Institutionen för datavetenskap
Department of Computer and Information Science

Master's Thesis

Investigation of Automated Terminal Interoperability Test

Niklas Brammer

Reg Nr: LIU-IDA/LITH-EX-A--08/009--SE
Linköping 2008

Linköpings universitet
INSTITUTE OF TECHNOLOGY

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Institutionen för datavetenskap
Department of Computer and Information Science

Master's Thesis

Investigation of Automated Terminal Interoperability Test

Niklas Brammer

Reg Nr: LIU-IDA/LITH-EX-A--08/009--SE
Linköping 2008

Supervisor: **Rani Iskender**
Ericsson AB

Examiner: **Erik Larsson**
IDA, Linköpings universitet

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Avdelning, Institution
Division, Department

IDA
Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Datum
Date

2008-03-26

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ ______________

Rapporttyp
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ ______________

ISBN
—

ISRN
LIU-IDA/LITH-EX-A--08/009--SE

Serietitel och serienummer
Title of series, numbering

ISSN
—

URL för elektronisk version

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-10139

Titel
Title

Undersökning av automatiserad interoperabilitetstest av mobila terminaler
Investigation of Automated Terminal Interoperability Test

Författare
Author

Niklas Brammer

Sammanfattning
Abstract

In order to develop and secure the functionality of its cellular communications systems, Ericsson deals with numerous R&D and I&V activities. One important aspect is interoperability with mobile terminals from different vendors on the world market. Therefore Ericsson co-operates with mobile platform and user equipment manufacturers. These companies visit the interoperability developmental testing (IoDT) laboratories in Linköping to test their developmental products and prototypes in order to certify compliance with Ericsson's products. The knowledge exchange is mutual, Ericsson as well as the user equipment manufacturers benefit from the co-operation.

The goal of this master's thesis performed at Ericsson AB is to suggest areas in which the IoDT testing can be automated in order to minimize time consuming and tedious work tasks. Primarily the search should be aimed at replacing manual tasks in use today.

The thesis suggests a number of IoDT tasks that might be subject for automation. Among these one is chosen for implementation. The thesis also includes an implementation part. The task that has been chosen for implementation is the network verification after base station controller software upgrade procedure. This is not a core IoDT function but it entails a lot of work, and is often performed.

The automation project is also supposed to act as a springboard for future automation within IoDT. The forthcoming LTE standard will require a lot of IoDT testing, and therefore the automation capabilities should be investigated. The thesis shows that automation work is possible, and that the startup process is straightforward. Existing tools are easy to use, and well supported. The network verification automated test scope has been successful.

Nyckelord
Keywords

GSM, UMTS, LTE, Interoperability Test, Test Automation

Abstract

In order to develop and secure the functionality of its cellular communications systems, Ericsson deals with numerous R&D and I&V activities. One important aspect is interoperability with mobile terminals from different vendors on the world market. Therefore Ericsson co-operates with mobile platform and user equipment manufacturers. These companies visit the interoperability developmental testing (IoDT) laboratories in Linköping to test their developmental products and prototypes in order to certify compliance with Ericsson's products. The knowledge exchange is mutual, Ericsson as well as the user equipment manufacturers benefit from the co-operation.

The goal of this master's thesis performed at Ericsson AB is to suggest areas in which the IoDT testing can be automated in order to minimize time consuming and tedious work tasks. Primarily the search should be aimed at replacing manual tasks in use today.

The thesis suggests a number of IoDT tasks that might be subject for automation. Among these one is chosen for implementation. The thesis also includes an implementation part. The task that has been chosen for implementation is the network verification after base station controller software upgrade procedure. This is not a core IoDT function but it entails a lot of work, and is often performed.

The automation project is also supposed to act as a springboard for future automation within IoDT. The forthcoming LTE standard will require a lot of IoDT testing, and therefore the automation capabilities should be investigated. The thesis shows that automation work is possible, and that the startup process is straightforward. Existing tools are easy to use, and well supported. The network verification automated test scope has been successful.

Acknowledgments

With no particular order in mind, I would like to express my gratitude to the following people that have made this thesis work possible.

My colleagues in the IoDT team for always supporting me. Especially I want to thank my supervisor Rani Iskender, and my manager Roland Sevegran for giving me the opportunity to perform this thesis at Ericsson.

The kind people from the Automatic Testing Team and Ericsson Test Environments for supporting me with automation issues and test tool problems.

My examiner Erik Larsson at IDA, Linköping University.

Finally I would like to thank my opponent David Abrahamsson for valuable comments and improvement suggestions and Ron Brammer for proofreading this report.

Abbreviations

Throughout this thesis the following abbreviations will be used. They are also to be defined at their first appearance in the report.

1G First Generation

2G Second Generation

3G Third Generation

3GPP 3rd Generation Partnership Project

AMPS Advanced Mobile Phone Service

AT Attention (Hayes Command Set)

ATD Automated Test Design

ATE Automated Test Environment

ATLM Automated Test Life-Cycle Methodology

BSC Base Station Controller

BSS Base Station System

BTS Base Transceiver Station

CCN Cellular Coaxial Network

CDMA2000 Code Division Multiple Access 2000

CN Core Network

CS Circuit Switched

DSL Digital Subscriber Line

EDGE Enhanced Data Rates For GSM Evolution

EGPRS Enhanced General Packet Radio Service

EMP Ericsson Mobile Platforms

eNodeB Evolved UTRAN NodeB

EPC Evolved Packet Core

ETSI European Telecommunications Standards Institute

GGSN Gateway GPRS Support Node

GMSC Gateway Mobile Switching Center

GPRS General Packet Radio Service

GSM Global System for Mobile communications

GUI Graphical User Interface

HLR Home Location Register

HSCSD High-Speed Circuit-Switched Data

HSDPA High Speed Downlink Packet Access

HSPA High Speed Packet Access

HSS Home Subscriber Server

HSUPA High Speed Uplink Packet Access

I&V Integration and Verification

IMEI International Mobile Equipment Identity

IMSI International Mobile Subscriber Identity

IMTS Improved Mobile Telephone Service

IoDT Interoperability Developmental Testing

IP Internet Protocol

IRAT Inter Radio Access Technologies

ISDN Integrated Services Digital Network

KPI Key Performance Indicators

LAN Local Area Network

LTE Long Term Evolution

MIMO Multiple Input / Multiple Output

MML Man-Machine Language

MS Mobile Station

MSC Mobile Switching Center

MSISDN Mobile Subscriber ISDN

MTS Mobile Telephone Service

NGMN Next Generation Mobile Networks

NMT Nordisk MobilTelefoni

OFDM Orthogonal Frequency Division Multiplexing

PC Personal Computer

PCU Packet Control Unit

PLMN Public Land Mobile Network

PS Packet Switched

PSTN Public Switched Telephone Network

R&D Research and Development

RAN Radio Access Network

RF Radio Frequency

RNC Radio Network Controller

SAE System Architecture Evolution

SGSN Serving GPRS Support Node

SIM Subscriber Identity Module

SMS Short Message Service

STP System Test Plant

TACS Total Access Communications System

THC Test Harness Core

UE User Equipment

UMB Ultra Mobile Broadband

UMTS Universal Mobile Telecommunications System

USB Universal Serial Bus

UTRAN UMTS Terrestrial Radio Access Network

VLR Visitor Location Register

VoIP Voice over IP

WCDMA Wideband Code Division Multiple Access

WiMAX Worldwide Interoperability for Microwave Access

XML eXtensible Markup Language

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter aims to give the reader an introduction to the subject of this report. Furthermore it describes the purpose of the thesis and requested goals as well as how they are planned to be met. Finally the outline of the thesis is presented.

1.1 Background

In order to develop and secure the functionality of its GSM (Global System for Mobile communications) and UMTS (Universal Mobile Telecommunications System) systems, Ericsson deals with numerous R&D (Research and Development) and I&V (Integration and Verification) activities. One important aspect is interoperability with mobile terminals from different vendors on the world market. Consequently there is a special unit within the company, IoDT (Interoperability Developmental Testing), that deals with these matters. Mobile platform and user equipment manufacturers visit the IoDT test laboratories in Linköping to test their developmental products and prototypes in order to certify compliance with Ericsson's products. The knowledge exchange is mutual. Ericsson, as well as the user equipment manufacturers, benefit from the co-operation. Universally functional GSM and UMTS systems are in the interest of both Ericsson, the equipment manufacturers and the network operators.

To make the aforementioned tests possible to conduct, a GSM/UMTS infrastructural environment has been set up at the plant in Linköping. The test environment is used in many different areas. Testing in other sections of the company is highly automated, and it is requested that departments that have not yet followed also investigate the possibility to make their testing more efficient by means of automation. Test automation will be described in chapter 3, and IoDT automation in particular will be further investigated in chapter 3.4.

As within other technology areas, mobile telephony faces an evolution at an urgent pace. The forthcoming mobile standard LTE (Long Term Evolution) will soon be established, and the IoDT test laboratories need to have an environment prepared for tasks that might be involved. Ericsson aims to be a pioneer actor within LTE, taking an important part in the development and standardization work. In the dawn of unfolding the new technology, IoDT will play an important role in I&V testing of different products. There will be many new network elements, and interoperability testing will be of big importance in the initial phase.

1.2 Purpose

The goal of this master's thesis performed at Ericsson AB is to suggest areas in which the IoDT testing can be automated in order to minimize time consuming and tedious work tasks. Primarily the search should be aimed at replacing manual tasks in use today. If considered feasible, the thesis work will also include an implementation part. Secondarily future tasks involving LTE should be investigated and suggested. As the specifications for LTE are yet to be established, the I&V work, and specifically IoDT assignments, will be significant in the first phase. Since the same tests are foreseen to be run many times, test automation is desirable.

1.3 Research Approach

Having established the purpose above, this thesis aims to answer the following main questions:

- What tasks in use today in the IoDT are suitable, and possible to automate?
- Are there any other tasks outside the IoDT area that are feasible to conduct in an automated manner in the IoDT laboratories?
- What needs to be done in order to enable the IoDT laboratories to facilitate LTE test automation?

1.4 Methodology

In order to answer the questions in the previous section the following work is planned to be carried out:

- Investigate what has previously been done in the area of automated testing at Ericsson.
- Analyze which work tasks that are possible to automate within IoDT.
- Analyze whether there are other work tasks outside the IoDT area that can be automated in the IoDT laboratories.
- Automate manual test cases and test preparations, provided that the study suggests tasks that are reasonable to implement within the given time frame.
- Investigate how automatic test cases can be deployed from the genesis of LTE, the next generation cellular network.

1.5 Boundaries

Today, Ericsson utilizes a multi-vendor, multi-platform capable test environment called THC (Test Harness Core). It has been decided that THC also is to be used with the forthcoming LTE standard. Thus it is important for the company that the scope of this thesis is limited to apply to the existing THC test environment.

Primarily GSM features are tested in the Linköping IoDT laboratories today. However some IRAT (Inter Radio Access Technologies) handovers are involved in the testing, and therefore UMTS cells are available in the test environment. Interoperability in the future system LTE will be tested in Linköping before long and is thus subject to investigation. The first part of this thesis will focus on GSM features and the second part will aim to present an overview of LTE test automation. However, since the standards of LTE are yet to be determined, the work within the area is limited. LTE work in this thesis will thus mainly focus on an overview of possible automation solutions.

1.6 Target Audience

This thesis work may be of interest for various readers, but mainly it is intended for the project initiator. That is, the IoDT team at Ericsson AB in Linköping. The background chapters provide a good introduction to cellular mobile telecommunications, both historical and topographical, for anyone interested. The test automation chapters may be of interest for readers that work with automated test cases oriented toward software and hardware coaction. The report is written, so it should in some extent contribute even to a reader unfamiliar with the subjects covered.

1.7 Related Work

Since automation of test tasks currently is underway in many areas within Ericsson, work has already been performed. The aforementioned THC framework is used as an umbrella that gathers and interconnects several different automation tools. One of the tools, Mobitec, used in this thesis, has been refined and improved as a result of another master's thesis project at Ericsson [28]. As mentioned in the next section, in-house tools are also well documented, even though these documents are not categorized as related to this thesis.

1.8 Reference Literature

A mix of white papers from different manufacturers and standardization organizations, journal articles, books and Ericsson internal documents have been used when collecting background facts for this thesis. Ericsson internal documents referred to are unfortunately confidential, and can only be accessed through Ericsson's internal network.

1.9 Outline

This report consists of a few main sections. In the second chapter a background to cellular telecommunications and IoDT is presented. This chapter includes a historical overview of cellular telephony, a description of the GSM and UMTS networks and an introduction to IoDT.

The third chapter presents a theoretical base for test automation. It also describes tasks that have been automated within Ericsson previously, as well as possible IoDT tasks to automate.

The fourth chapter describes the implementation part. It gives arguments for the selected task for implementation and outlines the implementation process.

Finally the results of the work is presented. Conclusions and future work is discussed.

Chapter 2

Background

This chapter will present the reader with background facts on cellular telecommunication regarding different cellular systems, network topography and traffic cases. IoDT's mission and work will also be presented.

