Management of Telecommunication Systems Design and Development: Learning from Disruptive Innovations in 3G Mobile Systems

by

Yujiro Mochizuki

Bachelor of Arts in Economics, Keio University, 1996

Submitted to the Alfred P. Sloan School of Management in Partial Fulfillment of the Requirements for the Degree of Master of Science in Management of Technology at the Massachusetts Institute of Technology June 2004

© 2004, Yujiro Mochizuki. All rights reserved. The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part.

Signature of Author: ____________________________

Sloan School of Management

May 7, 2004

Certified by: ____________________________

Michael A. Cusumano

Sloan Distinguished Professor of Management Sloan School of Management

Thesis Supervisor

Accepted by: ____________________________

David A. Weber

Director, Management of Technology Program Sloan School of Management
Management of Telecommunication Systems Design and Development: Learning from Disruptive Innovations in 3G Mobile Systems

by

Yujiro Mochizuki

Submitted to the Alfred P. Sloan School of Management on May 7, 2004 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Management of Technology

ABSTRACT

Disruptive innovations in network systems are forcing mobile carriers to manage difficult network software development. Within the limited development time, mobile carriers are required to develop large scale, high quality and robust network software, and accomplish smooth transition/upgrading. In the drastic transition of the network architecture, network carriers often fail to develop platform network software. Compared with KDDI, NTT DoCoMo (DoCoMo) has been coping with network migration problems. The critical difference in both mobile carriers is whether the platform-based approach was applied sufficiently in the network software development. DoCoMo’s insufficient platform-based approach led to migration difficulties. On the other hand, the consistent platform-based approach allowed KDDI to enjoy successful migration.

In order to identify the critical factors in network software development related to this problem, this thesis (1) explores the characteristics of network innovations and (2) analyzes the platform innovations in mobile network software. The ultimate goal of this thesis is to (3) suggest how incumbent mobile carriers can avoid the potential threats and develop the platform network software in future disruptive network innovations. In the future, management of telecommunication systems design and development will face more challenging and more complicated migration because the network concepts and architecture will be completely different from those in the existing 3G networks. The lessons learned from the case studies of DoCoMo and KDDI suggest ideas that mobile carriers can utilize to address tough situations. The platform-based approach definitely will help the migration strategies of mobile carriers and reduce the potential threats in current and future network systems.

Thesis Supervisor: Michael A. Cusumano
Title: Sloan Distinguished Professor of Management, Sloan School of Management
Acknowledgements

In every phase of this study, many people supported me with many suggestions and words of encouragement. I would never have been able to accomplish this work without these people.

First and foremost, I would like to present my deep and sincere gratitude to my thesis advisor, Professor Michael A. Cusumano for providing me opportunities to work with him, and for his essential guidance, great insights and valuable support throughout the course of this study.

I wish to express my gratitude to Professor Yoshiko Takenaka of Keio University, who encouraged me to pursue further studies at MIT Sloan School of Management. I also would like to express my deep appreciation to Mr. David A. Weber for his great support and encouragement.

Finally, I am deeply grateful to my parents, Takeo and Ikuko, for your constant support and love since my childhood. I will never forget the Zen comments you taught me.

Do not think.
The moon appears when the clouds are gone.
All the time it has been there in the sky
So perfectly clear.

-Zen Comments on the Mumonkan by Zenkei Shibayama

Yujiro Mochizuki
Boston, Massachusetts
May 2004
# Table of Contents

1 Introduction .......................................................................................................................... 10

1.1 Purpose of This Thesis ..................................................................................................... 14

1.2 Scope of This Thesis ....................................................................................................... 16

1.3 Overview of the Core Network ...................................................................................... 17

1.4 Thesis Structure ............................................................................................................. 21

2 Services and Technological Characteristics of Mobile Telecommunications .............. 23

2.1 Service and Technology Trends .................................................................................... 23

2.2 Service and Technology Trends as an S-curve (Continuity and Discontinuity) ....... 25

2.2.1 Technology Trends: Generation to Generation ....................................................... 25

2.2.2 Technology Trends: Same Generation ................................................................... 28

2.3 Service Architectures in Mobile Telecommunications ................................................. 31

2.4 Unique Characteristics of Telecommunication Systems .............................................. 32

2.4.1 Reliability and Availability ..................................................................................... 32

