Bayes*, who proved a special case of what is called *Bayes' theorem*. The term *Bayesian*, however, came into use only around 1950 [84]. It provides an approach to the treatment of uncertainty with incomplete and inaccurate available data to produce inferences.

A Bayesian network is a Directed Acyclic Graph (DAG) in which each node represents a random variable to which conditional probabilities are associated, given all the possible combinations of values of the variables represented by the directly preceding nodes; an edge in this graph represents conditional probabilities between the variables corresponding to the interconnected nodes [34][84].

The terms *subjective probability*, *personal probability*, *epistemic probability* and *logical probability* describe some of the schools of thought, which are customarily called "Bayesian". These terms overlap but there are differences of emphasis. *A subjective probability* expresses the degree of belief of an expert related to the occurrence of a given event, based on the information this person has available up to the moment. The use of subjective probabilities is very often the only resource in situations where analytical or experimental data is very hard or even impossible to obtain.

It is possible sometimes to evaluate conditional probabilities from empirical data obtained from the past behaviour of the network service provider under study. Given a Bayesian network and a set of evidences it is possible to evaluate the network, that is, to calculate the conditional probability associated with each node, given the evidences observed up to the moment. Generally speaking, this is a NP-hard problem [85] but with the use of appropriate heuristics and depending on the problem dealt with; networks containing thousands of nodes may be evaluated in an acceptable time.

2.8 Related Work

Network faults modeling usually involves network faults identification, localization, detection and predictions. A review of network faults identification and detection is presented by Lazar at al [112]. Yang et al [102] presents a behavioural fault model where they apply statistical methods to model network faults. Koutsoukos et al [103] presents a model for monitoring and diagnosis of faults in sensor-rich hybrid systems. Network faults are related to cells by Hartmann et al [104]. Niemann et al [41] presents detection of parametric faults and Hajj et al [78][79] presents detection of network faults. Most of the above authors failed to relate network faults to network services and failed to demonstrate how their models could perform under complex distributed network environment. For this lack, this work presents a hybrid technique of network faults modeling using probabilistic methods and artificial intelligence techniques.

Thottan et al [105] presents a proactive anomaly detection using distributed intelligent agents and faults prediction at the network layer using intelligent agents in [106]. They used intelligent agents, throbbing technique and Bayesian Belief Network (BNN) in network faults detection and prediction. The deployed agents obtain relevant MIB data, provide temporally and spatially correlated predictive alarms, and time correlated abnormal changes in the individual MIB variables. Their testing showed successful prediction rate of seven faults out of nine faults with a prediction time in the order of minutes. However, the authors failed to relate faults to network services; this work provides the relationship, uses mobile intelligent agents and gives better network faults prediction rate in minutes.

A dynamic Bayesian belief network for intelligent fault management systems is provided by Sterritt et al [110]. They explored ways of applying the Bayesian Belief Network in fault prediction. Bayesian reasoning techniques is used to perform fault localization in complex communication systems using dynamic, ambiguous, uncertain, or incorrect information about the system structure and state by Steinder et al [5].

Pissinou et al [107] used mobile agents to automate the fault management in wireless and mobile networks. Their work delved on detection of faults and added mobile agents to automate the recovery from faults. Bieszczad et al [53] discussed the potential uses of mobile agents in network management and Eleftheriou et al [109] explored the use of mobile agents in network management systems. Both works used mobile agents in fault analysis as well fault detection. Hood et al [108][109][110][111] presents a proactive network fault detection using intelligent agents. They used Bayesian network for theoretical framework within which a prior knowledge was used to determine structure and learn the normal behaviour of the measurement of variables. Their system could detect seven out of ten faults before they occurred in minutes. However, the above authors failed to relate network faults to services.

Biologically inspired modeling has attracted many researchers who have used mostly insect-like (ant, bee, etc) behaviour to model and perform network management. White et al [117] presents the use of multi-agent system that relies on Swarm Intelligence, in particular ant-like trail laying behaviour for fault detection in communications network. In this case, each agent is 'chemically' inspired and proposes 'chemical' interaction as the principal mechanism for inter-swarm communication. 'Chemical' messages have two attributes, a label and a concentration with the latter defining a 'weight' that leads the process of behaviour selection. Agents within a given swarm have behaviour that is inspired by the foraging activities of ants with each agent capable of simple actions, while no single agent has knowledge of the global goal. The creation of chemical trails is proposed as the primary mechanism used for distributed problem solving arising from the self-organization of swarms of agents. The approach has been applied to fault management [119]. However, this work uses mobile intelligent agents that rely on the behaviour of the honeybees. More specifically the prediction feature of honeybees to predict nodes which are about to experience network faults.

An approach proposed which is based on societies of small, biologically inspired and relatively simple agents that need to cooperate to deliver the intelligence needed for the diagnosis of network faults [117]. A number of types of such tiny agents are injected in the network, each addressing one aspect of the problem and using observations to confirm or disprove a specific hypothesis. The solution to the problem emerges through the integration of the hypotheses results of each agent type [118][119]. However, these approaches though some used Bayesian network and mobile agents; they fail to relate network faults to services. In this work honeybee-like mobile agents are used to predict the network faults before they occurred (c.f. Chapter 6).

2.9 Summary

As the public telephone networks grew, preventative maintenance and planning for network growth became a significant concern. The possibly detrimental effects of adding new network elements to the overall network were often poorly understood until well after the elements had been deployed. Resulting network failures were typically not acute in nature. Rather, they would happen gradually over time and, as a result, be very difficult to troubleshoot. This would in turn often require another costly upgrade of network elements.

In this Chapter, cellular network OSS review was presented. A review of network faults management, software intelligent agents, network service dependency, network faults modeling and related work were presented. In the next chapter, network services and network service dependency modeling is provided.

Chapter 3

SERVICE AND NETWORK SERVICE DEPENDENCY MODELING

In this Chapter, the basics of services and applications are presented in Section 3.1. In Section 3.2, cellular network service management is discussed. Issues affecting cellular network services are presented in Section 3.3. Relationship between network faults and services is presented in Section 3.4. Cellular network service dependency basics, types and benefits are presented in Section 3.5. In Section 3.6, cellular network service dependency models are presented. In Section 3.7, experimental results are presented with summary being drawn in subsequent section.

3.1 Network Services

Network service is a crucial and very important resource in a cellular network. For any cellular network service provider to retain and acquire new subscribers, it should be generic, cost effective, fair robust, reliable and have high performance connectivity among a large number of communication devices (i.e., computers, wireless terminals, etc), for the highest customer satisfaction.

3.1.1 Definition of Services and Applications

The ITU-T defines telecommunication services as [121]: A *service* represents telecommunication capabilities that the customer buys or leases from a service provider. Service is an abstraction of the network-element-oriented or equipment-oriented view. Identical services can be provided by different network elements, and different services can be provided by the same network elements. A set of functions and facilities offered to a user by a provider is known as service. One service can serve several consumers and a server is always its execution environment.

Application on the other hand is used as the generic term that represents a set of features, combining communication and document processing, on which end users may perform operations. Applications may depend on working methods, and on allowed processing of documents. Open interchange of process-able documents and co-operative working are examples of applications [121]. An *application* is a program that a user directly interacts with. An application utilizes services and might incorporate modules to fulfill its tasks. The application is not restricted to a special environment to run in.

3.1.2 Typical Services and Applications Foreseen for 4G Networks

Typical services and applications foreseen for 4G networks will include, service and bearer capabilities, including speech, data, supplementary services, security and mobility provided for UMTS services, and further applications (i.e., multi-person gaming over the network, high definition audio, high multimedia applications, etc). Video and audio services may be provided as applications that directly access services and bearer capabilities. Other services/applications include voice over IP (VoIP), voice over ATM (VoATM), videoconferencing, video on demand (VOD), broadcast video, network games, Internet access, virtual private network (VPN) access, Internet Protocol TV (IPTV), and account access/private applications.

In future the 4G services will put consumer in control by enabling personal, custom, on-demand viewing of entertainment, e-learning, video games, etc. Individuals will choose what they want to hear, see, or be entertained by on their own, and people will no longer have to plan around preconceived broadcast schedules for home entertainment. Customers will eventually be able to decide their own schedules for home entertainment. 4G will also allow customers to easily create their own content, personalize it, and distribute it for viewing on TVs, PCs, remote laptops, and mobile phones and other wireless devices around the world, instantly.

3.1.3 Characteristics of Cellular Network Services

Cellular network services are tricky entities that do not have specific characteristics that apply to all of them. Different cellular network services posses’ different features that preserve the identity of each cellular network service.

However, there are a few features that most cellular network services have in common such as [124][125], services can use one or more media of transmission; most services being offered by cellular service providers are easily programmable and flexible to the needs of customers; most services are easily accessible with cost and legal permission to use them; cellular services are randomly initiated and executed.

3.1.4 Classification of Network Services

A set of applications with similar or common set of characteristics (cf. Section 3.1.3) can be classified as a service. Generally cellular network services can be classified into data, voice and multimedia according to ITU-T I.211 [121]. Network services in digital form are called data. Network services in 'vocal chord' form are called voice and are regarded as the oldest cluster of network services. However, the latest type of network services, which are normally composed of pictures, videos, text and/or sound are called multimedia. Multimedia services consume a lot of bandwidth and require powerful devices to be able to receive and send them. Figure 3.1 shows classes of network services with examples.

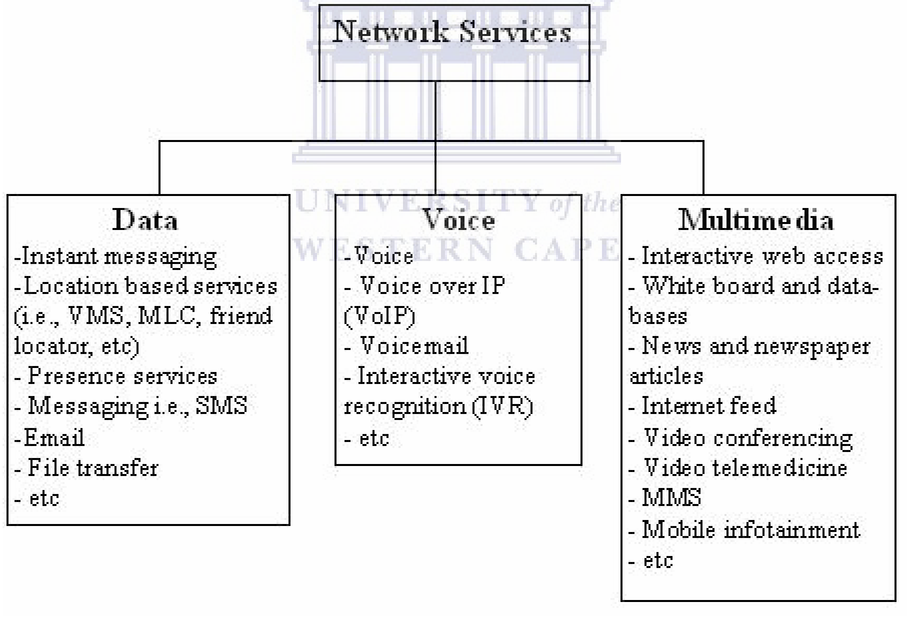


Figure 3.1: Classes of Network Services

3.1.5 Services Life-cycle

Services go through various defined stages, which start with planning, signing (analyze) before one can start the provision of the service, which may encounter errors during this period and such errors must be corrected. The service can be maintained till it requires a major improvement. Then the process starts again from plan, sign, provision and maintenance, hence service life cycle as shown in Figure 3.2 below.

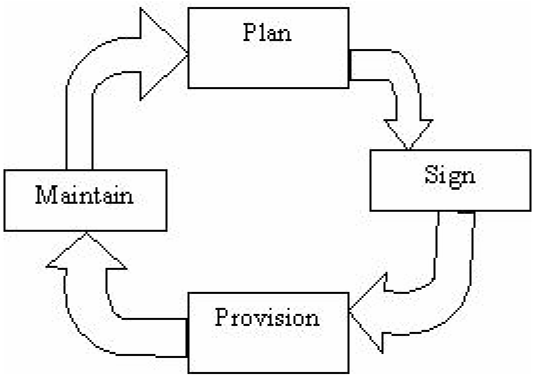


Figure 3.2: Cellular Network Service Life Cycle

The stages of service life cycle are:

- *Plan* – this is the first stage and this is where the target sites are identified in market penetration plan with the actual versus planned implementation success being measured. Then the results are reported, which can be according to region, city or sales representatives.

- *Sign* – the rent rates are analyzed by geographic region to establish pricing benchmarks before negotiating site arrangements with landlords and property managers. Then the agreement is signed with track of all information about leases, buildings, facilities and contacts.

- *Provision* – services are built and implemented into signed buildings. Construction can be managed and status tracked using building, service and/or geographical location. Then sell new services to increase sales and profitability.

- *Maintain* – property managers are notified of lease, insurance expiry dates, up to date contact information, site documentation, SLA details and payment schedule. Then ensure that emergency contacts are up-to-date and readily available to OSS group.

3.2 Cellular Network Service Management

Network services require proper management for easier and better utilization. Network service management revolves around the technical employees who design, develop, monitor, analyze, coordinate, maintain, and optimize the networks. OSS functions performed by network specialists include[4][121]: (1) design and document network solutions necessary for the installation and repair of cellular systems and components by field personnel, (2) monitor, test, optimize, and perform preventive maintenance of cellular networks, (3) coordinate efforts to maintain, repair, and install circuitry and equipment in the networks, including investigating and resolving complex service breakdowns or equipment malfunctions, (4) develop procedures for the documentation, back-up operation, and servicing of the networks; and (5) analyze network operations.

3.2.1 Quality of Service and Security

Quality of Service (QoS) is defined as “a set of quality requirements on the collective behaviour of one or more objects” [124]. All aspects of a cellular network connection including; service response time, loss, echo, signal-to-noise ratio, loudness levels, cross-talk, frequency response, interrupts, etc comprises quality of service requirements.

The ability to control QoS offers operators ways to differentiate themselves from other competitors by being able to launch new revenue-generating services faster and to bundle service packages for different user segments. Absence of this ability remains prevalent with most fixed and wireless ISPs today, making them resort to “first come first served” or “all you can use”’ service models.

In spite of the promising revenue opportunities, 3G wireless operators are facing many challenges: very expensive license cost, market implosion, expensive infrastructure investment, and a new type of market with new players. QoS control can be one of the key relievers against the pressures mounted by these challenges. Cost control, early return on investment, new revenues from high-margin services and high-tier subscribers, brand loyalty, customer retention, and cost reduction via network consolidation are possible benefits for employing QoS control in the network from the beginning.

Service quality can be measured using signal quality, service availability, service reliability, restoration time, and service restorability. Signal quality is mainly represented by the signal-to-noise ratio (SNR), bit error rate (BER), and other factors, and is affected by the transmission equipment characteristics.

Security of network services provided is central to customer attraction and retention. Network services are often vulnerable to a range of security risks such as, spoofing, sniffing, session hijacking, etc. These security risks can be addressed by secure accounting data, secure user authentication, and message encryption.

3.2.2 Management of Level of Service Agreement

Level of Service Agreement (LSA) is a contract between the cellular network service provider and customer that specifies the QoS level that can be expected. It includes the expected behavior of the service and the parameters for QoS. Availability, reliability, latency, service provisioning and customer support are some of the service level parameters that can be measured using the statistical data. Reporting of SLA compliance by customers will ensure that QoS management is up to the standard agreed.

In order to remain competitive, cellular network service providers must offer guarantees not just in terms of availability, but also in terms of performance guarantees such as response time, performance, utilization, security and throughput.

3.3 Issues Affecting Cellular Network Services

The issues that affect provision of quality network services are summarized as follows:

3.4.1 Technical Advances and Network Issues

Technological advances in the transmission of voice, video, and data are fostering fundamental changes in the telecommunications industry. For example, large local telephone companies plan to offer video services in competition with cable and broadcast television, while cable television companies plan to offer local telephone service over their wires in competition with the local telephone companies. Challenges that come with this change include: (1) managing the transition to a more competitive local telecommunications marketplace; (2) ensuring that all consumers have access to affordable services as competition develops; (3) ensuring that the information superhighway provides adequate security, privacy, reliability, and interoperability. (4) Ensuring that previous investments on the obsolete technology are recouped before they are rendered outdated.

With the integration of different networks and technologies, any service to be developed and deployed to the customers must meet the *standards* (ISO, IEEE, etc) that specify rules and criteria to follow in service development. These standards also define the interoperability norms between different networks, as well as determine the complexity of the service to be developed and deployed.

The service contents should be enabled in different configurations for *interoperability*. For instance, the problem of integrating Voice over IP (VoIP) gateways with existing PBXs (Public Branch eXchanges), switches and routers often results in a complex configuration. This is due to a lack of experienced technicians, and the closed architectures of most PSTN network elements.

Scalability as a network issue may be dealt with by increasing the port density of each gateway, as the number of users increase, however sufficient business, service, network and element management tools and processes are not available, making scalability of general operations difficult.

The *unreliability* of the networks i.e. Internet is the main market restraint of VoIP. There are many QoS issues experienced by packet-switched networks that do not affect circuit-switched networks. Acceptable sound quality has become expected on the PSTN, whereas VoIP is an immature technology, experiencing many problems in this area. The Internet is a best-effort network, where variable latencies and dropped packets occur. Because a voice service requires real-time transmission, VoIP often results in a heavily degraded QoS. Unreliability will also be a cause for concerns in areas such as public safety. PSTNs are usually well engineered and very reliable where general services are rarely unavailable, and emergency services are extremely reliable.

Echo is caused by signal reflections in a hybrid circuit that is converting between a four-wire circuit and a two-wire circuit. It is present in all telephone networks, but is acceptable in a circuit switched network because round-trip delays are usually short enough to go unnoticed. In packet networks, the roundtrip delay is almost always noticeable, and an annoying reflection of the speaker’s voice can be heard in his ear. Although echo-cancellation can be used to filter out the echo, this is often insufficient, usually with only a prediction of the actual echo removed.

Talker-overlap – occurs when one-way delay exceeds about 200msecs.

3.4.2 Regulatory and Policy Issues

Each and every country has laid down rules by Telecommunication Acts that guard the network services being provided to its members. Most of these acts differentiate the *Basic* and *Enhanced* network services types. *Basic Services* are "the transmission of information the user is choosing without change in the form or content." Even this is still in most cases subjected to common carrier regulation. *Enhanced Services* are information services that involve "transforming, processing, retrieving, utilizing or making available information via telecommunications." Common carrier regulations may not apply.

Telecommunication services are also handicapped by *political concerns* and *worries about national security, social order and international relations.* The Government and other organization requirements in the form of *laws*, *policies*, *regulations*, *standards*, *guidelines*, *directives*, *communications*, *orders* and/or other types of documents for services accessibility.

3.4.3 Financial Issues

It is very expensive to install certain telecommunication equipment. Even Governments of various countries, mostly developing countries find it enormously expensive to install such equipment. For example, ISPs are usually accessible in industrial and business locations or urbanized areas but their extension to rural areas and remote places is costly and tedious, thus limiting access for poorer and under–privileged citizens to Internet services. The cost of developing and operating a network is very expensive forcing them to charge high prices in order to reap back the benefits.

Traffic is doubling about every five to ten months i.e., Internet, thereby forcing most ISPs to increase prices to keep up with the growing demand. *Poor infrastructure* and even sometimes lack of it lead to poor provision and no provision of services respectively. The *financial befits* normally arise from increased use of the services (i.e., increased use of websites, SMS, etc), direct cost savings, cost considerations of initial costs, and on-going costs/operation costs, etc.

Companies often cite reusability and faster development time factors that translate to short and long-term cost saving. For instance, Integration projects are very expensive, and web services will help reduce costs by 30% or more. The investments in using XML and web services are incremental in most cases. Reaching out to new customers brings in new revenue streams. This is one of the major reasons that firms can justify their investment in e-business standards.

The value of any telecommunications network increases exponentially according to the *number of users/computers* attached to it [123]. For example, as more and more IP equipment is attached to the Internet, the benefits derived from being part of a VoIP network will increase, fostering further demand.

The *source of capital* for the establishment of new services may affect services. The source can be internal where local banks can lend money for such investment or external where foreign investors can come in but this is determined by other factors, i.e., whether intellectual property rights are honored within a stable system of law.

3.4.4 Customer and Cultural Issues

The diversity in language, low literacy rate and other cultural factors may also affect telecommunication services being offered across the World. For example, Internet services need high literacy rate. A highly *stratified culture* is not easy for services to penetrate through.

A strong *nationalistic nation* that tends to resist 'foreign' influences including the provision of information and communication originating from elsewhere will greatly affect provision of such services. Other cultural issues include, stable and viable legal framework within which intellectual property rights, privacy and patents can be protected; vertical authority relationships that foster small, networked, cooperative arrangements among firms; Trust where business relationships can easily be developed outside tribes, families and other social institutions; and lastly the understanding of the concept of information either as a public or private good.

Customer requirements, which include up to date technology compliant services that can serve diversely demanding inter-network operations may be affected by lack of inter-network cooperation, i.e., roaming services offered by mobile networks. The supply affects the service provision positively when for instance, the cost of electronic equipment decrease then a lot or same quantity and quality can be gotten at cheaper prices. This leads to saving on cost of purchase [122]. The cost of wavelength is predicted to go down substantially [123]. The volume of traffic will always affect the cost. It is always high during the working hours of the day say from 0800HRS to 1700HRS. For someone to use any of the services at this time, you have to pay at a higher rate as compared to other hours thereby affecting the use of such services at that time.

3.4.5 Speed, Service and Quality of Service (QoS) Issues

The speeds of some of the services do affect the provision and preference of services by customers. For example, customers who play games (infotainment service) prefer faster broadband access and therefore are dropping dialup as a means of Internet access. The server load and network type employed affect the speed of the service, i.e., IP networks experiences propagation delay, network delay, accumulation delay and processing delay.

The s*ervice development and maintenance time* – most integrated projects that trigger the use of standards are inter-organizational information and strategic information systems. Therefore, the decision to adopt these standards is of strategic importance. Faster development cycles, an often-quoted benefit of using these standards, enable firms to gain a competitive edge because of shorter time-to-market of new products and services.

The degree in which a service is perceived as difficult to understand and use is known as *service complexity*. The more complex service development is, the more time it takes to develop it out and the more it becomes expensive to deploy and maintain. This in turn affects its provision to the customers. Standards are one of the factors that affect the complexity of a service in that a complex standard, leads to a more complex service.

The *hardware and software* used will always determine the quality of services that are being offered over any platform/network. More robust software will always deliver high quality services to customers. However, hardware is constantly improving and falling in price, compression techniques are becoming more sophisticated, and underlying transport mechanisms are advancing the speed of data transfer.

3.4 Relate Faults to Services

In this section, the relationship between network services and four network faults encountered by a certain cellular network service provider is presented.

3.4.1 Transmission

Transmission fault occurrence is highest at 75% (cf. Appendix B) among the four network faults studied. Transmission fault occurs as a result of Mobile Switching Center (MSC) failure, switch failure, software failure, etc. If for example, transmission fault occurs at the MSC, the whole area covered by this MSC will not offer any service. SMS, VoIP, Voice message, and many other services will be affected. Hence transmission fault may affect all services that emanate from the area affected. It might not only affect services at this area but also cause more problems to the whole network i.e., congestion, loss of packets, etc. The location-based services at this area will definitely be affected. Services routed through the area affected can be re-routed through the unaffected areas.

3.4.2 Power

Power fault causes more impact than any other network fault (cf. Appendix B). It causes an unplanned outage. This outage must be reversed in the quickest time possible either by restoring the power supply or by using the alternative power sources, i.e., generator. During this period of blackout, the network will be totally unusable. It will affect all services that originate from that particular area, the services terminating at that point (some can be retrieved later once the power is restored if the MSC is not affected i.e., voice mail, SMS, etc).

3.4.3 Multiplexer

Network links normally employ equipment like repeaters, bridges, gateways, and multiplexers. Multiplexer is a device for taking several separate digital data streams and combining them together into one data stream of a higher data rate. This allows multiple data streams to be carried from one place to another over one physical link, which saves cost. However, if multiplexer malfunctions or fails to work, this cost may not be saved. Real-time services are the most affected. Path blockage and routing problems will occur as a result of links being ineffective to function. Packets will have to find an alternative routes leading to network congestion. In the long run the fault will have an impact on the whole network.

3.4.4 Cell

An area covered by the entire network is divided into smaller units called cells. Cells may encounter power failure, links breakage, and even transceiver may sometimes be knocked out of operation. If this does happen then the services emanating and terminating at that cell will be affected. A service terminating at the affected cell may be stored (i.e., SMS, voicemail, etc) and sent to the recipient after the restoration of the cell or when the customer moves to another working cell. Cell is the least fault to occur (in terms of frequency of occurrence) (cf. Appendix B) in this cellular network service provider under study. It is represented by DN/BEADN in the figures shown appendix B.

3.5 Cellular network services dependency

When a component requires a service performed by another component in order for it to execute its functions, this relationship between the two components is called dependency. The detailed description of network service dependency giving definition, types and benefits is already presented in Chapter 2, Section 2.6.

In order to drive the problem solving process that is cellular network faults prediction, a model of cellular network faults, or a concept of services and dependencies between them is required. A cellular network service can be defined as a set of functionalities, which are offered by a cellular network service provider to a customer at a customer provider interface (i.e., mobile handset) with an agreed quality of service. A service can depend on one or more resources and a resource can be used by one or more services. To ensure high quality of services is provided, it is necessary to react accurately to faults occurring in one or more components that provide such resources. This can be achieved by determining the dependencies between different services, dependencies between services and resources and dependencies on the resource level. It will be better also to bring to clarity what an interface and a component stand for. *A component* is a non-trivial, nearly independent, and replaceable part of a cellular network system that fulfils a clear function in the context of a well-defined architecture. A component conforms to and provides the physical realization of a set of interfaces. An *interface* is a collection of operations that are used to specify a service of a component. It focuses upon the behaviour, not the structure of a given service. Figure 3.3 shows a model of cellular network environment showing dependency.

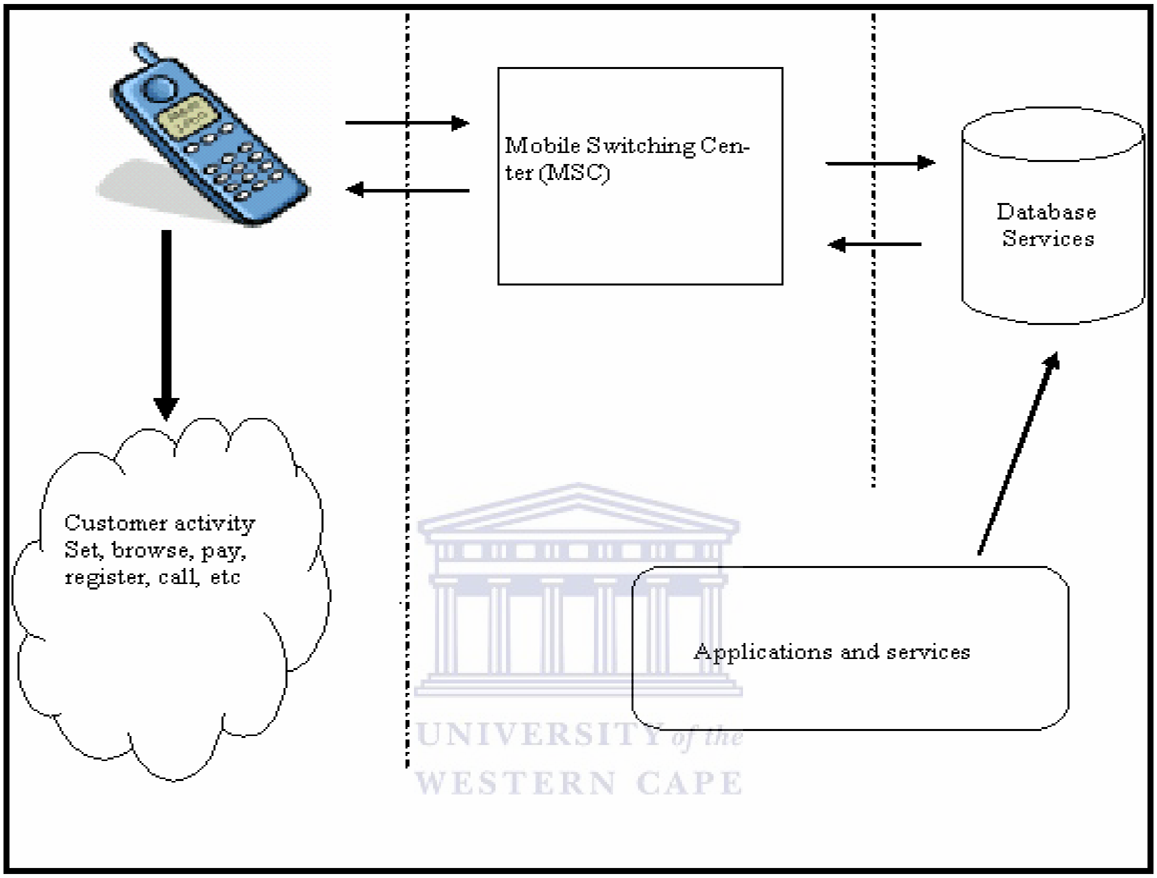

Mobile Switching Center (MSC)
Database Services
Customer activity Set, browse, pay, register, call, etc
Applications and services

Figure 3.3: Model of Cellular network Environment showing Dependency

3.5.1 Dynamics and Dependency Life Cycle

3.5.1.1 Cellular Network Service Dependency Dynamics

Cellular network service dependency changes as variables within the cellular network setup changes. The changes are normally caused by resources (network components, services, power, etc) becoming unavailable due to network faults, resources may migrate, or may be upgraded. In a cellular network, the components and/or managed objects that represent the resources may be many. The change of dependency that may occur as a result of fault in a cellular network is termed as *cellular network dependency dynamics*.

Cellular network services can be modeled as node, communication and precedence constraints between services as directed edge and the model can be expressed as a

Directed Acyclic Graph (DAG). Let the service be *S*, $S = \{s_1, s_2, s_3, ..., s_N\}$. A complex cellular network system may offer *N* number of services. A service depends on resource(s) *R*, where $R = \{r_1, r_2, r_3, ..., r_j\}$ as is the shown diagram in Figure 3.4a. A resource can be network link, component, or other services. An edge between two services, S_a and S_b is given by S_{ab}, which expresses the dependent relation between S_a and S_b. Given service S_N the set of parent services is denoted as $pred(S_N)$, and the set of children services is denoted as $succ(S_N)$. A service S_N is called entry service if $| pred(S_N) |= 0$ and an exit service if $| succ(S_N) |= 0$. Therefore a network with *N* services depending on *R* heterogeneous resources. It is essential to map the set of *N* services in the DAG into *R* heterogeneous available resources in order to avoid the faulty components supporting the resources required by the services.