2.1 Cellular Telecommunications History

Cellular telecommunication systems have been subject to an exceedingly rapid evolution, ever since the introduction to the general public in the beginning of the 1980s. In this section some theoretical foundations which will aid the reader in understanding terminology regarding different systems will be presented. This will be accomplished by presenting a historical odyssey of some major milestones of radio communication and cellular telecommunication networks.

2.1.1 Early Radio Communications Systems

The history of mobile telephony out-dates the previously mentioned eighties public introduction by several decades. Some milestones of radio communications worth mentioning are presented below. For a more extensive listing, refer to [16, 22, 30].

The first successful long range transmission of human voice over radio was, according to [6], accomplished by Reginald Fesseden in 1906, when he on Christmas Eve played his violin for astonished telegraph operators at sea. Before this, Morse code was the only information possible to transmit.

In [16], it is claimed that it took an additional fifteen years before the radio technology was to be used for voice in scale. In 1921 The Detroit Police Department incorporated a 2 MHz one way vehicular mobile radio system. Due to the one-way nature of the system, it was necessary to find a wired phone in order to respond to radio calls.

In the 1930's the first two-way mobile system was launched in the United States. It incorporated push-to-talk[1] technology [16] and consequently it was still

[1] Half-duplex, a button is pushed to transmit.

not possible for both participants to talk simultaneously. However, the possibility for both participants to transmit was a great step forward.

The first commercial system, according to many, was launched 1946 in St. Louis by AT&T. The system, named MTS (Mobile Telephone Service), required the use of operators for dialing numbers. Furthermore, the number of channels was limited [6]. The system was in use for many years and in 1964 IMTS (Improved Mobile Telephone Service) was released, with improvements such as the elimination of push-to-talk and features like direct dialing and automatic channel selection [22]. The major problem was still the lack of capacity. In 1976, the New York waiting list for IMTS still exceeded 3,500 potential subscribers [30]. Similar mobile systems emerged around the world at the time.

Cellular telephony systems were discussed as early as 1947 by Bell Laboratories' D.H. Ring, according to [22]. The basic concepts were the same as the ones used today with, for example, frequency reuse and handovers. The first ever cellular system was introduced in 1969, as a pay phone on the Metroliner train running between New York and Washington, D.C.

2.1.2 First Generation

In the late 1970s and early 1980s, cellular phone systems such as AMPS (Advanced Mobile Phone Service) in North America and TACS (Total Access Communications System) in Great Britain started to be rolled out. The first large scale international cellular network NMT (Nordisk MobilTelefoni) will be described in the following section.

NMT

The first NMT system was paradoxically, considering its name, introduced in Saudi Arabia in 1981 [22]. After its first introduction in the middle east NMT networks were also deployed in the Scandinavian countries. As the first multinational cellular telephony system, its true benefit amongst other techniques was the capability of both national and international roaming. A feature that is commonplace in today's networks. At this time, in general, it was typical that different 1G (First Generation) cellular networks worked well autonomously but not in conjunction with other networks.

2.1.3 Second Generation

The major difference between 1G and 2G (Second Generation) cellular mobile telephony is the radio signaling. 1G systems use analog signaling, while 2G networks use digital signaling in the air interface. However both 1G and 2G systems use digital signaling to connect the radio towers with the rest of the system [37]. There are several 2G systems in use. However GSM is by far most widely deployed. Newer cellular phones tend to support several frequencies used in different parts of the world.

GSM

The foundation for GSM was laid in 1982, when 26 European national phone companies together started the development of the standard. The abbreviation GSM first represented Groupe Spécial Mobile, the name of the creating organization [22]. The first standard was launched in 1991 [16]. The GSM system has grown extremely rapidly and is still today the largest mobile telephony system. During 2006, the number of worldwide active GSM subscribers passed two billion [23, 24]. Even though the development of the third generation cellular system is well underway and the advent of the fourth generation lies in the near future, GSM will be the largest system for many years to come.

The first GSM release, contained features such as calling, SMS (Short Message Service), international roaming and basic fax and data services [16]. The flora of features has grown with the many new releases. The following sections will present major addendums, mainly data features, to the GSM standard. For a listing of 3GPP's (Third Generation Partnership Project) releases, refer to Appendix A. For a complete listing of releases and features added, see [2].

GSM Data

The original data functionality in GSM used CS (Circuit Switched) technology and had a maximum speed of 9.6 kbit/s. The demand for greater speeds was soon met with HSCSD (High-Speed Circuit-Switched Data) that multiplied the speed by allocating an increased number of time-slots per call [16]. Even though HSCSD was a great improvement to the original 9.6 kbit/s data speed, with theoretical speeds of between 28.8 kbit/s and 43.2 kbit/s, it had its drawbacks. The resource waste was unequivocal, since such a call would require the use of four GSM channels simultaneously. The need for a PS (Packet Switched) technology was evident [22].

GPRS

GPRS (General Packet Radio Service) was introduced in GSM Release 97 and is a technology that incorporates packet switching instead of demanding a closed circuit for the sending of data [38]. The result is that resources are not constantly reserved when a connection is established. GPRS collects data in packets, which are sent over the network when bandwidth is available. Resources are only requested when needed. GPRS is sometimes referred to as a 2.5 generation technology

EDGE

A further development of the GPRS technology is the EGPRS (Enhanced General Packet Radio Service) or EDGE (Enhanced Data rates for GSM Evolution), which improves the data transmission rates and data transmission reliability even further. EDGE incorporates a new modulation method which increases capacity on the air interface [16]. While the implementation of GPRS in the GSM networks only required a software update in the BSC (Base Station Controller), EDGE also requires a hardware upgrade [30]. Even though EDGE technically is a 3G (Third

Generation) technology it is sometimes referred to as a 2.75 generation technology, since it is used in the GSM network.

2.1.4 Third Generation

Cellular telephony systems have been optimized for voice since the beginning [20]. In the future, however, data traffic will be dominant in the networks, and therefore research focus is directed toward different technologies to improve data transfer rates. During 2007, the amount of packet data traffic has started to exceed voice traffic in 3G networks and the evolution toward even more data traffic is predicted to continue.

UMTS

As data services had been foreseen to take over the majority of traffic in cellular networks the convergence of these with voice services has been an important aspect from the start when designing the UMTS network [38]. From 1999, 3GPP has been responsible for standardization of both GSM and UMTS functionality. Release 99 included specifications for both GSM and the brand new UMTS access network UTRAN (UMTS Terrestrial Radio Access Network).

WCDMA (Wideband Code Division Multiple Access) is a UMTS technology, in which users are separated by means of a unique code assignment instead of by different time slots and frequencies. The bandwidth has also been greatly improved compared to GSM. The original WCDMA specification allowed a download speed of 384 kbit/s [38]. The terms UMTS and WCDMA are usually used interchangeably when referring to the network as a whole.

HSPA

HSPA (High Speed Packet Access) is a growing technology for high speed data transfers in the 3G networks [25]. Since UMTS Release 5, the downlink protocol HSDPA (High Speed Downlink Packet Access) has been launched in many networks worldwide [38]. Speeds achieved by HSDPA top 14.4 Mbit/s but most network operators provide speeds up to 3.6 Mbit/s or 7.2 Mbit/s.

HSUPA

HSUPA (High Speed Uplink Packet Access) is a feature that enhances the upload speeds from the mobile devices even further [33]. The HSDPA upload speed of 384 kbit/s has been increased to a maximum of 5.7 mbit/s. For the first time, the wireless cellular system will be a true competitor to wired techniques such as DSL (Digital Subscriber Line) networks that use the PSTN (Public Switched Telephone Network) for Internet access. HSUPA is not yet widely deployed. Most 3G WCDMA operators are eventually expected to adopt HSDPA and HSUPA since these technologies provide high value for the end users at a marginal incremental cost.

2.1.5 Beyond 3G

The trend has, as earlier mentioned, moved from high volume speech and low volume high speed data to the reversed relation [29]. Traffic has moved from using the circuit switched domain to incorporate packet switched traffic. Finally, earlier isolated networks have been developed so that inter-working between networks has been improved.

The evolution of GSM and UMTS networks has shown that a new generation mobile technology takes about ten years to develop and standardize. Before the technology is technically mature, another decade is required [29]. Since GSM was released in the beginning of the 1990s and UMTS in the end of the 1990s, the time is ripe for the next generation.

UMTS has been introduced i parallel with existing GSM networks [29]. In fact, much of the CN (Core Network) infrastructure is the same for the two systems. Even though the next generation systems will be launched soon, they need to be able to co-exist with 2G and 3G systems for many years.

Competing Standards

Worldwide operators work in an extremely competitive environment. They need to make the most of their current investments in 2G and 3G networks. This is why some of the world's leading operators have united with equipment vendors and research institutes and formed NGMN (Next Generation Mobile Networks) [32]. NGMN has established clear performance targets, recommendations and deployment scenarios for a future wide-area mobile broadband network. Some of the main demands are the efficient reuse of existing assets, both infrastructural and frequency spectrum and that the next generation has no impact on the current HSPA road map [20].

The main competing standards are all adopting a similar air interface technology, OFDM (Orthogonal Frequency Division Multiplexing), and claim comparable performance. The following sections introduce LTE, UMB (Ultra Mobile Broadband), and WiMAX (Worldwide Interoperability for Microwave Access) – the main competitors for the next generation mobile networks [36].

CDMA2000

One major player in the field is the CDMA2000 (Code Division Multiple Access 2000) system. It is optimized for wireless data, and current networks have a spectral efficiency similar to the one of HSUPA. One of the big problems with today's CDMA systems is the prevention of users from using voice and high-speed data services simultaneously, whereas this is possible with UMTS/HSPA. Eventually this will be solved by implementing VoIP (Voice over Internet Protocol) [4]. The emerging standard based on CDMA2000 is known as UMB [36].

Many operators already have CDMA2000 systems in use. Among them the Swedish operator Nordisk Mobiltelefon, that reuse old NMT frequencies. According to them their CDMA2000 coverage in Sweden exceeds the UMTS coverage by far [34].

The main drawback for CDMA2000 based systems is the lack of users. The GSM/UMTS systems add more users every year, than the entire current base of CDMA2000 users [4].

WiMAX

WiMAX was from the beginning a fixed wireless access specification. However, a mobile version has now been developed [36]. Earlier versions can be used in both licensed and unlicensed bands, whereas the latest addition, mobile WiMAX, has cellular features and operates in licensed bands. According to [4], any potential advantages of WiMAX do not justify replacing 3G systems. Instead WiMAX is being discussed as replacement for landline installations in developing countries.

Despite some significant radio innovations, WiMAX faces problems with spectrum, economies of scale and technology. Very few operators have access to sufficient WiMAX spectrum to provide widespread coverage [4].

LTE

LTE is the successor to GSM and UMTS and is developed by 3GPP – the consortium of Asian, European and North American telecommunications standards organizations [35]. Ericsson has been, and is one of the main proponents for LTE.

The main advantage of LTE is that it builds on existing GSM and UMTS networks [35]. Thereby it provides easy migration from the extremely wide-spread systems, providing an opportunity for network operators to save money on investment costs. LTE will be further explained in the following section.

2.1.6 3GPP Long Term Evolution

Ericsson has chosen the track of LTE and is currently one of the pioneers in development of standards within the area. NGMN has some criteria set for fulfilling their requirements on a new standard. LTE fulfills these criteria, even though the technology is not yet officially recognized by NGMN [20]. Some of the main targets when developing LTE are [12]:

- Significantly higher data transfer rates (100 Mbit/s downlink and 50 Mbit/s uplink).
- Three to four times higher average throughput compared to current HSDPA systems with enhanced uplink.
- Improved spectrum efficiency.
- Significantly reduced control and user plane latency.
- Reduced cost for operator and end user.
- Spectrum flexibility, allowing networks to be smoothly migrated into other frequency bands, for example the ones used by 2G networks today.

- Smooth introduction – ability to co-exist with current 3G radio access technologies.

Since the standardization of LTE is underway, all proposals may not necessarily be met in the final standard. However, some of the features that most likely are to be included will be discussed here.

All IP Network

The LTE network will use simplified core and transport networks, easier to build, maintain and extend with new services. These networks will be completely IP-based (Internet Protocol) [20]. The evolution of the existing GSM/WCDMA core networks will provide a flatter network architecture, designed to efficiently support any IP-based service.

Advanced Antenna Technology

Multi-antenna technologies imply that multiple antennas are used at the sender and/or receiver [9]. Multi-antenna technologies can be used to improve cell capacity and coverage and to provide higher per-user data rates. The use of multiple antennas can also provide additional diversity against fading on the radio channels.

If multiple antennas are used both at transmitter and receiver, multiple parallel communication channels can be set up over the radio interface [9]. This provides the possibility for higher data rates within a limited bandwidth without a significant loss of radio coverage. This feature is referred to as MIMO (Multiple Input Multiple Output).