2.4.2 Compatibility ............................................................................................................ 33

2.4.3 Homogeneity and Heterogeneity ........................................................................... 34

2.4.4 Scale ....................................................................................................................... 35

2.4.5 Real-time Response ............................................................................................... 36

2.4.6 Geographical Distribution of the System ............................................................... 37

3 Platform Innovation in Network Software ...................................................................... 38

3.1 Platform Principles and Perspectives from the Literature .......................................... 39

3.1.1 Component Standardization .................................................................................... 41

3.1.2 Product Architecture ............................................................................................. 42

3.1.3 Product Platform ................................................................................................... 43
Appendix: Abbreviations and Terminologies ................................................................. 107

Bibliography ......................................................................................................................... 109
# List of Figures

Figure 1-1: Market Share of Japanese Mobile Subscribers (Total as of March 2004)........ 11
Figure 1-2: Market Share of Japanese 3G Mobile Subscribers (Total as of March 2004).... 12
Figure 1-3: Evolution of the Mobile Systems in Japan............................................. 13
Figure 1-4: Evolution of Mobile Systems in Europe and the US ............................... 14
Figure 1-5: Core Network in 3G (UMTS) Systems................................................ 19
Figure 1-6: Thesis Structure.................................................................................... 22
Figure 2-1: Technology Trends in Mobile Systems (Generation to Generation)........... 28
Figure 2-2: Technology Trends in Mobile Systems (in the Same Generation).............. 30
Figure 2-3: Compatibility in Telecommunication Systems...................................... 34
Figure 3-1: Component Standardization in 3G System (DoCoMo)............................. 42
Figure 3-2: Basic Structure of Call Handling Subsystems...................................... 44
Figure 3-3: Product Family Evolution..................................................................... 49
Figure 3-4: Typology of Platform Change in Product Family Evolution................... 50
Figure 3-5: Original Platform of Network Software................................................. 51
Figure 3-6: Platform Extension of Network Software.............................................. 52
Figure 3-7: Platform Renewal of Network Software............................................... 53
Figure 3-8: Requirements for Network Software.................................................... 54
Figure 3-9: Complicated Interfaces....................................................................... 56
Figure 4-1: The Discontinuity of Generation in DoCoMo and KDDI Networks............ 65
Figure 4-2: Number of Subscribers (DoCoMo)...................................................... 68
Figure 4-3: Number of Subscribers (KDDI)........................................................... 68
Figure 4-4: Roadmap of New Mobile Systems (DoCoMo)....................................... 69
Figure 4-5: Roadmap of New Mobile Systems (KDDI) ................................................. 69
Figure 4-6: Network Software Development of DoCoMo ............................................ 71
Figure 4-7: Network Software Development of KDDI ................................................ 71
Figure 4-8: Waterfall Model ...................................................................................... 74
Figure 4-9: The New Platform of the DoCoMo's Network Software (HLR). .................. 78
Figure 4-10: The Complex Platform Architecture of the UMTS Platform .................. 79
Figure 4-11: Number of Subscribers (W-CDMA, DoCoMo) ....................................... 81
Figure 4-12: The More Complex Platform Architecture of the UMTS Platform .......... 82
Figure 4-13: New Platform Architecture: Global Roaming and Integration of the SCP ..... 86
Figure 4-14: The Development Size of Software (Example of One Project) ............... 88
Figure 4-15: The Accumulated Development Size of Software (Example of One Project) 88
Figure 5-1: The Gap Created by Complex Architecture ............................................. 94
Figure 5-2: Phases of Platform Innovation ................................................................. 95
Figure 5-3: Discontinuities among Three Network Generations ................................. 100
List of Tables

Table 1-1: Network Entities in 3G Systems ................................................................. 18
Table 3-1: Major Differences among Subsystems (Call Functions) ......................... 45
Table 4-1: Comparison of Mobile Systems ..................................................................... 62
Table 4-2: Required Changes in Network Software ....................................................... 65
Table 4-3: Differences in Software Development Strategy ............................................. 73
Table 4-4: The Number of Dynamic Steps in PDC and UMTS (HLR) ......................... 76
Table 4-5: The Relationship between Reuse Rate and Defect Rate (VLR/MSC/HLR) .... 84
Table 4-6: Low Quality Due to Rapid Creation ............................................................... 89
Table 4-7: CPU Usage under the High-Load Conditions (%) ......................................... 89
Table 5-1: Required Changes in Network Software to 4G ............................................. 91
1 Introduction