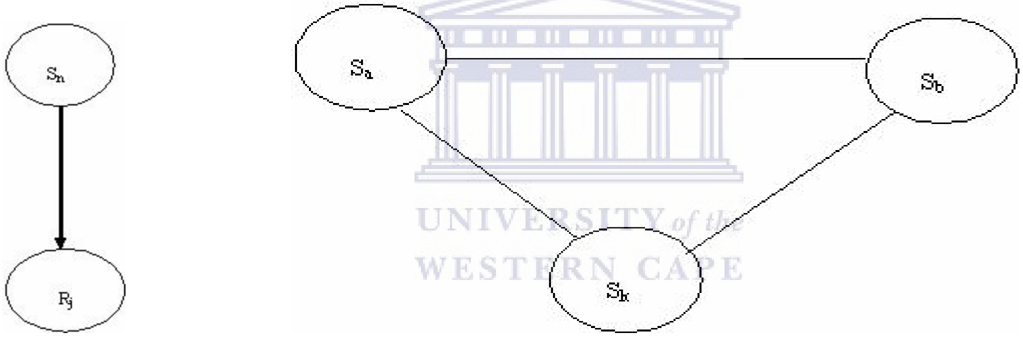


Figure 3.4: a) Service and Resource Dependency (Right) b) Services Dependency

In explaining the cellular network dependency dynamics, the dependency relation between S_a and S_b may change, for example, if S_b malfunctions then S_a (will also fail in normal circumstances) but in this case S_a may use another service say S_k, which offers the same resources for its operation as shown in Figure 3.4b. Also if S_a depends on a particular route (link) R_l then with the failure of R_l, S_a is also expected to fail. But this may not be the case because S_a may use another route to complete the execution.

The system implementation takes into consideration this dependency dynamics with the cellular network system. It therefore means that the dependency models presented in this thesis are dynamic in nature and robust. For a system to fail, it means all the alternative dependencies are exhausted. The main causes of dependency dynamics include: cellular network faults, which may cause the cellular network resources to appear and disappear during the system lifetime; deployment of new sub-systems; change of resource availability; re-negotiation of new service level agreements, etc.

However, it is worth noting that most of the dependencies are fairly permanent and only change when there is deemed fault with one of the main antecedent. This is the main interest in studying how cellular network faults may change the dependency and its subsequent effects on the reliability of the cellular network services.

3.5.1.2 Cellular Network Dependency Binding

The three main variables, faults, F where $F \rightarrow \{f_1, f_2, f_3,..., f_i\}$, resources, R, where $R \rightarrow \{r_1, r_2, r_3,..., r_j\}$, and services, S, where $S \rightarrow \{s_1, s_2, s_3,..., s_N\}$ depicting the relationship between them, which can be one-to-one, one-to-many and many-to-many. The dependency only exists when cardinality between dependent and antecedent is exactly one. The maximum cardinality between the objects is infinity. The relationship between the variables can take either of the following two sets:

(a) The network faults relates to resources directly as follows:

i. A set of faults can affect a set of resources in the network, $F \rightarrow R$

ii. A set of faults can affect one or a particular resource in the network, $F \rightarrow r_j$

iii. One or a particular fault can affect a set of resources in the network, $f_i \rightarrow R$

iv. One or a particular fault can affect one or a particular resource in the network, $f_i \rightarrow r_j$

(b) The network services depend on network resources directly as follows:

i. A set of services depend on a set of resources in the network, $R \leftarrow S$

ii. A set of services depend on one or a particular resource in the network, $r_j \leftarrow S$

iii. One or a particular service depends on a set of resources in the network, $R \leftarrow s_N$

iv. One or a particular service depends on one or a particular resource in the network, $r_j \leftarrow s_N$

A binding which can be static or dynamic would occur with the knowledge of cardinality. A *static binding* is where the dependency bindings cannot change at run time and the dependent service is guaranteed to be present the entire time the resource is available, whereas *dynamic binding* is where the dependency bindings can change at run time and service availability cannot be guaranteed. Network services would be affected differently by different types of bindings as summarized in Table 3.1:

Table 3.1: Different types of service dependency binding

Binding Type	**Semantics of the dependency type**
One-to-One, static	A service is bound to one resource, any change invalidates the service.
One-to-One, dynamic	A service is bound to one resource; changes do not invalidate the service as long as it can be bound to another resource.
One-to-Many, static	A service is bound to at least one resource, any change invalidates the service.
One-to-Many, dynamic	A service is bound to at least one resource; changes do not invalidate the service as long as the binding count is not zero.
Many-to-Many, static	A set of services are bound to a set of available resources at the time of binding, changes invalidates the services.
Many-to-Many, dynamic	A set of services are bound to a set of all available resources at the time of binding, as resources become available/unavailable they are bound/unbound to/from the services, the services never becomes invalid.

3.5.1.3 Cellular Network Service Dependency Life Cycle

Cellular network environment is very dynamic in nature and so the dependency evolving through FIVE phases, referred to as *dependency lifecycle*. The phases include [185]:

1. **Initiate**: This is the initial phase where the dependency is initiated when service consumption is signaled. The initial parameter values are received at this stage. For example, when you initiate a call, first the signal is acquired to have connection to the

MSC, and then database connection is initiated to establish whether you have enough units to continue with the service consumption.

2. **Acquire**: existing services, resources, and common (known) faults are acquired by the dependency for mapping purposes at this stage. New and old dependencies are ascertained mainly for consumption purposes, i.e., after service, S_a signaled service, S_k for dependency and received positive answer, then it acquires the resources in readiness for the dependency mapping to complete the service consumption through S_{ak}.
3. **Start Map**: This stage triggers the start of dependency mapping.
4. **Map**: new resources may be added to the dependency pool during this phase. Existing resources and dependency parameters may be removed or updated to ensure the dependency dynamics are maintained. Dependency mapping can be affected by these changes, and so must be resolved continuously for a robust cellular network system.
5. **Stop**: The dependency is terminated at this stage.

The dependency life cycle continues by initiating another dependency. The semantics of the dependency are implemented in the system, which correlates the network services to network faults. The service dependency life cycle is shown in Figure 3.5.

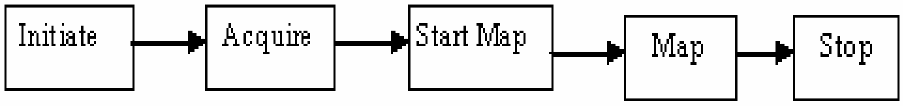


Figure 3.5: Dependency Life Cycle

3.6 Cellular Network Service Dependency Models

A cellular network environment can be logically modeled as layers of resources (i.e., services, applications and other software and hardware components) that cooperate to deliver an end-to-end service. Services or components in one layer depend on functions provided by components in a lower-supporting layer. Failures occurring in one layer affect the functioning of dependent components in another layer. The dynamics of service dependency are considered for cellular network faults prediction purposes, because significant changes in the overall system behaviour are detected through emerging or disappearing dependencies. An understanding about network resources is important in service dependency modeling as explained in the following section.

3.6.1 Network Resources

Any physical or virtual component in a network system with limited availability may be referred to as a network resource. It can be any internal or external device connected to a computer system.

Network resources can also be referred to as various parts of the network (hardware and software) which support each other by combining or individually to provide specific functions within the network environment. Network service is created by these functions.

Table 3.2: Summary of Faults, Resources and Services correspondingly

Faults	**Resources**	**Common Services**
Multiplexer	Link (lines), electrical power, Multiplexer adaptor, External Bus Interface (EBI) cable, conversion kit, port module, etc	Voice, VoIP, Videoconferencing, etc
Power	Generator, Electrical power sources, Electrical switches, transmission lines, etc	SMS, MMS, Voice, VoIP, Email, etc. Virtually all services are affected.
Transmission	Link, cables, multiplexer, network name resolution, ISDN switches, ISDN lines, Gateways, etc	Affect real-time services Affects services in a serial connection May delay services such as SMSs, etc
Cell	Link, Electrical power, cables, multiplexer, etc	VoIP, SMS, MMS, Email, Internet, Video conferencing, etc
Time Out	Link, RAM, multiplexer, etc	SMS, MMS, Email, VoIP, Video conferencing, etc
Run Time Error	RAM, Link, Switches, etc	VoIP, voice, Video conferencing, etc
Out of Range	Signals, wireless access point, Internet, etc	Voice, VoIP, email, SMS, MMS, etc

Network resources are network elements that support services. A network resource can be basic or elementary. *A basic* network resource is the smallest element that supports a network service. It is scalar in nature and cannot be split further down. It supports the service with all its parts as a whole. A combination of two or more basic network resources to offer a function to a service is called *elementary* network resource. An elementary network resource cannot be used without any of the basic element parts. Table 3.2 above summarizes the list of faults, resources and services correspondingly.

Network elements (resources) support services through service access points (SAP) and port accesses. The services dependency is modeled keeping in mind the current changes happening to the network environment to proactively detect faults before they impact end users. Ontology is implemented in the system which facilitates the mapping of faults and services to list faults that are likely to cause network services failure.

3.6.2 Network Service Availability Models

One of the main aims of this work is to develop a reliable service based OSS where services would be available to users whenever they want to consume them. Availability of network services depends on availability of network resources to support them to carry out their functions. For example, an end-to-end service availability would depend on availability of service source, network, link, and availability of the destination device. Network service availability is a combined availability of the network parts (elements) supporting the service(s). The combined availability is a product of the availability of all the network parts involved. This can be defined as:

$$SA = S(s) * N(s) * L(s) * Sw(s) * D(s) \qquad \text{(Equation 3.1)}$$

Where SA – Service Availability

$S(s)$ – Availability of service source

$N(s)$ – Availability of network

$L(s)$ – Availability of link

$Sw(s)$ – Availability of software

$D(s)$ – Availability of destination device

Equation 3.1 also means that the combined availability of the network is always lower than the availability of its individual components (resources). It is important to note that when network is available, the services being offered will also be available and vice-versa. Therefore, network availability directly impacts on service availability. Simply put,

$$NA = RA = SA \quad where \quad p(F) = 0 \qquad \text{(Equation 3.2)}$$

Where NA – Network Availability

RA – Resource availability

SA – Service availability

$p(F) = 0$ - is a probability of fault occurrence is 0 indicating fault-free network

The network availability at time, t for network service, s may be defined in terms of several parameters that includes network reliability $R(t,s)$, network maintainability $M(t)$ and fault effects $F(t)$ where,

$$NA(t,s) = R(t,s) + F(t,s) * M(t,s) \qquad \text{(Equation 3.3)}$$

Where $F(t,s) = 1 - R(t,s)$ (Equation 3.4)

However, the network reliability depends upon the reliability of many components that make up the network; i.e., network link, power, software, switches, and services. These set of components (resources) can be represented by $R \rightarrow (r_1, r_2, r_3, ..., r_j)$. Rewriting Equation 3.4 to know faults effects at time *t*, for service *s*, is

$$F(t,s) = 1 - (R(t,s))^j \qquad \text{(Equation 3.5)}$$

A given set of resources consisting of R members can be constructed with $R_\pi (\pi = 1,2,3,..., j)$ homogenous sub-populations.

$$\sum_{\pi=1}^{j} R_\pi = R \qquad \text{(Equation 3.6)}$$

A homogenous sub-population R_π is defined by the verifiable assumption that its members exhibit the same probabilistic decision behaviour. However, these set of resources are affected by network faults. Network faults are errors that occur frequently within the network elements impairing their operations. Network faults may render the network elements unusable or partially working thereby diminishing partly or wholly the resources ability to carry out its functions depending on the faults impact.

A set of network faults $F \rightarrow (f_1, f_2, f_3,..., f_i)$, can affect the network resources, R:

$$\alpha : F \rightarrow R \qquad \text{(Equation 3.7)}$$

Where the domain α is the set *F*, the target of α is the set *R*

The range or image of α, written rng α, is

$$rng\alpha = \{r \in R \,|\, (f, r) \in \alpha \;\; for \quad some \quad f \in F\}$$

$$= \{r \in R \,|\, r = \alpha(f) \;\; for \quad some \quad f \in F\} \qquad \text{(Equation 3.8)}$$

Therefore the function has its range of resources as the target of network faults given by $rng\alpha = R;$ that is every $r \in R$ is of the form $r = \alpha(f)$ for some $r \in R$. Equivalently for any $r \in R$, the equation $r = \alpha(x)$ has a solution $x \in R$. The affects on network resources are transferred to network services with the function:

$$w : R \rightarrow S \qquad \text{(Equation 3.9)}$$

The composition of α and w is the function

$$\alpha \circ w : F \rightarrow S \qquad \text{(Equation 3.10)}$$

Equation 3.10 is defined by

$$(\alpha \circ w)(f) = \alpha(w(f)) \;\; for \quad all \quad f \in F \qquad \text{(Equation 3.11)}$$

The simulation results are provided in Chapter 6.

3.7 Summary

Network services are the ‘commodity’ that network service providers are ‘selling’ to the customers. This is the connection point between these two parties and the agreement based on price; service quality level and other terms must be explicitly defined.

In this Chapter, the classification, characteristics, and management of network services were discussed. The issues affecting cellular network services were highlighted. The relationship between cellular network faults and services were also discussed. The network service dependency basics, types, benefits and dynamics were discussed. The network service dependency models were presented. In the next chapter, cellular network faults prediction using Bayesian network models is presented.

Chapter 4

FAULTS MODELING USING BAYESIAN NETWORK

In this Chapter, classification of cellular network faults is provided in Section 4.1. Some of the cellular network faults models are provided in Section 4.2. Section 4.3 describes the assumptions for models application. Cellular network faults modeling processes are presented in Section 4.4. Section 4.5, provides cellular network fault prediction using Bayesian networks. Characteristics of a good model are explored in Section 4.6. Section 4.7, explains the construction of the structure of the Bayesian network. Section 4.8, explores factors affecting the performance of Bayesian networks with summary being drawn in subsequent Section.

4.1 Classification of Faults

In addition to the definitions given in Section 2.1.1, a fault is regarded as an abnormal and/or an accidental condition that is either caused by a defective network element, problem in the network layer or at sub-system level. Such problems often cause a previously functional unit to fail. A cellular network fault can also be viewed as a defect that causes a reproducible or catastrophic malfunction. A reproducible malfunction is one that occurs consistently under the same circumstances.

Cellular network faults may also cause malfunctions and outages. Malfunctions are mostly experienced when the software and hardware are working with some errors. However, outages are often manifested when the software and hardware are completely knocked out; they will not be working at all. When this occurs, ESPs will not only lose revenues but also the customers may shun away. However, in case of outages, ESPs may device contingent plans that are meant to improve services by ensuring that the duration of each outage is kept to minimum time possible.

Cellular network faults are classified as *malfunctions* and *outages*. The model gives an overview of cellular network faults types that are commonly experienced by a cellular network service provider under study. Figure 4.1 depicts the classification of cellular network faults.

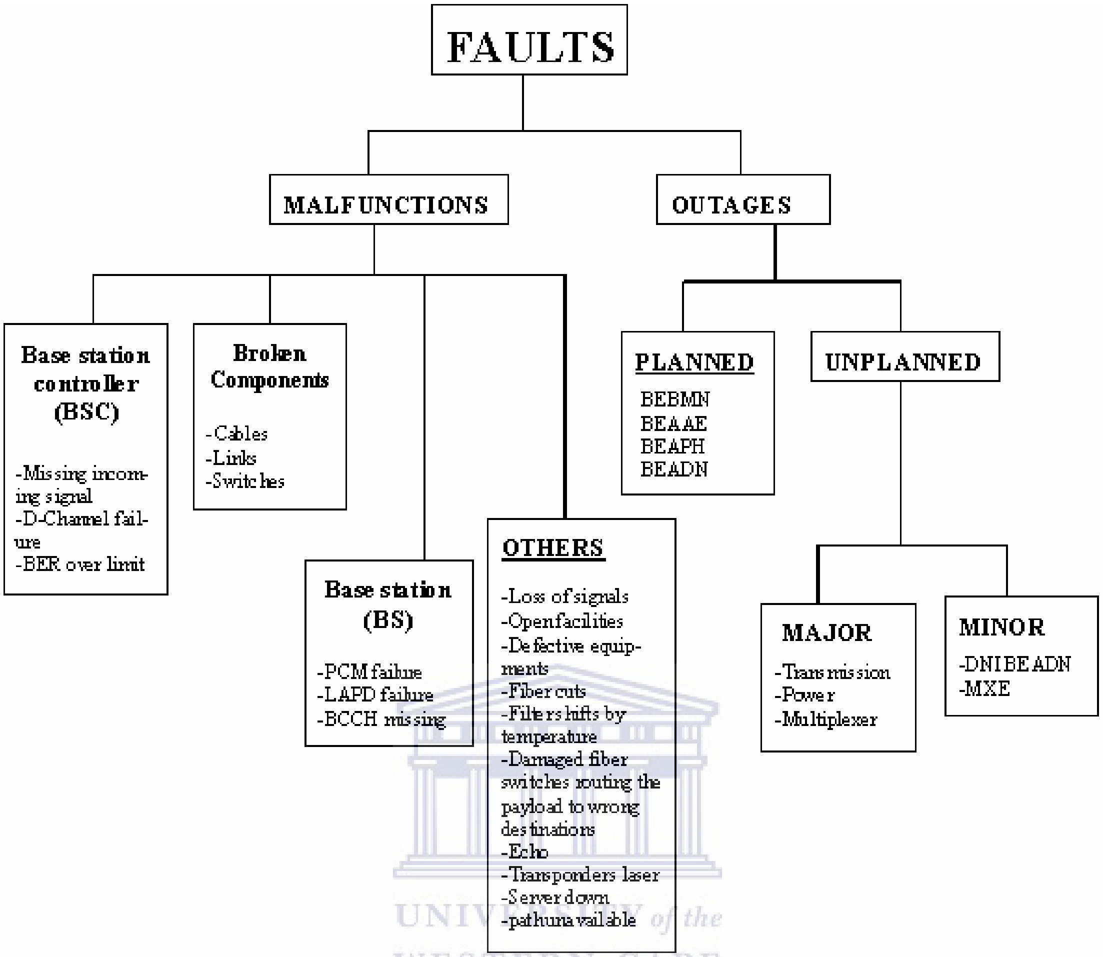


Figure 4.1: Classes of cellular network faults

4.2 Faults Modeling Using Bayesian Network

In cellular network systems, an error may occur as a result of either malfunctioning of multiple components or just one component failure which replicates itself over the network. This kind of failure is often common and sometimes very difficult to detect. The network service dependency (c.f. Section 3.5) shows how network faults can affect the network resources, which in turn affects the quality of services provided. Network faults can be modeled using Bayesian networks.

4.2.1 Why Bayesian Networks?

The main reasons why Bayesian networks was chosen for network faults prediction include [111][126][127]:

- *Mathematical support*: the Bayesian networks count on a solid mathematical support, which allows the analysis of the model in view of the knowledge of its performance and precision before an implementation is carried out.
- *Robustness*: approximate answers can be obtained, even when the existing information are incomplete or imprecise whenever new information become available, the Bayesian networks allow a corresponding improvement in the precision of the correlation results.
- The facilities are readily available for the construction of the Bayesian network.
- Bayesian networks have the capacity to identify, in polynomial time, all the conditional independence relationships that are extracted from the information gained by the Bayesian network structure.
- The capacity for non-monotonic reasoning, through which previously obtained conclusions may be withdrawn as a consequence of the knowledge of new information.
- The capacity to carry out inferences on the present state of telecommunication networks from the combination of: a) statistical data empirically surveyed during the network functioning, b) subjective probabilities supplied by specialists, and c) information (that is, “evidences” or “alarms”) received from the telecommunications network, in real time.
- It is simple and effective.

4.2.2 Concerns about Bayesian Networks

Although the use of Bayesian network for knowledge generation from data produces good results on some benchmark data sets, there are still some concerns. These include [131][132]:

- *Node ordering requirement* - many Bayesian network learning algorithms require additional information, which is mostly an ordering of the nodes to reduce the search space. Unfortunately, this information is not always available.
- *Computational Complexity* - practically all Bayesian network learners are slow, both in theory and in practice. For example, most dependency-analysis based algorithms require an exponential numbers of "conditional independence" tests.
- *Lack of publicly available learning tools* - although there are many algorithms for this learning task, very few systems for learning Bayesian networks systems are publicly available. Even fewer systems can be applied to real-world data-mining applications where the data sets often have hundreds of variables and millions of records.

4.2.3 Application Areas of Bayesian Networks

Bayesian networks have been utilized in solving problems in several areas in which uncertainty is a key factor. These areas include [131][132]:

- Fault diagnosis in optical communications networks, diesel engines, etc.
- Establish the location of a fault in complex devices such as aircrafts, trains, etc.
- Retrieval of information, according to users' area of interest.
- Debugging of complex computer programs.
- Coding, representing and discovering knowledge, through some processes that seek new knowledge on a given domain based on inferences on new data and/or on the knowledge already available.
- In proactive detection of abnormal behavior in a computer network.
- Bayesian techniques are integrated into automated categorization of unwanted spam, the formation of medical diagnoses, virus detection, and several other uses that advance compatible business objectives.

4.2.4 Basics of Bayesian Networks

Bayesian networks are Directed Acyclic Graphs (DAG) with nodes representing random variables and edges denote the existence of direct causal influence among the variables connected by them. Conditional probability is used to quantify the level of influence that these variables exert on each other. In most cases the causal connections in the Bayesian networks are not absolute. Due to the complexity inherent to the system under study, it is convenient when it is difficult to attribute deterministic causal relations among the variables.

A random variable may be *continuous or discrete*. It is discrete when it has a finite or countable number of states and continuous when the number of possible states is infinite. For instance, a random variable denominated “Power Voltage” may be defined as being continuous, if it can assume any value in the range of 0 to 250 volts, for example, while “Multiplexer Switch”, whose possible states may be on or off is a discrete variable (and also binary, since it has only two possible states).

A probability distribution is a function that assigns probabilities to events or propositions. A distribution is called a discrete distribution if it can be defined using a countable set of non-continuous values, for instance, using a specific subset of the integer values. A distribution is called a continuous distribution if it has a continuous function. Polynomial and exponential functions are two examples of continuous functions. Distribution curves provide important information about the possible values that can be assumed by a random variable. For example, if X is a random variable, corresponding probability distribution assigns to the interval [a, b] the probability $p[a \leq X \leq b]$, i.e., probability that variable X will take a value in the interval [a, b]. Probability distribution of the variable X can be uniquely described using a cumulative distribution function $F(x)$. Using mathematical model $F(x)$ can be defined as shown in equation 4.1:

$$F(x) = p[X \leq x] \qquad \text{(Equation 4.1)}$$

A continuous probability density distribution can be expressed by probability density function (pdf) which is a non-negative Lebesgue integrable function *f* defined on the reals such that:

$$p[a \leq X \leq b] = \int_a^b f(x)dx \qquad \text{(Equation 4.2)}$$

4.2.4.1 Joint Probabilities Distribution

A combination of values of a random set of variables may be used to define a specific configuration. For example, let X be a set of random variables$\{X_1, X_2,..., X_n\}$. The number of possible configurations is given by the product$n_1 * n_2 * ... * n_n$, where $n_1, n_2,..., n_n$correspond to the number of possible states for each of the variables$X_1, X_2,..., X_n$. The number of possible combinations is 2^n in this particular case, where$X_1, X_2,..., X_n$represents binary values. The joint probabilities distribution of X is defined as$p(x_1, x_2,..., x_n)$, for all the possible configurations. Therefore, if$X = \{X_1, X_2\}$, where X_1 and X_2 are binary variables, the joint distribution will contain the following probabilities, corresponding to four possible configurations:

$$p(x_1, x_2), p(x_1, -x_2), p(-x_1, x_2), p(-x_1, -x_2). \qquad \text{(Equation 4.3)}$$

The sum of these probabilities must be equal to 1. This means that these probabilities are exhaustive and mutually exclusive. Apart from that, the probability associated with the occurrence of at least one among two possible configurations$A$ and $B$is equal to the sum of the probabilities associated with A and B, that is,

$$p(A \bigcup B) = p(A) + p(B) \text{ and } p(A \bigcap B) = 0. \qquad \text{(Equation 4.4)}$$

4.2.4.2 Local Probabilities Distribution

A joint probability distribution $p(X_1, X_2,..., X_n)$ is represented by a set of variables $X = \{X_1, X_2,..., X_n\}$in a Bayesian network. This distribution has the advantage of reducing the number of possible scenarios that should be specified. The reduction is achievable because of a set of local probabilities associated to each variable constituting a Bayesian network. It also specifies the level of independence associated with each of the variables indicated in the network structure. From this information, it is possible to construct an equation that represents the joint distribution. Figure 4.2 below shows an example of a Bayesian network with 16 nodes corresponding to discrete variables of two or three states each. The three states are *normal*, *uncertain* and *abnormal*. In this case, not considering the causal independence information built into the Bayesian network, the number of probabilities to be considered would be (2 x 3 x 3 x 3) ^{4} = 8503056. However, only 158 (i.e., (2+3+18+9) + (6+9+18+9) x 3) probabilities must be specified. This is as a result of Bayesian network structure making explicit causal independences.

Consequently, the complexity associated to the construction of a Bayesian network, which is measured by the number of probabilities to be specified, grows linearly with the number of network nodes. Lack of information on causal independences would make the number of probabilities to be specified grow exponentially in relation to number of nodes. In a case study of data from a certain cellular network service provider, Power (Po), Cell (C), Multiplexer (M) and Transmission (T) faults form *n* nodes with Power (having good, weak and blackout as states), Cell (normal, uncertain and abnormal states), Multiplexer (Ok and faulty states) and Transmission (normal, uncertain and abnormal states).

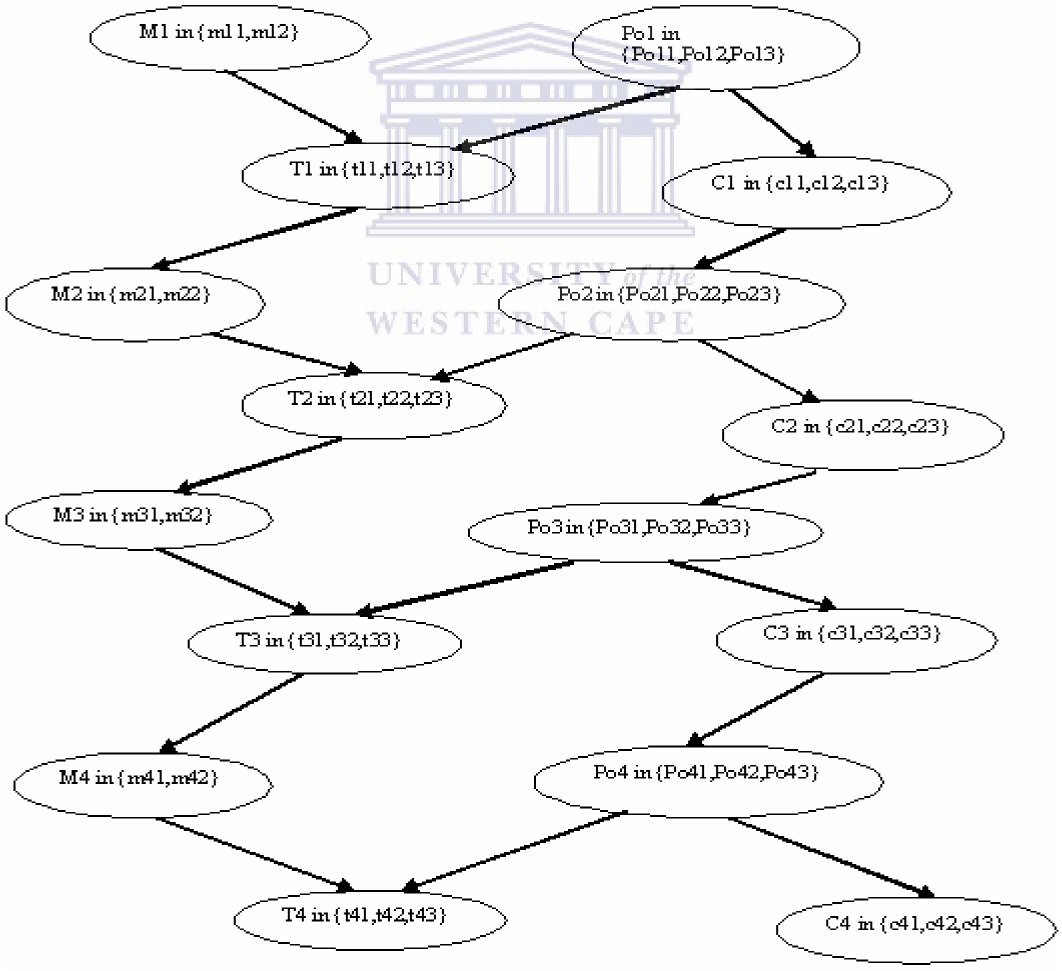


Figure 4.2: Example of Bayesian network with 16 nodes

In the study, the network fault variables taken included Power, Multiplexer, Transmission and cell. The above network shown in Figure 4.2 was formed using the four network fault variables. However, by taking the simplest network as in Figure 4.3 below, which considers only two network fault variables i.e., Power (Po) and Multiplexer (Mux). The reasoning here is based on the simple understanding that if there is power outage or any other problem, then the multiplexer will be affected. This is because a multiplexer requires power in order to discharge its functions which principally involves transmission of signals. The relationship determines its operational status. At each instant of time, the multiplexer’s operational status may either be normal, uncertain or abnormal. The operational status assumed by a multiplexer will depend on the Mux and Power states. Prior probabilities of Power (Po), Cell (C), Transmission (T) and Mux network fault variables were computed as 0.36%, 0.42%, 76.57% and 22.63% respectively (cf. Appendix A). Using this information, the conditional probabilities for each possible outcome given the states of Power and Multiplexer can be computed. Sufficient power supply is one of the factors that ensure multiplexer functions well. If the voltage of power supply is measured, the conditional probabilities are given as shown in Figure 4.3.