Flat Network Architecture

To support the new packet data capabilities described earlier an evolved core network is being developed. The evolution is known as SAE (System Architecture Evolution). A major goal when designing the LTE RAN (Radio Access Network) has been to reduce the number of different nodes to one. This has been accomplished by moving functionality to the enhanced base stations, eNodeB (Evolved UTRAN NodeB). The eNodeB has inherited RNC (Radio Network Controller) functionality from the WCDMA architecture. Fewer nodes in the network provide lower latency [9]. The proposed network structure will be presented in chapter 2.2.7.

OFDM Air Interface

As mentioned earlier OFDM is used for several wired and wireless technologies, for example WiMAX and digital video broadcasting. For LTE, OFDM is used as the downlink transmission scheme. OFDM, which is a kind of multi-carrier transmission, uses a relatively large number of narrow band sub-carriers. OFDM has high spectral efficiency[2] [9]. The lack of sensitivity to time synchronization errors also makes it suitable for broadcast services, such as television [8].

[2] The amount of information that can be transmitted over a given bandwidth.

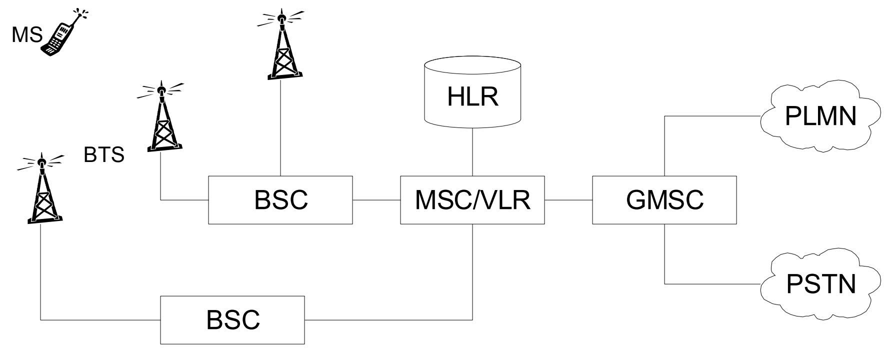

Figure 2.1. Simplified GSM Network Structure

2.2 Cellular Network Topography

In order to understand the nodes and interfaces that need to be analyzed when testing, an overview of the GSM system will be presented in this section. An introduction will also be given to the 3G network structure, which differs from GSM, even though some of the nodes are the same. New LTE networks will significantly differ from the previous ones. Discrepancies will be sorted out in the following subsections. The reader should keep in mind that simplified views of the networks are used.

2.2.1 GSM Network Structure

The GSM network consists of the RAN, also known as the BSS (Base Station System) and the core network, also known as the switching system [15]. The RAN consists of BTSs (Base Transceiver Stations) and BSCs. One BSC can control many BTSs. The core network includes the MSC (Mobile Switching Center), with connections to other networks, and other supporting nodes (GPRS will be presented in section 2.2.2). The GSM structure is shown in Figure 2.1.

Mobile Station

The MS (Mobile Station) is the user part of the system, it could be for example a mobile phone or a PC (Personal Computer) card [37]. The MS connects the user with the network. In the GSM system the MS also includes a SIM (Subscriber Identity Module) card which contains user data. The SIM card can easily be moved between different MSs. Each MS also has its unique identifier, the IMEI (International Mobile Equipment Identity) number.

Base Transceiver Station

The BTS, or simply base station, is the most visible element of the GSM system. This is since it often includes a large antenna system [38]. The BTS is also the most numerous element in the network. BTSs can cover cells of different sizes depending on the number of simultaneous users. In theory a cell can cover a radius of 35 km.

Base Station Controller

The BSC is responsible for connection establishment, release and maintenance of all cells connected to it [38]. When a subscriber, for example, wants to make a call or send an SMS, the MS sends a channel request message to the BSC. The BSC then checks for resources and allocates a channel in the BTS. Handovers are also managed by the BSC. More about traffic scenarios in 2.2.5. The BSC includes a TRAU (Transcoding and Rate Adaption Unit). The TRAU is responsible for compression and decompression of the voice data streams. One BSC can serve many BTSs.

MSC & GMSC

The MSC is the central part of a mobile telecommunication network [37]. Each PLMN[3] (Public Land Mobile Network) typically has a few MSCs depending on the number of subscribers. All connections between subscribers are managed by the MSC. A connection to the PSTN is also needed, and this is accomplished by a GMSC (Gateway Mobile Switching Center).

VLR & HLR

Every MSC has a VLR (Visitor Location Register) which keeps track of all currently served subscribers [38]. The data is copied from the subscribers' HLR (Home Location Register). The main reason why the VLR is used, is to reduce traffic between the MSCs and HLRs. When a user moves between different MSCs, the user data is first copied to the new VLR and then removed from the previous one.

The HLR is the subscriber database of the GSM network [38]. Each HLR contains data about the network's users, and their available services. Each subscriber has a unique IMSI (International Mobile Subscriber Identity) number, which is stored in the HLR and in the users SIM card. The IMSI is used for most subscriber-related signaling in the network [38]. Logically, there is only one HLR per GSM network, even though it may be physically split up [5].

2.2.2 GPRS/EDGE

Packet data over GPRS is an expansion to the GSM network [26]. The new structure is presented in Figure 2.2. Since GPRS systems were introduced, the BSC also includes a PCU (Packet Control Unit). It can be installed as part of the BSS at different locations in the network. Most common, though, is that the PCU resides

[3] In Sweden, for example: Telia, Tele2, Telenor.

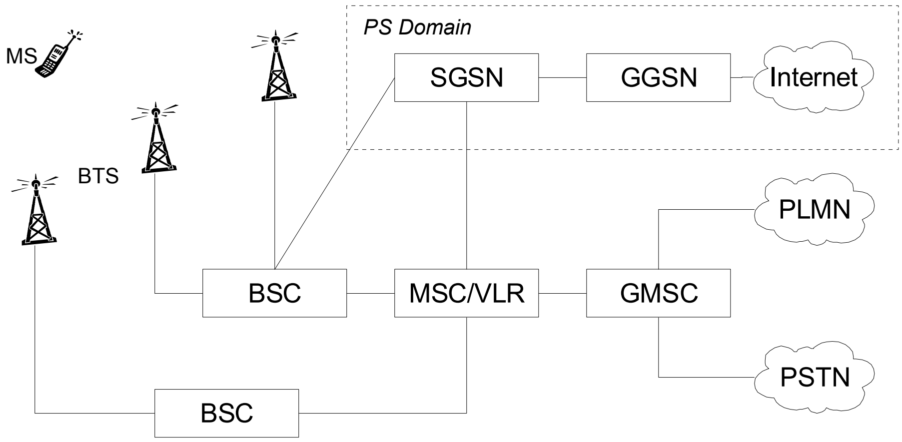

Figure 2.2. Simplified GPRS Network Structure

as a few additional expansion cards in the BSC. The packet control unit handles the PS data traffic through the BSC. Two more elements have also been introduced in the network, the SGSN (Serving GPRS Support Node) and the GGSN (Gateway GPRS Support Node). These are presented in the following sections. For a thorough introduction to GPRS, refer to [26].

Serving GPRS Support Node

The SGSN is a part of the GPRS core network and serves many PCUs and BSSs [26]. Put simply, the SGSN in the GPRS network, performs the same task as the MSC and the VLR in circuit switched GSM. In addition to these, it also handles GPRS specific functions.

Gateway GPRS Support Node

The GGSN is the gateway between SGSNs and the rest of the IP network (primarily the Internet) [26]. Externally the GGSN acts as a normal router. An operator will need at least one GGSN to operate a GPRS network. The geographical coverage does not decide the need for additional SGSNs, the number of users and the payload does. The GGSN, together with the SGSN handles charging in the GPRS network.

2.2.3 System Management

The OSS (Operation and Support System) is used for GSM system administration [15]. OSS supports the operator with functions such as mobile subscriber, cellular network administration and alarm handling.

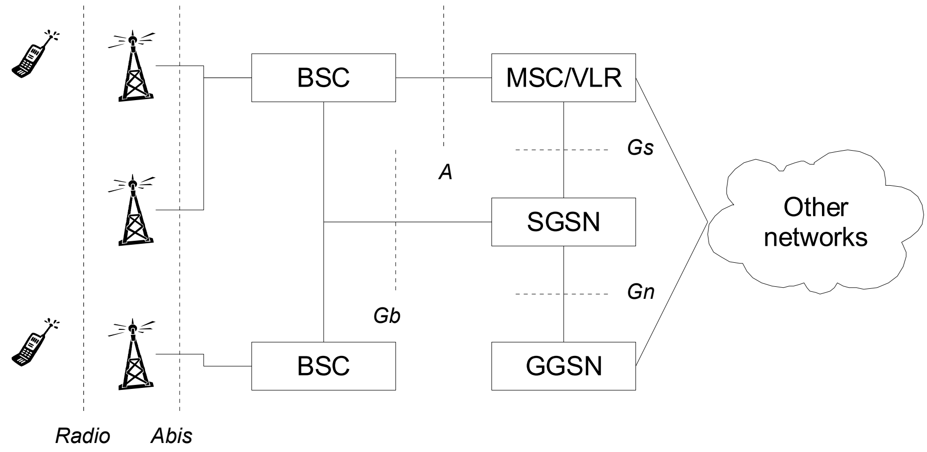

Figure 2.3. GSM Interfaces – Simplified

2.2.4 GSM Network Interfaces

All nodes in the GSM network are connected by means of standardized interfaces [15]. Some of these are used for signaling, others for payload, and some for both. The idea has been that different vendors' equipment can be used in different nodes, but still communicate without problems. This idea has not always worked well in practice though. In this work, focus is set on the signaling parts of the interfaces. See Figure 2.3 for a simplified illustration. The following GSM interfaces are of interest in this thesis work:

- The Abis interface, which connects the BTS with the BSC.
- The A interface, which connects the BSC with the MSC and is used for CS.
- The Gb interface, which connects the BSC/PCU with the SGSN and is used for PS.

2.2.5 GSM Traffic Cases

In cellular systems some typical traffic cases exist. In idle mode the MS is registered in the network but not in use. In active mode the MS is registered and in use. Some typical scenarios are explained in this chapter. For a complete listing and details explained, refer to [15].

Idle Mode

When in idle mode the phone is powered on, but not currently in a call [15]. Typical traffic cases in idle mode are IMSI attach, location area updating, changing cells within the same location area and IMSI detach.

Network Attach

When an MS is switched on, the IMSI attach procedure is executed [15]. This means that the MS sends an attach message to the network, indicating that it is now in idle mode. The VLR checks if the subscriber is known. If not, the subscriber's HLR is paged for a copy of the subscription information. The VLR then updates the MS's status to idle, and sends an acknowledgment to the MS.

Network Roaming

All cellular networks consist of individual cells, known as BTSs [15]. Each base station covers a small geographical area which is part of a uniquely identified LA (Location Area). Mobile subscribers may roam within LAs, between LAs, to a new serving BSC or even to a new MSC.

The MS's current location is stored in the VLR [15]. If a MS changes cells within an LA the network is not updated. If the MS detects that it is in a new LA it informs the network. Depending on if the new LA is served by the same BSC or MSC the location updating procedure is executed in different ways.

Network Detach

IMSI detach enables the MS to tell the network that it is powering off [15]. This will stop the network from paging the MS when it is no longer switched on. The MS is marked as switched off in the VLR. The HLR is not informed and no acknowledgment is sent to the MS.

If for some reason the detach message is not received, the MS may improperly be marked as attached [15]. The use of periodic registration and implicit detach will avoid this, since the MS will be implicitly determined detached when its periodic registration messages are not received by the VLR.

Active Mode

Active mode scenarios describe traffic cases such as setting up a call, disconnecting a call and moving between cells while in a call [15]. Active mode traffic cases are mainly handled by the BSC.

Call from MS

When initiating a call from an MS, the MS first requests a signaling channel, which is allocated by the BSC [15]. When the signaling channel is established, the MS sends a call set-up request to the MSC. The MS is marked as active in the VLR and the BSC is instructed to allocate a traffic channel. The dialed phone number is then forwarded to the appropriate exchange and the connection is established, provided that the recipient answers the call.

Call to MS

When calling an MS the exact location of the subscriber is unknown [15]. Because of this, the MS must first be located by means of paging before the call can be set up.

When an MS is dialed, for example from the PSTN, the first node that is reached in the recipient's network is the home GMSC [15]. The GMSC examines the MSISDN (Mobile Subscriber ISDN) number, that is the subscriber's phone number, to determine in which HLR the MS is registered. The HLR is queried in order to be able to route the call to the correct MSC. The HLR converts the MSISDN to an IMSI which is used for this purpose. The serving MSC knows which LA the MS currently resides in. A paging message is sent to the BSCs serving that LA. This paging message is forwarded to all BTSs in the area, which in turn transmit the message over the air interface. When the MS receives the paging message it requests a signaling channel. When this is set up, a traffic channel is requested. The mobile phone rings.

Handover

The process of changing cells while in a call is called handover [15]. Measurements are constantly made by the MS and BTS to ensure that the cell with the strongest reception is selected. The MS measures downlink signal strength both on the active cell and on neighboring cells with the intention of choosing the one with the best reception. The uplink measurements are made by the BTS. The measurements are sent to the BSC, which decides if a handover is necessary. Handovers are not only used when moving between cells, they can also be used for load balancing.

2.2.6 UMTS Network Structure

The UMTS network structure is quite similar to the one of GSM [38]. The concept of base stations and controllers has been adapted from GSM. The BTS is however called NodeB and the BSC successor is called RNC. A difference between the NodeB and the BTS is that the NodeB is capable of serving cells not transmitted from the same antenna site. The RNC is like the BSC connected to the MSC. Both GSM and UMTS radio systems can be connected to the same CN. The MS has also received a new name in the UMTS network, UE (User Equipment) , in order to reflect the extended use of other devices such as PC cards and USB (Universal Serial Bus) dongles. The UMTS network structure is presented in Figure 2.4.

2.2.7 Proposed LTE Network Structure

The standards for LTE are at the time of writing this report not established. However, a lot of work has been done, and ideas of how the nodes in the system will be set up are started to be synchronized through 3GPP.

As mentioned earlier, the approach until now has been to split the mobile telephone network into radio access networks and core networks. With LTE this approach will change [9]. The demands for lower cost and improved efficiency

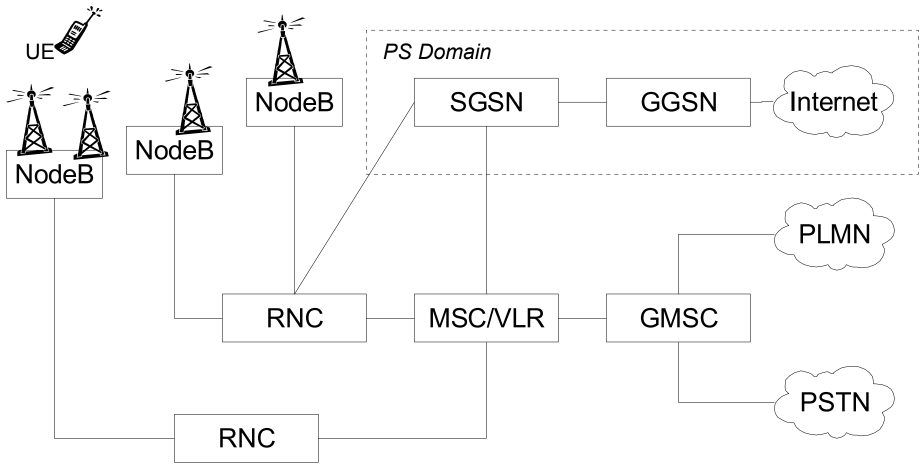

Figure 2.4. Simplified UMTS Network Structure

suggests a simpler architecture with fewer nodes and interfaces. The idea is to move time-critical functions from the RNC/BSC to the base stations while moving routing and internetworking functionality to a new single core network node called EPC (Evolved Packet Core). The new base station node, the eNodeB will handle handover decisions and scheduling of users in both uplink and downlink in its cells. Just like in the UMTS case, the cells served by an eNodeB do not necessarily have the same antenna site.

The structure of the new core network is a major evolution from the CN of GSM and has therefore been renamed to EPC [9]. From the start the EPC was supposed to consist of only one node. However the HSS (Home Subscriber Server), LTE's version of the HLR, has been kept outside the node. The proposed LTE network structure is presented in Figure 2.5.

2.2.8 Backwards Compatibility

Infrastructural investments imply high costs for network operators. Consequently it is important that several generations of mobile systems can coexist, and be upgraded. In the previous section, it was stated that GSM and UMTS can utilize the same core network. This is one example. It has also been important that old MSs can be used with newer network equipment.

The approach to LTE, however, has not focused on backward compatibility [9]. This despite the fact that the evolution is driven by 3GPP and mainly the same companies as the ones behind WCDMA and HSPA. The earlier protocols will be an important foundation for the LTE design since some features, such as the support for ISDN (Integrated Services Digital Network) services, will no longer be used. Instead new features will be developed. However, many WCDMA and HSPA features that are considered good will be kept. The idea is that WCDMA and

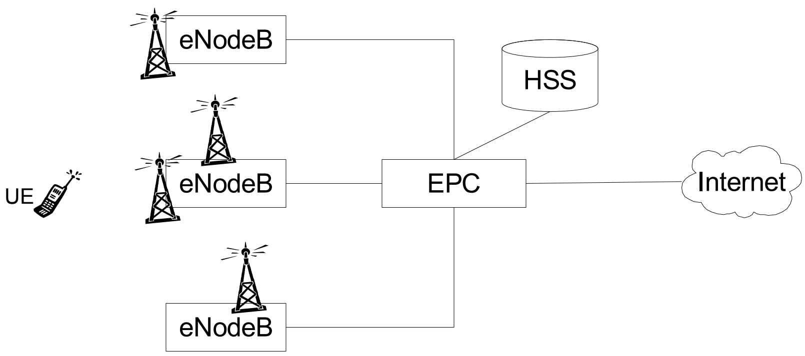

Figure 2.5. Proposed LTE Network Structure

LTE networks will exist in parallel, with new LTE cells introduced gradually where needed.

2.3 Interoperability Developmental Testing

Interoperability testing verifies that the tested application runs without faults in its live environment. The operation should not impact adversely on other systems and vice versa [40]. When developing communications systems at Ericsson, the interoperability developmental testing is an important step, the interaction between systems should work flawlessly. This section gives an introduction to how the IoDT environment is set up and how the work within IoDT is conducted.

2.3.1 Assignment

The GSM and UMTS cellular telephony systems are comprised of numerous standards and have since 1998 been controlled by 3GPP [3]. 3GPP is a collaboration between telecommunication associations around the world, among them ETSI (European Telecommunications Standards Institute) [3]. ETSI was one of the main actors in establishing the first GSM standards [38].

Even though different manufacturers of network and terminal equipment are compelled to follow the standards, problem-free interoperability between different vendors' equipment is not always evident [18]. In order to minimize issues with Ericsson's equipment the company has interoperability test laboratories available for their customers' use. The general idea is that the interoperability developmental testing will guarantee that new features in the Ericsson network equipment and other vendors' mobile platforms should inter-operate harmoniously, when released.

2.3.2 Test Environment

The test infrastructure at the Linköping plant consists of a majority of the nodes present in a GSM and UMTS network [17]. It also comprises a large number of MSs/UE, both placed in mobile racks[4] and standalone units. The nodes not available in Linköping can be accessed remotely at other test plants in Sweden if needed. The test network also has its own PLMN service provider, which can assign SIM cards to the HLR and so on. One difference between the test plant network and live networks is that all RF (Radio Frequency) connections are achieved by the use of coaxial cables instead of the air interface with antennas. This is done to minimize interference with adjacent equipment and to make sure that the test environment is sealed in the building.

The IoDT part of the test infrastructure is physically separated from the rest of the test plant. This is to enable the hosting of external customers, and to let them use the equipment without jeopardizing the company's confidentiality. The IoDT test area consists of three STPs (System Test Plants). Each STP has a patch panel which is connected to BTSs (which in turn are connected to BSCs and so on) in another part of the building. It is possible to patch through several cells, either GSM or UMTS to the STP. IoDT has its own dedicated BSS part of the network, but shares the core network with other users of the test plant. Most of the work with the CN, for example changing different parameters, can be done remotely. Some tasks though, require physical access. Examples of such tasks are when cables need to be moved in order to reroute connections [17].

The layout of the IoDT test plant is shown in Figure 2.6 [14]. As seen in the figure; the interfaces connecting different nodes can be monitored by protocol analyzers. This is not true for the *virtual* (cable implemented) air interface though [31]. Several different protocol analyzers are available. A description of the interfaces connecting different nodes can be found in section 2.2.4. Fading equipment is also installed in order to control signal attenuation, and thereby executing handover scenarios and other cell changing tasks. The fading equipment can be either a *blue box* with knobs that are turned in order to manage the attenuation on different cells, or a computer controlled CCN (Cellular Coaxial Network). A CCN has been installed in an IoDT STP during this thesis work, so that these tasks can be successfully automated. The CCN system also has an intuitive GUI (Graphical User Interface), if manual control is required. The CCN will be further described in chapter 3.3.1. In addition, signal generators can be used to add disruptive signals to the radio interface. Oscilloscopes are also available, and these can be used to analyze radio signals.

2.3.3 Current Tasks

Today Ericsson performs IoDT work with several mobile platform vendors [18]. The co-operation with EMP (Ericsson Mobile Platforms) is quite obvious, but also other partners, for example Nokia and NXP, participate in the IoDT testing. The partners normally visit Linköping a couple of days to perform testing of a specific

[4]Mobitec Controlled, see 3.3.1

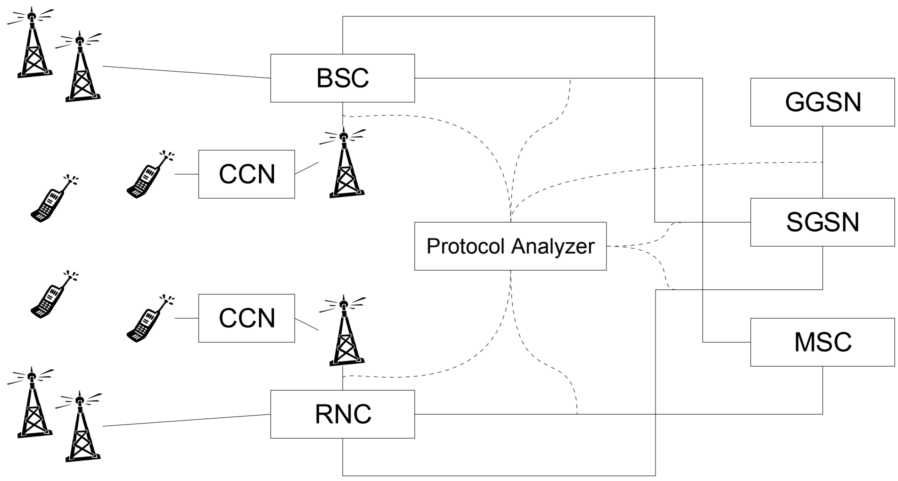

Figure 2.6. IoDT Lab Setup

feature. Usually they are in contact with their software developers back home and receive firmware updates to their products continuously during testing, in order to implement workarounds to problems directly during testing.

After feature testing has been done in IoDT, multi-vendor terminal verification is done elsewhere [18]. This is a larger scale testing activity. The testing identifies necessary changes so as to reach optimum configuration for a successful multi-vendor environment.

One important task for IoDT is to supply different BSS projects with terminals with requested functionality [18]. These can be found in co-operation with platform vendors. For example many phones include features that by default are disabled, but can be enabled in order to test new functionality. This is because the same mobile platform is used in many telephone models manufactured by different vendors. These activities will be further explained in chapter 3.4.1.

2.3.4 Future Tasks

Today the IoDT STPs are not solely utilized for IoDT tasks. Even though GSM IoDT testing is still of importance, as new features that require testing are released regularly, discussions with colleagues give that GSM testing is declining. When LTE testing is launched the actual interoperability developmental testing of terminals is anticipated to rise, since a lot of testing is required during the beginning of the new product development.

Today, the testing is only done during daytime when staff is present. This means that the equipment, which is leased by the hour at high rates, is not used during nights and weekends. One idea is that this equipment also can be used outside office hours, if automation is implemented. As will be discussed in chapter 3.4, today's IoDT testing is not feasible to conduct without supervision. What could be done, though, is to run long-duration tests that do not require user input. Examples may be load testing or stability testing. It must be discussed though, whether the testing can be applied to the IoDT mission, otherwise it should be performed in other departments of the company.

Chapter 3

Test Automation

Before deciding whether testing activities should be automated or not, there are some aspects that should be considered. When determined that automation is the road to take, it is important to approach it correctly. Some basic theories are presented in this chapter, among with a presentation of some previously automated tasks within Ericsson. A summary of tasks possible to automate within IoDT is also presented.

3.1 Automated Testing Background

Automated testing is the result achieved by automating the manual testing process currently in use. To make this possible, a formalized manual testing process is required. Such a process should at least include [41]:

1. Detailed test cases with predictable expected results.
2. A standalone test environment, including a test database, which is restorable to a known state, so that the test cases may be repeated each time modifications has been made.

If the testing before automation implementation only implies leaving the subject for test to a group of users or specialists that evaluate it in some ad hoc manner it is not subject for test automation [41]. The work must be structured. The real use of automated testing is in regression testing[1]. This means that a database of test cases must be developed. This suite of tests is then run every time a change has been made to the test object. This to ensure that it does not produce unexpected results after modification.

An automated test script is a program [41]. Test developers should follow the same rules and standards that apply to software development. Because of this, it is advantageous if the test writer is also a programmer, or even better, a technically skilled person with knowledge in both testing and programming.

[1]Testing a modified program in order to ensure that new bugs have not been introduced in functionality that previously worked as desired.