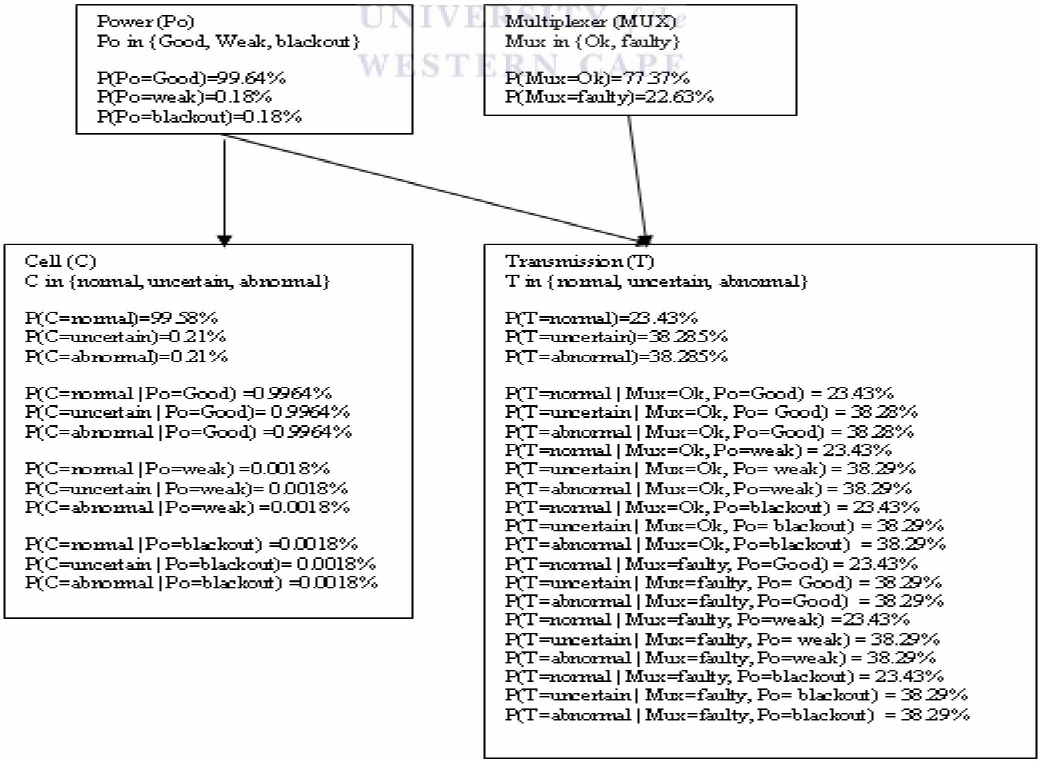

Power (Po)
Po in {Good, Weak, blackout}
P(Po=Good)=99.64%
P(Po=weak)=0.18%
P(Po=blackout)=0.18%
Multiplexer (MUX)
Mux in {Ok, faulty}
P(Mux=Ok)=77.37%
P(Mux=faulty)=22.63%
Cell (C)
C in {normal, uncertain, abnormal}
P(C=normal)=99.58%
P(C=uncertain)=0.21%
P(C=abnormal)=0.21%
P(C=normal | Po=Good) =0.9964%
P(C=uncertain | Po=Good)= 0.9964%
P(C=abnormal | Po=Good) =0.9964%
P(C=normal | Po=weak) =0.0018%
P(C=uncertain | Po=weak)= 0.0018%
P(C=abnormal | Po=weak) =0.0018%
P(C=normal | Po=blackout) =0.0018%
P(C=uncertain | Po=blackout)= 0.0018%
P(C=abnormal | Po=blackout) =0.0018%
Transmission (T)
T in {normal, uncertain, abnormal}
P(T=normal)=23.43%
P(T=uncertain)=38.285%
P(T=abnormal)=38.285%
P(T=normal | Mux=Ok, Po=Good) = 23.43%
P(T=uncertain | Mux=Ok, Po= Good) = 38.28%
P(T=abnormal | Mux=Ok, Po=Good) = 38.28%
P(T=normal | Mux=Ok, Po=weak) = 23.43%
P(T=uncertain | Mux=Ok, Po= weak) = 38.29%
P(T=abnormal | Mux=Ok, Po=weak) = 38.29%
P(T=normal | Mux=Ok, Po=blackout) = 23.43%
P(T=uncertain | Mux=Ok, Po= blackout) = 38.29%
P(T=abnormal | Mux=Ok, Po=blackout) = 38.29%
P(T=normal | Mux=faulty, Po=Good) = 23.43%
P(T=uncertain | Mux=faulty, Po= Good) = 38.29%
P(T=abnormal | Mux=faulty, Po=Good) = 38.29%
P(T=normal | Mux=faulty, Po=weak) =23.43%
P(T=uncertain | Mux=faulty, Po= weak) = 38.29%
P(T=abnormal | Mux=faulty, Po=weak) = 38.29%
P(T=normal | Mux=faulty, Po=blackout) = 23.43%
P(T=uncertain | Mux=faulty, Po= blackout) = 38.29%
P(T=abnormal | Mux=faulty, Po=blackout) = 38.29%

Figure 4.3: An example of a Bayesian network

4.2.4.3 Local Conditional Probabilities Distribution

An edge departing from node *A* to node *B* in a Bayesian network forms a parent-child relationship. Node *A* is a direct predecessor (or 'parent') of node *B*, which is a direct descendant (or 'child') of node *A*. Local prior probabilities are ones associated to a node that does not have parents. These probabilities are not driven from values specified by any other network variable. Local a posteriori probabilities are ones associated to the other nodes, which are related to the occurrence of a certain pattern of events of the direct predecessors of the node.

For example, Figure 4.2, the *prior* probabilities to be specified are: $p(M1 = m11), p(M1 = m12), p(Po1 = Po11), p(Po1 = Po12), p(Po1 = Po13)$. All the rest are a *posteriori* probabilities, i.e., $p(T1 = t11 \mid M1 = m11; Po1 = Po11)$ which is the conditional probability of *T1* assuming the value *t11*, given that *M1* is worth *m11* and *Po1* is worth *Po11*.

A set of conditional probabilities constitute local distribution of a posteriori probabilities. For instance, the distribution of probabilities associated to the node *T* of Figure 4.3 is constituted by the set of 18 conditional probabilities shown inside the associated rectangle. A set of probabilities that constitute a local probabilities distribution is also called link matrix [130].

4.2.4.4 Alternatives for the Distribution of Local Probabilities

A fragment of a hypothetical Bayesian network is presented as an example in Figure 4.4 below. If in each discrete variable, that is, *Po*, *C*, *Mux*, *T*, there are two or three possible values, the specification of the local probabilities distribution for the *T* node will contain $3^3 + 3^2 = 36$ probabilities.

Hence, the size of the link matrix grows exponentially with the number of parent variables. The size of the link matrix is generally given by the number of local conditional probabilities that should be specified. It is further difficult, even for an expert, to evaluate each one of these probabilities in situations where parent variables have little else in common apart from sharing a child node.

The appearance of alternative proposals for modeling of multi-causal interactions in situations where causal independence among conditioning factors may be assumed are motivated by these and other reasons.

A disjunctive interaction or a “Noisy-OR gate” is one of these models, which occurs when a certain event may be caused by any one among a set of conditions. The probability of that event occurring does not decrease even when several of the possible conditions occur simultaneously.

A combination of node and several “noise” nodes (can be parent or “cause” nodes) constitute causal independence node. In the Noisy-OR model, the combination function corresponds to the logical OR that has been applied on the individual outcome of each cause and on the effect of a “background” node. In the absence of any of the causes, the node is related to the probabilities distribution of the combination node.

Many numbers of states may constitute the parent nodes of a Noisy-OR node, but one of them must correspond to the absence of the cause associated to the node. The distribution of probabilities in a Noisy-OR node should be binary in nature.

Noisy-MAX is a model which is similar in every aspect to the Noisy-OR model, apart from the fact that the probabilities distributions are not necessarily binary.

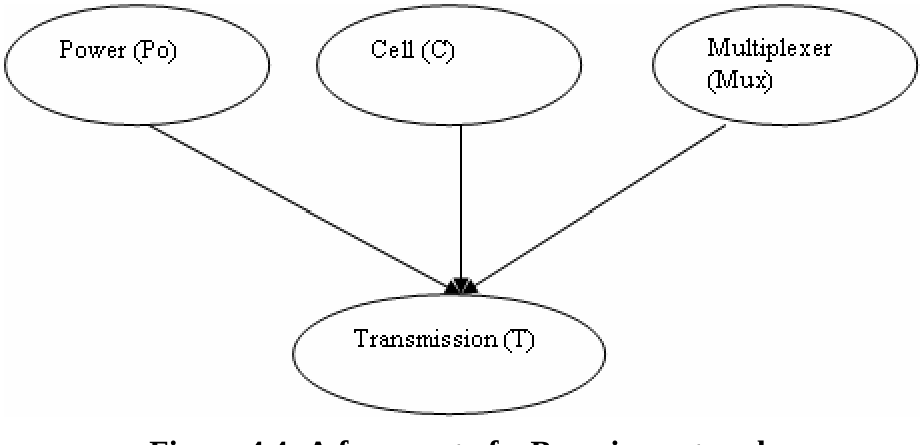


Figure 4.4: A fragment of a Bayesian network

To demonstrate recently introduced concepts, *T* node is represented as shown in Figure 4.4 and the possible states for each one of the variables involved presented in Table 4.1.

Table 4.1: Definition of the possible states of variables Po, C, Mux and T

<table>
<tr><th colspan="4">Variables</th></tr>
<tr><th>Po</th><th>C</th><th>Mux</th><th>T</th></tr>
<tr><td>Good (G)</td><td>Normal (N)</td><td>Ok</td><td>Normal (N)</td></tr>
<tr><td>Weak (W)</td><td>Uncertain (U)</td><td>Faulty (F)</td><td>Uncertain (U)</td></tr>
<tr><td>Blackout (B)</td><td>Abnormal (A)</td><td></td><td>Abnormal (A)</td></tr>
</table>

Using the Noisy-MAX distribution for *T* node, only 27 probabilities should be specified, using the probabilities that are given in Table 4.2. This represents a substantial reduction when compared to the 36 probabilities that characterize a conditional distribution.

Table 4.2: Probabilities distribution for node T.

<table>
<tr><th>Parent nodes</th><th colspan="3">Probabilities</th></tr>
<tr><td>Power (Po)</td><td>P(T=n | Po=g)
P(T=u | Po=g)
P(T=a | Po=g)</td><td>P(T=n | Po=w)
P(T=u | Po=w)
P(T=a | Po=w)</td><td>P(T=n | Po=b)
P(T=u | Po=b)
P(T=a | Po=b)</td></tr>
<tr><td>Cell (C)</td><td>P(T=n | C=n)
P(T=u | C=n)
P(T=a | C=n)</td><td>P(T=n | C=u)
P(T=u | C=u)
P(T=a | C=u)</td><td>P(T=n | C=a)
P(T=u | C=a)
P(T=a | C=a)</td></tr>
<tr><td>Multiplexer (Mux)</td><td>P(T=n | Mux=ok)
P(T=u | Mux=ok)
P(T=a | Mux=ok)</td><td>P(T=n | Mux=f)
P(T=u | Mux=f)
P(T=a | Mux=f)</td><td></td></tr>
<tr><td>Transmission (T)</td><td>P(T=n)
P(T=u)
P(T=a)</td><td></td><td></td></tr>
</table>

4.3 Assumptions

For the models to perform some of the functions there were assumptions made as a condition for the model application in this work. These include:

- It was assumed that it is possible to define a discrete random variable values representative of the state for each network element corresponding to a node in the graph of the cellular network model.
- There is an integral network management system that collects information, from which values are attributed, in real time, to the variables mentioned at element level.
- The managed cellular network service provider is modeled in conformity with the ITU-T rules.
- The prior probabilities related to the variables of each root node, and the local probabilities related to the variables of the other nodes, may as well be alternatively attributed to:
    - through the *relative frequencies* of the corresponding events which are calculated from the data collected by the network management system
    - by using *relative likeliness*. This consists of estimating the probabilities from the subjective judgment of an expert. This method will be useful whenever there is not enough data to permit the estimation of the relative frequencies, which may occur due to the low frequency of the phenomena observed, or even due to the nonexistence of sufficient network management resources.

4.4 Cellular Network Faults Modeling Process

The cellular network fault modeling process consists of five independent processes. These processes are indefinitely repeated and take into account the existence of the assumptions outlined in Section 4.3. These include:

- Network fault alarms acquisition by the network management system
- Classification of the network fault alarms received, according to time windows and originating network element.

- Correlation at network element level, from network fault alarms originally generated by the network elements or obtained through other correlation processes, depending on the correlation topology adopted. The existence of this process is not mandatory at a first moment, as it can be gradually implemented in each network element, according to necessity and taking into consideration peculiarities of each one of them.
- Random variables corresponding to each network element are updated, according to the state of these network elements, which is given by the network fault alarms received and by the result of the correlations carried out on these fault alarms.
- Network fault alarm correlation at the cellular network level, through the evaluation of the new probabilities associated to the fault states defined for each network element, in view of the evidences available at each moment.

4.5 Cellular Network Fault Prediction Using Bayesian Networks

In a Bayesian network, each network fault variable corresponds to one of the predefined states and the probability of each event occurring is evaluated during a diagnosis session (c.f. Section 4.2). Any variable of a Bayesian network may also be defined as an observation node only if its state is observable during a diagnosis session. It should be noted that the state of a node may be an observable but at the same time its associated variable may contain fault states.

4.5.1 Basics of Cellular Network Faults Prediction

The rigorous process of determining what will happen under specific conditions can be referred to as prediction. A cellular network fault is an abnormal operation that significantly degrades performance of an active entity in the network or disrupts communication. All errors are not faults since protocols can handle some of the errors. Generally faults may be indicated by an abnormally high error rate. Therefore cellular network fault prediction is the process of determining which cellular network fault will occur under specific conditions. For example, given that a multiplexer is in good working condition, the only reason it may fail to execute its functions is when there is power blackout. Therefore a conditional probability for this occurrence is used to predict the possibility of a multiplexer failure whenever there was a power failure. This estimation process gradually generates a belief that the multiplexer may fail to execute its functions as a result of power failure.

The purpose of cellular network faults prediction is to enable timely and successful high-level service failure prediction or proactive failure correction, and thereby increasing the chances for proactive error correction before failures set in. This leads to preventative maintenance, which consists of deciding whether or not to maintain a system according to its states. Such measures may help to reduce the cost of maintenance since overstocking of spare parts and over repairing would be avoided. Figure 4.5 shows the construction of a Bayesian network that predicts network faults.

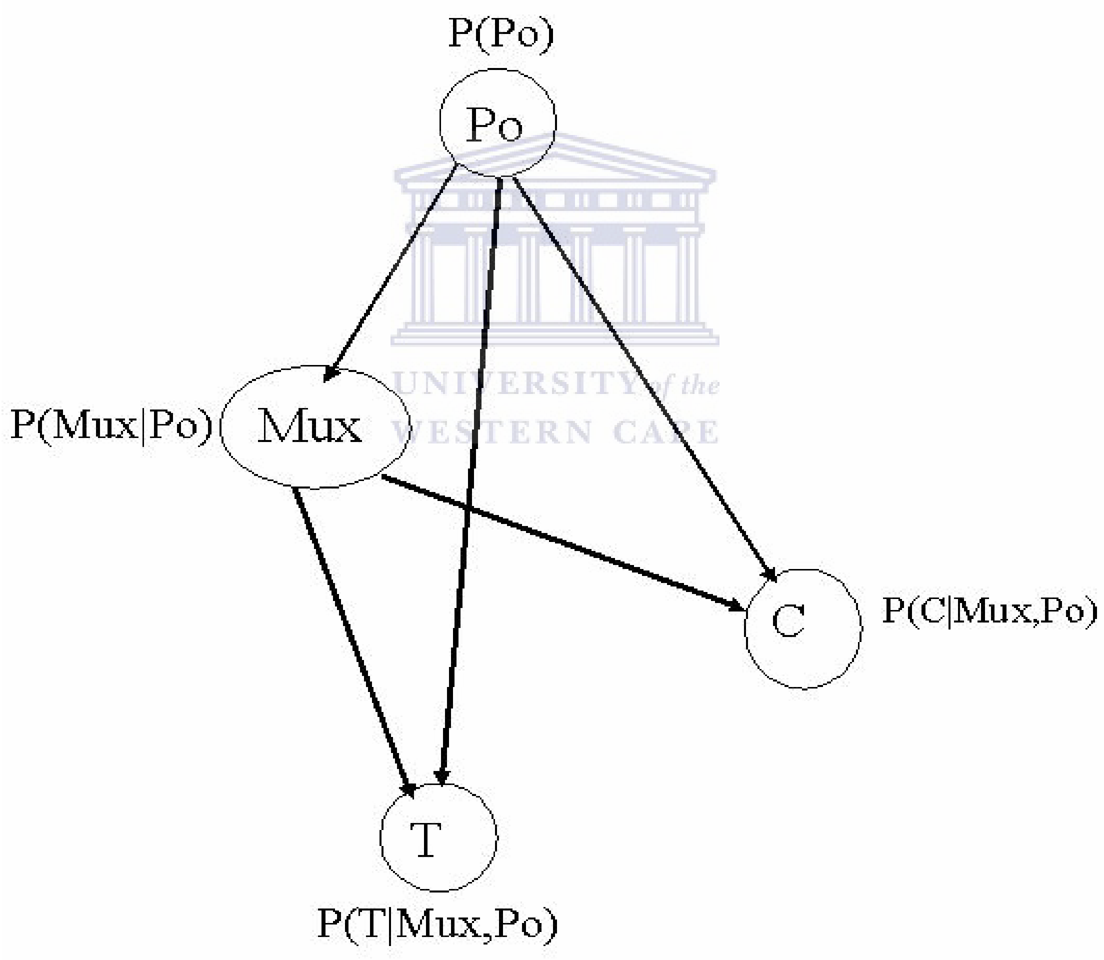


Figure 4.5: A Bayesian network for network fault prediction

4.5.2 Why Cellular Network Faults Prediction?

Fault prediction brings a number of benefits to the cellular network service providers. Some of these include [111][126][126][127][128]:

(1) Accurate fault prediction models support project planning and steering.

(2) Faults prediction helps network managers in re-routing of network traffic. Network managers can take corrective action before the faults occur, thereby ensuring services reliability and availability over the network.

(3) Decision making, i.e., whether to buy from a particular vendor or not, whether to buy a particular hardware or not, etc. It may also be decided that for elements with a high-predicted fault-proneness, say, above 25%, the element design shall undergo quality assurance (QA) activities such as inspections, extensive unit testing, etc. QA tests help to improve the quality of system with each fault that is discovered and corrected.

(4) Operations cost will be minimized if network faults are found as soon as they occur. Faults which are discovered early are cheaper to repair and hence such a scenario leads to offering of cheap and reliable services.

(5) The fault prediction models provide a mapping from a hard to interpret design measurement data to easily interpreted external quality data.

(6) Fault prediction models provide a sound method to combine multiple factors into one cohesive model i.e., take into account the various factors that make certain network elements fault-prone.

(7) Highly accurate fault prediction models can be beneficial when highlighting trouble areas in cellular network system.

4.5.3 Uncertainty Causing Factors

Cellular networks being dynamic in nature and as has been demonstrated in this thesis, uncertainty is inherently associated with the cellular network faults prediction process. This uncertainty is largely driven by the possibility of including erroneous data. There are basically four main sources of data errors. These are:

- The influence of factors not captured by the managed system model

- The imprecision in the calculation of the probabilities distribution values
- The imprecision in the capture and transference of the alarms.
- Imprecision in the information obtained from other correlating processes.

4.5.4 Cellular Network Faults Prediction Models

The joint probabilities distribution $p(x_1, x_2, ...x_n)$ for a Bayesian network may be obtained as a product of the local probabilities distributions for each random variable. For instance, the Bayesian network of Figure 4.4 in which joint distribution $p(Po, Mux, C, T)$ may be calculated as:

$$p(Po, Mux, C, T) = p(Po) \times p(Mux) \times p(C \mid Po, Mux) \times p(T \mid Mux) \quad \text{(Equation 4.5)}$$

The probability that a set of variables $Y \subset X$ constituted by the variables $X_m, ..., X_p \in \{X_1, X_2, ..., X_n\}$, assumes the configuration $y = \{X_m = x_m, ..., X_p = x_p\}$ is given by the sum of all the probabilities of the joint distribution of X where $X_m = x_m, ..., X_p = x_p$.

Using a Bayesian network of Figure 4.4 as an example, the probability that, in the modeled system, the power is good $(Po = good)$ and the transmission is abnormal $(T = abnormal)$ is calculated to be 38.15%. The probability that transmission is abnormal (*T*=*abnormal*) in the same network is 38.28% (c.f. Appendix A).

A conditional probability may be calculated by using the formula,

$$p(Mux \mid Po) = \frac{p(Mux) \times p(Po \mid Mux)}{p(Po)} \quad \text{(Equation 4.6)}$$

according to which the probability for the occurrence of Mux, given that Po occurred, is given by the quotient between the probability of simultaneous occurrence of Mux and Po and the probability of occurrence of Po.

Hence if one knows a set of evidences $e = \{X_m = x_m, ..., X_p = x_p\}$, constituted by all the known values of the random variables of a Bayesian network, where $\{X_m,...,X_p\} \subset X = \{X_1, X_2,...,X_n\}$, the calculation of the probability (or ‘belief’) that a variable $X_k \notin \{X_m,...,X_p\}$ assumes the value x_k is given by:

$$p(X_k = x_k \mid e) = \frac{p(X_k = x_k) \times p(e \mid X_k = x_k)}{p(e)} \qquad \text{(Equation 4.7)}$$

The above derivations may be illustrated using Bayesian network of Figure 4.3. The belief that power is good given that $e = \{T = abnormal\}$, which is the set of all the known evidences, is given by:

$$p(Po = Good \mid T = abnormal) = \frac{p(Po = good) \times p(Po = good \mid T = abnormal)}{p(T = abnormal)} \text{(Equation 4.8)}$$

By using the previously calculated probability values:

$$p(Po = Good \mid T = abnormal) = \frac{0.3815}{0.3828} = 0.9966 \approx 99.66\%$$

Supposing now that new evidence is known, multiplexer is faulty and totally knocked out, new belief that power is good is calculated using equation below:

$$p(Po = Good \mid T = abnormal, Mux = \ faulty) = \frac{p(Po = Good, T = abnormal, Mux = \ faulty)}{p(T = abnormal, Mux = \ faulty)}$$

The capacity for non-monotonic reasoning of the Bayesian networks is demonstrated by the above presented examples. While the only known evidence was that the Multiplexer was faulty, the belief that power was good was of 99.66% as it is known that the transmission was abnormal. With new evidence and information being discovered, the belief could be recalculated. This belief would grow more with the discovery of more new information about other network fault variables becoming available. For example, if the Cell is found to be in normal state is made available. The belief could be recalculated using equation below:

$$p(Po = good \mid T = abnormal, Mux = \ faulty, C = normal) =$$

$$\frac{p(Po = good, T = abnormal, Mux = \ faulty, C = normal)}{p(T = abnormal, Mux = \ faulty, C = normal)}$$

$$p(Po = good \mid T = abnormal, Mux = \ faulty, C = normal)$$

For more probability calculations refer Appendix A.

4.6 Characteristics of a good Model

The following are the main characteristics of a good model:

(1) The irrelevant aspects of the system must hidden by a good model for a given objective to be achieved.

(2) State of a network element should be modeled by a random variable whose value is attributed based on information supplied by a network management system. This may show the causal propagation of faults among the network elements that compose system.

(3) The network elements should send the faults alarms directly to the MIB after the construction of the Bayesian network that models the system is done.

(4) The model should use diagnostic inference, predictive inference and inter-causal inference. The above characteristics are met by the models derived in this work.

4.7 Construction of the Structure of the Bayesian Network

In this Section a practical aspects of creating a Bayesian network model for a cellular network fault diagnosis purposes is discussed. The information needed for the model construction comes from empirical data, which consist of network faults that were logged into the database. The source provides a simplified view of reality, which must be further simplified by the model. The key problem is to construct the model so that all the important aspects of system reality from the point of view of the diagnostic process are captured.

In balancing the cost of the model development with model fidelity, the following three steps are followed in model construction [131]:

- Choosing the variables and the states of each one of them,
- Constructing the structure of the Bayesian network, that is the directed a cyclical graph containing the information on the independence among variables,
- Assigning probability values (specifying local probabilities distribution) for each variable.

4.7.1 Problem Diagnosis

The evaluation of the diagnostic problem at hand is the first step towards the construction of the structure of a Bayesian network. A complex cellular network is sub-divided into sub-networks as the first step. These sub-networks are sub-divided further into multi-network layers and refined up to the point at which each sub-network instance corresponds to an individual element in the network.

An existence of a directed edge (N_i, N_j) departing from node N_i and reaching node N_j indicates that N_i depends on N_j in a network. This means that a fault in N_j may, potentially, have its effects directly propagated up to N_i. The possibility of direct propagation of the effects of a fault, from a node N_j to a node N_i is modeled by a directed edge (N_i, N_j) in a Bayesian network as shown in Figure 4.6 below.

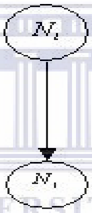


Figure 4.6: Two functionally dependent networks

The structure of a Bayesian network is reached by inverting the directions of the arrows in the directed edges of the cellular network model that is, replacing each edge (N_i, N_j) by an edge (N_j, N_i).

The network will be represented with nodes corresponding to the physical means occupying the upper part of the graph, because by convention, edges of a Bayesian network will always point from top down as shown in Figure 4.6.

4.7.2 Definitions of the Variables and their States

Each sub-network must be represented by a single discrete variable with arbitrary, though finite, number of possible values, allowing the states of the sub-network modeled by these variables to be represented at the required detailed level.

It may be convenient sometimes to associate an arbitrary number of discrete variables to a single sub-network. For example, suppose that a sub-network SR is associated with variables $X_1, X_2,..., X_n$ to with $Sx_1, Sx_2,..., Sx_n$ possible states respectively. For this to happen it is possible to define a variable X_{SR} with $Sx_1 \times Sx_2 \times .... \times Sx_n$ possible states; to each one of these states it must be assigned one of the possible combinations of the states of the original variables. For example, Table 4.1 presents the possible states of the four variables that were used in this study.

The random variables have values defined which must be collectively exhaustive and mutually exclusive. For instance, if $\{a_1, a_2,...., a_n\}$ is a set of states defined for a variable, and $p(a_1), p(a_2),....p(a_n)$ are the probabilities associated to each one of these values, then $p(a_1) + p(a_2)+,...., p(a_n) = 1$. Table 4.2 shows possible probabilities for node T.

Modeling challenge is frequently encountered at this stage of the construction of a Bayesian network. As an example, *normal*, *uncertain* and *abnormal* states are defined for transmission (T) variable of this network [Table 4.1].

4.7.3 Specification of the Local Probabilities Distributions

Suppose N_i network utilizes the services of the N_j network [Figure 4.6], then a fault in the N_j network may produce consequences on the N_i network, but this rarely happens. Some faults in the network N_j may occur in parts of the network that do not contribute to the service offered to the network N_i. The fault may occur in a part of the network, sometimes, which presents active redundancy, in such a way that the fault becomes transparent to the network N_i. A fault may also be perceived by network N_i through a reduction in the performance of the service offered by network N_j where a fault occurred.

It is generally difficult to analytically determine the degree of dependence of the network N_i in relation to the network N_j without a deep knowledge of the architecture and of the relationship among them and of the functionality of the two networks. Conditional probability can be used suitably to model the occurrence of an exception situation in the N_i network, as a consequence of an event (i.e., a fault) in the N_j network. This may be calculated from experimental data collected by the network management system or estimated by experts. Therefore, local probabilities distribution function for each variable of the Bayesian network will have to be specified.

4.8 Factors Affecting the Performance of Bayesian Networks

There are three important factors that affect the performance of a probabilistic inference algorithm in Bayesian networks. These include:

(1) The total number of edges. In the worst case this comes up to $O(n^2)$ where n is the number of nodes of the network as shown in Figure 4.7.

(2) The number of parent nodes, per node. The size of the link matrix associated to a node grows exponentially with the number of parent variables (cf. Section 4.2.4).

(3) The number of states per random variable. The total number of possible configurations in a Bayesian network with n nodes (or variables) and k states per variable are k^n.

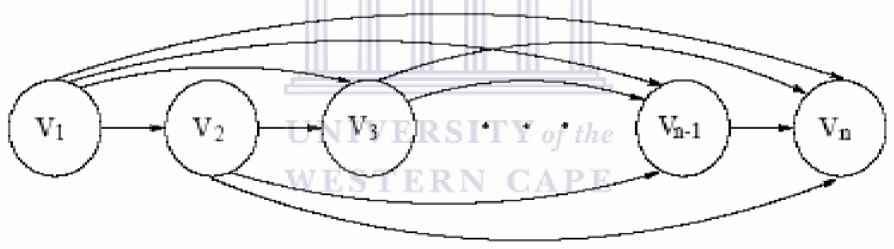


Figure 4.7: A Bayesian network with $n(n-1)/2$ Edges [32]

4.9 Summary

Cellular network service providers lose a lot of revenue due to cellular network faults. The revenues are lost as a result of customer churn, lost call times and short messages not delivered. In this Chapter, the classification of cellular network faults was presented. Basic concepts of Bayesian networks are presented with reasons for its use, concerns and application areas being explored. The models, assumptions, process and reasons for cellular network faults prediction using Bayesian networks were presented. Characteristics of a good model with the uncertainty-causing factors in the process were presented. Lastly, the construction of the structure of the Bayesian networks and factors affecting the performance of Bayesian networks were discussed.

Chapter 5

CELLULAR NETWORK FAULTS PREDICTION USING MOBILE INTELLIGENT AGENTS

In this Chapter, basic concepts of biologically inspired modeling are presented in Section 5.1. Mobile intelligent agent modeling is presented in Section 5.2. In Section 5.3, particle swarm optimization algorithm is presented. Components of the system architecture are presented in Section 5.4. The PSO solutions for cellular network environment are presented in Section 5.5, while some concluding remarks are highlighted in Section 5.6.

5.1 Basic Concepts of Biologically Inspired Modeling

Nature is full of knowledge and ways of doing things. In the animal kingdom, animals have different ways of achieving different tasks. Some animals collectively do their tasks in a group called swarm. The birds flock, fish school, slime molds aggregate, as some of the examples. Table 5.1 provides a summary of examples of swarming animals and insects. The benefits of swarming for each individual and/or group of insects or animals are explained.

Table 5.1: Examples of swarm intelligence that can be found throughout nature

Swarm	**Behaviour**	**Individual benefit**	**Collective Benefit**
Birds	Flocking	Reduces drag	Maximize flock travel
Fish	Schooling	Enhances foraging	Defence from predators
Slime Mold	Aggregating	Survive famines	Find food quicker
Ants	Foraging	Enhances foraging	Finding food faster
Termites	Nest building	Shelter	Climate-controlled incubator
Bees	Swarming	Enhances foraging	Find nectar quicker, task specialization, cooperation, etc

The benefits derived from swarming have been the driving force behind the use of honeybee behaviour as a modeling taxonomy in this work.

5.1.1 Honeybees

The two well-known classes of honeybees are European bees (*Apis mellifera*) and Africanized bees (*Apis mellifera scutellata*). Physiologically, every bee is able to carry out every task coming up in a hive but there seem to be specialization among the individual bees. As shown in Figure 5.1, bees go through all behavioural roles from egg to forager bee in their life (along thick arrows). The time they spend in a certain role differs. Bees follow the thin arrows when selecting a new activity and when they finish a task they generally return to the 'patrolling' activity before choosing a new task.

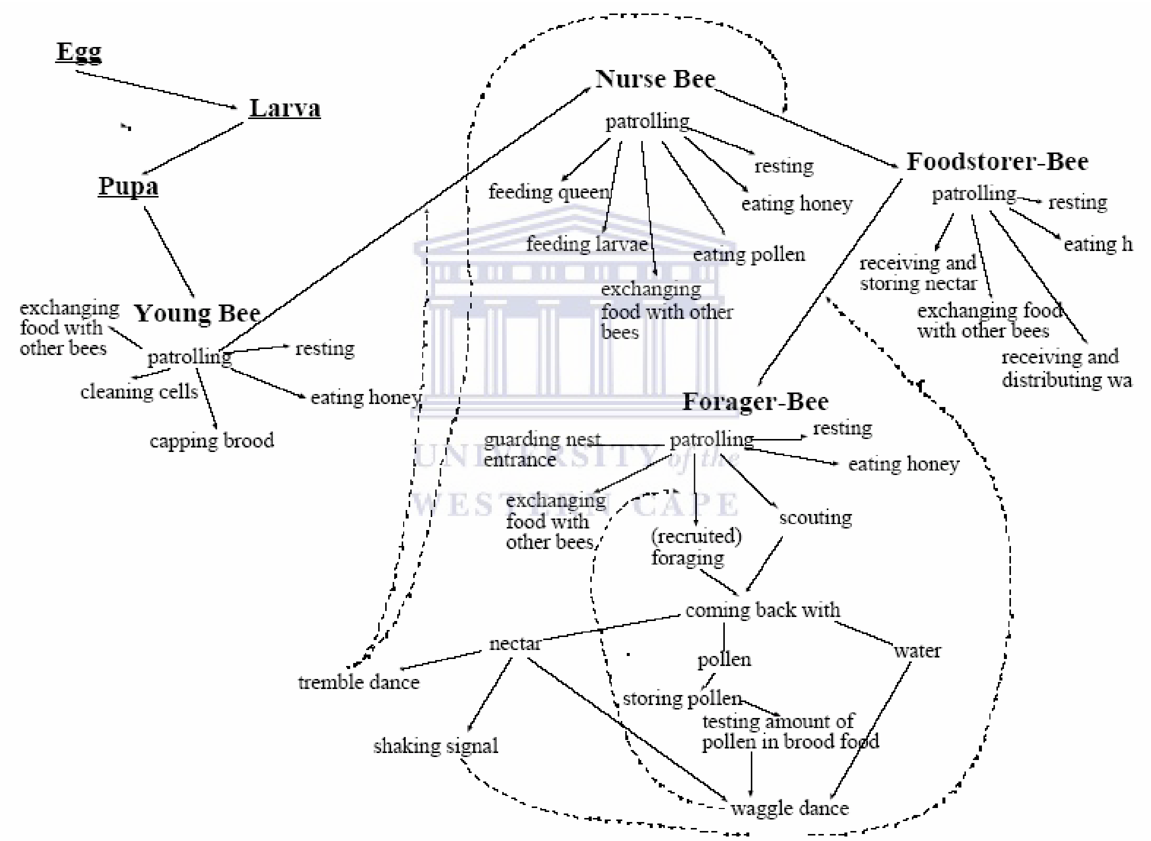


Figure 5.1: Schematic diagram of bees' behavioural roles and activities [133].

However, task selection by worker bees change as they age. Drones usually live five to ten weeks. Workers usually live about fifty days and all the workers are females. Queens live an average of about three years and there is only one surviving queen bee in each colony, which mates with many drones (male bees), and may lay 1500 eggs per day. The pheromone released by the queen identifies her as the queen. A new beehive is started when the beehive is overpopulated. Swarming is caused by too much warm or cold weather. Only one-queen bee rules the colony so when the two queens reach the adult stage, they battle to the death for control of the hive.

5.1.2 Why Honeybees?

A honeybee colony has many features that are desirable in cellular networks [133][154]:

- Efficient allocation of foraging force to multiple food sources;
- Different types of foragers for each commodity;
- Foragers evaluate the quality of food sources visited and then recruit optimum number of foragers for their food source by dancing on a dance floor inside the hive;
- There is no central control;
- Foragers try to optimize the energetic efficiency of nectar collection and foragers take decisions without any global knowledge of the environment.
- Bees also tend to perform tasks, which are nearer them. The probability with which a bee engages in a certain task is proportional to the distance of the stimulus. This feature can be used to monitor troublesome nodes within the cellular network. The MIA can be kept near or at the troublesome node and this will ensure that the task will be performed.

5.1.3 Concerns about Honeybees

The main difficulty in modeling cellular network systems using the bee-like behaviour is lack of research tools [133]. It is often very difficult to design an individual agent behaving like a honeybee as well as the effects it will have on the other agents. It is costly, time consuming and tedious to experiment with such agents. The researchers have not yet fully understood many mechanisms, from biological point of view, that influence honeybees.

5.1.4 A General model of Honeybee Behaviour

The general bee behaviour within the hive and outside the hive is used in this work to model the cellular network system. An adult bee performs tasks such as brood tending, storing, retrieving and distributing honey and pollen as well as communication and foraging. The behavioural role of each bee dictates the task it performs, the cues it perceives in its environment, and its response thresholds to those cues. Thus every bee reacts only on local cues and its internal state.

Behavioural roles of bees can change during the bee's life as a result of age, signals received (such as shaking signal or tremble dance performed by forager bees), internal predisposition and amount of time they need to find task, which is a measure of how needed the completion of this particular task is at the moment. The bees have an internal parameter, which can be seen as analogous to the juvenile hormone titer, which determines which tasks they are able to perform. This parameter increases with the bees' age, but the increase depends also on signals and cues received.

Bees in any 'role' can engage in a number of activities. The selection of the next activity depends on external stimuli and a 'threshold' for this special activity. Generally the probability with which a bee engages in a certain task is proportional to the distance of the stimulus, e.g. the larva that has to be fed, and to the reciprocal value of its threshold for that activity. These thresholds can thus be used to model differences in genetic predisposition of individual bees to do certain tasks.

Figure 5.1 above shows the schematic representation of the tasks and roles in the model. The activities in the different behavioral roles can be seen, as well as the feedback loops which influence the time at which workers switch roles. The time at which workers switch roles implicitly regulate the amount of workers allocated to each role (dotted arrows). The sequence of roles is as follows: A bee starts its life when an egg is laid by the queen. If it is fed enough, it then develops into a larva, pupates, and finally hatches as an adult 'young bee'. Young bees and 'nurse bees' (who care for brood and feed the queen) mainly engage in tasks in the brood area, while other adult bees (food-storers and foragers) often stay near the nest exit.

All activities taking place inside the hive are modeled explicitly, i.e. the bees move around in the hive while engaged in their tasks. The foraging of bees however, taking place outside the hive, is only modeled insofar as bees leave the hive and return with a load of honey, pollen or water. The time this takes and the amount brought back depend on whether the bee was scouting (then both values are chosen randomly) or was a recruited forager, in which case values are determined by the information given to the bee during the recruitment (waggle dance).

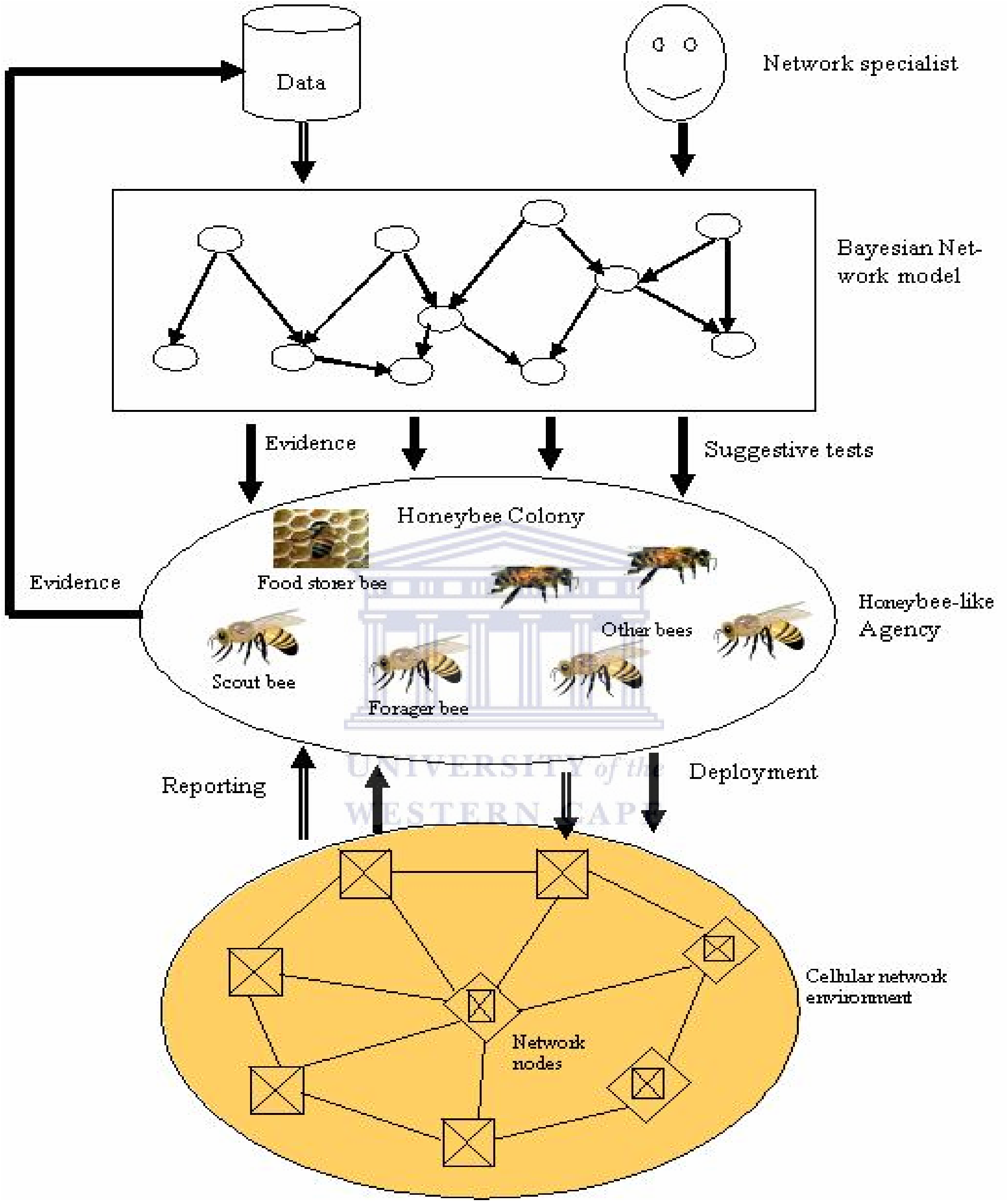


Figure 5.2: A proposed MORSBOSS system Architecture

In case of this thesis, the behaviour of the bees outside the hive and how it identifies the warm nectar is more important for cellular network fault prediction. The model is discussed in the next Section.

5.2 Mobile Intelligent Agent Modeling

5.2.1 Mobile Intelligent Agent System Architecture

Cellular network faults can be detected by bee-like mobile intelligent agents, which could move from one node to another in a cellular network. The bee-like mobile intelligent agent would visit various network nodes (flowers) in search of faulty node (nectar) in the network environment. Nectar in flowers has got color and degree of temperature as attributes. These two attributes are dependent on each other. The warmer flowers are indicated by darker colors and provide highest metabolism reward to the bees [134].

The bees use the color of the flowers to predict the warmer flowers with the highest metabolism reward to them [134]. The honeybees use this knowledge (prediction mechanism) to move from one flower to another in search of nectar with the highest metabolism reward. In this work, the Mobile Intelligent Agent (MIA) resembles the honeybees and would use the Bayesian network prediction models (cf. Section 4.3) to predict a node in the cellular network that is most likely to be faulty.

The MIA moves from node to node-gathering information. The MIA would predict and move to a node with highest fault occurrence. The network node with the highest fault occurrence would remain under MIA ‘microscopic eyes’ for it can always be faulty in most cases. The high fault occurrence is computed and determined using Bayesian network model (prior probability) from the fault logs stored in the database. These are known faults that can be predicted using the information stored about them in the database. However, a wireless local area network was constructed bringing in additional network faults for this study (c.f. Section 6.5.3).

Agents in our system can be described by the tuple, ***MIA = (C, V, W, MDF, M)***. They have uniform architecture consisting of five components as discussed below:

5.2.1.1 Color

The honeybees use the color of the flowers to predict the flower with the highest metabolism reward (rich in nectar and warmth). There are a range of flower colors used in this work from white to black indicating the color of each network node under observation. The color of a network node corresponds to the probability value of that particular network node. The color of the network node (flower) can be used to select profitable node (c.f. Section 5.5.5) and for Migration Decision Function (MDF). The function is based on the probability of each network element under scrutiny. The probability values and color values are stored in the database and would always be updated automatically after a specified time window (cf. Section 6.6).

Table 5.2: The Color coding for different Probability ranges

Probability range	Verb	Color
0.0	Impossible	White
0.0 – 0.1	Very unlikely	Yellow
0.1 – 0.25	Unlikely	Orange
0.25 – 0.4	Fairly unlikely	Peach
0.4 – 0.5	Less likely than not	Pink
0.5	As likely as not	Red
0.5 – 0.6	More likely than not	Brown
0.6 – 0.75	Fairly likely	Green
0.75 – 0.9	Likely	Blue
0.9 – 1.0	Very likely	Purple
1.0	Certainty	Black

In this work, the colors were chosen depending upon the probability values range. The brightest color (white) indicates that honeybees may not visit such a flower (node). This symbolizes that it is impossible for such a node to experience network fault of any nature. While the darkest color (black) indicates that honeybees would visit such a node. It symbolizes certainty in network fault occurrence to that node. In case of two of more network nodes bearing the same colors, the MIA would use the intrinsic probability values to move to the most profitable node in the network. Table 5.2 above shows a summary of flower colors mapped according to probability values of fault occurrence.

5.2.1.2 Vision

Like human beings, honeybees have eyes, which they use to see where they are going. The eyesight of honeybees is so accurate and it holds the key to a breakthrough in micro-cameras for use in surgery, medical diagnostics and even mobile phones [135]. Honeybees can see flowers very clearly just like human beings, identifying flowers using flower colour. The honeybees then predict the flower that is more profitable (rich in nectar and warmth) based on the flower colour. The darker flowers are more profitable to the honeybees as compared to brighter ones [134].

5.2.1.3 Warmth

The higher the frequency of network fault occurrence at a particular network node, the warmer the node and the more it will be preferred by MIAs. Probability values are computed using Bayesian network models (cf. Section 4.3). Honeybees not only search for warmer flowers for direct metabolic reward but also nectar. This decision can be made based on frequency of occurrence of faults which is represented by the colour of the flower. At a particular point in time it can be relative to the expected warmth. In case of a situation where the colour of two or more flowers happens to be the same, then the honeybee would move to the closest flower. In the same manner MIA would move to the node with higher frequency of fault occurrence and in case of two or more nodes having the same frequency of occurrence, MIA would move to the closest node. Just like honeybees, MIA would perform the task at hand quicker, ensuring the networks availability.

5.2.1.4 Migration Decision Function (MDF)

This is the functional value the MIA uses to decide on whether to move to another network node or to stay at the very node where it resides. The MIA can also move from one node to another to gather information based on the history of that particular network node. Whenever MIA detects a fault at a particular network node, the MIA sends a message to other MIAs who then move to work in-groups (cooperate) to accomplish the task. The task here can be to repair, to inform the network engineer or to redirect all the tasks at this particular node to other nodes.

The posterior probability (c.f. Section 4.3) calculated using the Bayesian network model is here used to determine the MDF. A difference of 0.1 can make an agent move from one network node to another, for example, if *Mux =0.34, C=0.21*, *Po=0.16, Tx=0.19* at a particular time *T,* and the highest recorded is of *Mux* node at that moment, then most MIAs would move there for it is more profitable. After a time window elapses then an update occurs which may change the values of the nodes as *Mux=0.25*, *C=0.12*, *Po=0.58*, *Tx=0.21*. This would make the MIAs move from *Mux* node to *Po* node. They can also pre-empt the fault by learning and keeping this history of movement in their memory (as discussed in the next sub-section).

The MDF is a function or rule set that is used to determine where an agent should visit next. The MDF typically uses the warmth, which determines the colour of a flower and link cost information in order to determine the next hop in its journey through the network or may simply follow a hard coded route through the network. This latter migration strategy is often referred to as an itinerary in the mobile agent literature. Alternatively, when migrating, the agent may use the default migration node available to it. But we want to ensure that an agent moves as fast as possible to a node which experiences danger.

5.2.1.5 Memory

Just like human beings, honeybees have a working memory [136]. The working memory of the honeybees makes them aware of the environment, in which they operate, the events that took place and what to do when such an event occurs again. By keeping all these information in the memory, the honeybees can predict the flowers with warmer nectar using flower colour and previous experience. The MIA stores the colour and the degree of warmth. It also stores the MDF which it uses to determine when to move from one node to the other. Though some of this information is stored in the database, symbolic information can also be stored in the memory of the MIA, which can only be used by the MIA. The MIAs that store information in their memory cannot communicate such information to the environment. The memory in each MIA enables MIA to remember the best position of the search space visited in the past [133]. Thus its movement is an aggregated acceleration towards its best previously visited position and towards the best individual of a topological neighbourhood.

5.2.1.6 Operation

In an insect world scenario, honeybees will move to the flowers in search of nectar. They just not only want nectar for food but for warmth also. This kind of flower, which can offer warmth and food is said to be of more reward to the honeybees than any other flower. This kind of flower is also known as profitable flower (node).

The MIA will reside in the server. The MIA has in its memory the experience and probability values, which would be indicated by colour. Depending on the colour indicator, the agents will move to a particular node with darker colour indicated by more warmth and nectar. In the similar way, MIA will move from the server to network elements (node) where the frequency of fault occurrence is higher. In this manner MIAs get to monitor the node to report any anomaly whenever it occurs; solve anomalies that it can within the shortest time possible; inform the engineer of the anomaly; reroute all the network traffic to avoid loss of data packets passing through the faulty node and lastly predict network fault before it occurs. Just like honeybees, the MIAs work in groups (cooperate) to complete the task at hand.

It is worth stating that in case of two or more anomalies occurring in the network at the same time, the MIA will replicate (clone) itself according to the number of affected elements in order to ensure that the problem at each of the affected nodes is solved. As well in case there are two or more nodes with the same level of fault occurrence whereby exhibiting the same color, MIA will replicate (clone) itself and move with great speed to these affected nodes to perform its functions.

The operation of the MIA becomes complicated and it is said to be NP-hard complete problem. The MIA clones itself into many (swarm), which in a large cellular network may be up to 10,000 MIAs. During the movement of these MIAs collision among them should be avoided. The speed of movement also needs to be optimized. All these can be achieved by using particle swarm optimization algorithm, discussed in the next Section.

5.3 Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique inspired by social behaviour of birds flocking [137] or fish schooling [138] [139]. Developed by Eberhart and Kennedy in 1995 [140][141][142][143][144][145], PSO was originally designed to simulate birds seeking food, which is defined as a ‘cornfield vector’ [140] but was found to be a good optimizer [146].

PSO is initialized with a population of random solutions, called particles [147]. Each particle in PSO moves over the search space at a velocity dynamically adjusted according to the historical behaviours of the particle and its companions. All particles have fitness values and velocities. The particles have the tendency to fly towards the better search area over the course of search process. The popular examples are floys [148] and boids [137][149].

5.3.1 The PSO Algorithm

The common scenario of a group of birds randomly searching for food is used here to explain the algorithm. There is only one piece of food in the area being searched and all birds do not know where the food is. But they know how far the food is in each iteration and follow the bird, which is nearest to the food as the best effective way to find food.

In this scenario, each single solution is a ‘bird’ (called particle in PSO) in the search. All particles have fitness values evaluated by the fitness function to be optimized and have velocities directing the flying of the particles. The particles fly through the problem space by following the current optimum particles.

The PSO is initialized with a group of random particles (called solutions) and then searches for optima by updating generations. In each iteration, each particle is updated by following two ‘best’ values. The first one is the best solution (fitness) achieved so far by each particle (fitness value is also stored). This value is also referred to as personal best or simply *pbest*. Another is the best value, obtained so far by any particle in the population. This best value is a global best and called *gbest*. When a particle takes part of the population as its topological neighbours, the best value is a local best and is called *lbest*. After finding the two best values, the particle updates its velocity and positions according to the equation 5.1 and 5.2.

Basic algorithm as proposed by Kennedy and Eberhart in 1995[140]:

x_k^i - Particle position

v_k^i - Particle velocity

P_k^i - Best “remembered” individual particle position

P_k^g - Best “remembered” swarm position

c_1, c_2 - Cognitive and social parameters

r_1, r_2 - Random numbers between 0 and 1

The velocity is calculated as follows:

$$v_{k+1}^i = v_k^i + c_1 r_1 (P_k^i - x_k^i) + c_2 r_2 (P_k^g - x_k^i). \qquad \text{(Equation 5.1)}$$

Position of individual particle is updated as follows:

$$x_{k+1}^i = x_k^i + v_{k+1}^i \qquad \text{(Equation 5.2)}$$

Algorithm 1: Particle Swarm Optimization Algorithm

```
1. Initialize
        (a) Set constants $k_{\max}, c_1, c_2$.
        (b) Randomly initialize particle positions $x_0^i \in D$ in IR$^n$ for i=1,…,p.
        (c) Randomly initialize particle velocities $0 \le v_0^i \le v_0^{\max}$ for i=1,…,p.
        (d) Set k = 1
2. Optimize
        (a) Evaluate function value $f_k^i$ using design space coordinates $x_k^i$
        (b) If $f_k^i \le f_{best}^i$ then $f_{best}^i = f_k^i$, $p_k^i = x_k^i$
        (c) If $f_k^i \le f_{best}^g$ then $f_{best}^g = f_k^i$, $p_k^g = x_k^i$
        (d) If stopping condition is satisfied then goto 3.
        (e) Update all particle velocities $v_k^i$ for i=1,…,p
        (f) Update all particle positions $x_k^i$ for i=1,…,p
        (g) Increment k.
        (h) Goto 2(a).
3. Report results
4. Terminate.
```

5.3.2 Why PSO Technique?

There are a number of reasons why PSO is chosen ahead of other search strategies such as simulated annealing [150][151], genetic algorithm [152], and evolutionary algorithm [153] for this work. These include [142][154][155][156]:

- Insensitive to scaling of design variables
- Simple implementation
- Easy parallelized for concurrent processing
- Derivative free
- Very few algorithm parameters
- Very efficient global search algorithm
- Has been successfully applied in many areas
- Computational feasibility
- Effectiveness
- Consistency in performance
- Systems developed using PSO are very robust, being relatively insensitive to noisy and/or missing data.
- PSO has implicit ability to track changing optima (suitable for fault prediction in cellular network environment).

However, PSO’s slow convergence in refined search stage (weak local search ability) and PSO as one of the swarm algorithms, it uses multiple searching points and do not utilize design sensitivity information. It inherently depends upon extremely high computational cost since it caters for a very large number of particle evaluation processes.

5.3.3 Application areas

PSO technique has been applied in many areas including [137][140][145]:

- Training of neural networks, which include, identification of Parkinson’s disease, extraction of rules from fuzzy networks and image recognition, etc.

- Optimization of electric power distribution networks
- Structural optimization, which include, optimal shape and sizing design, topology optimization, etc.
- Process biochemistry
- System identification in biomechanics

5.4 Components of the System Architecture

The main components of proposed system architecture include database, Bayesian networks, Honeybee-like agency, cellular network environment and network specialist. These components are discussed below:

5.4.1 Database

Stores the fault logs (alarms), the probability values associated to each network node, and acts as the memory of the MIAs where MIAs refer to for certain decision making, i.e., MDF and alarm triggering decisions (for more on database cf. Section 6.6.2).

5.4.2 Bayesian networks

Bayesian network is used to extract information from the raw faults data (which is an input for the system). The probabilities (prior, conditional, posterior, etc) are calculated based on historical faults data. The calculation is done using the models generated in Chapter 4. From these calculations the network node with higher frequency of fault occurrence would be noticed at this stage. This forms what is called evidence. This evidence would also indicate which network node would be affected as a result of causal relations. This evidence would also indicate the cellular network services, which would be affected by the network fault. Each network fault (evidence) has a suggestive measure on how to solve it. Evidence and suggestive tests are vital information, which would be passed on as input to the honeybee colony.

5.4.3 Honeybee-like Agency

This consists of MIAs specializing in different tasks just like honeybees. Just like the food stocker, the bee would store food into the sectors of the comb so the MIA would store information into the database. This information would be in the form of evidence stating where a fault occurred, the starting time and ending time. The forager bee in the meantime would be roaming visiting the network nodes (cf. Section 5.2) gathering information. The current state of the network would always be reported to the other honeybees after a time window. This would enable the other bees to perform their tasks. For example, after a time window new evidences may emerge that needs special attention. It may be a new different kind of fault or it may be an evidence indicating a fatal error is about to occur. In the case where a fault is located by the scouting bee, it recruits the foragers (MIAs) to help in the task. This is referred to as *deployment* in this architecture.

5.4.4 Cellular network environment

Cellular network environment consist of the hardware and software. The hardware may include but not limited to network nodes, the switches, the multiplexers, the transmission cables/lines, routers, and gateways, among other hardware components. The environment is constantly monitored by the MIAs just like honeybees forage monitors their food. They move from one node to another depending on the history and colour of the node. In this manner the process would continue thereby ending up with a 'healthier' cellular network system.

5.4.5 The Network specialist

Network specialist is an engineer responsible for the network operations support system (OSS). With a user interface that displays the state of each and every network node, the network specialist is able to manage the nodes effectively. The network specialist would in certain cases sanction the repair of the network depending on the 'health' or the deemed catastrophe. The process flow is shown using flow diagram in Figure 5.3.

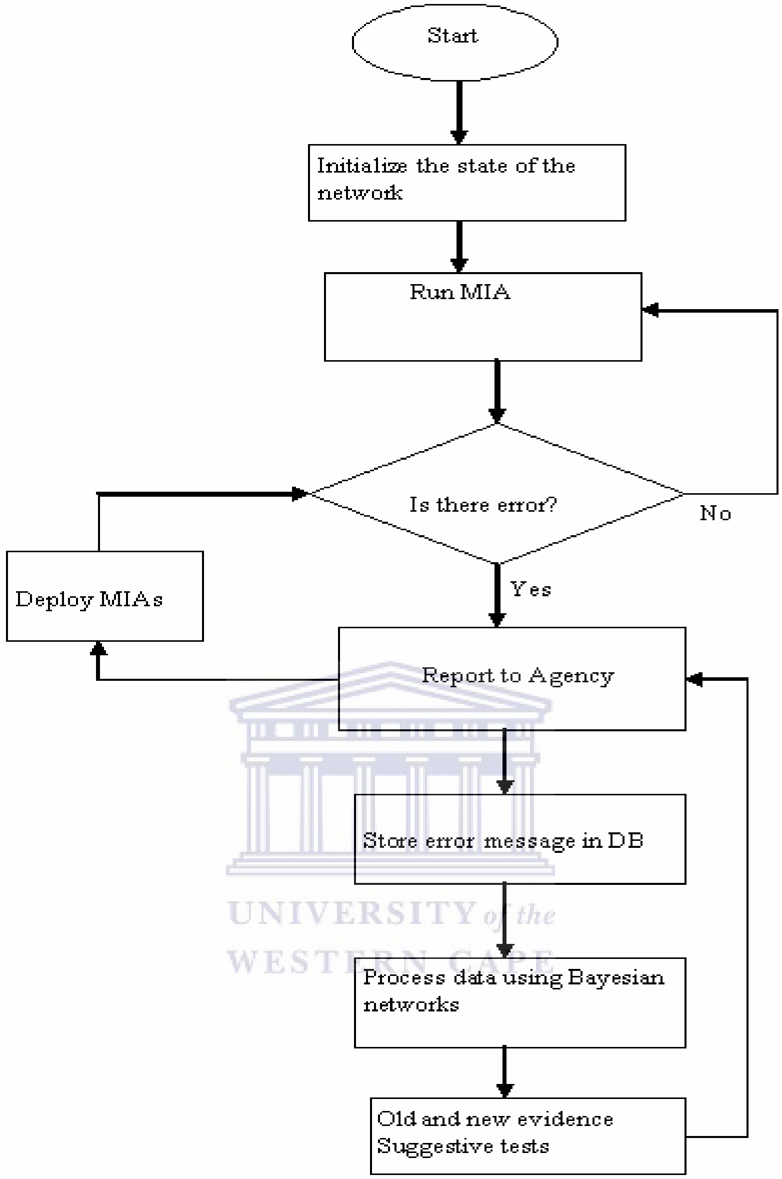

Start
Initialize the state of the network
Run MIA
Is there error?
No
Yes
Deploy MIAs
Report to Agency
Store error message in DB
Process data using Bayesian networks
Old and new evidence Suggestive tests

Figure 5.3: Flow Diagram of the system Architecture

5.5 PSO Solutions for Cellular Network Environment

In applying the PSO algorithm to cellular network fault prediction, detection of changes in the environment and how to react to these changes are considered in this Section.

5.5.1 Cellular Network Environment Change Detection

The cellular network environment is dynamic in nature and detection of change in the environment needs to be done almost immediately when it occurs. The real-time change detection therefore requires that the optimization algorithm (PSO) respond appropriately to adapt to these changes. An automated approach is applied to detect changes based on information received from the cellular network environment.

The use of sentry particle(s) is proposed by Carlisle and Dozier [157][58][159], which is used to detect change in its local environment. The sentry particle stores a copy of its most recent fitness value. The fitness of the sentry is re-evaluated and compared with its previous stored value at the start of the next iteration. A change occurs if there is a difference in the fitness values. However the use of sentry particle increases the computational complexity of the algorithm. Monitoring the fitness of global best position is proposed by Hu and Eberhart [156]. In this case change detection is based on globally provided information. Monitoring the global second-best position [160] increases the accuracy of change detection and limits the occurrence of false alarms.

In this work, environment change detection is done periodically after a time window and the fitness of the globally best particle. The fitness value will be calculated after a time window and compared to the previous fitness value of the particle. If there is a difference then change has been detected. At this moment the global best position would also be detected signifying a change in the environment. This method is not computationally demanding and it is very efficient.

5.5.2 Response to Cellular Network Environment Changes

There can be two kinds of changes that can be detected in the cellular network environment. The positive change occurs when the level of severity of the fault occurrence registers a decrease and the negative change when an increase is registered. In the event that a negative change is detected in the cellular network environment, the system responds by raising alarm if the threshold is reached. The response would be inertia weight update, which would be adjusted according to equation 5.3 [161]:

$$w_i(t+1) = w(0) + (w(n_t) - w(0))\frac{e^{m_i(t)} - 1}{e^{m_i(t)} + 1} \qquad \text{(Equation 5.3)}$$

Where the relative improvement, m_i is estimated as:

$$m_i(t) = \frac{f(y_i(t)) - f(x_i(t))}{f(y_i(t)) + f(x_i(t))} \qquad \text{(Equation 5.4)}$$

with $w(n_t) \approx 0.5$ and $w(0) < 1$

The swarm is reinitialized by setting all particle positions to new random positions after which the personal and neighbourhood best positions are recalculated. After the change in environment, a local search can be done around the previous optimum. It is always believed that an offending node in the network can always offend.