3.1.1 Test Cases

Test cases consist of three parts: inputs, outputs and order of execution [7]. Inputs may be data entered on a keyboard, but also data received from interfacing systems and devices. Data read from files or databases are also considered to be inputs. So is the system state when data arrives, and the environment within which the system executes.

The obvious output is data displayed on a computer screen [7]. Output data can also be sent to interfacing systems and external devices. Data can be written to files or databases, and the system state and environment may be modified by the system's execution.

According to [7], there are several types of programs, processes and data that provide the test designer with the expected output and result of a test. These sources of expected results are called oracles. Some oracles are presented below:

- Kiddie oracles – just execute the program and observe how it behaves. If it looks right, it must be right.
- Regression test suites – run the program and compare the output to previously recorded test results achieved when testing an earlier version of the program.
- Validated data – run the program and compare the results with a standard such as a table, formula or another defined type of valid output.
- Purchased test suites – run the program against a standardized test suite which has been previously created and validated.
- Existing program – run the program and compare the output to an earlier, working version of the program.

Regarding order of execution there are two different approaches according to [7], cascading test cases and independent test cases:

- Cascading test cases – test cases can build on each other. First testing of one part or feature of the tested system is executed. The system is then left in a state so that the subsequent test case can be executed. The advantage of cascading test cases is that each test case typically is less complex. The disadvantage is that if one test case fails, the following ones may be invalid.
- Independent test cases – each test case is completely stand-alone. The tests do not build on each other and do not require the successful execution of a previous test. The advantage is that any number of tests can be executed in any order. The disadvantage is that the tests tend to be complex and difficult to design, create and maintain.

3.1.2 Types of Testing

Testing is divided into two types; black box testing and white box testing [7]. In black box testing, the test cases are based on the requirements and specifications only. The tester takes an external perspective of the test object to derive test cases. Valid and invalid inputs are determined as well as the correct output. With more extensive systems, the different parts grow bigger and more complex and therefore opens up for the use of black box testing to simplify. Black box testing requires no knowledge of the internal paths, structure or implementation of the tested software.

White box testing uses an internal perspective of the system to design test cases based on internal structure [7]. As white box testing is based on internal paths and implementation of the tested system, it requires deeper programming skills.

3.1.3 Testing Levels

According to [7], testing is typically performed on four different levels: unit testing, integration testing, system testing and acceptance testing. These levels are not applicable for all applications. They assume that the time between developing units and system integration is significant.

Unit Test

A unit is the smallest piece of software that a programmer develops [7]. The unit is typically the work of one developer and is stored in a single file. Unit testing is the practice of validating that this individual unit of source code is working properly. Unit testing is typically done by developers and not by software testers or end-users.

Integration Test

In the integration testing, units are assembled into subsystems and finally into complete systems and tested together [7]. Units may function well when isolated but failure might be the result when parts are put together to form larger systems. The purpose of integration testing is to validate that each interface that connects different units functions as expected.

System Test

The system comprises all software that make up the product as delivered to the customer [7]. It may also include hardware. The system testing activity aims to find errors that occur at the highest level of integration. Typically the system test includes many different types of tests. These could be functionality, usability, security, reliability and so on.

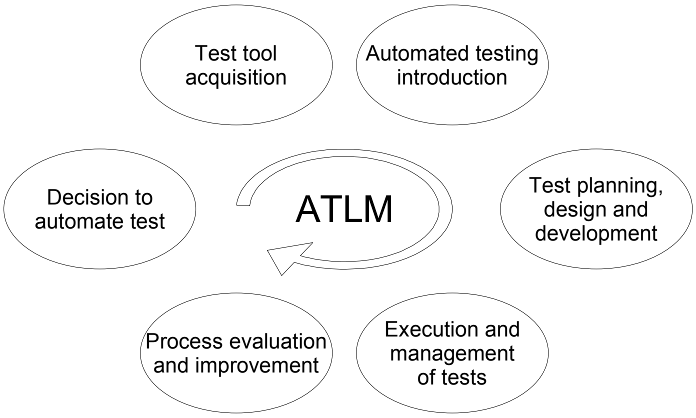

Figure 3.1. Automated Test Life-Cycle Methodology

Acceptance Test

Acceptance testing is a practice that aims to evaluate whether the product will be accepted by the potential customers [7]. It is in the customers' interest that exhaustive acceptance testing is performed, while the vendor would like a minimal acceptance testing effort, that still enables the product to be exchanged for money.

3.2 Automated Software Test Framework

When the decision to automate testing is taken, the decision makers may not be aware of what the introduction of a test tool in a project implies [11]. By using a systematic approach it is possible to arrange and execute testing and related activities in a manner that maximizes test coverage within the resource limits. According to ATLM (Automated Test Life-Cycle Methodology), as presented in [11], the steps in Figure 3.1 should be included when deciding on and implementing automated testing. ATLM is a structured methodology that aims to ensure successful implementation of automated software testing. The methodology is similar to the one of application development, where the user is engaged in the development cycle at an early stage. An incremental fashion is used, where the end-user is involved throughout the analysis, design development and testing of each build version of the program. The steps of ATLM will be briefly introduced in the following subsections.

Decision

The first step is to decide whether to automate or not – an ever so important decision [11]. In order to make an accurate decision, some aspects need to be taken into account. First, false expectations of the automation need to be overcome.

Some deceptive expectations on test automation are common. The following list presents some facts on these:

- Test plans are not automatically generated.
- No single test tool will fit all applications.
- Test efforts will not immediately be reduced.
- Test schedule will not necessarily be reduced.
- Test tools are not necessarily easy to use.
- There is no such thing as 100 % automated test coverage.

Despite these caveats, automated testing may provide several benefits when correctly implemented [11]. Given the prerequisites, the test engineer must evaluate whether the given benefits fit the required improvements, and if the automation provides a logical fit in the organizations needs. Significant benefits that might be reached if the criteria are fulfilled are:

- Production of a reliable system.
- Improvement of the quality of the test effort.
- Reduction of test effort by minimizing the test schedule, test time that is.

In order to realize test automation, one of the first tasks is to acquire management support [11]. In order to get this, the management's understanding of the application needing automated testing may need to be adjusted. For example the return of investment might need to be argued for, if the test tools imply investment costs.

Acquisition

In an ideal situation a test tool is selected that fits the organization's system engineering environment at the early stages of the system's development cycle [11]. In reality, this is often accomplished after the projects have a detailed system design in place.

The test tool selection process may be time consuming, and support from management is required [11]. Therefore a detailed proposal should be presented, to convince management of the need of a test tool. When approval is granted, a methodical approach is needed in order to select the best tool. The organization's system engineering environment must be reviewed. Based on the review criteria, a tool evaluation domain should be defined. Available test tools then need to be investigated, and compared. This may be accomplished by scoring of the competing tools, and their features. If no single tool has all the features required, several tools may be required. When one or more tools have been selected, a hands-on tool evaluation should be conducted in order to determine which tool (or tools) to use in a pilot project.

Introduction

The test tool introduction phase includes the steps necessary to introduce automated testing to a new project [11]. Test goals, objectives and strategies need to be defined and test processes need to be documented, and communicated to all the users. The test tool introduction to its future users is nearly as important as the tool selection, since new technologies usually are met by skepticism by their users. Major unplanned adjustments can be avoided by developing and following a strategy for the system roll-out. Use of a strategy also boosts productivity from the start, since it is more likely that the users will use the equipment correctly.

The test tool consideration phase investigates whether company generic test tools can be used in the current project or if new tools need to be obtained to achieve the desired functionality [11].

Test Planning, Design and Development

This phase consists of planning and preparation of the automated testing, analysis and design of test cases, and development of these [11].

The test planning part of ATLM includes all activities required in the test program [11]. It should ensure that testing processes, methodologies, people, tools, schedule and equipment are well organized and work together efficiently. The test plan should include both definition of test requirements, as well as a suggested approach to how the requirements should be met. Important elements of the test planning include milestone planning, test program activities and documentation.

The test environment setup is also a part of the test planning [11]. It involves planning, tracking and management of the the setup activities. There is probably a need to purchase new testing equipment, and lead times may be long. The test team needs to schedule environment setup activities such as hardware installation, software and network configuration, integration of, and installation of test environment resources.

When developing software, requirements must first be set up [11]. The same needs to be done when developing test designs. They should be clearly defined and documented, so that everyone that uses them also understands them. A test requirement analysis will define the requirement statements. As tests should be reusable, repeatable and maintainable, test development standards should be followed. The test design addresses the need to define the number of tests to be performed, the way of testing, and the test conditions that need to be met.

The test development architecture provides a framework for test development preparation activities necessary to create test procedures [11]. It shows major steps that need to be taken in order to develop the test cases. Before the development starts, existing test procedures should be analyzed in order to determine if existing test cases within the automation infrastructure can be reused.

In order to be able to run the tests the testing environment needs to be sufficient [11]. Hardware and software should be dimensioned to meet the needs of the testing tools. The physical environment must also be set for the testing activities. The physical environment includes elements such as premises for testing and infrastructural assets such as furniture. Security must also be addressed.

Execution

At this stage, when test design and development has been completed, it is time for test execution [11]. With the test plan finished and test environment operational and consistent with the guidelines, as defined during test design, it is time to execute the tests defined in the test program.

During execution the test team needs to follow a test procedure execution schedule, which is defined in the test plan [11]. Plans for unit, integration, system and user acceptance testing are executed, and together make up the required testing activity for the system as a whole.

Key indicators of the test coverage, progress and quality are provided to the test manager by means of metrics [11]. During black box testing these metrics focus on the breadth of testing, including the amount of demonstrated functionality and the amount of testing that has been performed. During white box testing the depth of testing is measured, by collecting data relative to path coverage and test coverage.

Review and Assessment

In order to enable improvements of the automated test activities, review and assessment need to be done throughout the testing life cycle [11]. Following the test execution, review of the test performance should be performed in order to determine where improvements can be implemented, so that performance can be enhanced in future testing. This is the final phase of the ATLM model.

Throughout the test program, various test metrics are collected [11]. The focus of the test program is to determine if the tested application satisfies the test criteria and is ready for production.

In addition to the evaluation of the test program, the effectiveness of the process should be studied [11]. The return of investment should be investigated. Was the implementation of automated testing really beneficial? To support this assessment the automation benefits should be measured throughout the test life-cycle.

3.3 Test Automation at Ericsson

Throughout Ericsson it has been requested that test tasks should be automated in order to make testing more efficient. At the Linköping site this has been carried out in several different areas [19]. There is a special group, the ATD (Automated Test Design) team, which is responsible for test automation. The ATD team manages the so called ATE (Automated Test Environment), based on the THC framework. THC is an Ericsson product that can execute test cases automatically by accessing different software or hardware resources, while logging the results during test case execution.

3.3.1 Previously Automated Tasks

Several *resource factories* have been developed by ATD in order to to facilitate the use of THC to control different nodes and tools in the network [19]. Resource factories are interfaces that connect tools and applications with THC. They are developed by the ATD team in order to promote the use of THC within different areas of the company. A resource factory can be described as an adapter between a test tool (or a system) that provides resources, and test cases within THC. All tool specific adaptations are implemented in the factory in order to be able to communicate with the rest of THC through standard interfaces. Hence a factory acts, in most cases, as a proxy for a number of resources. Some of the resource factories will be presented in the following subsections. Focus will be laid on ones that are adequate to use within the requested IoDT test automation. The factories listed control the software supplied with different products, which in turn controls the hardware in their respective systems.

Mobitec

Mobitec is a tool that controls mobile telephones from a PC [28]. Mobitec uses AT (Attention, Hayes Command Set) commands [1] to query the USB connected telephone. AT is a standardized command set for querying modems, and most manufacturers have adapted to it. Despite this, some manufacturers use their own AT commands in addition to the standard set. However, Mobitec has the ability to send custom AT commands. The efficiency of the testing might rely on the availability of certain commands, such as reset and power off/on commands. The telephone is installed as a modem and a dial-up connection is configured. It is possible to either send simple AT commands to the MS or to utilize XML (eXtensible Markup Language) scripts to execute several commands sequentially. Mobitec can be run independently but can also be controlled by THC. There is a Mobitec resource factory available in ATE. A study of current features and features that possibly will be required in the future can be found in [28]. Mobitec is mainly used to generate traffic with several mobile phones, typically placed in a rack.

Cellular Coaxial Network

One of the cornerstones of a cellular network is, as earlier mentioned, the use of several base stations, which form cells that users can move between while in a call. There are of course networks available for testing this in a natural environment with BTSs. However, the simulation environment CCN is much more convenient to use. It consists of a box with controllable fading equipment. Depending on model, it can can handle up to eight BTSs and eight MS groups, and simulate movements of the MS groups to trigger handovers[2] [21]. The CCN equipment may be controlled remotely via a GUI [10] or through THC.

[2]Described in 2.2.5

MML Command Control

In order to communicate with a BSC, MML (Man-Machine Language) commands are used [15]. There are more than 5,000 commands with several options and switches that can be entered in sequence to achieve different results. One way of accomplishing this is to put several commands in a file and run them as a batch job. Another way is to use an application like WinFIOL [13]. WinFIOL is a family of tools specially designed to enable users to tap into Ericsson products, such as AXE switches and BSCs to unleash their full capabilities. THC features a factory that can execute MML commands in order to change BSC parameters during test execution.