5.5.3 Performance Measurement in Cellular Network Environment

The performance measurement of the dynamic environment such as cellular network system can be estimated using either the reporting on performance of an algorithm after a fixed number of iterations. This, however, will not be an accurate performance measure under particular environmental conditions. Another measure is to produce a performance profile over all iterations.

A performance measurement for dynamic environment is provided by Morrison [162] with respect to evolutionary algorithms, which describes the performance of the algorithm over a representative sample of possible environment changes. The performance measure is valid under two conditions as:

(1) the algorithm has a reasonable recovering time for all types of environment changes,
(2) the fitness of the optimum can be assumed to be restricted to a relatively small range of values.

The performance measure is given by [162]:

$$PF_C = \frac{\sum_{m=1}^{n_{PF}} \left(\frac{\sum_{t=1}^{n_t} PF(t)}{n_t} \right)}{n_{PF}} \qquad \text{(Equation 5.5)}$$

Where PF_C is collective fitness, $PF(t)$ is the best solution for time step t, n_t is the total number of time steps (iterations), and n_{PF} is the total number of simulations on the same dynamic problem. For a specific simulation, m, the value of PF_{C_m} is the average performance over exposure to a representative sample of environment changes.

The above performance measure is applicable to PSO, where

$$F(t) = \hat{y}(t) \qquad \text{(Equation 5.6)}$$

In a cellular network, faults occur randomly and some of these faults appear in the proximity of the other faults. It is always believed to be a propagation of one or more faults to other network components. This can be approximated by *association* measure, which is simply shown by the size of the numbers. For example, in a case where a target fault f_x is often associated with fault f_y, then a big number of fault f_x's would be found in fault f_y's.

However, Euclidean distance, in which if one point A in an n-dimensional space is defined by coordinates $(a_1, a_2, ...a_n)$, and another, B, has coordinates $(b_1, b_2, ...b_n)$, then the distance between the two points is:

$$D_{AB} = \sqrt{\sum (a_i - b_i)^2} \qquad \text{(Equation 5.7)}$$

This distance can help in estimation of the time to travel to the faulty node. The MIA will use this distance information to move to the nearest node in case of it being faulty. This in turn will improve the performance of the system.

5.5.4 Parameter Encoding

MIAs performing different tasks help to maintain a "healthier" network. MIAs operate in the same environment (colony) provided by the JADE-LEAP software. For example, the network monitoring agents, the alarm raising agents and the recruiting agents all work for the purpose of network's 'health'. Every MIA in the colony is viewed as a particle. Let the number of particles in a colony C, be n, the dimension of particle position be d, $d = 2$. k_i^j denotes the particle i in colony C with the jth dimension. This colony can be described by matrix $n \times d$ as follows:

$$C(n, d) = \begin{vmatrix} k_1^1 & k_1^d \\ k_2^1 & k_2^d \\ ... & ... \\ k_n^1 & k_n^d \end{vmatrix} \qquad \text{(Equation 5.8)}$$

The superiority of individual particle or solution is evaluated by the speed and position that characterize the particle. The MIAs need not only close up to the faulty node target, but also avoid obstacles and colliding into each other as it moves to the target point. This process forces the MIA to adapt new values (position and speed). Then the target function is obtained by:

$$Fn = \int_0^0 (w_1 \, | \, er(l) \, | + w_2 \frac{V_{p1} \times V_{p2} \times ... \times V_{pi}}{l^i}) dl \tag{Equation 5.9}$$

Where $er(l)$ is the system error, V_{pi} is expected value of searching points during the process of moving to the target, l is the distance to the target, w_1, w_2 are the weights.

Algorithm Flow

After parameter encoding and determination of target function, the PSO algorithm in MORSBOSS reads as follows:

Algorithm 2: Adapted PSO for MORSBOSS
1. Initialize the position and speed of every agent in the swarm
2. Calculate the adapted values of every MIA using equation 5.9
3. Compare the adapted values of every MIA with the self best position experienced so far. This may be regarded as the best position currently
4. Compare the adapted values of every MIA with the best position of the whole swarm. This may be regarded as the best position currently.
5. Update the speed and position of every particle using equation 5.1 and 5.2
6. If the ending condition is not satisfied (via the maximum iteration times in advance), then go to 2.
7. End

5.5.5 Profitable node

The profitable node is indeed the faultiest node under the network environment in surveillance. Depending on the time and situation of the cellular network environment, profitability is both relative and dynamic. Therefore in making the decision to move to the most profitable node, a MIA considers the relative profitability ($\Pr$) for the i^{th} node of state s. This can be calculated using:

$$\Pr \ = \ P_i \ + \sum_{j \in x} \tilde{P}_{ij} \qquad \text{(Equation 5.10)}$$

where $\tilde{P}_{ij} = P_{ij}$ for $i < j,$ and $\tilde{P}_{ij} = P_{ji}$ for $i > j.$ P_i and P_{ij} are profit coefficients.

5.5.6 Noisy and Cellular Network Environment

A cellular network environment sometimes experiences noise. During noisy period it is almost impossible or time consuming to obtain precise values. This is especially true if the function and gradient values depend on the results of numerical simulations. The function values obtained are frequently corrupted by noise. In the presence of additive Gaussian noise, for almost all the methods considered by Arnold [163] there exists a noise strength beyond which convergence becomes unreliable. This effect leads to stagnation of the search process. Information about $f(x)$ is obtained in the form of $f^{\eta}(x)$, where $f^{\eta}(x)$ is an approximation to the true function value $f(x)$, contaminated by small amount of noise η. The function values are obtained, for the additive noise case, as [164]:

$$f^{\eta}(x) = f(x) + \eta \qquad \text{(Equation 5.11)}$$

The multiplicative noise is obtained as shown in equation 5.12:

$$f^{\eta}(x) = f(x)(1+\eta), \quad 1+\eta > 0, \qquad \text{(Equation 5.12)}$$

where η is a Gaussian noise term with zero mean and standard deviation σ

$$\eta \sim N(0, \sigma^2) \qquad \text{(Equation 5.13)}$$

Equation 5.13 is used to approximate the relative stochastic errors that characterize the test problems. Assuming a normal noise distribution is used as an approximation of reality based on the maximum entropy principle [165], the probability density function of the noise reads

$$p(\eta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2}\left(\frac{\eta}{\sigma}\right)^2 \right] \qquad \text{(Equation 5.14)}$$

To obtain the η, we apply the method of Box and Muller [166], using various noise strengths (standard deviation) σ. The assumption of the normal distribution is not mandatory in this work. However, the measurement errors in nature and technology are very often modeled using this distribution.

The global minimum of the objective function potentially moves within the search space. The movement can be simulated by applying a rotation of the whole space and /or adding an offset to the global minimum. The presence of noise seems to contribute to the avoidance of local minima and the detection of the global minimum of the objective function by the PSO.

5.6 Summary

In this Chapter biologically inspired modeling was presented. Taxonomy of honeybees operations were used to model the network using mobile intelligent agents. Mobile intelligent agent system architecture is developed based on various functions as colour, vision, warmth, MDF and memory. The components of the system architecture were incorporated into the PSO algorithm. In the next chapter, experimental setup, process and results are presented.

Chapter 6

IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this Chapter, the software and technologies used in the implementation of the MORSBOSS system are presented in Section 6.1. In Section 6.2, similarities between intelligent agent architectures are presented. The reasons for the choice of JADE are discussed in Section 6.3 while the challenges of embedding Mobile Agents into Cellular Network Devices are presented in Section 6.4. In Section 6.5, scenarios of experiments are presented. Experimental setup is presented in Section 6.6. In Section 6.7, the experiment process is discussed. Results and discussion is presented in Section 6.8 with summary of the chapter in the subsequent section.

6.1 Implementation Technologies

Cellular network faults prediction prototype called MORSBOSS was implemented using the Java Agent DEvelopment framework (JADE 3.5), MYSQL server 5.0 (database), Java Development Kit (JDK 1.6.0), Java Runtime Environment (JRE 1.6.0), Apache ant 1.7.0, JavaCC 4.0, Sun Java Wireless Toolkit 2.5.0, LEAP 3.5 as the technologies. The backend was developed using MYSQL server 5.0 with the remaining technologies being used in rules and user interface implementation.

6.2 Why JADE Architecture?

A large number of mobile architectures are available today. Most of these architectures offer suitable runtime environment for mobile agents as well as high-level programming Applications Programming Interface (API) supporting mobile agent capabilities. The work presented in this thesis was done using JADE-LEAP (Java Agent DEvelopment framework and Lightweight Extensible Agent Platform) [167][168] (mobile agent platform) owned and distributed by Telecom Italia [167][169]. JADE was chosen for a combination of benefits that it offers. These include [170]: (1) Simplicity in usage and agent programming, (2) Good online community support and documentation, (3) Support for the FIPA [171] standards, (4) Efficient and tolerant of faulty programming and (5) It is an open source (free), etc.

However, it is worth noting that JADE does not provide support for migration between different execution environments as a drawback.

JADE has already been integrated into different major architectures, such as J2EE and .NET allowing JADE to execute multi-platform proactive applications. JADE can take J2EE [172] for servers, J2SE [172] for PCs and desktop computers, Personal Java (Pjava) [173] for wireless mobile devices supporting Pjava (i.e., PDA) and J2ME with CLDC and MIDP for mobile devices supporting MIDP [174] (i.e., cell phones) as shown in Figure 6.1.

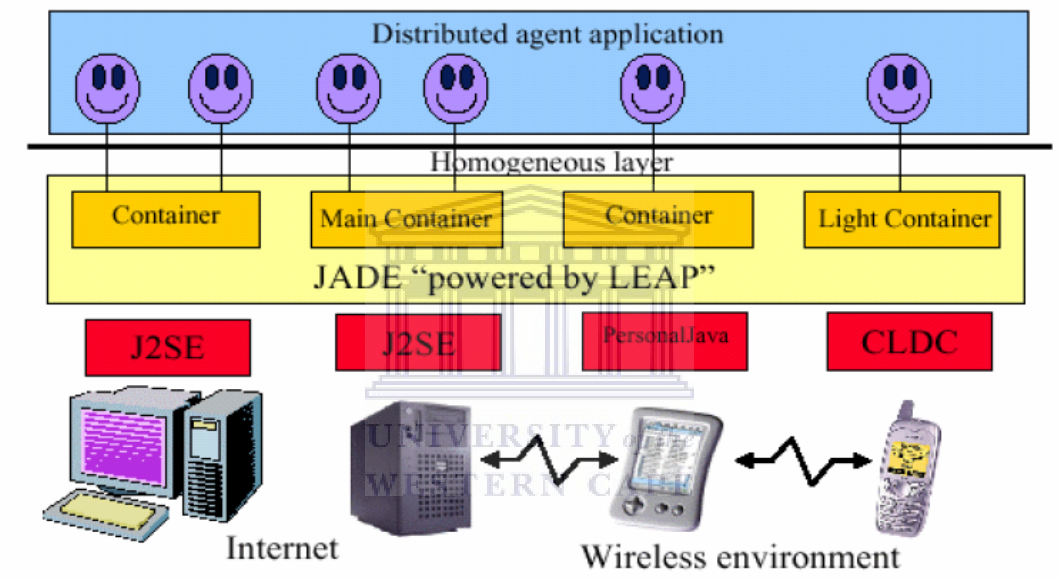


Figure 6.1: JADE-LEAP agents' architecture, from JADE-LEAP user's guide

A MIA is a JADE-LEAP agent (c.f. Figure 6.1), which is mobile with autonomy and high degree of collaboration. The system automatically creates a new Client Agent (CA) to act on behalf of the user once a new user logs onto the system. The agent name is unique, and based on the user ID. The CA is able to obtain and maintain the user's personal information and preferences, and to react to various incoming and outgoing messages and requests intended for the user. The CA is automatically removed from the system once the user log outs.

6.3 Challenges in embedding Mobile Agents into Cellular Network Devices

Embedding software agents in a variety of mobile devices can be a challenging mainly because of the diversity of existing hardware and software settings. However, the deployment of JADE and in particular of the LEAP libraries [168] allows the creation and deployment of light-weighted agents in a variety of mobile devices in a pretty straightforward way. The more challenging, and still open issue concerns the integration of agents in the providers' network infrastructure. Besides the diversity of network equipment and technologies, the complexity arises from the need of possibly adapting current operators' policies and procedures in terms of service and service level agreement management, adopted security mechanisms and deployed billing/accounting frameworks [175].

6.4 Scenarios of Experiments

The commonly offered services by the cellular network service providers include voice, short message service (SMS), video, MMS, among others. In most cases customers who possess cellular phones are expected to consume the services on offer. In case of a network failure it's the customers who would first experience the repercussions.

In the experimental scenario, the set-up network is able to conduct communication between the nodes. The network can also facilitate a number of services, i.e., surfing the internet, downloading of files, communication (sending and receiving of messages), etc. The solution designed in this Chapter can also suit a similar situation in cellular network environment. The operation performance of the network system is expected to be normal, just like in any other network.

The evaluation of cellular networks can be done by simulation methods using software like MATLAB [176], NS-2 [177], and OPNET [178]. Simulations allow repetition of experiments and altering of parameter values in such a way that each set of parameter values represent the different conditions under which the network system may be deployed. Simulations are an easy way of measuring the performance of the network dynamics because it can be done with little logistics, few equipment and limited human manpower. However, simulations may not give accurate representation of the network in a real life environment. Simulations do not capture certain real-world operations of the cellular network systems.

A real network environment can be set-up to evaluate the performance of the network. This method is bound to give you the real performance measurement of the network. However, it is very expensive and takes a lot of time to implement the software, install and/or run the experiments. The size of the network to be built is also another challenge because of inadequate space, financial constraints and limited expertise of people who are required to help in the handling of devices.

The setup was done indoors involving eight devices. The access point was setup covering a radius of 10 meters. All the operations could be carried out within this radius without the risk of encountering the problem of fading signals. However, if a node moves beyond the 10 meters radius, range related faults that are related to fading signals at a node would be detected.

6.5 Experimental Setup

There are five different equipments used in carrying out the experiment. These equipments were sourced from local vendors. A detailed description of the equipments is given in Table 6.1.

6.5.1 The Hardware and Software Components

A total of eight devices were used in the experiment forming eight nodes. The first PC was used as server where all the operations of the execution are taking place. A wireless LAN card was installed for it to communicate with other devices. The second PC, which was also equipped with a wireless LAN card, was used as one of the nodes. The two handheld iPAQs and two Laptops were also used as nodes within the network. These nodes are regarded as managed nodes for experimentation purposes. Affordability and availability of the devices influenced their choice for use in this experiment. The devices are rich in documentation which can be used to customize their operations easily to accommodate the system implemented and perform the experiment at hand.

Table 6.1: Specification of devices used in Experiment

No. of items	Device model	Description
2	iPAQ h5450	Intel PXA250, 400MHz, 64MB RAM, 48MB ROM, 64K color transflective TFT screen & SDIO card slot.
1	Wireless Router WL-552	3Com OfficeConnect ADSL Wireless 54Mbps 11g Firewall Router. Made to comply with IEEE 802.3 ISO/DIS 8802/3
1	Acer Travel mate 290 series Laptop	Intel (R) Pentium (R) M CPU,1.50 GHz, 512MB RAM
1	PC	Intel (R) Pentium Dual-core CPU, 1.80GHz, 0.98 GB RAM.
1	CISCO AIRONET 1200 SERIES WIRELESS ACCESS POINT	Simultaneous support for 2.4GHz and 5 GHz radios supports single- & dual-bad configuration.
1	PC	Intel (R) Pentium (R) 4 CPU, 3.00GHz, 512MB RAM.
1	HP Compaq nx5000 Laptop	Intel (R) Pentium (R) M CPU,1500MHz, 512MB RAM

The iPAQs used have integrated wireless features, including Bluetooth, 802.11 Wi-Fi, and consumer infrared functionality. The storage memory of the iPAQs can be increased with the built-in SDIO card. The iPAQs have WindowsCE 3.0 operating system. The first PC was equipped with SMC2802w 2.4 GHz 54Mbps wireless PCI Adapter. The second PC was equipped with AIR-PCI350 2.4 GHz DS 11Mbps wireless LAN Adapter. Both the PCs run Windows XP Professional with service pack 2. The Acer laptop runs Windows XP Home Edition with service pack 2. It was also fitted with the Intel (R) PRO/Wireless 2200BG Network Connection. The HP Laptop runs Windows XP Professional with service pack 2. It was fitted with Intel (R) PRO/Wireless LAN 2100 3B Mini PCI Adaptor. The configurations of the laptops allow them to communicate wirelessly as nodes within a single network.

The software used includes JADE [169], JADE-LEAP [167][168], j2se [172], Pjava [173], MIDP [174], and jdk1.6 [172] as explained in Section 6.1. Java Agent

DEvelopment (JADE) [169] framework is a FIPA-compliant java classes that allow developers to build multi-agent systems. It comes with graphical tools for administration purposes that make complex tasks simpler. LEAP [168] is an add-on which replaces some parts of JADE kernel thereby creating a modified environment called JADE-LEAP [167][168]. JADE-LEAP allows agent implementation in mobile devices with limited resources, i.e., iPAQs. Microsoft Visual Basic 6.0 was used to program a random power turn off of PCs used for the experiment. After the hardware and software assembling, it is very important to ensure the wireless connection is secured.

6.5.2 Making and Establishing Wireless Connectivity

Point to point protocol (PPP) and Bluetooth are used for the making and establishing the connection in [167]. The experiment [167] only considered the communication between PC and PDAs. However, this work employs some of the techniques with more devices connected together using wireless access point forming a wireless local area network. The experimental set-up divides the devices into service-provider and service-consumer relationship. Taking PC-1 as the service provider (server), the other devices were made to consume the services offered by this PC. In this case the PC communicates wirelessly with the rest of the other devices in the constructed network via the access point once it establishes connection. Working at a higher communication level, JADE-LEAP establishes TCP/IP connections between containers. However, the physical means of providing connection was provided by [167]. In this work, the devices integrated with Bluetooth were connected to wireless LAN using Wi-Fi. vxUtil [179] was used in order to verify the connection between Bluetooth and Wi-Fi. Bluetooth and Wi-Fi were chosen because of their availability, affordability and support of the devices used. It is worth to note that it does not matter which type of connection is used. The agents' code could still move and execute within the constructed network without any modification. The connection range was limited to 10 meters of radius. Despite the limitation, it was still suitable for the purpose of the academic exercise.

The establishment of wireless connection between PC and PDAs is a challenging process. However using Microsoft ActiveSync 4.0 [180] application, it was possible to establish wireless connection. This happened after installing Plugfree 2.0 [181] and NetBEUI in the PC. Registry modification of the PDAs was done then set-up the IP address, subnet mask, and ActiveSync in the PC and PDAs. The devices were then connected by running Plugfree on the PDAs and PC. ActiveSync starts automatically on both the PDAs and PC signaling the connection of the devices using PPP via Bluetooth and Wi-Fi. At this point JADE-LEAP can be started on the PC and PDAs execution of the MIA is then initiated.

The aim of this work is to predict cellular network faults before they occur. Prediction is based on detected network faults data, which are stored in the database. Therefore, network fault detection is very crucial to this work. However, network faults need to be injected to the network environment.

6.5.3 Network Faults Injection

After the establishment of wireless connection, various services were tested over the network to check its operability. Text messages, voice communication, video conferencing, etc were transacted over the network. Tests showed that the network was functioning properly.

JADE-LEAP has a built-in mechanism of detecting network faults that knock out node containers in real time. However some of the network faults are intermittent, i.e., delay, logical faults which may crop up as a result of structural or functional errors. It is very difficult to inject some of these faults to test the MIA deployed in the constructed WLAN. However, some of these network faults may occur without being injected into the constructed WLAN.

Network faults injected were representative of the actual network faults that occurred within the system and the additional software required to inject network faults would not affect the functional behaviour of the system in response to the injected faults as some of the assumptions made.

Though it is very difficult to create failure scenarios of large and complex systems (i.e., cellular network system) [181], network fault injection is very vital for identifying the reliability bottlenecks, studying network system behaviour in the presence of network faults, determining the coverage of error detection and recovery mechanisms, evaluating the effectiveness of fault tolerance mechanisms and performance loss, etc.

The network faults injected are grouped into three types as hardware, software and execution faults.

6.5.3.1 Hardware Faults

Hardware faults are ones that affect operations of hardware parts of network components. Injection of these hardware based faults are fast, not intrusive, can access locations that are hard to be accessed by other means, no model development or validation is required, experiments can be done in near real-time, and are better suited for low level network faults. However, hardware fault injections introduce high risk of damage to the devices, low portability, time consuming, requires special purpose hardware, limited observability and controllability, some of the methods for injecting hardware faults are complex and limited set of injectable faults [182].

Some of the hardware faults include hardware breakage, hardware bending, hardware rust, power surge, short circuit and power blackout. The power blackout fault is a total cut of electricity supply to the network device. The fault renders the device unusable since the device depends on power to function. It is important to note that some devices are fitted with internal power supply in the form of internal battery, UPS, among others. In such cases the internal batteries were removed to enforce the fault injection. Power blackout was programmed using Ms Visual Basic 6.0 for applications to turn off the targeted network devices at random automatically. Due to the dangers associated with power surge, hardware breakage, bending, rust and short circuit faults and financial constraints they could not be injected.

6.5.3.2 Software Faults

Software faults on the other hand were flexible and cheaper to inject as compared to hardware faults. The real software running on the target device is modified to inject faults. However, software faults may not be injected to inaccessible locations, and may overload the system thereby decreasing the performance of the whole system. A timer is used to trigger a fault injection at the expiry of a predetermined time. A program that allows targeted injection of fault before a particular instruction is implemented.

6.5.3.3 Execution Faults

Execution faults are the network faults that occur during the execution of the system.

6.5.3.4 Multiplexer

The data from a certain cellular network service provider, with network fault variables show multiplexer type of fault. It is one of the major faults together with Power and Transmission that were experienced during the operation of the very network. Multiplexer is a device that combines multiple analog message signals or digital data streams into one signal. This fault was difficult to inject. However, it was injected by way of cutting off the electricity supply to the device to impair its operation.

6.5.3.5 Transmission

The transmission errors were grouped together. It was one of the major faults in the network under study. Other network faults may cause transmission failures. In most cases it is caused by broken network links, power outage, and device breakage, among others. Therefore the injection of this fault would automatically take place with the injection of the other faults like power, multiplexer, etc. The transmission or rather the completion of any initiated call would not be completed as a symptom. This fault was not injected because the injection of the other faults took care of it. However, transmission error can be injected using VBScript which would stop and start a node automatically thereby causing the messages sent to that particular node to bounce back. It would also cause the messages being sent from that particular node not to reach their destination due to loss of connection.

6.5.3.6 Cell

Cell is one of the minor network faults experienced by the service provider under study. It is a case when a particular cell fails to perform some of its functions. The functions may be call routing, connect a customer to the base station, among others. These functions would be affected by transmission, power and multiplexer failures within the cell.

6.5.3.7 Time-out

A case where an operation takes longer time than required to execute, a time out error would be sent to the database. In this case, simple messages were exchanged between devices in the constructed network. During this operation, a message may take longer to reach the targeted host.

6.5.3.8 Run Time Error

During the execution of the system, breakpoints were inserted into the classes, which could cause errors. This error could impair the execution process of the MIA.

6.5.3.9 Out of Range

Out of range network fault was injected manually. The portable network devices were moved further from the access point causing signal to fade until the device could not receive any signal. It was the safest and easiest fault to inject but very tedious.

The network faults injected are summarized in Table 6.2 below.

Table 6.2: Network fault-injection implementation methods

Hardware	**Software**	**Execution**
Power blackout	Time out	Run time error
Multiplexer	Cell	Out of range
Transmission		

6.5.4 Independent and Dependent Network Faults

The independent network faults are the ones that could not cause another network fault when injected. The dependent network faults are the ones that could cause or be caused to occur by another.

In this experiment, Power fault could cause Multiplexer, transmission, cell and out of range network faults without injecting them. For example, a device can go off due to power failure and may cause transmission fault. This makes power fault the most hazardous and with more impact on the network than others that were injected. Power fault is therefore the one that needs more attention than other network faults. Out of range failure could cause transmission fault. Run time error failure could cause transmission fault. Multiplexer and time out failures could cause transmission fault.

Independent fault in this experiment was power failure. Power failure could cause multiplexer, transmission, cell and out of range network faults without injecting them but none of them, could cause power fault. This makes power fault independent and the main cause of other network faults.

6.5.5 The Network Fault Detection

The detection of network faults in a constructed wireless network was made possible by JADE framework [183] and PSO algorithm (cf. Section 5.5). JADE-LEAP executes in containers [168][169][183]. The containers depend on the Main Container, which coordinates all the nodes and govern the whole platform. In this work, PC-1, which was acting as a host to the MIA would execute implemented system in the Main Container. Then MIA was started in other devices forming ordinary containers that were numbered from 1 to 5 as shown in Figure 6.3. In this experiment, ordinary containers 1 to 5 represent PC-2, laptop-1, laptop-2, iPAQ-1 and iPAQ-2 respectively. The containers arrange themselves in a logical ring so that whenever one of them fails, the others will notice and act accordingly [183]. JADE platform has a star topology but as Main Container replicates itself whenever it’s affected by fault. This in turn changes the topology into a ring of stars [183].

The availability of all the peripheral containers of the main container is constantly monitored by *UDPNodeMonitoringService* based on UDP ping packets that are periodically sent by the peripheral containers. JADE platform allows either permanent TCP or UDP connection to the monitored remote nodes [183]. In this work UDP was preferred because it can accommodate higher number of containers, and support connection interruptions. UDP connection is more scalable and allows containers to be out of site for sometimes [183]. For example, PC-2 with container-1 running in it suddenly shuts down due to power failure, Main Container in the PC-1 would notice within 2 seconds because peripheral containers constantly send UDP ping messages to the Main Container. The Main Container would wait for 5 seconds without receiving ping messages from a peripheral container before it is marked *temporary unreachable*. The peripheral container is automatically removed from the platform if it does not send any ping messages within a minute [183]. At this point a message is sent to the database reporting a detection of a fault. This message consists of the *faultsLogID*, *nodeID*, and *startTime*. The *endTime* is logged when a faulty container is restored. This is when the duration of the fault can be calculated.

JADE framework has a service that supports the detection of container, additions and deletions from the Main Container node and updating of address lists of all peripheral containers involved. It can either be done by activating the 'Address-Notification service' on all Main Container nodes and on the peripheral containers or by passing the address list to a starting peripheral container using `-smaddrs` command line argument [183]. The later approach is used in this thesis because it eases notification traffic directed towards peripheral containers. It assumes a fixed list of Main Container nodes before hand. In this work one Main Container node is assumed with five ordinary containers as shown in Figure 6.2.

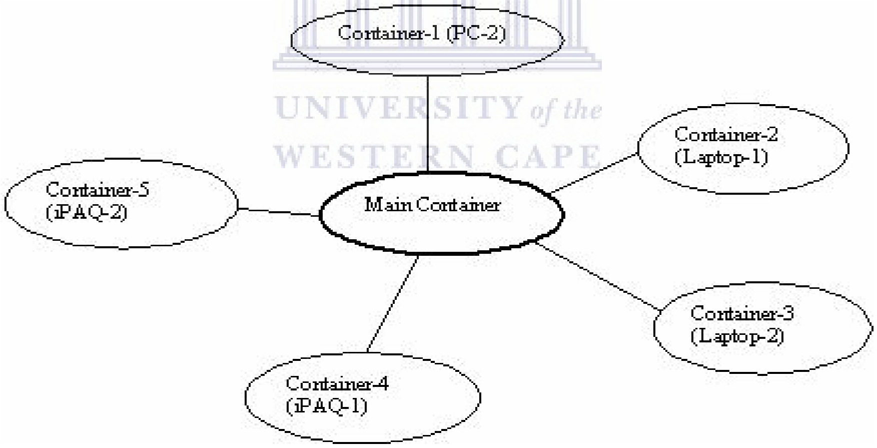


Figure 6.2: The Logical Connection of MIA

6.5.6 Detection of Unknown Network Faults

The system was trained by data (network faults log) from a certain cellular network service provider. The common network faults variables included Multiplexer, Power, Cell, and Transmission. During the execution of the system (iterations), the MIA could detect and report these network faults before they occurred. The detection mechanism is explained in Section 5.5.1.

Together with the four common network faults injected, unknown (new) network faults were injected to test how the system could detect and predict them. MIA could detect 98% of old and new or unknown network faults that were injected for the first time. However, new network faults must be learned by the MIA before it could predict them. The network fault prediction algorithm is based on the detected faults information stored in the database.

6.5.7 The Constructed Wireless Network Environment

The physical connection of the constructed wireless local area network (WLAN) involves the host PC, which is connected to the wireless router using Unshielded Twisted Pair (UTP) cable. The remaining devices connect to the network wirelessly via the wireless router as shown in Figure 6.3.

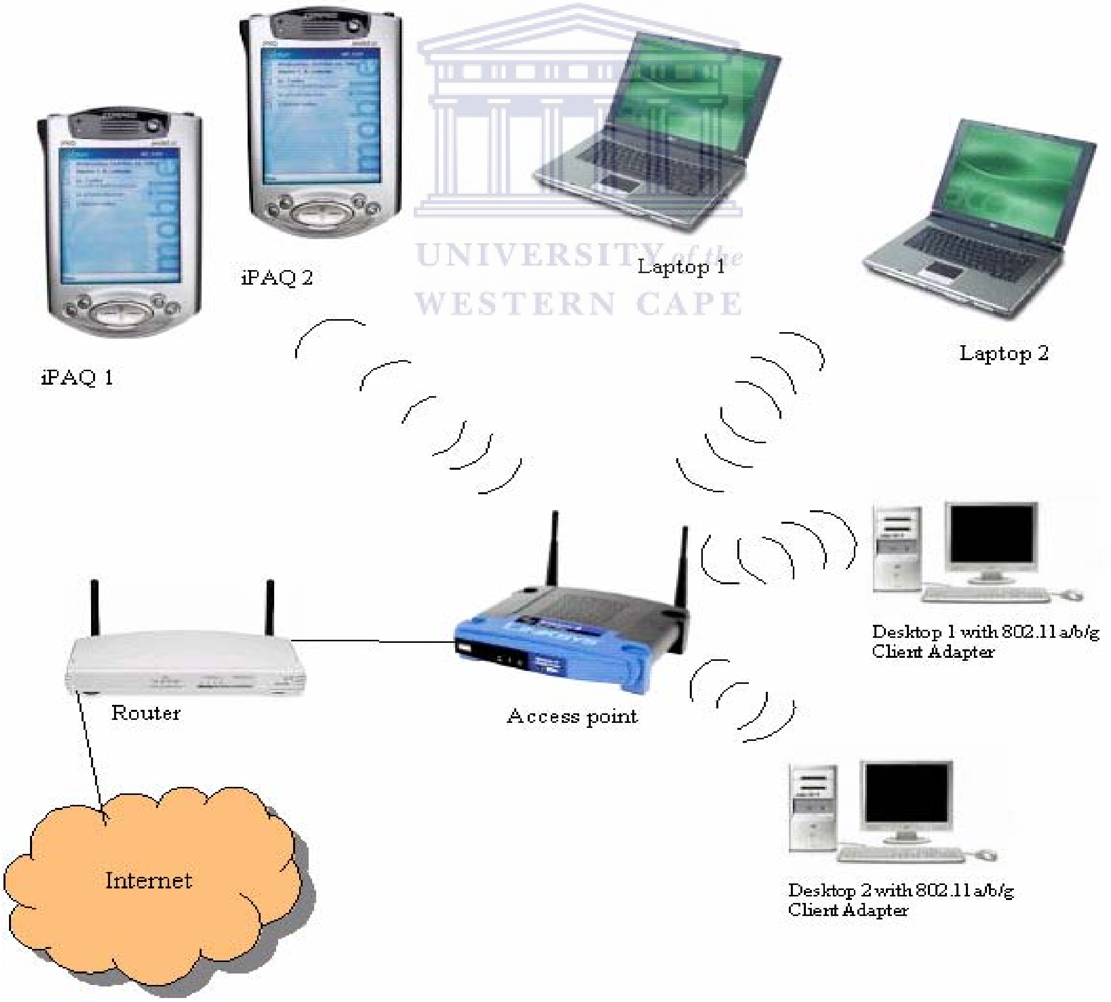

iPAQ 2
iPAQ 1
Laptop 1
Laptop 2
Desktop 1 with 802.11 a/b/g Client Adapter
Router
Access point
Internet
Desktop 2 with 802.11 a/b/g Client Adapter

Figure 6.3: Wireless network Environment Set-up

To test the network, simple messages were sent from one node to another. This test showed that messages took the acceptable normal time to move from the sending node to the receiving one. The messages sent included email, network messaging, file transfer, file sharing, etc. The test showed that the network is ‘healthy’ and working properly.

6.6 The experiment

Having setup and established connection (cf. Section 6.5.2), the experiment was carried out by injecting faults to the WLAN. It was important for the system to detect these faults as they occur, and based on the faults data; report them before they occurred. The MIA would move from one node to another depending on the colour indicated by that particular node (cf. Chapter 5).

Faults were injected at random to the nodes within the WLAN. To inject faults randomly and automatically, for example, PC-2 was programmed (using Ms Visual Basic 6.0) to automatically shut down. The program injecting power fault in the PC-2 comes with an option of stopping the shutdown. This option would be available for 5 seconds on the screen before it shuts down automatically. In the event that this fault (power failure in PC-2) is predicted to occur by MIA developed it can be stopped once the engineer is alerted of the imminent fault. The other faults include moving the device (one of the nodes, i.e., iPAQ) out of range, denial of migration, inaccessible node, unavailable resource, remote communication failure, and logic unavailable among others. Some of these faults are crude and simple, but it is important to note that the methods served well for the purpose of this work.

The execution of the system was done in iterations. Iteration represents a period of interruptible execution of the MIA under observation. A total of 400 iterations were made during a period of 3 months of intensive testing. Each iteration time took a period of 30 minutes. During each iteration times the MIA is expected to perform its functions. These include log in the detected cellular network faults into the database, calculate the ‘belief’ (equation 4.7) of each variable based on the fault logs, logs in a new cellular network fault and check on cellular network services affected by the fault detected as well as cellular network services that are going to be affected by the reported cellular network fault, among other functions. All these functions are repeatedly performed after a time window by the MIA.

6.6.1 Time Window

The 'belief' update is done after every five minutes. This time window was derived from the real data obtained from a certain cellular network service provider. The data showed five different types of fault variables. These include transmission, multiplexer, cell, power, and others. The time window is the average time-taken after the last fault (TALF). Table 6.3 gives a summary of the faults timing analysis with TALF value of 677.27 minutes. However, TALF value gotten was too long to wait for since time was a constraint. It would mean that network faults that occur after 10 minutes as shown in Table 6.3 would not be taken into consideration by the system.

Table 6.3: Network Faults Timing Analysis

Fault Type (variable)	**Start**	**End**	**Duration**	**Time After Last Fault**
Transmision (T)	Wed 16 Jan 2002,17:04	Thu 17 Jan 2002, 19:32	1,588	-
Others (O)	Thu 17 Jan 2002,00:26	Thu 17 Jan 2002,00:34	8	442 minutes
O	Thu 17 Jan 2002, 01:14	Thu 17 Jan 2002, 01:22	8	48 minutes
O	Thu 17 Jan 2002, 02:31	Thu 17 Jan 2002,02:29	8	77 minutes
O	Thu 17 Jan 2002, 02:41	Thu 17 Jan 2002, 02:49	8	10 minutes
Mux	Fri 18 Jan 2002, 17:12	Fri 18 Jan 2002, 20:17	185	2311 minutes
T	Sat 19 Jan 2002, 00:05	Sat 19 Jan 2002, 05:18	313	413 minutes
T	Sat 19 Jan 2002, 19:05	Sat 19 Jan 2002, 22:15	190	1140 minutes
O	Mon 21 Jan 2002, 10:48	Mon21 Jan 2002, 11:36	48	2383 minutes
Mux	Mon 21 Jan 2002, 13:37	Mon 21 Jan 2002,04:27	890	169 minutes
Cell ©	Mon 21 Jan 2002, 15:50	Mon 21 Jan 2002,16:10	20	133 minutes
T	Tue 22 Jan 2002, 13:44	Tue 22 Jan 2002, 00:52	516	1315 minutes
T	Tue 22 Jan 2002, 20:12	Tue 22 Jan 2002, 04:41	509	384 minutes
Power (Po)	Wed 23 Jan 2002, 16:51	Wed 23 Jan 2002,17:08	17	1239 minutes
T	Wed 23 Jan 2002, 18:26	Wed 23 Jan 2002,03:04	518	95 minutes
		Average time window		677.27minutes

The computed TALF is way too long even for a single run as per the design of this experiment. It therefore means that time window has to be adjusted to suit the experimental design. The time window of 5 minutes was arrived at after considering the experimental condition and real data.

6.6.2 The MORSBOSS Database

The database developed consists of five tables. The tables include *customers*, *faults*, *faultslog*, *resources*, and *services*. The *customers* table contains personal details of customers with the service provider, i.e., names, sex, date of birth and the service (*serviceID)* the customer has subscribed. *Faults* table include details like fault name, description, probability of occurrence, resolution when it occurred, the current state and colour. The *faultsLog* table is used to log in the faults as they occur in the network. The details include the fault that has occurred (denoted by faultID), start time and end time. Duration is calculated by subtracting start time from end time. The *faults* table uses *faultslog* table for its values. *Resources* table include resource name, the fault that frequently affects it, and what resource it depends on and its life expectancy. Lastly *services* table has the service name, requirements, resource it depends on and rate. The entity relationship diagram with details of fields in each table is shown in Figure 6.4.

The network faults would be logged into *faultslog* table. The MIA would compute the prior probability of the fault variables based on the information from *faultslog* table. The network fault variables would be aggregated and then conditional and joint probabilities are computed based on prior probabilities of the network fault variables. The common network faults are stored in the faults table. A network fault may affect (reduce, or cut-off) resources that support network services on offer. This effect is felt by the customers in the end. The customers would report the service anomalies, which are just the symptoms of the network faults affecting the resources that support network services. This relationship is shown in Figure 6.4.

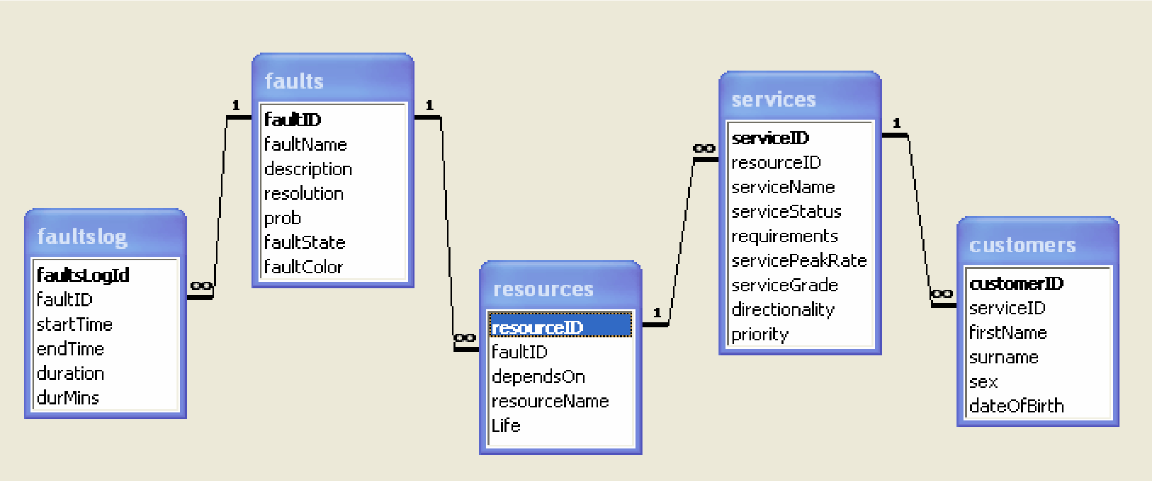
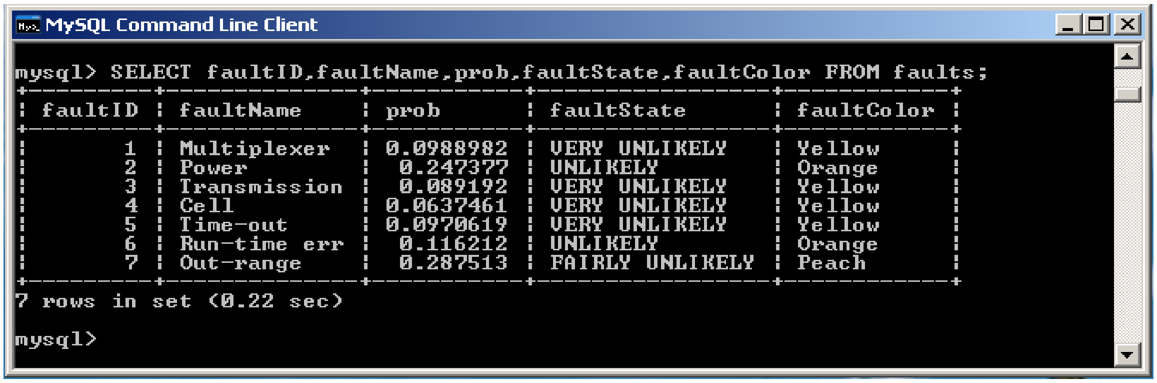


Figure 6.4: Entity Relationship Diagram

The faults data from a certain cellular network service provider helped in design and development of the database. Data collection was conducted during the operation of the system, which was stored in the designed database. The execution of the MIA yielded certain results that are discussed here after.

6.7 RESULTS AND DISCUSSION

In this Section, experimental results are provided. During each iteration time the MIA would update the 'belief' after a time window of 5 minutes, by logging in the faults and computing the new 'belief' of each and every variable. The most common variables based on the data received from a certain network service provider include Multiplexer, Power, Transmission and cell. However, seven different network faults were injected to the constructed WLAN. The node status is stored in the database and bears a typical format as shown in Figure 6.5.



```
mysql> SELECT faultID,faultName,prob,faultState,faultColor FROM faults;
+---------+---------------+-----------+-----------------+------------+
| faultID | faultName     | prob      | faultState      | faultColor |
+---------+---------------+-----------+-----------------+------------+
|       1 | Multiplexer   | 0.0988982 | VERY UNLIKELY   | Yellow     |
|       2 | Power         |  0.247377 | UNLIKELY        | Orange     |
|       3 | Transmission  |  0.089192 | VERY UNLIKELY   | Yellow     |
|       4 | Cell          | 0.0637461 | VERY UNLIKELY   | Yellow     |
|       5 | Time-out      | 0.0970619 | VERY UNLIKELY   | Yellow     |
|       6 | Run-time err  |  0.116212 | UNLIKELY        | Orange     |
|       7 | Out-range     |  0.287513 | FAIRLY UNLIKELY | Peach      |
+---------+---------------+-----------+-----------------+------------+
7 rows in set (0.22 sec)

mysql>
```

Figure 6.5: Seven Common Network Faults with Probability, State and Color

The customers consume network services, whose performances are affected by the network faults. Dependency on network services arises (cf. Chapter 3). In case of foreseen network fault, which is anticipated to affect certain network services, MIA may inform valuable consumers to such services by sending a message to them. Network engineer also receives a message detailing the imminent network fault(s). The typical message to the customers who consume the affected services looks like:

“Dear Customer, Fault ID: 1 has been created for service ID: 1 MORSBOSS will endeavour to resolve the problem ASAP. Please do not reply this SMS”

During the execution of each iteration time, the probability of a fault occurring was computed. During certain iteration times, the probability was very low as can be seen in some iteration times (i.e., 52nd, 68th, 69th, 104th, 126th, 151st, 300th, 326th, 381st, etc) network fault occurrence were negligible. There is a reducing trend of fault occurrence which can be observed. The frequency of network fault occurrence reduced tremendously from 0.977324 to 0.024581 during the 2nd and 385th iteration times respectively. The reduction symbolized the improvement of MIA’s performance. Robust nature of the MIA and prompt reporting of network faults before they occurred. Probability of fault occurrence in each iteration time is shown in Figure 6.6.

The network faults were reported before they occurred. Most of the network faults were reported between 8 and 13 minutes before they occurred. Power fault could be reported 13 minutes before it occurred. As shown in Figure 6.7, power fault occurrence reduces with the iteration times being made. The occurrence is almost zero from 275th to 400th iteration made. The occurrences of false alarms were taken into consideration. False alarms were computed using posterior probability. A false alarm of 8% was detected at the beginning of the experiment. It reduced to 5% but this can still be reduced further by improvement of MIA and more training of the software.

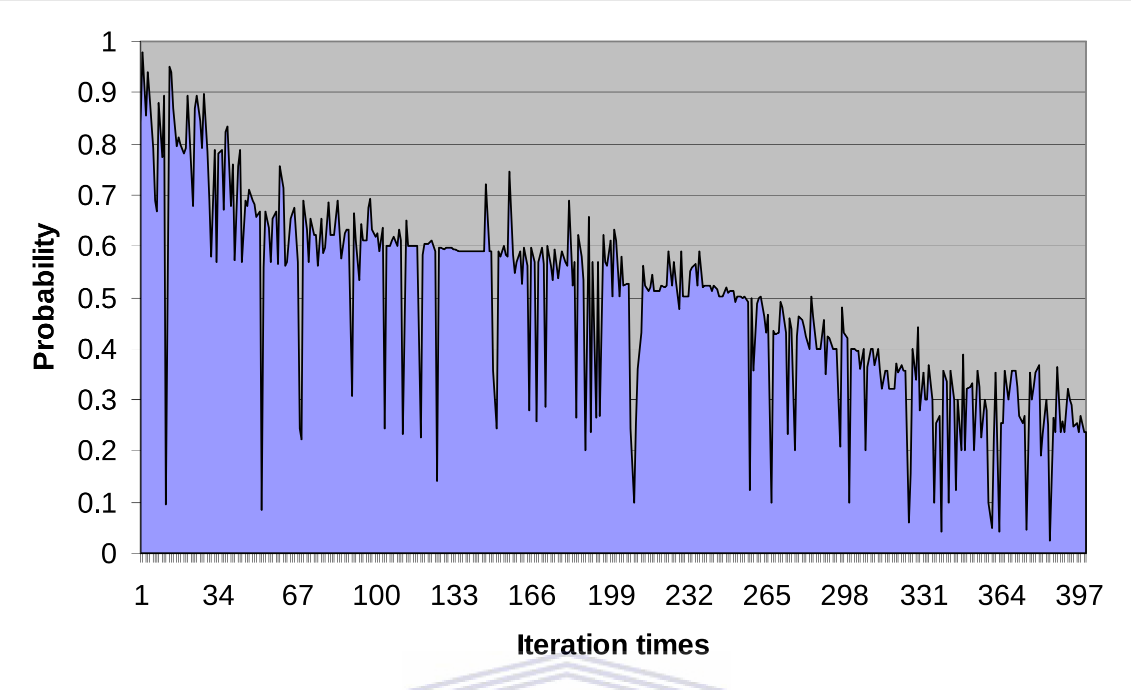


Figure 6.6: Probability of Network Faults in each of the 400 Iteration times made.

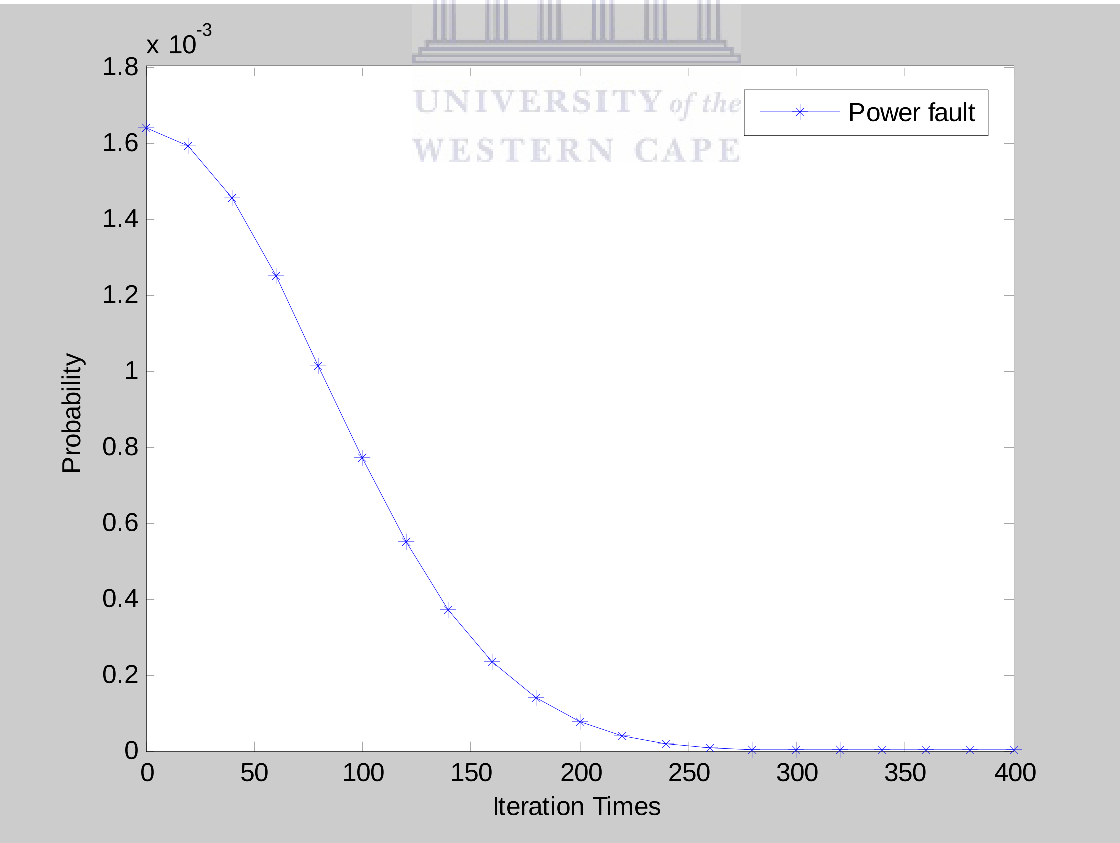


Figure 6.7: The Power Fault

Utility of MIA and network services i.e., voice and SMS were measured. Utility of MIA is the number of reported network faults before they occurred divided by the number of network faults that appeared during the 400 iteration times that were made. The network service utility (voice and SMS) were measured using the pattern of consumption of the services. Voice service utility is the number of calls that were connected divided by the number of calls attempted or that were supposed to be made after 400 iteration times. SMS utility is the number of SMSs delivered divided by the number of SMSs sent during the 400 iteration times (cf. Appendix B). The graph shows voice service having very low utility under the presence of network faults. While SMS service has a high utility of up to 75%. This is because SMS can be stored and delivered later to the recipients. However, it is impossible to send SMS from the faulty node. Figure 6.8 shows voice service utility. SMS service utility is shown in Figure 6.9.

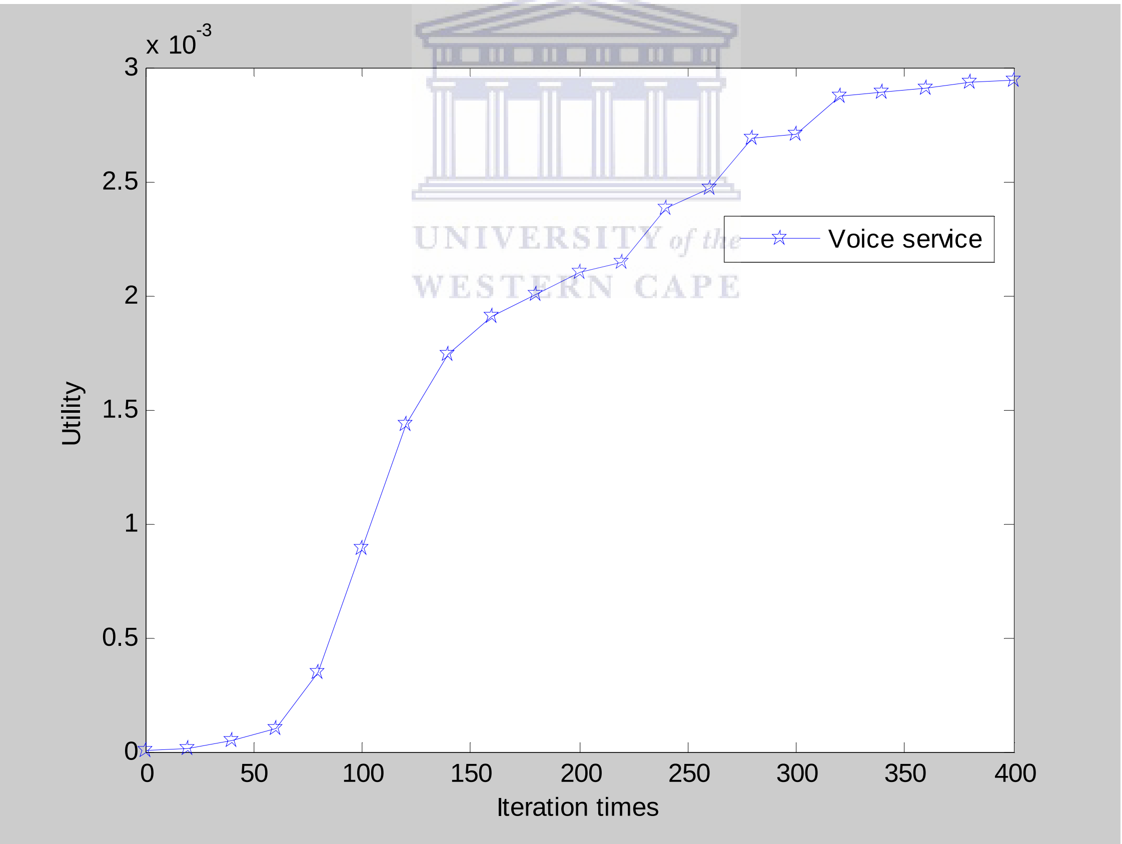


Figure 6.8: Network Service (voice service) Utility

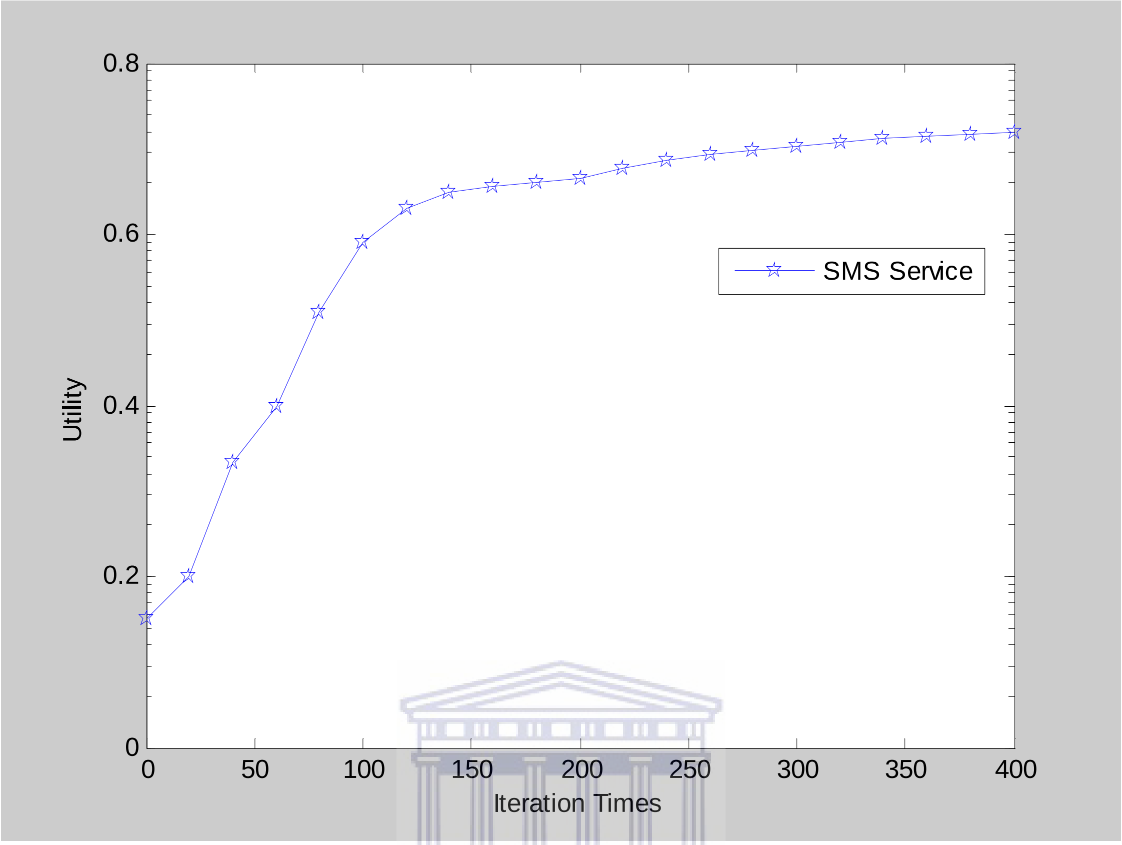


Figure 6.9: Network Service (SMS) Utility

Network service dependency (cf. Chapter 3) comes into question at this point. Taking a case study of network voice service, for example. A number of assumptions are made in computing the values based on derived models of network service dependency. Software availability was assumed to be 100%, other network faults were not considered except power fault, etc. Source availability, network availability, link availability, software availability, and destination device availability are 0.9958, 0.2343, 0.7737, 1.0, and 0.9958 respectively. Equation 3.1 was used to compute Service Availability (SA) with 17.982% being obtained. Network availability (NA) was computed using equation 3.3 and a value of 70.62% was obtained.

However, according to equation 3.2, service availability is supposed to be equals to network availability. This is not the case here and it can be attributed to assumptions made, network fault's impact and other factors which are beyond the scope of this work.

Network faults effects on network services at time *t* is given by Equation 3.5. The value of 99.95% shows that network faults directly affect network services. The margin of 0.05% can be attributed to noise. For more on computation see Appendix C.

The utility of network service (in this case voice) improves with the reduction of network fault (in this case power) occurrence. Network faults occurrence and network service utility matched at the 100th run. This point is called acceptance point as shown in Figure 6.10.

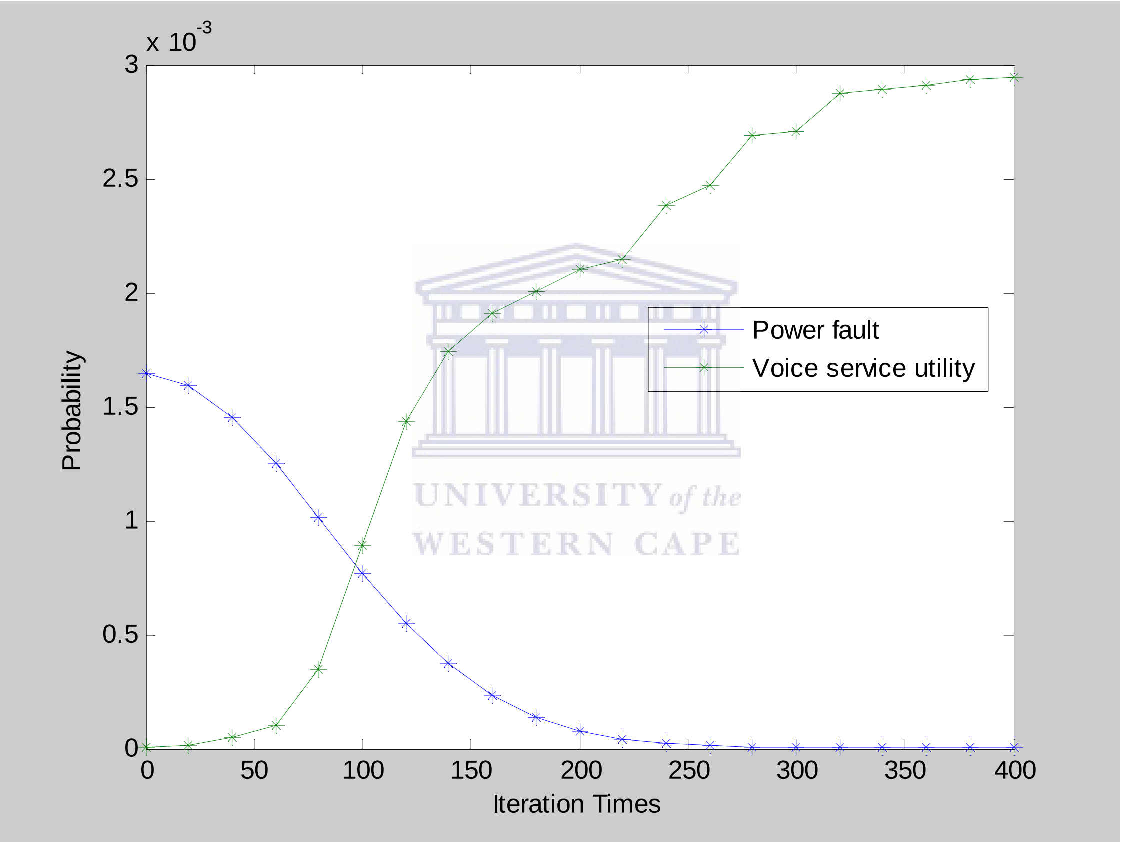


Figure 6.10: Network Service vs Network faults (Dependency)

The network faults injected are shown in Figure 6.11. Time-out fault is the least fault to occur in the network while all the other network faults occur in more or less the same pattern. The network faults occurrences diminish and almost to zero after 275th iteration time. This change can be attributed to the following factors:

- Shaky environment at the beginning of the experiment.