Nethawk Protocol Analyzer

As described in chapter 2.2.5 different interfaces exist between different network nodes. In order to analyze data sent over these interfaces, a commercial tool called Nethawk [31] is used. Nethawk is a PC based application that connects to different GSM interfaces in order to analyze the signaling and data throughput. See Figure 2.3 for information on GSM interfaces. Depending on which information is requested it is possible to analyze both signaling and traffic channels. If unencrypted, it is possible to read the transferred data, and protocol messages.

3.3.2 Test Support

Beside the resource factories supplied in ATE, the automated test environment also feature several tools to ease the use for programmers and testers. These include both in-house developed tools, open source tools, and commercial tools. Some components of the environment are introduced below.

Code Development Framework

When developing test scenarios for THC, Java is the programming language used. Eclipse [39], an integrated development environment for Java, is used to simplify both writing and compiling the code. There are many test case developers. To simplify the work, code can be stored in a common library readily accessible for anyone using the system. This provides users with both inspiration and help to solve problems. It also enables users to reuse already written code, rather than starting from scratch every time.

Revision Control

In order to keep files secure, a revisioning system called ClearCase [27] is used. ClearCase keeps track of changes made and maintains a complete database of old file revisions so that rollback can be executed if problems occur. ClearCase is connected to the code library, and also prevents users from editing the same files simultaneously by mistake.

Tester GUI

A GUI is used when the test cases are to be executed. Within the GUI, resources are reserved and test cases are chosen. If the test cases require user input, as stated in the code, the input is applied as test properties in the GUI. Furthermore, features like test scheduling are available. During test execution information is printed out to aid the tester in analyzing problems and following the test execution details. The program can also send an e-mail returning the test verdict to the tester when finished.

3.4 IoDT Test Automation

In order to decide whether automation is feasible within IoDT, the ATLM framework, as presented in 3.2 has been studied. When the thesis work was started the task was quite clear: to find appropriate tasks to automate within IoDT and preferably implement one of these. Therefore, focus will be put on some of the items in the framework. Since the automation decision was already taken, this thesis will aim to choose an appropriate task to automate instead. The test tool to use was also already decided, since it has been used earlier at Ericsson, and accordingly it is well supported.

The approach to the ATLM framework in [11], mainly focuses on the testing of software during development. In IoDT software is tested in conjunction with hardware. The network equipment hardware is upgraded now and then, and the installed software is updated, or rolled back regularly since different releases are needed for different tests. Customers' MSs that are tested are often prototypes, both in hardware and software. Problems that arise during test can consequently be either software or hardware oriented, and they can lie either in Ericsson's network equipment or in the terminals connected to the network for interoperability test.

3.4.1 Automation Scope

The dilemma with IoDT testing activities, according to some testers, is that the same tests seldom are run during longer periods, or need to be rerun many times. This, at least, is true for the current stage of GSM development. The standard is still developed with new releases and features, but these features are not tested to an extent that would gain from test automation. Despite this, annoying work tasks that are done repetitively and therefore are subject to automation have come up when discussing IoDT test automation. Note that some of the activities below cannot be used in actual testing, but rather as supportive functions that need to be done, and require a great deal of time to perform. The ideas below, for activities possible to automate, have arisen when discussing the topic with IoDT team members.

Also observe that even though the scenarios below are all suggested by IoDT team members, they are not necessarily within the IoDT scope. However, they are appropriate to run in a laboratory environment similar to the one in IoDT.

As automation is the subject of this report, and appropriate IoDT automation tasks have not been easy to find, some non-IoDT suggestions will also be included. Extended non-IoDT tests could possibly be run during non-office hours, when the labs are not used for IoDT work.

Scenario A – Feature Testing

An obvious task that would be desirable to automate is the daily basis testing activities performed in co-operation with IoDT partners. The testing scope basically includes features that have been decided in advance. This means that test cases could be written beforehand. However, since the tests performed differ a lot from one time to another, the reuse of code and test cases may not be possible. Also, since the tests are done in co-operation with the platform vendors, discussion is important during tests and new testing ideas may emerge. Predefining of tests is also difficult since developmental products may not be controllable by IoDT systems and software, and hence need to be manually controlled.

Even though platforms and MSs might be officially released, they are not necessarily standardized by means of, for example, computer communication. A universal tool that could adapt different means of communication would be desirable, but is deemed difficult to implement. This since it would need to be expanded for new products continuously.

Scenario B – Network Verification After BSC Software Upgrade

When performing different feature tests in IoDT, the software version in the BSC often needs to be changed. The BSC software is developed in-house, and non-released software versions are usually used within IoDT for testing of new features. Software versions are typically changed every time a new feature is to be tested in the lab, which means that it may be changed at up to a weekly basis. The upload of a different software version to the BSC is initiated with ease and needs no further supervision. However, after the upload is completed, a test of basic features needs to be done in the lab. This in order to make sure that that the system is up and running with basic functionality.

The network verification procedure is quite time-consuming today since the process needs to be done manually. In general, the equipment works if the software has been used before. Some parameters might have been messed up though, and need to be set. After this the procedure has to be done again. This in order to exclude basic functionality as a factor for errors, when testing new functionality.

The steps included are among others to find an appropriate MS and SIM-card and verify basic radio coverage. Some basic functions that require verification are GSM circuit switched voice traffic and packet switched data traffic. Handover scenarios also need to be tested. The test procedure includes plenty of log analyzing.

Scenario C – Maintenance of MS/UE Database

One of the tasks of IoDT is to supply different BSS projects with MSs/UE equipped with requested functionality. Usually the functions asked for are not included in the phone specifications, and hence the phones need to be tested to verify compliance. IoDT keeps a database of phones of different make and model, with detailed specifications.

The work of extracting this information is today done manually. To accomplish this, requested features are activated in the BSC and MS/UE, simulations are made and logging performed on signaling interfaces. Log analysis shows if the features are available in the MS/UE. This information is entered in the MS/UE database for future reference.

Scenario D – Log Extraction and Analysis

When performing tests logs are almost always taken. Usually only logs on the signaling interfaces are of interest. Even though logs are only taken on signaling interfaces and not on traffic interfaces, the log size grows big very fast. The analysis of the produced logs can be lengthy, even if the printouts sought for are known in advance. The logs produced by the primarily used log tool Nethawk, are extracted in a binary format that is difficult to resolve if not using the supplied Nethawk software.

It would be desirable to, in an automated manner, extract only the information needed for specified purposes from the logs created by Nethawk. An automatically generated graphical representation would be appropriate in some cases. The analysis of logs would be much simplified if only the necessary data would be presented to the tester.

There are already some in-house tools that can accomplish this, but these are not remotely controllable by THC. An adequate future THC development task could be to build factories for these tools.

Scenario E – Check Whether MS/UE Fulfill Basic Requirements

When customers come to the IoDT lab with prototype products to test new functionality it is important that the terminals fulfill basic requirements. Depending on the development stage it is not obvious that they do. With new technology though, as with for example LTE testing, the prototypes may be unstable as they are in an early stage of development. IoDT would in this case preferably need to test the terminal for basic functionality, before the customer arrives for expensive and time-consuming testing with a product that is not yet ready for testing. This testing would need to be done many times with different vendors.

Scenario F – MS/UE Supported by Network

Consider a network operator that plans to launch a new feature in its network, for example adaptive multi-rate wideband, for enhanced speech quality or packet switched handover, that enables PS handovers even when using time-critical applications. They will want to certify that some specified phones are compatible with the network. Maybe they will even sell these terminals, as branded phones to a discounted retail price. If the network operator has a lot of Ericsson infrastructure it would be appropriate to test the different terminals at Ericsson premises with available logging tools before launch. Comparisons and benchmarking of several different products could also be performed.

Scenario G – Which MS/UE to Choose

This scenario is similar to the previous one. A network operator might want to purchase a large number of USB modems, for example, to sell with data subscriptions to a subsidized price. In order to choose the best product several KPI (Key Performance Indicator) tests could be made. The operator, of course, is interested in the product with the best value. The decision needs to be made after analyzing the hardware requirements. To enable long term tests of several products automation would be appropriate.

Scenario H – Stability and Performance

Stability and performance tests could also be run in the IoDT lab. This could preferably be done when the office is unmanned, and therefore would require some kind of automation.

The stability and performance of the cellular network is tested elsewhere, by using racks with multiple phones simulating traffic. However, the stability and performance of MSs could be tested in IoDT-like facilities. However this is also a task that is outside IoDT's main mission.

Chapter 4

Automation Implementation

This chapter will explain which IoDT task among the ones presented in chapter 3.4.1, that has been selected for automation and why. The method and steps of implementation will also be described.

The ATLM framework, as presented in 3.2, will in some extent be applied to the automation implementation process. Even though ATLM's main target application is purely automated software development testing, an adaption to the current testing implementation will be attempted. The headers will mainly be used as springboards for unfolding and discussion of the different topics. The procedure will be described in the following steps, that reflect the steps of the model.

4.1 Decision

In the ATLM framework, the decision item argues whether a chosen task should be automated or not. In this work, however, the automation decision was already taken when the project was launched. The approach has instead been to find an appropriate task to automate – among many. The major goal has been to minimize the work effort and the test schedule of the chosen task.

Among the suggested tasks in chapter 3.4.1, several are rather complex to implement, even though they appear appropriate to automate at first sight. Also, some of the tasks are not really IoDT's business at all. Table 4.1 summarizes the tasks presented and divides them into three groups: tasks that are *Core IoDT*, tasks that are *IoDT Supportive Functions*, and tasks that are *Non-IoDT*, that is, outside of IoDT's scope. The selection and grouping has been based both on the task automation ability, its importance in daily work, and its ability to relieve pressure from daily IoDT tasks.

Feature Testing, the task that mainly reflects daily basis work in IoDT is today not feasible to automate, since the performed tests differ a lot from one time to another. Furthermore, as many of the phones tested are just prototypes, manual debugging and troubleshooting is often required. This implies that the desired unsupervised execution of tests would be an utopia.

Core IoDT	IoDT Support	Non-IoDT
Feature Testing	Network Verification	MS/UE Supported by Network
	Maintenance of MS/UE Database	Which MS/UE to Choose
	Log Extraction and Analysis	Stability and Performance
	Check Whether MS/UE Fulfill Basic Requirements	

Table 4.1. Automation Tasks

IoDT supportive tasks, as for example *Maintenance of MS/UE Database* and *Check Whether MS/UE Fulfill Basic Requirements* are complicated to automate, as different handsets not necessarily are compatible and therefore cannot be controlled in the same manner. This means, among other things, that new software would need to be installed and configured for every single new phone model used in the tests.

Log Extraction and Analysis is a task that would be appropriate to work with. However, it has been set aside at this stage, to leave room for automation more closely connected to the use of MSs/UE. A script for log information extraction will be used in this thesis work however. This script could provide a start for future work within the area.

The non-IoDT tasks *MS/UE Supported by Network*, *Which MS/UE to Choose* and *Stability and Performance* are of interest, even though problems previously described – mainly incompatibility issues – apply to these as well. The main reason why these suggestions have been ignored is however that they are outside of IoDT's scope.

Task Selection Verdict

Network Verification After BSC Software Upgrade has been chosen for initial automation. The main reason why this was selected is that the same hardware, such as terminals and logging tools, can be used for the required tests, even over time. In order to complete the tests we need to control several different hardware nodes, like MSs, BSCs, protocol analyzers and phone movement simulators. When the test cases have been prepared, it should be possible to run them without major modification, provided that the same hardware and SIM cards are used.

It will most likely be possible to decrease the test time schedule. Even though the tests may need to be run many times, the startup time will be greatly improved, since many problems will be avoided. This is mainly because the automated test introduction will also imply standardization of, for example, MSs used for the verification, and universally preprogrammed configuration files for different test tools.

Management support is already acquired. This thesis project has been initialized by the manager, and different solutions have been discussed during the project startup and progress.

4.2 Acquisition

The test tool selection process has been straightforward. As mentioned in chapter 1.5, one of the requirements for this thesis work has been to use the in-house test platform THC. A quick analysis of the tool and connected platform has shown that this is the best way to proceed. This is because application specific tools are already developed. Support is also available at an arms reach, and functionality may be added without additional costs.

Since it was already decided which test tool to use before the project started the tool evaluation and selection process work effort has been minimal. The major difficulty has been to adapt the different parts to the project, in order to make them work in concert.

4.3 Introduction

The methodology that has been adopted while deciding how to implement the automated testing has been to study the way of working before automation. Focus has been laid on surveying the steps that have been repeatedly done. During the introduction of the automation project, dialog has been maintained with the team members currently using the system. The project has been met with some skepticism, considering new tools that are claimed to be more complicated to use, when working outside the automation scope in the lab. These problems have been solved by keeping the possibility to use the affected legacy tools in parallel with new tools. As mentioned in the previous section, it is not necessary to purchase new testing tools, since what is needed is already available in-house.