- Re-enforcement of power equipment and supply units. Most of the equipment used have got inbuilt power source, i.e., iPAQs and Laptops do come with built-in batteries that ensure the continuity of their operations even with power failures.
- It can also be attributed to MIA's learning and adaptive nature. By having experience of the network faults injected, it is able to report and inform the engineer in advance of their occurrence. This in turn enables the engineer to keep the network fault-free.

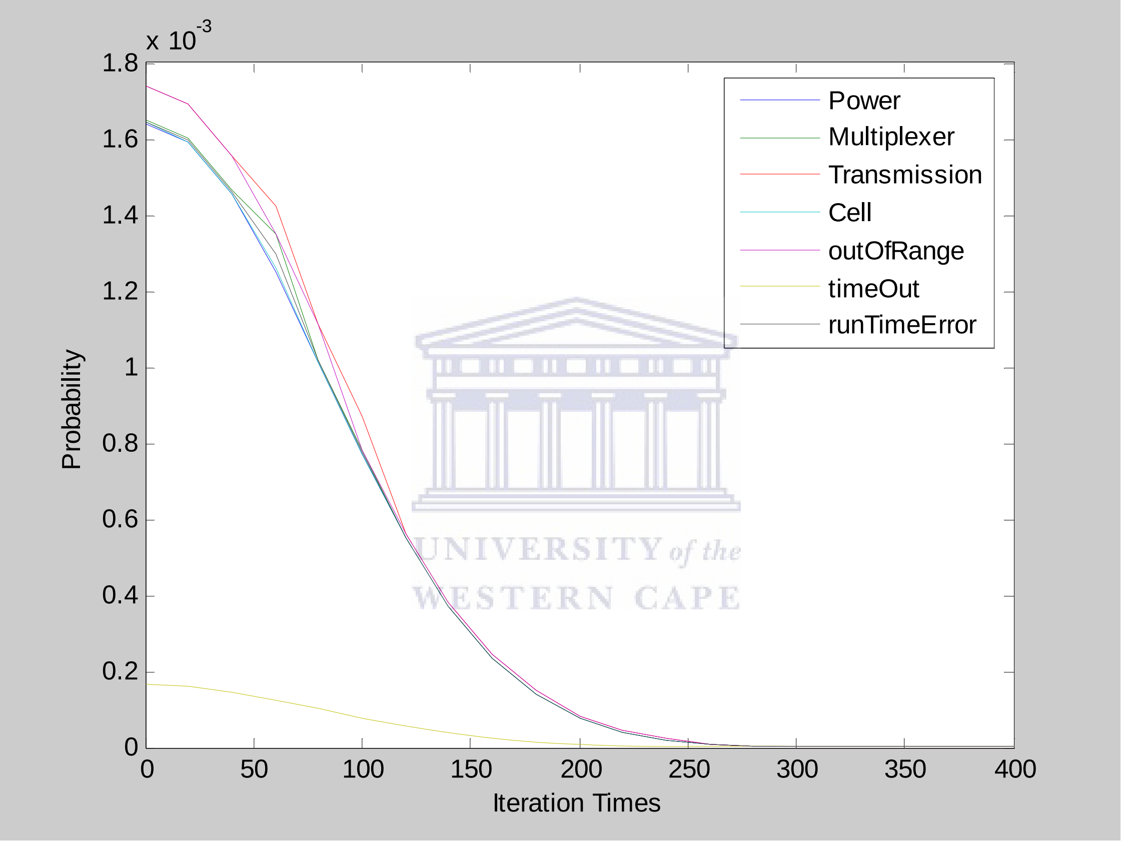


Figure 6.11: Network Faults Occurrence Showing all Faults

Figure 6.12 below shows the utility of the MIA deployed. It shows utility of 79% which can be improved on by fine tuning the software and performing more training of the MIA. However, this is one of the best results that can be obtained given the nature of the experiment and algorithms involved. It is also the best result as compared to literature and work done, which the author has seen in this area.

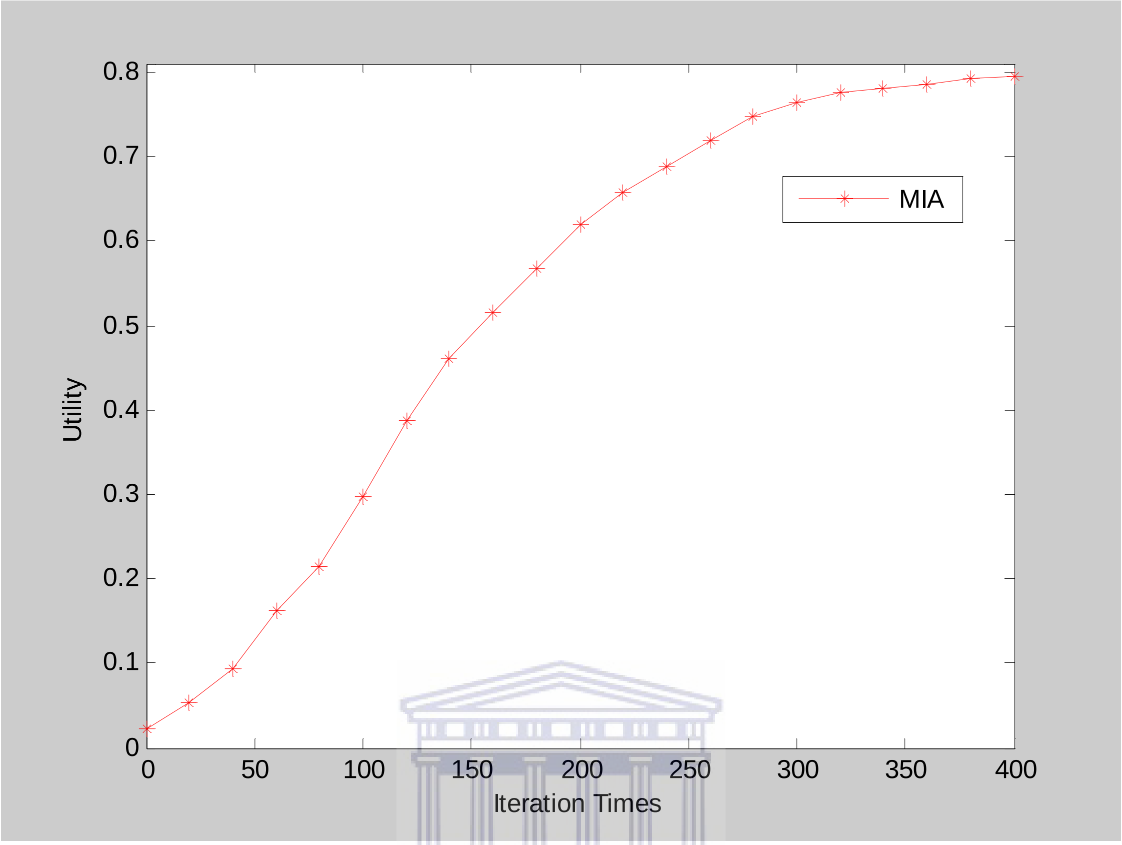


Figure 6.12: MIAs Utility

6.8 Summary

In this Chapter, the implementation software and technologies were presented. Challenges of embedding Mobile Agents into cellular network devices were presented. Experimental scenario, setup, and the process were presented. Impressive results of 98% detection rate, 5% false alarm rate, 8 to 13 minutes reporting of faults before they occurred and 79% utility rate of the MIA deployed were presented.

Chapter 7

CONCLUSION AND FUTURE WORK

In this Chapter, a summary of the work presented in this thesis is provided. A summarized discussion of the results obtained and challenges faced are presented. The future directions that this work may take for further improvement are also presented.

7.1 Conclusion

The underlying theme of this thesis is identification, classification, detection and prediction of cellular network faults using state of the art technologies, methods and algorithms. The state of the art technologies involves the use of Bayesian network as the framework for knowledge representation and evidence propagation. The use of biological behaviour of honeybees as analogy of the mobile intelligent agents used for monitoring and detection of network faults is one of its kind. The application of swarm optimization algorithm as a way of implementing the honeybee operation was presented. These methods and technologies ensured that a robust and heterogeneous system was designed and developed, which is fault-tolerant and effective.

This thesis began with an overview of the problem, highlighting the concerns and challenges of telecommunications network management.

Chapter 2 presented a panoramic view of cellular network faults management system. The evolution of network faults management system and telecommunication services architectures are presented. Software intelligent agent systems are discussed, detailing the benefits of their use in cellular network application. Overview of cellular network service dependency and network faults modeling are presented. The latest related works done in the area are also discussed.

Chapter 3 presented network services basics, classification, and issues affecting them. The relationship between network faults and services was established using data from a certain cellular network service provider. The network service dependency models were given detailing the kind of services that could significantly be affected by particular network faults. Simulation results of network service dependency were provided. It is through dependency that critical network services which could be affected by the common network faults were identified and attended in order to enhance the maintenance and provision of a robust system that in turn helps to keep the customers happy.

Chapter 4 presented faults modeling using Bayesian network theory. Using data from a certain cellular network service provider, the chapter started by classifying network faults using the faults logs that were provided. Bayesian network theory was explained stating the reasons for use, concerns and other areas of application. Cellular network faults prediction models were derived using Bayesian networks. The construction of the structure of the Bayesian network was discussed. The network faults prediction models formed the engine of the mobile intelligent agency.

Chapter 5 explains the mobile intelligent agency system development. The MIA developed was biologically inspired by the behaviour of the honeybees. The movement and control of the MIA required the application of swarm optimization algorithm. The proposed system architecture showing the most important components forming the solution architecture provided in this thesis were presented in this chapter.

Chapter 6 presented the implementation and experimental results of the proposed solution architecture. The software and technologies used were presented. The scenarios of experiments, the experimental setup and the experimental process were presented showing the state of the art details of how the work was done. A wireless local area network was setup involving eight devices that communicated with each other. Seven different network faults were injected into the network in order to verify the MIA’s ability to detect and predict the network faults before they occurred. The obtained results showed 98% detection rate of unknown and known network faults. The network faults (i.e., power) could be reported as early as 13 minutes before they occurred. The MIA showed utility of 79%. The MIA was scalable, robust and could execute certain functions concurrently ensuring faster reporting and execution of tasks.

7.2 Future Work

7.2.1 Performance of the MIA

In this work, the network faults prediction was the main theme and therefore the performance of the MIA was not evaluated. Even though the results showed a very good performance of the system developed, it could not be ascertained how it performs against other similar applications developed under different frameworks. It could therefore be very interesting to further investigate the performance of the MIA developed under JADE in comparison to other applications developed under different intelligent software frameworks (i.e., JADEX, Grasshopper, Aglets, cougaar, Ajanta, Tacoma, etc).

The computational complexity involved in the MIA, because of the use of Bayesian networks may have reduced the performance of the system. It could also be interesting to investigate how the performance can be improved may be by using another model (i.e., Fuzzy logic, neural networks, etc) instead of Bayesian networks.

7.2.2 Self Corrective MIA

The MIAs developed could detect and predict the network faults before they occurred. The MIAs could also move from one node to another within the network environment visiting mostly the 'notorious nodes' (nodes with high frequency of fault occurrence). In case of network fault, the MIA would report to the network specialist, valuable customers by sending messages that are presented as faults log to the database. However, MIAs may collide or experience run time error or any other problem thereby rendering the whole process unattainable. This anomaly needs to be corrected by the system immediately. It therefore, requires a self correcting MIA in case of any errors. This area was not investigated and therefore requires further investigation. The JADE framework used in this work has a fault-tolerance mechanism but that may not be enough in case of major errors. Further work may also investigate to what extent the fault-tolerance can sustain the operations of the MIA developed under such an environment.

8 REFERENCES

[1] Morris Sloman, editor, Network and Distributed Systems Management, Addison-Wesley, Wokingham, England, June 1994, pp. 455-480, ISBN 0-201-62745-0.

[2] ITU-T. Recommendation M.3010: Principles for a Telecommunications Management Network, May 1996.

[3] Alico Pras, Bert-Jan van Beijnum, Ron Sprenkels, "Introduction to TMN", CTIT Technical Report 99-09, University of Twente, Netherlands, April 1999. Also available in the Web: http://www.simpleweb.org/tutorials/tmn/index.html

[4] Masahiko Matsushita, "Telecommunication Management Network", NTT Review, Vol. 3 No. 4, July 1991, Page 117 – 122.

[5] ISO website: http://www.iso.org/iso/en/ISOOnline.frontpage

[6] ISO 9595, "Information Processing Systems – Open Systems Interconnection (OSI) – Common Management Information Service Definition", Geneva, 1990.

[7] ISO 9596, "Information Processing Systems – OSI – Common Management Information Protocol", Geneva 1991.

[8] ISO DIS 10165-1, "Information Processing Systems – OSI – Structure of Management Information – Part 1: Management Information Model", Geneva, 1993.

[9] ITU-T Recommendation X.700: Management framework for Open Systems Interconnection (OSI) for CCITT applications, September 1992.

[10] ITU-T TMN, Recommendation M.3100, 1993.

[11] ITU-T, Recommendation M.3010: Principles for a Telecommunications Management Network May 1996.

[12] ITU-T Recommendation M.3000: Overview of TMN recommendations, Oct. 1994.

[13] ITU-T Recommendations, M.3010 principles for a telecommunications management network, Feb. 2000.

[14] ITU-T Recommendation X.701: Information technology – Open Systems Interconnection – systems management overview, 1992.

[15] ITU-T Recommendation X.722: Information technology – OSI – structure of management information: Guidelines for the Definition of Managed Objects, 1992.

[16] ITU-T Recommendation M.3100: Generic network information model, July 1995.

[17] ITU-T Recommendation X.720: Information technology – OSI – structure of management information: Management information model, January 1992.

[18] ITU-T Recommendation M.3180: Catalogue of TMN management information, October 1992.

[19] ITU-T Recommendation M.3200: TMN management services: Overview, October 1992.

[20] ITU-T Recommendation X.710: Common Management Information Service definition for CCITT applications, 1991.

[21] ITU-T Recommendation X.721: Information technology – OSI – structure of management information: Definition of management information, 1992.

[22] ITU-T Recommendation M.3020: TMN interface specification methodology, October 1992.

[23] ITU-T Recommendation M.3300: TMN management capabilities presented at the F interface, October 1992.

[24] ITU-T Recommendation M.3400: TMN management functions, October 1992.

[25] ITU-T Recommendation Q.811: Lower layer protocol profiles for the Q3 interface, March 1993.

[26] ITU-T Recommendation Q.812: Upper layer protocol profiles for the Q3 interface, March 1993.

[27] ITU-T Recommendation X.711: Common Management Information Protocol specification for CCITT applications, 1991.

[28] ITU-T Recommendation X.723: Information technology – OSI – structure of management information: Generic management information, November 1993.

[29] ITU-T Recommendation X.730: Information technology – Open Systems Interconnection – systems management: Object management function, 1992.

[30] ITU-T Recommendation X.733: Information technology – Open Systems Interconnection – systems management: Alarm reporting function, 1992.

[31] ITU-T Recommendation X.734: Information technology – Open Systems Interconnection – systems management: Event report management function, 1993.

[32] ITU-T Recommendation X.790: Data networks and open system communications – trouble management function for ITU applications, November 1995.

[33] Michael H. Chawner and Robert D. H. Wu, "Network Management Architecture in ITS Telecommunications Networks", iMPath Networks, Nepean, Ontario, 2000.

[34] D.M. Meira, "A Model for Alarm Correlation in Telecommunications Networks", PhD Thesis, Federal University of Minas Gerais, Belo Horizonte, Brazil, Nov. 1997.

[35] The Insight Research Cooperation, "Wireless Operations Support Systems 2005-2010", and Available: http://www.insight-corp.com/reports/woss05.asp, Nov. 2005.

[36] Kornel Terplan, "Telecom Operations Management Solutions With Netexpert", CRC Press, Jun 1, 1998.

[37] The Insight Research Cooperation, Operations Support Systems (OSS) 2002-2007", Available: http://www.insight-corp.com/reports/oss2002.asp, Dec. 2002.

[38] Aidarous, S. and Plevyak, T., "Telecommunications Network Management: Technologies and Implementations", Wiley-IEEE Press, December 1997.

[39] Tereza Cristina Melo de Brito Carvalho, Vivian Bastos Dias and Christiane Marie Schweitzer, "Network Administration and Management Model – A Proposal for Corporate Network", Department of Computer and Digital Systems Engineering, University of Sao Paulo, 2000.

[40] Subramanian Mani, "Network Management – Principles and Practice, Addison-Wesley Longman, Inc., 2000.

[41] Henrik Niemann and Jakob Stoustrup, "Detection of Parametric Faults", Proceedings of 15th World Congress IFAC, Barcelona, 11th June 2004.

[42] Roy Sterritt, Dave Bustard, Andrew McCrea, “Automatic Computing Correlation for Fault Management System Evolution”, In Proceedings of IEEE International conference industrial information, Banff, Alberta, Canada, pp.240-247, Aug. 2003.

[43] 3GPP Websites – http://www.3gpp.org/ and http://www.3gpp2.org/ , June 2006.

[44] ETSI website – http://www.etsi.org/, June 2006.

[45] Parlay Group: Parlay API Specifications 4.0. Available: http://www.parlay.org/specs/index.asp, Jan 2004.

[46] Randy H. Katz and Anthony D. Joseph, “A Revolutionary Confederated Service Architecture for Future Telecommunications Systems”, MICRO Program Proposal, Computer Scie. Division, EECS Dept, University of California, Berkeley, March 2001.

[47] P. Stone and M. Veloso, “Multiagent systems: A survey from a machine learning perspective”, Autonomous Robots, vol. 8, pp.345-383, 1996.

[48] Wooldridge, M. and N.R. Jennings, "Agents Theories, Architectures, and Languages: a Survey”, in Intelligent Agents, Berlin: Springer-Verlag, pp. 1-22, 1995.

[49] M. Weiss, “A gentle introduction to agents and their applications”, http://www.magma.ca/~mrw/agents/, August 24, 2006.

[50] Wooldridge, M., *“An introduction to multi-agent systems”*, John Wiley Ed., 2002. ISBN 0-471-49691-X.

[51] M. J. Wooldridge and N. R. Jennings, “Intelligent Agents: Theory and Practice”, The Knowledge Engineering Review, vol. 10(2), pp. 115-152, 1995.

[52] D. B. Lange, “*Mobile Objects and Mobile Agents: The Future of Distributed Computing?”* , Lecture Notes in Computer Science, Volume 1445, pages 1-12, 1998.

[53] A. Bieszczad, B. Pagurek, T. White, “*Mobile Agents for Network Management*”, IEEE Communications Surveys, Fourth Quarter 1998, vol. 1, no. 1, pp. 2-9, 1998.

[54] T. Papaioannou, “Mobile Information Agents in Cyberspace: State of the Art and Vision”, In Proc. 4th International Workshop CIA2000, Boston, MA, USA, 7-9 July 2000.

[55] D. Kotz, R. Gray, “Mobile Agents and the Future of the Internet”, in ACM Operaing Systems Review, 33(3), pp.7-13, August 1999.

[56] C. G. Harrison, D. M. Chess, A. Kershenbaum, “Mobile Agents: Are they a good idea?” Technical Report, IBM Research Division, Watson Research Center, March 1995.

[57] Jennings, N.R, & Wooldridge, M., Application of Intelligent Agents, http://agents.umbc.edu/introduction/jennings98.pdf, Date Accessed: June 2006.

[58] FIPA, “FIPA ACL Message Structure Specification”, version E, October 2001.

[59] FIPA, “FIPA Abstract Architecture Specification”, version J, February 2002.

[60] G. Vigna, ed., “Mobile Agents and Security”, volume 1419 of Lecture Notes in Computer Science, Springer-Verlag, 1998.

[61] W. Shen, D.H. Norrie, “Agent-Based Systems for Intelligent Manufacturing: A state-of-the-Art Survey”, *Knowledge and Information Systems*, vol. 1, pp.129-156, 1999.

[62] G. Fleury, J-Y. Goujon, M. Gourgand and P. Lacomme, “Multi-Agent Approach for Manufacturing systems Optimization”, 1st International Conference on Practical Applications of Intelligent Agents and Multi-Agents, London, pp.225-244, April 1996.

[63] G. Lanzola, S. Folasconi and M. Stefanelli, “Cooperating Agents Implementing Distributed Patient Management”, 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, pp.218-232, Jan. 1996.

[64] Bhavna Orgun, “Interoperability in Heterogeneous Medical Information Systems Using Smart Mobile Agents and HL7 (EMAGS)”, Masters Thesis, Dept. of Computing, Macquarie University, Australia, June 2003.

[65] M. S. Fox, J. F. Chionglo and Barbuceanu, “The Integrated Supply Chain Management System”, Department of Industrial Engineering, University of Toronto, Toronto, Canada, Internal Report, 1993.

[66] V. Jamwal and Iyer S., “Mobile agents for effective structuring of large-scale distributed applications”, http://www.it.iitb.ernet.in/~sri/papers/ma-icse01.pdf, Aug.2006.

[67] F. M. T. Braizier, J. Treur, B. Dunin-Keplicz and N. R. Jennings, “DESIRE: Modeling multi-agent systems in a compositional formal framework”, Int. Journal of Cooperative Information Systems, vol. 6, pp. 67-94, 1997.

[68] D. Kinny, M. Georgeff and A. Rao, “A Methodology and Modelling Technique for Systems of BDI Agents”, 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindohoven, The Netherlands, pp. 56-71, January 22-25, 1996.

[69] JADE, http://jade.tilab.com/, June 2006.

[70] Antonio Moreno, Aida Valls, Alexandre Viejo, “Using JADE-LEAP to implement agents in mobile devices”, 2002.

[71] Object Management Group, “Mobile Agent System Interoperability Facility”, 1998.

[72] Galliers, J. R., “A Strategic Framework for Multi-Agent Cooperative Dialogue”, in *Proceedings of the Eighth European Conference on Artificial Intelligence (ECAI’88)*, August, Munich, Germany, 415-420, 1988.

[73] Rosenschein, J. S. and M. R. Genesereth, “Deals among Rational Agents, in *Proceedings* of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, CA, 91-99, 1985.

[74] Christian Ensel, “A Scalable Approach to Automated Service Dependency Modeling in Heterogeneous Environments”, Proc. 5th International Enterprise Distributed Object Computing Conference, pp.128-139, Seattle, WA, USA, 2001.

[75] D. Caswell and S. Ramanathan. “Using service models for management of Internet services”. In *HP Technical Report HPL-1999-43, HP Laboratories,* Palo Alto, California, USA, March 1999.

[76] Andreas Hanemann, David Schmitz, Martin Sailer, “A Framework for Failure Impact Analysis and Recovery with Respect to Service Level Agreements”, Proc. of IEEE International Conference on Services Computing, Vol.2, pp.49-58, 2005.

[77] ETSI Guide, Final drafts ETSI EG 202 009-3 V1.1.0 (2001-12) User Group; Quality of telecom services; Part 3: Template for Service Level Agreements (SLA). 2001.

[78] Hassan Hajji, B. H. Far and Jingde Cheng, “Detection of Network Faults and Performance Problems”, Proc. of the Internet Conference, Osaka, Japan, Nov. 2001.

[79] Hassan Hajji, Behrouz H. Far, “Continous Network Monitoring for Fast Detection of Performance Problems”, Proceedings of 2001 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, July 2001.

[80] Yan Lin and Marek J. Druzdzel, “Computational Advantages of Relevance Reasoning in Bayesian Belief Networks”, In Proceedings of the Thirteenth Annual Conference in Uncertainty in Artificial Intelligence (UAI-97), Pages 342-350, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1997.

[81] Lotfi A. Zadeh, In the engineering journal, Proceedings of the IRE, 1962.

[82] Lotfi A. Zadeh, Fuzzy sets, Information and Control, 8:338-353, 1965.

[83] J. Walrand and P. Varaiya, “High-performance Communication Networks”, Second edition, Morgan Kaufmann, 2000.

[84] D. Heckerman, Christopher Meek and Gregory Cooper, “A Bayesian Approach to Causal Discovery”, Technical Report, MSR-TR-97-05, Microsoft Research, Feb. 1997.

[85] D. Chickering, D. Heckerman, and C. Meek, Large-Sample Learning of Bayesian Networks is NP-Hard, *Journal of Machine Learning Research.* V5:pp.1287-1330, 2004.

[86] Duda, R.O., Hart, P.E., Stork, D.G. Pattern classification (2nd edition), Wiley, 2001, ISBN 0471056693.

[87] B. Widrow, “DARPA Neural Network Study”, AFCEA International Press, 1989.

[88] T. Mitchell, "Decision Tree Learning", in T. Mitchell, *Machine Learning*, The McGraw-Hill Companies, Inc., pp. 52-78, 1997.

[89] P. Winston, "Learning by Building Identification Trees", in P. Winston, *Artificial Intelligence*, Addison-Wesley Publishing Company, pp. 423-442, 1992.

[90] Roni Khardon and Dan Roth, “Defaults and Relevance in Model Based Reasoning”, Artificial Intelligence, Vol. 97, Number 1-2, pp. 169-193, 1997.

[91] Roni Khardon, Heikki Mannila and Dan Roth, “Reasoning with Examples: Propositional Formulae and Database Dependencies”, Acta Inf, 36(4), pp.267–286, 1999.

[92] Watson, Ian. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann, July 1997, ISBN: 9781558604629.

[93] Aamodt, Agnar, and Enric Plaza. "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches" *Artificial Intelligence Communications* 7, no. 1, pp. 39-52, 1994.

[94] Althoff, Klaus-Dieter, Ralph Bergmann, and L. Karl Branting (Eds), *Case-Based Reasoning Research and Development: Proceedings of the Third International Conference on Case-Based Reasoning*. Berlin, Germany, July 27-30,1999.

[95] Pat Langley and Herbert A. Simon, “Applications of Machine Learning and Rule Induction”, Communication ACM Vol. 38(11): pp. 54 – 64, November 1995.

[96] Daniel D. Corkill, “Blackboard Systems”, AI Expert 6(9): pp.40 – 47, Sept.1991.

[97] J. Frey and L. Lewis, “Multi-level reasoning for managing distributed enterprises and their networks”, In Integrated Network Management V, pp.5-16, 1997.

[98] C. Williamson, E. Halepovic, H. Sun and Y. Wu, “Characterization of CDMA2000 Cellular Data Network Traffic”, LCN, pp.712–719, Nov. 2005.

[99] F. M. Donini, M. Lenzerini, D. Nardi, F. Pirri, and M. Schaerf, “Non-monotonic reasoning”, *Artificial Intelligence Review*, Vol. 4(3), pp.163-210, 1990.

[100] Irene Katzela, A. T. Bouloutas, and S. Calo, “Comparison of distributed fault identification schemes in communication networks”, Technical report, IBM Corp., T.J. Watson Research Center, Yorktown Heights, NY, USA, January 1996.

[101] David Heckerman, Abe Mamdani, and Michael P. Wellman, “Real-world applications of Bayesian networks”, Communications of the ACM, 38(3), pp.24-26, March 1995.

[102] Z. R. Yang and M. Zwolinski, “A methodology for statistical Behavioural Fault Modeling”, Department of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK.

[103] X. Koutsoukos, F. Zhao, H. Haussecker, J. Reich and P. Cheung, *Fault Modeling for Monitoring and Diagnosis of Sensor-Rich Hybrid Systems,* Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA.

[104] J. Hartmann, B. Schieffer and U. Sparmann, *COFS–A cell oriented fault simulator*, Proceedings of the European Simulation Multi-conference, pp.424-429, 1992.

[105] Marina Thottan and C. Ji, “Proactive Anomaly Detection Using Distributed Intelligent Agents”, IEEE Net-work, Sept./Oct. 1998.

[106] Marina Thottan and C. Ji, “Fault Prediction at the Network Layer using Intelligent Agents”, Proc. IEEE/IFIP Integrated Network Management, May 1999, Boston, MA.

[107] Niki Pissinou et al., “Mobile Agents to Automate Fault Management in Wireless and Mobile Networks”, IPDPS Workshops, Lecture Notes in Computer Science, Vol.1800, pp.1296-1300, Springer, 2000.

[108] Andrzej Bieszczad, Benard Pagurek and Tony White, “Mobile Agents for Network Management”, IEEE Communications Surveys and Tutorials, 1(1), 1998.

[109] George Eleftheriou and Alex Galis, “Mobile Intelligent Agents for Network Management Systems”, Proceedings London Communication Symposium, 2000.

[110] Sterritt, R; Marshall, A H; Shapcott, C M; McClean, S I; *Exploring dynamic Bayesian Belief Networks for intelligent fault management systems*; Proc. IEEE Int. Conf. Syst. Man Cybern. Vol. 5, pp. 3646-3652. 2000

[111] Cynthia S. Hood and Chuanyi Ji; *Proactive Network Fault Detection*; In Proceedings of the IEEE INFOCOM, pp. 1139-1146, Kobe, Japan, April 1997.

[112] A. Lazar, W. Wang, and R. Deng, “Models and algorithms for network fault detection and identification: A review”, In *Proc. IEEE ICC*, Singapore, pp.999-1003, November 1992.

[113] B. Gruschke, “Integrated event management: Event correlation using dependency graphs” Proc. 9th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM 98), pp.130-141, October 1998.

[114] D. Caswell and S. Ramanathan, “Using service models for management of Internet Services”, In HP Technical Report HPL-1999-43, HP Laboratories, Palo Alto, California, USA, March 1999.

[115] M. Gupta, A. Neogi, M. Agarwal, and G. Kar, “Discovering Dynamic Dependencies in Enterprise Environments for Problem Determination” Lecture Notes in Computer Science, Vol. 2867, Springer Berlin, pp.125-166, 2003.

[116] Humberto Cervantes and Richard S. Hall, “Autonomous Adaptation to Dynamic Availability Using a Service-Oriented Component Model”, ICSE, pp.614-623, 2004.

[117] T. White, B. Pagurek, F. Oppacher, “ Connection Management using Adaptive Agents” , Proc. International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’ 98), pp. 802-809, Las Vegas, July 13-16, 1998.

[118] T. White, B. Pagurek, “Towards Multi-Swarm Problem Solving in Networks”, Proc. 3rd International Conference on Multi-Agent Systems (ICMAS’ 98), July 1998.

[119] T. White, A. Bieszczad, B. Pagurek, “Distributed Fault Location in Networks using Mobile Agents” Proc. International Workshop on Agents in Telecommunications Applications (IATA’ 98), Paris, France, 4-7 July, 1998, pp.130-141.

[120] Anders Holst, “The Use of a Bayesian Neural Network Model for Classification Tasks”, Thesis, Studies of Artificial Neural Systems, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden, September 1997.

[121] ITU-T Recommendation I.211, “B-ISDN Services Aspects,” Geneva, Switzerland, 1993.

[122] Moore’s Law, found at - http://www.intel.com/labs/eml/, Accessed on July 2004.

[123] Metcalfe’s Law available at: http://www.mgt.smsu.edu/mgt487/mgtissue/newstrat/metcalfe.htm, July 2004.

[124] ITU-T Study Group 2, Teletraffic Engineering Handbook, 2006. Also available at http://www.com.dtu.dk/teletraffic/handbook/telenook.pdf, Accessed on Aug. 2004.

[125] S.V. Kartalopoulos, "A Global Multi-Satellite Network for Multi-Media and PCS Service with Fault and Disaster Avoidance Characteristics, ICC (2), pp.694-698, 1997.

[126] Cynthia S. Hood and Chuanyi Ji, "*Automated proactive anomaly detection*", In Integrated Network Management V, California, USA, Vol.86, pp.688-699, 1997.

[127] Cynthia S. Hood, Chuanyi Ji. "Intelligent Agents for Proactive Fault Detection," IEEE Internet Computing, vol. 2(2), pp. 65-72, March/April 1998.

[128] Danyluk, A. and F. Provost, "Telecommunications Network Diagnosis", In W. Kloesgen and J. Zytkow (eds.), *Handbook of Knowledge Discovery and Data Mining, Oxford University Press, 2002.*

[129] Eugene Charniak, "Bayesian networks without Tears", AI Magazine, 50-63, 1991.

[130] Judea Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference", Artificial Intelligence Vol. 48(1), pp. 117-124, 1991.

[131] David Heckerman, "Bayesian Networks for Knowledge Discovery", Advances in Knowledge Discovery and Data Mining, pp.273-305, 1996.

[132] Heckerman, D., "A Tutorial on Learning with Bayesian Networks", In Learning in Graphical Models, pp. 301-354, MIT Press, 1999.

[133] Anna Dornhaus, Franziska Klugl, Frank Puppe, and Jurgen Tautz, "Task Selection in Honeybees – Experiments Using Multi-Agent Simulation", July 10 1999.

[134] Lars, Chittka, Adrian Dyer, Heather Whitney, Sarah Arnold and Beverley Glover, "Bees get a buzz from warm flowers", source Internet **http://news.bbc.co.uk/2/hi/science/nature/5236334.stm**, In Journal Nature, 03/08/2006.

[135] http://www.theage.com.au/news/National/Bees-vision-could-improve-microsurgery/2007/05/14/1178995052343.html, Accessed on 17th May 2007.

[136] http://www.sciencenews.org/articles/20050402/fob6.asp, Accessed on 17/5/2007.

[137] C.W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model", Computer Graphics, Vol. 21(4), pp. 25-34, July 1987.

[138] B.L. Partridge, “The Structure and Function of Fish Schools”, Scientific American, Vol. 246, pp.114-123, June 1982.

[139] T.J. Pitcher, B.L. Partridge, and C.S. Wardle, “Blind Fish Can School”, Science, 194:964, 1976.

[140] Kennedy, J. and Eberhart, R. C. Particle Swarm optimization. Proc. IEEE int'l conf. on neural networks Vol. IV, pp.1942-1948. IEEE service center, Piscataway, NJ, 1995.

[141] Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the sixth international symposium on micro machine and human science pp. 39-43. IEEE service center, Piscataway, NJ, Nagoya, Japan, 1995.

[142] Eberhart, R. C., P. K. Simpson, and R. W. Dobbins, Computational Intelligence PC Tools. Boston, MA: Academic Press Professional, 1996.

[143] http://www.engr.iupui.edu/~eberhart/, Accessed on 20th May 2007.

[144] www.particleswarm.net/JK/, Accessed on 20th May 2007.

[145] Eberhart, R. C. and Shi, Y. Particle swarm optimization: developments, applications and resources. Proc. congress on evolutionary computation 2001 IEEE service center, Piscataway, NJ. Seoul, Korea, 2001.

[146] http://www.engr.iupui.edu/~shi/Coference/psopap4.html, Accessed on 12/5/2007.

[147] Hu, X., Shi Y., and Eberhart, R.C. Recent Advences in Particle Swarm, Congress on evolutionary Computation, Portland, Oregon, pp.90-97, 19-23 June, 2004.

[148] http://www.aridolan.com, Accessed on 21st May 2007.

[149] http://www.red3d.com/cwr/boids/, Accessed on 21st May 2007.

[150] Corana, A., M. Marchesi, C. Martini, and S. Ridella, Minimizing Multimodal Functions of Continuous variables with the Simulated Annealing algorithm. ACM Transactions on Mathematical Software, Vol.13 (3), pp.262–280, 1987.

[151] Kirkpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi, Optimization by simulated annealing, Science Vol. 220, pp.671–680, 1983.

[152] J. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, Cambridge, MA, 1975.

[153] Michalewicz, Z, "Genetic Algorithms + Data Structures = Evolution Programs", 3rd ed., Artificial Intelligence, Springer-Verlag, 1996.

[154] A. Carlisle and G. Dozier, "Adapting Particle Swarm Optimization to Dynamic Environments", Proc. Int. Conference on Artificial Intelligence, pp.429-434, 2000.

[155] R.C. Eberhart and Y. Shi, "Tracking and Optimizing Dynamic Systems with Particle Swarms", Proc. IEEE Congress on Evolutionary Computation, vol.1, pp.27-30. IEEE Press May 2001.

[156] X. Hu and R.C.Eberhart, "Tracking Dynamic Systems with PSO: Where's the Cheese?" Proc. Workshop on Particle Swarm Optimization, pages 80-83, 2001.

[157] A. Carlisle, "Applying the Particle Swarm Optimizer to Non-Stationary Environments", PhD thesis, Auburn University, 2002.

[158] A. Carlisle and G. Dozier, "Tracking changing Extrema with Particle Swarm Optimizer", Technical Report, CSSE01-08, Auburn University, 2001.

[159] A. Carlisle and G. Dozier, "Tracking Changing Extrema with Adaptive Particle Swarm Optimizer", Proc. 5th Biannual World Automation Congress, pp.265-270, 2002.

[160] X. Hu and R. C. Eberhart, "Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems", Proc. IEEE Congress on Evolutionary Computation, Vol.2, pp.1666-1670, May 2002.

[161] M. Clerc, "Think Locally, Act Locally: The Way of Life of Cheap-PSO, an Adaptive PSO", Technical Report, http://clerc.maurice.free.fr/pso/, 2000.

[162] R.W. Morrison, "Performance Measurement in dynamic Environments", In GECCO 2003: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference, pp. 99-102, AAAI, 11 July 2003.

[163] Arnold D. V., “Local Performance of Evolution Strategies in the Presence of Noise”, Ph.D. thesis, Department of Computer Science, University of Dortmund, Germany, 2001.

[164] Elster C and Neumaier A., “A Method of trust region type for minimizing noisy functions”, Computing Vol. 58, pp. 31-46, 1997.

[165] Bayer H-G, “Evolutionary Algorithms in noisy environments: Theoretical issues and guidelines for practice”, Computation Methods Appl. Mech. Engrg. Vol. 186, pp.239-269, 2000.

[166] Box GEP and Muller ME, “A note on the generation of random normal deviates”, Ann. Math. Statistics Vol. 29, pp. 610-611, 1958.

[167] Antonio Moreno, Aida Valls, Alexandre Viejo, “Using JADE-LEAP to implement agents in mobile devices”, 2002.

[168] JADE-LEAP, http://leap.crm-paris.com/, Accessed on March 2007.

[169] JADE website, JADE “Java Agent DEvelopment Framework –” Home page for the JADE, http://jade.tilab.com/ , March 2006.

[170] Josef Altmann, Franz Gruber, Ludwig Klug, Wolfgang Stockner, Edgar Weippl, “Using Mobile Agents in Real World: A Survey and Evaluation of Agent Platforms”, Workshop on Infrastructure for Agents, MAS and Scalable MAS, pp. 33-39, June 2001.

[171] Foundation for Intelligent Physical Agents Specifications 2000, Available at http://www.fipa.org., Accessed on March 2007.

[172] Sun, http://www.sun.com, Accessed on March 2007.

[173] Pjava, Personal Java. http://java.sun.com/products/personaljava/pj-emulation.html

[174] Mobile Information Device Profile, http://java.sun.com/products/midp

[175] IEEE. IEEE Wireless Communications: Revolution toward 4G Mobile Communication Systems, Volume 10, August 2003.

[176] MATLAB - http://www.mathworks.com/ Accessed on 20 August 2007.

[177] NS-2 Simulator - http://www.isi.edu/nsnam/ns/ Accessed on 20 August 2007.

[178] OPNET Simulator - http://www.opnet.com/ Accessed on 20 August 2007.

[179] http://www.cam.com/vxutil_pers.html Accessed on 23 August 2007.

[180] http://www.pocketpcfaq.com/faqs/activesync/activesync4.0.htm, Accessed on 24 August 2007.

[181] http://support.fujitsu-siemens.com/com/support/downloads.html, Accessed on 24 August 2007.

[182] Mei-Chen Hsueh, Timothy K. Tsai, and Ravishankar K. Iyer, “Fault Injection Techniques and Tools”, IEEE Computer Volume 30, Issue 4, pp. 75-82, April 1997.

[183] JADE Administrator’s guide, February 2006.

[184] Theodora A. Varvarigou and Sudhir Ahuja, “MOFA: A Model for Fault & Availability in Complex Services”, IEEE Trans. On Reliability, Vol. 46, No. 2, 1997, pp. 222 – 232.

[185] Hong Cai, “A Two Steps Method For Analyzing Dependency of Business Services On IT Services Within A Service Life Cycle”, ICWS, pp.877-884, IEEE Computer Society, 2006.

[186] Judea Pearl, “CAUSALITY: Models, Reasoning, and Inference”, Cambridge University Press, 2000.

APPENDIX

Appendix A

Evaluation of Bayesian Networks

The joint probabilities distribution $P(x_1, x_2,...x_n)$ for a Bayesian network may be obtained through the product of the local probabilities distributions for each random variable. For example, the Bayesian network of Figure 4.4 in which the joint distribution $P(Po, Mux, C, T) = P(Po) \times P(Mux) \times P(C \mid Po, Mux) \times P(T \mid Mux)$, which is

$$
\begin{aligned}
& P(Po = Good, Mux = Ok, C = normal, T = normal) = P(Po = Good) \times P(Mux = Ok) \times \\
& P(C = normal \mid Po = Good, Mux = Ok) \times P(T = normal \mid Mux = Ok) \\
& = .9964 \times .7735 \times .9958 \times .2343 \\
& = 0.17982 \\
& = 17.982\,\%
\end{aligned}
$$

Probability that a set of variables $Y \subset X$ constituted by variables $X_m,...,X_p \in \{X_1, X_2,...,X_n\}$, assumes configuration $y = \{X_m = x_m,...,X_p = x_p\}$ is given by the sum of all probabilities of the X joint distribution in which $X_m = x_m,..., X_p = x_p$.

Taking again as an example the network of Figure 4.4, probability that, in the modeled system, power is good $(Po = good)$ and the transmission is abnormal $(T = abnormal)$ can be calculated as below:

$$
\begin{aligned}
& P(Po = good, T = abnormal) = \\
& P(Mux = ok, Po = good, T = abnormal, C = normal) + \\
& P(Mux = ok, Po = good, T = abnormal, C = uncertain) + \\
& P(Mux = ok, Po = good, T = abnormal, C = abnormal) + \\
& P(Mux = faulty, Po = good, T = abnormal, C = normal) + \\
& P(Mux = faulty, Po = good, T = abnormal, C = uncertain) + \\
& P(Mux = faulty, Po = good, T = abnormal, C = abnormal) = \\
& 0.7737 \times 0.9964 \times 0.38285 \times 0.9958 + \\
& 0.7737 \times 0.9964 \times 0.38285 \times 0.0021 + \\
& 0.7737 \times 0.9964 \times 0.38285 \times 0.0021 + \\
& 0.2263 \times 0.9964 \times 0.38285 \times 0.9958 + \\
& 0.2263 \times 0.9964 \times 0.38285 \times 0.0021 + \\
& 0.2263 \times 0.9964 \times 0.38285 \times 0.0021 = \\
& 0.2939 + 0.0006 + 0.0006 + 0.0860 + 0.0002 + 0.0002 = \\
& 0.3815 = 38.15\,\%
\end{aligned}
$$

Probability that transmission is abnormal (*T*=*abnormal*) in the same network is:

$$
\begin{aligned}
&P(T = abnormal) = \\
&P(Mux = ok, Po = good, T = abnormal, C = normal) + \\
&P(Mux = ok, Po = good, T = abnormal, C = uncertain) + \\
&P(Mux = ok, Po = good, T = abnormal, C = abnormal) + \\
&P(Mux = faulty, Po = good, T = abnormal, C = normal) + \\
&P(Mux = faulty, Po = good, T = abnormal, C = uncertain) + \\
&P(Mux = faulty, Po = good, T = abnormal, C = abnormal) + \\
&P(Mux = ok, Po = weak, T = abnormal, C = normal) + \\
&P(Mux = ok, Po = weak, T = abnormal, C = uncertain) + \\
&P(Mux = ok, Po = weak, T = abnormal, C = abnormal) + \\
&P(Mux = faulty, Po = weak, T = abnormal, C = normal) + \\
&P(Mux = faulty, Po = weak, T = abnormal, C = uncertain) + \\
&P(Mux = faulty, Po = weak, T = abnormal, C = abnormal) + \\
&P(Mux = ok, Po = blackout, T = abnormal, C = normal) + \\
&P(Mux = ok, Po = blackout, T = abnormal, C = uncertain) + \\
&P(Mux = ok, Po = blackout, T = abnormal, C = abnormal) + \\
&P(Mux = faulty, Po = blackout, T = abnormal, C = normal) + \\
&P(Mux = faulty, Po = blackout, T = abnormal, C = uncertain) + \\
&P(Mux = faulty, Po = blackout, T = abnormal, C = abnormal) = \\
&0.7737 \times 0.9964 \times 0.38285 \times 0.9958 + 0.7737 \times 0.9964 \times 0.38285 \times 0.0021 + \\
&0.7737 \times 0.9964 \times 0.38285 \times 0.0021 + 0.2263 \times 0.9964 \times 0.38285 \times 0.9958 + \\
&0.2263 \times 0.9964 \times 0.38285 \times 0.0021 + \\
&0.2263 \times 0.9964 \times 0.38285 \times 0.0021 + \\
&0.7737 \times 0.0018 \times 0.38285 \times 0.9958 + \\
&0.7737 \times 0.0018 \times 0.38285 \times 0.0021 + \\
&0.7737 \times 0.0018 \times 0.38285 \times 0.0021 + \\
&0.2263 \times 0.0018 \times 0.38285 \times 0.9958 + \\
&0.2263 \times 0.0018 \times 0.38285 \times 0.0021 + \\
&0.2263 \times 0.0018 \times 0.38285 \times 0.0021 + \\
&0.7737 \times 0.0018 \times 0.38285 \times 0.9958 + \\
&0.7737 \times 0.0018 \times 0.38285 \times 0.0021 + \\
&0.7737 \times 0.0018 \times 0.38285 \times 0.0021 + \\
&0.2263 \times 0.0018 \times 0.38285 \times 0.9958 + \\
&0.2263 \times 0.0018 \times 0.38285 \times 0.0021 + \\
&0.2263 \times 0.0018 \times 0.38285 \times 0.0021 = \\
&0.29390 + 0.00062 + 0.00062 + 0.08596 + 0.00018 + 0.00018 + 0.00053 + 0.000001 + \\
&0.000001 + 0.000155 + 0.0000003 + 0.0000003 + 0.0005309 + 0.0000011 + 0.0000011 + \\
&0.000155 + 0.00000033 + 0.00000033 = \\
&0.38283636 = 38.28\%
\end{aligned}
$$

A conditional probability may be calculated by using equation 4.6, according to which the probability for the occurrence of Mux, given that Po occurred, is given by the quotient between the probability of simultaneous occurrence of Mux and Po and the probability of occurrence of Po.

Therefore if one knows a set of evidences $e = \{X_m = x_m, ..., X_p = x_p\}$, constituted by all the known values of the random variables of a Bayesian network, where $\{X_m, ..., X_p\} \subset X = \{X_1, X_2, ..., X_n\}$, the calculation of the probability (or 'belief') that a variable $X_k \notin \{X_m, ..., X_p\}$ assumes the value x_k is given by equation 4.7.

To illustrate the above derivations, Bayesian network of Figure 4.3 is used. Supposing that $e = \{T = abnormal\}$ is the set of all the known evidences, the belief that the power is good is given by equation 4.8. By using the previously calculated probability values:

$$P(Po = Good \mid T = abnormal) = \frac{0.3815}{0.3828} = 0.9966 \approx 99.66\%$$

Supposing now that new evidence is known, multiplexer is faulty and totally knocked out, new belief that power is good is calculated as follows:

$$P(Po = Good \mid T = abnormal, Mux = faulty) = \frac{P(Po = Good, T = abnormal, Mux = faulty)}{P(T = abnormal, Mux = faulty)}$$

Where:

$$\begin{aligned}
&P(Po = good, T = abnormal, Mux = faulty) = \\
&P(Mux = faulty, Po = good, T = abnormal, C = normal) + \\
&P(Mux = faulty, Po = good, T = abnormal, C = uncertain) + \\
&P(Mux = faulty, Po = good, T = abnormal, C = abnormal) = \\
&0.2263 \times 0.9964 \times 0.38285 \times 0.9958 + \\
&0.2263 \times 0.9964 \times 0.38285 \times 0.0021 + \\
&0.2263 \times 0.9964 \times 0.38285 \times 0.0021 = \\
&0.0859645 + 0.0001813 + 0.0001813 = \\
&0.0863271 = 8.63271\%
\end{aligned}$$

$$
\begin{aligned}
&P(T = abnormal\ , Mux = faulty\ ) = \\
&P(T = abnormal\ , Po = good\ , Mux = faulty\ , C = normal\ ) + \\
&P(T = abnormal\ , Po = good\ , Mux = faulty\ , C = uncertain\ ) + \\
&P(T = abnormal\ , Po = good\ , Mux = faulty\ , C = abnormal\ ) + \\
&P(T = abnormal\ , Po = weak\ , Mux = faulty\ , C = normal\ ) + \\
&P(T = abnormal\ , Po = weak\ , Mux = faulty\ , C = uncertain\ ) + \\
&P(T = abnormal\ , Po = weak\ , Mux = faulty\ , C = abnormal\ ) + \\
&P(T = abnormal\ , Po = blackout\ , Mux = faulty\ , C = normal\ ) + \\
&P(T = abnormal\ , Po = blackout\ , Mux = faulty\ , C = uncertain\ ) + \\
&P(T = abnormal\ , Po = blackout\ , Mux = faulty\ , C = abnormal\ ) = \\
&0.2343 \times 0.9964 \times 0.2263 \times 0.9958 + \\
&0.2343 \times 0.9964 \times 0.2263 \times 0.0021 + \\
&0.2343 \times 0.9964 \times 0.2263 \times 0.0021 + \\
&0.2343 \times 0.0018 \times 0.2263 \times 0.9958 + \\
&0.2343 \times 0.0018 \times 0.2263 \times 0.0021 + \\
&0.2343 \times 0.0018 \times 0.2263 \times 0.0021 + \\
&0.2343 \times 0.0018 \times 0.2263 \times 0.9958 + \\
&0.2343 \times 0.0018 \times 0.2263 \times 0.0021 + \\
&0.2343 \times 0.0018 \times 0.2263 \times 0.0021 = \\
&0.0526093 + 0.0001109 + 0.0001109 + 0.0000950 + 0.0000002004 + \\
&0.0000002004 + 0.0000950 + 0.0000002004 + 0.0000002004 = \\
&0.053022 = 5.3022\ \%
\end{aligned}
$$

Therefore, $P(Po = Good \mid T = abnormal, Mux = faulty) = \dfrac{0.0863271}{0.053022} = 1.628$

NOTE: The value is more than 1, which is abnormal probability value gotten after the calculation of the example above. The reasons for this could be the assumption made on the value of some states.

Never the less, example presented above demonstrates capacity for non-monotonic reasoning of Bayesian networks, while the only known evidence was that Multiplexer was faulty, belief that power was good was of 99.66% as it is known that transmission was abnormal, the belief could be recalculated, having gone up to 162.8% (not correct value). This belief will grow even more as the information that the Cell is found to be in normal state is made available.

$$
\begin{aligned}
&P(Po = good \mid T = abnormal\ , Mux = faulty\ , C = normal\ ) = \\
&\frac{P(Po = good\ , T = abnormal\ , Mux = faulty\ , C = normal\ )}{P(T = abnormal\ , Mux = faulty\ , C = normal\ )}
\end{aligned}
$$

Where:

$$P(Po = good, T = abnormal, Mux = faulty, C = normal) =$$
$$0.9964 \times 0.38285 \times 0.2263 \times 0.9958 = 0.0859644 = 8.596\,\%$$

$$P(T = abnormal, Mux = faulty, C = normal) =$$
$$P(T = abnormal, Po = good, Mux = faulty, C = normal) +$$
$$P(T = abnormal, Po = weak, Mux = faulty, C = normal) +$$
$$P(T = abnormal, Po = blackout, Mux = faulty, C = normal) =$$
$$0.38285 \times 0.9964 \times 0.2263 \times 0.9958 +$$
$$0.38285 \times 0.0018 \times 0.2263 \times 0.9958 +$$
$$0.38285 \times 0.0018 \times 0.2263 \times 0.9958 =$$
$$0.0859644 + 0.0001553 + 0.0001553 =$$
$$0.086275 = 8.6275\,\%$$

Therefore,

$$P(Po = good \mid T = abnormal, Mux = faulty, C = normal) =$$
$$\frac{0.0859644}{0.086275} = 0.996399 \approx 99.6399\,\%$$

Appendix B

Impact of Network Faults

In this Section, simulation results based on faults logs from a certain network service provider are presented. There were four different faults that were taken into consideration. These are power, multiplexer, transmission and cell. The transmission fault occurred most and had more impact on the network operations as compared to the other network faults as shown in Figure B.1.

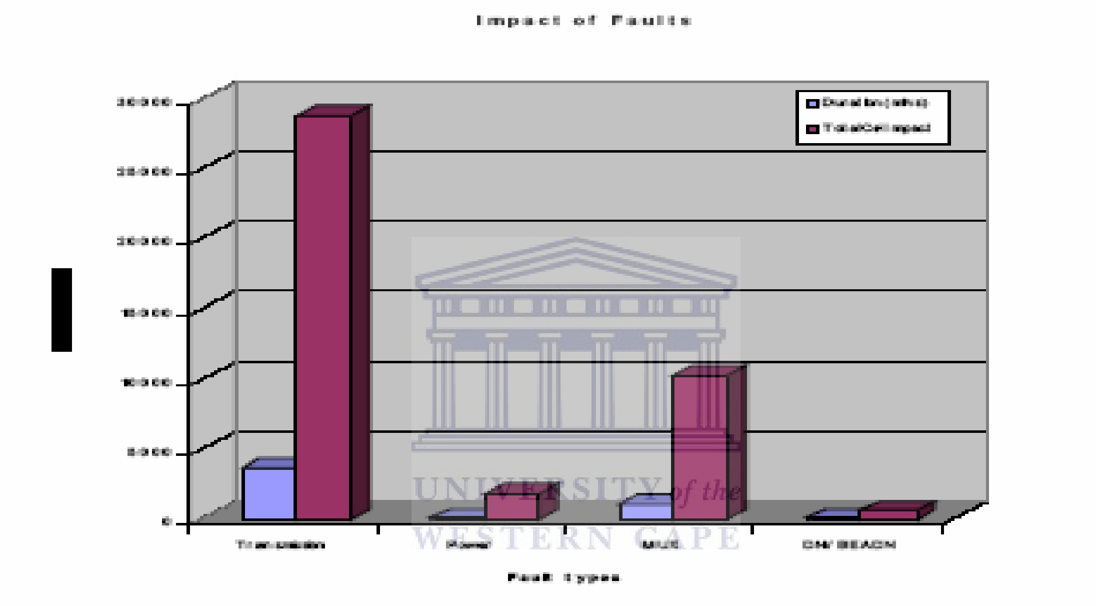


Figure B.1: Duration of Network Faults and Total Impact

Network fault variables are bundled into twos to see which pair of faults is likely to occur at once and cause devastating impact. Using product rule of probability for independent events the four common network faults were related to one another. Transmission and multiplexer faults have high probability (95%) of occurring together. Power and cell are the least likely to occur together at less than 5% as shown in Figure B.2.

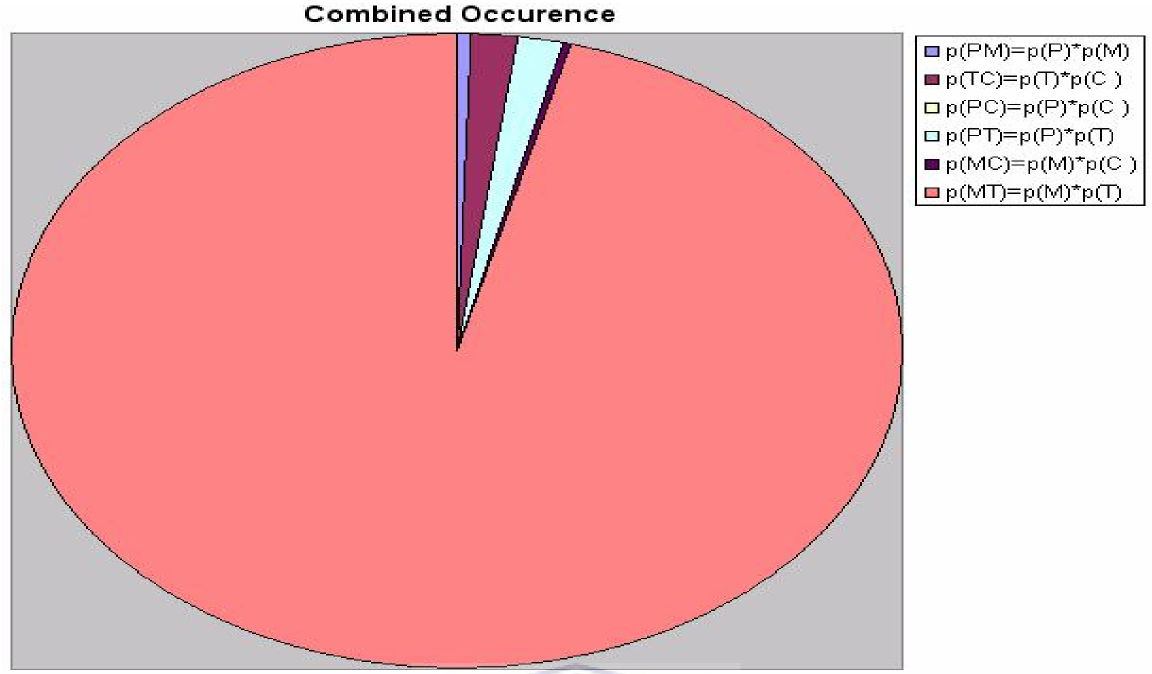


Figure B.2: Combined Network Faults Probability

Averages of 1027 SMSs were transacted every minute over the network. Computation of SMSs undelivered as a result of network faults were done, which shows that transmission caused more SMSs not to be delivered as compared to other network faults. This is shown in Figure B.3 and more details can be found in our publication titled, “Impacts and Cost of faults on Services in Cellular Networks”.

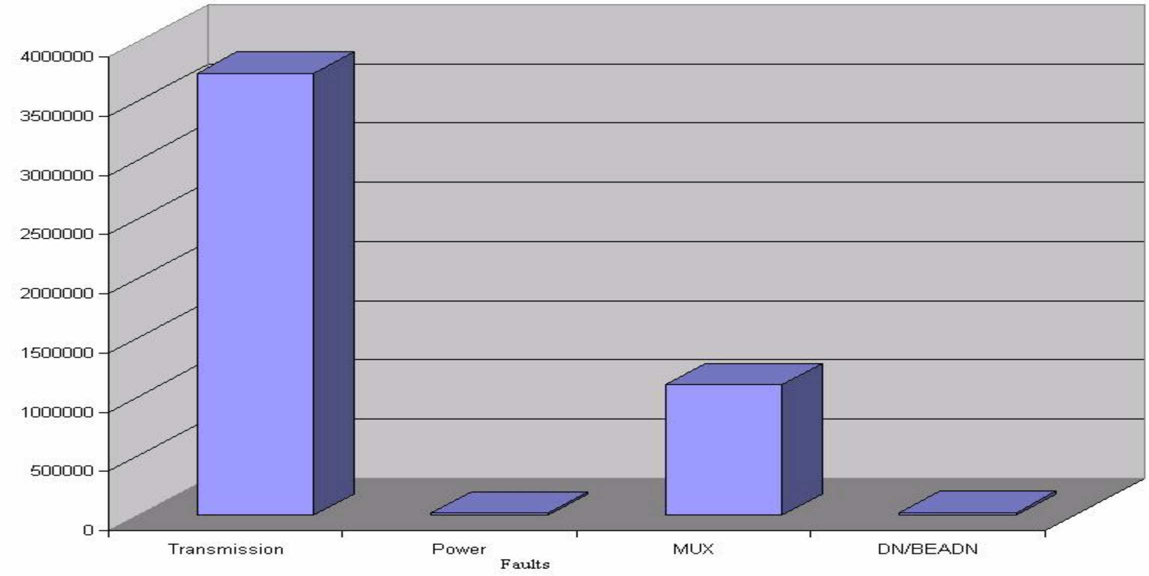


Figure B.3: Undelivered SMSes Caused by Network Faults

Appendix C

Network Service Dependency calculations

A number of resources were studied in connection to network services. For example, voice service, where an end to end connection is required for service consumption. For this to happen, the user must have a mobile/cell-phone and be situated in a cell where network signals are available for the linkage, which uses software to run all these devices to connect you to the destination device (cell phone), which also must be in a cell where network signals are available. Based on this logical argument and assuming that the user has money in his/her account for billing purposes, then it means voice service would depend on source (cell phone, cell), network, link, software, and destination device (cell phone, cell). Let software availability be 100%; network(transmission) be 23.43%; link(Mux) be 77.37%; Destination and originating source be 99.58%; therefore *voice service* availability can be computed using the Equation 3.1 as follows:

$$
\begin{aligned}
&SA = S(s) * N(s) * L(s) * Sw(s) * D(s) \\
&= 0.9958 * 0.2343 * 0.7737 * 1 * 0.9958 \\
&= 0.1797583732983324 \\
&= 17.976\%
\end{aligned}
$$

The network availability can be calculated using Equation 3.3 as follows:

$$
\begin{aligned}
&NA(t,s) = R(t,s) + F(t,s) * M(t,s) \\
&= (0.14845 + 0.8515494 * 0.655) \\
&= 0.706214857 \\
&= 70.62\%
\end{aligned}
$$

Network faults effects on network services can be computed using Equation 3.5 as follows:

$$
\begin{aligned}
&F(t,s) = 1 - (R(t,s))^{j} \\
&= (1 - (0.14845)^{4} \\
&= (1 - 0.00048564710894700625) \\
&= 0.99951435289105299375 \\
&= 99.95\%
\end{aligned}
$$

Appendix D

Equipment used in the experiment

Appendix E

Applications source code

The applications source code is available on request.