4.4 Test Planning, Design and Development

This item discusses the planning phase including design, development and technical requirements of the automation project.

The network verification test program can closest be compared to a *Validated Data* test suite, since the logs taken are compared to previously defined results. If the test equipment does not fail, the test verdict is set by extracting and analyzing appropriate messages from the logs taken during test execution.

The test program is categorized as a *Black Box* testing suite. No deeper understanding of functionality is taken into account. The test cases are rather based on specifications of the mobile network and predefined transfer methods. Therefore, not much consideration has been taken into account regarding different network nodes. Furthermore, the testing is classified as *Integration Testing*. This is because the BSC software is tested in conjunction with the other network nodes, so that the functionality of the complete network can be verified.

Test Planning

During the test planning, the test schedule has been considered. While trying to minimize the time to run the tests, the main goal has rather been to minimize user interaction. So, it has been of importance that the test suite can be started, and then be left without supervision until it has finished or a failure has been encountered. In the general case, the whole test will execute without interruption. The timespan may be 10–20 minutes. An error may be encountered early, and then the test will be halted. If the tester wants to know the verdict directly when finished, the e-mail service can be utilized. If used, the THC server will send an e-mail to the tester with the test verdict when finished, or when an error has been encountered.

Some primary requirements have been set up for the network verification procedure. These conditions have been kept in mind when planning for, and developing the test program.

1. The test cases should be reusable over time.
2. The test cases should be able to run without user interaction.
3. The test logs should clearly explain errors encountered.

Test Design

In order to be able to run either partial tests or the entire test suite, a split-up approach to the presented task has been chosen. The test structure is presented in Figure 4.1. As seen in the figure, the test consists of three parts. First, idle mode tests are performed. Secondly tests with circuit switched traffic, and finally tests with packet switched traffic. Thus, a *Cascading Test Scheme* has been used. The test suite is supposed to be halted if errors are encountered in any part of the testing, in order to save time and enable further testing directly.

The tests can easily be run independently, but when the implementation is finished they are supposed to be run sequentially. Each test has its inner verdict. If the test sequence is interrupted by an error, the execution should be stopped and the logs will give a clue of what caused the problem. The following sections explain the different parts of the network verification testing. See Appendix B for a more detailed test specification.

As explained in 2.2.5, a MS is in idle mode when it is powered on, but not currently in a call. To make sure that the MS is behaving correctly in idle mode, and to analyze the network attach procedure, the MS is first rebooted. If it connects correctly to a cell within coverage, the next step is to move it to another cell and check that it is transferred correctly. The movement is accomplished with support from the CCN.

As the testing suite utilizes a cascading testing scheme, the circuit switched traffic test case is to be executed when it has been verified that idle mode operation is working. The circuit switched testing includes automated set up of voice calls between cellular phones and handover scenarios while in calls.

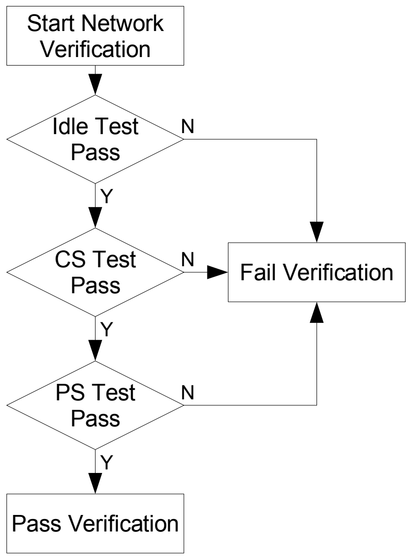

Figure 4.1. Test Execution Schedule

Packet switched traffic test scenarios are executed after the CS tests. First, a PDP context is attempted. The PDP context connects the phone to the GPRS network in order to acquire an IP address and prepare for data transfers. The PS testing also includes the upload and download of different size dummy files to and from an FTP server. Handovers are performed during download of large files.

The testing system will have several means of input. For one, correct data for phone numbers and SIM cards used need to be entered for the system to work. The log analyzer also needs input so that the correct configuration is used. This is accomplished by configuration files, and these are not supposed to be changed between testing sessions. This applies to several other inputs as well, for example configuration of the CCN and MS dial-up connections.

The main test output is presented in the ATE GUI, which also creates extensive log files for deeper analysis. However, each subsystem (Mobitec, CCN, Nethawk etc.) also produce their own log files. These can be useful if problems arise, and if the ATE GUI logs do not provide sufficient information to locate the problems. The Nethawk logs are queried so that a test verdict can be set. This is accomplished by using a Python script that extracts the necessary information, which then is compared with the desired output.

Test Development

During the development stage, existing test programs have been analyzed in order to make reuse of already existing code possible. As described in chapter 3.3.1, the

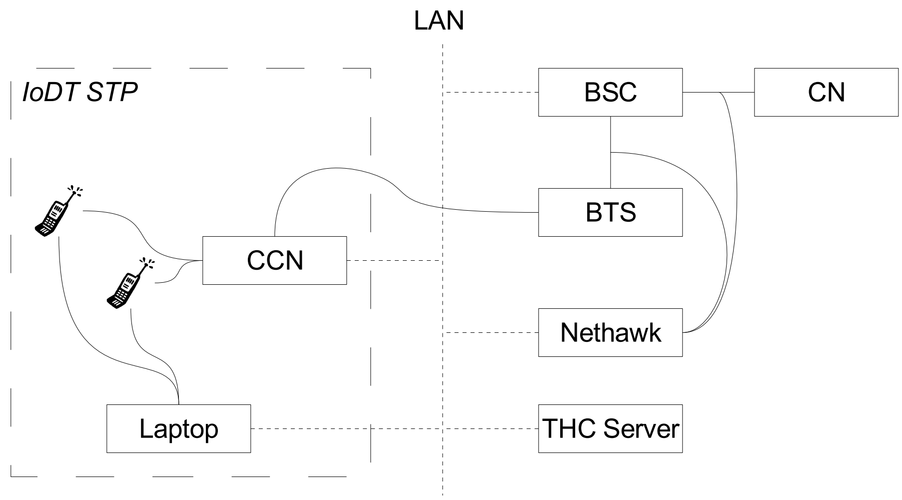

Figure 4.2. Network Verification Test Setup

required tools are already part of the automation framework. They have all been implemented in different automation projects previously. However, synchronization of testing tools, as in the case with the network verification, has not been found in the existing code data base, so a major part of the implementation has been to enable the tools to work in concert.

Technical Environment

The network verification procedure requires several hardware elements. A simple diagram of the setup is presented in Figure 4.2, and the elements needed are described in the list below. The system under test is mainly the BSC and particularly its software. Other network elements may also impact on the end functionality, but the purpose of this test tool is to verify that the network operates as requested, after the software in the BSC has been changed.

- MS – Two cellular phones are needed for the testing. One phone is enough for the PS tests as files are uploaded and downloaded to and from an FTP server. However, for SC calls two phones are needed. Landline phones are not controllable by Mobitec. The idea is to set two phones aside for this testing, since change of phone models and IMEI numbers require configuration changes which complicate the handling.
- PC – The tester's PC is the hub of the network verification infrastructure. The mobile phones used are connected to and controlled by this computer. It may also be used to remotely access the other systems needed for the testing. The Nethawk server is accessed via a Citrix interface and ATE features are accessed via Unix remote login.

- CCN – The CCN network will provide the testing with cell-to-cell movement features. A CCN has been installed in IoDTs STP and it consists of a hardware box with attenuators that control signal strength on different cells. The box is connected to a CCN server, which is accessed by the CCN THC factory.

- Nethawk – A Nethawk server is required for interface logging. The network verification process relies on the ability to take logs on interfaces, and by log extraction determining the correct functionality of the network. The Nethawk server is already installed in the lab environment. It consists of the server and probes for logging on the connecting node interfaces.

- THC Server – The THC server is already available in the lab environment and is the coordination point for the automated testing. In the preferred case, the THC server does its job without further user interaction than the one provided by ATE GUI.

- BTS – Two GSM BTSs, which are already available in the lab environment, are needed for the dump verification procedure. This is to enable phone movements and handover triggering. The frequency of the base stations should preferably be 900 MHz or 1800 MHz, but other standards could also be used, provided that they are supported by the MSs used.

- BSC – A BSC is also needed. Communication with the BSC may provide important information such as what kind of traffic is present in different cells. IoDT has a dedicated BSC.

- Core Network components – Core network components, such as an MSC and an SGSN, are also needed. These are available as shared resources at the test plant.

- Test Premises – One of the IoDT test plants will in the beginning be used for the network verification process. Today, mainly this STP is in use. The idea is that the network verification should be possible to run in other lab environments as well. If the project outcome is successful it opens up for investments in for example CCN equipment for the other IoDT laboratories.

4.5 Execution and Management

The test execution item in the ATLM model includes plans for unit testing, integration testing, system testing and acceptance testing as mentioned in chapter 3.2. This work, however, is focused on integration testing of the BSC and its software in conjunction with other network nodes.

The width of the testing is considered fair, and in accordance to the requirements. Even though the test suite could be extended to test more and deeper functionality, the purpose of this project has been to verify basic call and data handling. This testing is achieved with adequate results.

4.6 Review and Assessment

Most of the desired testing functionality has been possible to implement. However, some problems have been encountered during the work. The one problem that has yet not been solved is the telephone reset command. It is possible to send power on and power off AT commands from the PC to the phone. However, when the power off command has been sent, the USB connection is lost and it is not possible to power up the phone again. There is also an AT reset command which has been possible to use. This command does not appropriately reset the GPRS functionality, so after the AT reset command has been executed, it is not possible to attain a PDP context.

This is a known problem within MS/UE automation and a solution is being worked on. Since a phone cannot be run without its battery it is not sufficient to just connect it to mains power and break the power to power off. Instead a solution is being developed that will switch the mobile off and on in a regular manner, just by pushing the power button. Another solution to this problem could be to test other mobile brands and models, since they do not necessarily react the same to the reset and power on/off AT commands. Meantime, the approach used in this project is to query the tester to manually restart the phone when this step is reached in the test execution.

This will reduce the overall effectiveness of the test suite. However, the one maneuver that needs to be done manually is easy to accomplish. Except for this problem, the test suite is able to run from start to finish, provided that the test verdict is successful.

Chapter 5

Experimental Results

This chapter will present the results gained from the thesis work. The outcome of the testing should be evaluated in order to open up for future work within the area. Also, based on the implementation, the future of automation work within IoDT should be assessed.

For this thesis work, the network verification task was selected for implementation. The outcome of the automation project will be presented in section 5.1. Results from the other two research topics from chapter 1.3 are not based on experimental practices, but rather on experience gained from working at IoDT, and from knowledge communicated by colleagues. These findings will be presented in sections 5.2–5.3.

5.1 Automation Within IoDT

The test automation implementation part of the thesis has been successful, provided the expectations. The network verification automation process proved to be a good starting point, since it was quite easy to implement. However, the whole testing process has not yet been possible to automate, due to missing test tools. These tools are under development however.

Some of the false expectations, that were presented in chapter 3.2, have been observed. For example 100 percent test coverage and universal test tool items. The tools available presently do not provide complete coverage, even though the anticipation is that future tools will. This also implies that the expected reduction of test effort is not complete. The testing still needs some user interaction, even though it is minimal compared to the one of completely manual testing. Conclusively the total test effort has been reduced.

It is important to note though, that the network verification might still include application specific work that is not included in the automation scope. For example, when newly implemented features are to be tested, the verification might work perfectly well and produce the requested results of the basic testing. However, the new feature may not work as desired. In general, the activation of a new feature requires some troubleshooting which may take a significant amount of time in contrast to the network verification process. Thus the total time of the verification might still be lengthy. Some aspects for discussion are presented in Table 5.1. The discussion deals with improvements gained from implementing the automated test activity.

Looking back at the Core IoDT and IoDT Support columns in Table 4.1, several other tasks that could be automated are provided. These tend to be more difficult to implement but are all useful. Especially the feature testing item is of interest. Not mainly for GSM feature testing, but rather for future LTE testing described in chapter 5.3.

5.2 External Tasks in IoDT

As for external tasks, there is a summary of ideas in the Non-IoDT column in Table 4.1. These have briefly been discussed in the thesis, but have not been deemed feasible to implement at this stage. Focus should be laid on IoDT specific tasks. If more efficient use of the test environment is required in the future, these automation issues could be discussed again.

5.3 LTE Testing in IoDT

As of today, the IoDT laboratories are not set for LTE testing. The major 3GPP LTE road map is also delayed, and therefore mobile platform developers do not have prototypes ready for IoDT testing either. During this year, however, LTE development is anticipated to grow strong. An IoDT lab for LTE testing is being projected for, and will be set up when hardware is available and mature.

When Ericsson infrastructure and mobile platform vendors' products reach a state when the specifications start to be met, extensive integration and verification testing activities will be required. IoDT will play an important role in this testing. One aspect of this thesis work has been to get the automation work within IoDT kick-started, in order to show the possibilities and open up for further automation work.

The idea and requested outcome of LTE IoDT testing is that as much as possible will be automated at an early stage of the development. The complex of problems with general test cases in the current stage of GSM development, should be less obvious with LTE, as the same features presumably will be tested many times, with several different partners and hardware platforms. This may, however, lead to other problems.

As mentioned in chapter 1.5, it has been decided that the currently used THC test framework is to be used even in LTE testing. The functionality of the automation tools used today are not mainly required by IoDT, but by other testing functions within the company. For this thesis project available tools have been chosen. However, specific tools might be desired for IoDT. A dialog should continuously be held with the ATD team in order to help them develop THC connections to appropriate testing tools. It is a great advantage to have test tool developers in-house.

Property	**Verdict**
Test Effort	Test effort has been reduced as supervision of the testing activity is no longer needed. However, it is important to keep the used telephones and SIM cards handy between test sessions, since exchanging these will require configuration changes.
Test quality	Since the test suite is quite straightforward and not many tests are made, the reliability of the tested system is probably not significantly improved when test automation is implemented. That is, the tester will probably do the same tests manually, without forgetting important parts.
Execution Time	The execution time of the tasks included in the automation scope has been significantly reduced. This is because all tests are run in sequence without user interception.
Total Time	The experienced time, however, has not been significantly reduced. In the perfect case, everything works after network verification, but as discussed additional new features that are not covered by the network verification procedure may still require time consuming troubleshooting.
Stability	During the test sessions the automation program has proved to be stable. The same hardware has been used during test case development. For future testing, the use of the same hardware will also help to keep the test cases stable. However, some elements of the testing do not always work as desired. The THC server may need to be rebooted for one. This kind of error is not present when performing the testing manually.
Work Motivation	Work motivation is highly increased, if tedious manual procedures are automated. Even though the automated testing might need to be fine-tuned, this is more stimulating than doing the same time-consuming procedure every single time. As time is saved, the tester has more time to work with more stimulating tasks.

Table 5.1. Results

However, the manufacturers of some test tools, for example protocol analyzers must also be on track. Several tools used today, such as the Nethawk protocol analyzer, are produced by external manufacturers. Before ATD can make THC adaptations of the tools, the tools must be available. This may lead to unanticipated lead times.

The network verification procedure, as implemented in this thesis shows that automation is possible. In UMTS systems a similar approach to the network structure is in place. Instead of the BSC, there are RNC elements with different software versions. As described in the thesis the approach toward LTE will include fewer network elements. These will, however, require software, and the changing of versions is evident during system development. Therefore a test suite similar to the network verification procedure could be a good start for LTE test automation as well.

Chapter 6

Conclusions

The automation implementation has been successful, and has opened up for future work within the area. Focus should not be laid on GSM test automation, but rather on LTE automation from start – to the extent possible. Effort should not be laid on developing non-IoDT test automation cases, if these do not significantly reduce the work load for the IoDT team.

6.1 Future Work

The scope for future automation is big, and therefore opens up for future studies. At the end of 2008 the LTE technology should be standardized and more mature. IoDT testing in Linköping is not projected to start until then. In connection with this a new study on LTE automation would be appropriate. This could be put into practice by means of another master's thesis project.

Bibliography

[1] 3GPP. 3rd Generation Partnership Project; Technical Specification Group Terminals; AT command set for User Equipment (UE) (Release 6). 3GPP Technical Specification. Document number: 3GPP TS 27.007 V6.8.0, 2005.

[2] 3GPP. 3GPP specifications – releases (and phases and stages). URL: <http://www.3gpp.org/specs/releases.htm>, 2007. Accessed 2007-11-23.

[3] 3GPP. About 3GPP. URL: <http://www.3gpp.org/About/about.htm>, 2007. Accessed 2007-10-08.

[4] Rysavy Research/3G Americas. EDGE, HSPA and LTE – The Mobile Broadband Advantage. Rysavy Research/3G Americas White Paper, 2007.

[5] Regis J. Bates. *Broadband telecommunications handbook*, chapter 21. McGraw-Hill, 2000.

[6] R.F. Bellaver. Wireless: from Marconi to McCaw. *Technology and Society, 2000. University as a Bridge from Technology to Society. IEEE International Symposium on*, pages 197–200, 2000.

[7] Lee Copeland. *A Practitioner's Guide to Software Test Design*. Artech House, 2003.

[8] Erik Dahlman, Hannes Ekström, Anders Furuskär, Ylva Jading, Jonas Karlsson, Magnus Lundevall, and Stefan Parkvall. The 3G Long-Term Evolution – Radio Interface Concepts and Performance Evaluation. *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, 1:137–141, 7–10 May 2006.

[9] Erik Dahlman, Stefan Parkvall, Johan Sköld, and Per Beming. *3G Evolution, HSPA and LTE for Mobile Broadband*. Academic Press, 2007.

[10] Fredric Dahlqvist and Mikael Holmgren. Poseidon – GUI Prototype Implementation of a Mobile Network Mobility Simulator. Master's thesis, Umeå universitet, 2003.

[11] Elfriede Dustin, Jeff Rashka, and John Paul. *Automated Software Testing – Introduction, Management and Performance*. Addison–Wesley, 1999.

[12] Hannes Ekström, Anders Furuskär, Jonas Karlsson, Michael Meyer, Stefan Parkvall, Johan Torsner, and Mattias Wahlqvist. Technical Solutions for the 3G Long-Term Evolution. *IEEE Communications Magazine*, 44(3):38–45, March 2006.

[13] Ericsson. Function Specification – WinFIOL. Ericsson Internal Document. Document number: 155 17-CNZ 223 05 Uen Rev K, 2005.

[14] Ericsson. IoDT Lab Setup. Ericsson Internal Document. Document number: EAB/RJB-05:0600 Uen Rev PA3, 2005.

[15] Ericsson. GSM BSC Survey. Ericsson Internal Document. Document number: EAB/PJM-06:0225, 2006.

[16] Ericsson. GSM System Survey. Ericsson Internal Document. Document number: LZT 123 3321 R6A, 2006.

[17] Ericsson. Understanding the radio environment at LI/ETE. Ericsson Internal Document. Document number: ETE-06:000628 Uen, 2006.

[18] Ericsson. GSM IoDT Presentation. Ericsson Internal Document, 2007.

[19] Ericsson. Introduction to test automation with THC. Ericsson Internal Document. Document number: EAB/PJB-06:0450 Uen Rev PA13, 2007.

[20] Ericsson. Long Term Evolution (LTE): an introduction. Ericsson White Paper. Document number: 284 23-3124 Uen, 2007.

[21] Ericsson. Product specification CCN R3C. Ericsson Internal Document. Document number: ETE-07:001199 Uen, 2007.

[22] Tom Farley. Mobile telephone history. *Telektronikk*, 2005(3/4):22–34, 2005.

[23] GSM Association. GSM hits two billion milestone. GSM Association Press release, URL: `<http://www.gsmworld.com/news/press_2006/press06_29.shtml>`, 2006. Accessed 2007-11-05.

[24] GSM Association. GSM Association Web Site. URL: `<http://www.gsmworld.com>`, 2007. Accessed 2007-10-26.

[25] GSM Association. HSPA – Mobile Broadband Today. URL: `<http://hspa.gsmworld.com/about-hspa/default.asp>`, 2007. Accessed 2007-11-29.

[26] Gunnar Heine. *GPRS: Gateway to Third Generation Mobile Networks*. Artech House, 1999.

[27] IBM Rational Software. Rational ClearCase Web Site. URL: `<http://www.ibm.com/software/awdtools/clearcase/>`, 2008. Accessed 2008-01-09.

[28] Henrik Karlsson. Examining possible new requirements on a mobile controller. Master's thesis, Linköpings universitet, 2007.

[29] V. Kumar. Wireless communications “beyond 3G”. *Alcatel Telecommunications Review*, 2001(Q1):28–32, 2001.

[30] Andrew Miceli. *Wireless Technician's Handbook*. Artech House, 2003.

[31] NetHawk Oyj. NetHawk Web Site. URL: `<http://www.nethawk.fi>`, 2007. Accessed 2007-11-13.

[32] Next Generation Mobile Networks. NGMN web site. URL: `<http://www.ngmn.org>`, 2007. Accessed 2007-11-20.

[33] Nokia/Vodafone. 3G evolution – the trusted road ahead. Nokia/Vodafone White Paper. Nokia document number: 11292 – 09/2005, 2005.

[34] Nordisk Mobiltelefon. Nordisk Mobiltelefon Web Site. URL: `<http://www.nmt.net>`, 2007. Accessed 2007-12-06.

[35] Sixto Ortiz Jr. 4G Wireless Begins to Take Shape. *Computer*, 40(11):18–21, Nov. 2007.

[36] picoChip Designs Ltd. 3GPP Long-Term Evolution: fit or flawed? picoChip White Paper, 2007.

[37] Siegmund M. Redl, Matthias K. Weber, and Malcolm W. Oliphant. *An Introduction To GSM*. Artech House, 1995.

[38] Martin Sauter. *Communication Systems for the Mobile Information Society*. John Wiley & Sons, 2006.

[39] The Eclipse Foundation. Eclipse Web Site. URL: `<http://www.eclipse.org>`, 2008. Accessed 2008-01-09.

[40] John Watkins. *Testing IT: An Off-the-Shelf Software Testing Handbook*. Cambridge University Press, 2001.

[41] Keith Zambelich. Totally Data-Driven Automated Testing. Automated Testing Specialists Inc. White Paper, 1998.

Appendix A

3GPP Releases

GSM/EDGE Release	3G Release	Functional Freeze
Phase 1	-	1992
Phase 2	-	1995
Phase 2+ Release 1996	-	early 1997
Phase 2+ Release 1997	-	early 1998
Phase 2+ Release 1998	-	early 1999
Phase 2+ Release 1999	-	March 2000
-	Release 1999	March 2000
Phase 2+ Release 4	Release 4	March 2001
Phase 2+ Release 5	Release 5	March-June 2002
Phase 2+ Release 6	Release 6	December 2004-March 2005
Phase 2+ Release 7	Release 7	Stage 1 freeze September 2005, Stage 2 freeze September 2006, Stage 3 freeze Mar-Sep 2007
Phase 2+ Release 8	Release 8	Stage 1 freeze December 2007?

Appendix B

Network Verification Test Specification

This document specifies the steps taken in the network verification test program in a pseudocode fashion. The log verification steps set the test verdict. If one log fails to return the requested result, the execution is halted and the verdict *fail* is reported. Other error control is also performed during execution. If there is a problem with the testing equipment, the verdict *inconcise* is reported to the user. If all tests are passed without problems, the verdict *pass* is returned. The tool used for each step is also shown.

Idle Test

1. Start Abis logging (Nethawk).
2. Perform MS reset (Mobitec).
3. Stop Abis logging (Nethawk).
4. Verify log, attach procedure (Verification script).
5. Start Abis logging (Nethawk).
6. Move from cell A to cell B (CCN).
7. Stop Abis logging (Nethawk).
8. Verify log, change of cells procedure (Verification script).
9. Return idle test verdict.

Circuit Switched Test

1. Start Abis logging (Nethawk).
2. Set up a circuit switched call (Mobitec).
3. Check which cell the MSs are in (BSC/MML).
4. Stop logging (Nethawk).
5. Verify log, call procedure (Verification script).
6. Start Abis logging (Nethawk).
7. Set up call between phones (Mobitec).
8. Perform handover while in call (CCN).
9. Check which cell the MSs are in (BSC/MML).
10. Terminate call (Mobitec).
11. Stop logging (Nethawk).
12. Verify log, call/handover procedure (Verification script).
13. Return CS test verdict.

Packet Switched Test

1. Start Abis logging (Nethawk).
2. Perform PDP context (Mobitec).
3. Stop logging (Nethawk).
4. Verify log, PDP context procedure (Verification script).
5. Start Abis logging (Nethawk).
6. Perform FTP GPRS download (Mobitec).
7. Check which cell the MSs are in (BSC/MML).
8. Stop logging (Nethawk).
9. Verify log, FTP download procedure (Verification script).
10. Start Abis logging (Nethawk).
11. Perform FTP GPRS download, large file (Mobitec).
12. Check which cell the MSs are in (BSC/MML).

13. Perform handover during file download (CCN).
14. Check which cell the MSs are in (BSC/MML).
15. Verify log, FTP download/handover procedure (Verification script).
16. Start Abis logging (Nethawk).
17. Perform FTP GPRS upload (Mobitec).
18. Check which cell the MSs are in (BSC/MML).
19. Stop logging (Nethawk).
20. Verify log, FTP upload procedure (Verification script).
21. Start Abis logging (Nethawk).
22. Perform FTP GPRS upload, large file (Mobitec).
23. Check which cell the MSs are in (BSC/MML).
24. Perform handover during file upload (CCN).
25. Check which cell the MSs are in (BSC/MML).
26. Verify log, FTP upload/handover procedure (Verification script).
27. Return PS test verdict.
28. Return test suite verdict.