International Mobile Telecommunications-2000 (IMT-2000) technologies, an international 3G standard developed by the International Telecommunications Union (ITU), are expected to generate significant changes in the telecommunication services sector over the next ten years. 3G systems provide global mobility with a wide range of services including basic calls, short messages, movie email, Internet and high-speed data transmission services. 3G systems are expected to usher in social, economic and cultural changes that will affect people around the world. According to the ITU, approximately 120 licenses have already been granted to mobile carriers worldwide. The IMT-2000 standard comprises five different technologies that have been approved by the ITU, but most mobile carriers apply W-CDMA (Wideband Code Division Multiple Access) or CDMA2000 1X (Code Division Multiple Access 2000) to their networks.

The first mobile carrier worldwide to initiate 3G services was the Japanese firm, NTT DoCoMo, which is the largest mobile communications carrier with 46 million subscribers in Japan (Figure 1-1). DoCoMo launched FOMA (Freedom of Mobile multimedia Access) 3G services based on W-CDMA on October 1, 2001. KDDI, Japan's second largest mobile carrier with nearly 17 million subscribers, began its 3G services which uses CDMA2000 1X on April 1,
2002. Vodafone, Japan's third largest mobile telecommunications carrier with 15 million subscribers, initiated its Vodafone Global Standard 3G service, which uses W-CDMA and the latest version of 3GPP (Third Generation Partnership Project) standards, on December 20, 2002. More than 16 million subscribers enjoy 3G services in Japan (Figure 1-2). KDDI is a dominant mobile carrier (80.9%) in 3G mobile systems in Japan.

![Market share of Japanese mobile subscribers (All)](image)

<table>
<thead>
<tr>
<th></th>
<th>DoCoMo</th>
<th>KDDI</th>
<th>Vodafone</th>
<th>Tu-Ka¹</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of subscribers</td>
<td>45,926,700</td>
<td>16,958,800</td>
<td>15,002,400</td>
<td>3,631,800</td>
<td>81,519,700</td>
</tr>
<tr>
<td>Market share</td>
<td>56.3%</td>
<td>20.8%</td>
<td>18.4%</td>
<td>4.5%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Source: Telecommunications Carriers Association (TCA)

**Figure 1-1: Market Share of Japanese Mobile Subscribers (Total as of March 2004)**

¹ Tu-Ka is a subsidiary of the KDDI group.
Market share of Japanese 3G mobile subscribers

<table>
<thead>
<tr>
<th></th>
<th>DoCoMo</th>
<th>KDDI</th>
<th>Vodafone</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of</td>
<td>3,045,100</td>
<td>13,509,200</td>
<td>137,700</td>
<td>16,692,000</td>
</tr>
<tr>
<td>subscribers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market share</td>
<td>18.2%</td>
<td>80.9%</td>
<td>0.8%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Source: Telecommunications Carriers Association (TCA)

Figure 1-2: Market Share of Japanese 3G Mobile Subscribers (Total as of March 2004)

Moving towards 3G networks, mobile carriers need to upgrade not only their hardware but also their network software. The transition from 2G/2.5G to 3G is extremely challenging. Even if W-CDMA uses network architecture/protocols similar to those used in GSM networks, many European carriers have been applying evolutionary steps on the road to 3G (i.e., from GSM to GPRS, EDGE and W-CDMA).
Unlike European mobile carriers, Japanese mobile carriers had to migrate from completely different systems to 3G systems (Figure 1-3, Figure 1-4). Since 1993 DoCoMo has developed PDC as its 2G standard rather than GSM or CDMA. KDDI also has built its own PDC systems\textsuperscript{2} since 1994. PDC is a Japanese standard for digital mobile telephony (800 MHz and 1500 MHz bands). The network architecture, wireless access methods and network protocols entirely differ from those of other systems. Platform network software is a key factor that integrates two different networks and achieves radical network innovation.

\textbf{Figure 1-3: Evolution of the Mobile Systems in Japan}

\textsuperscript{2} Later in 1998, KDDI implemented cdmaOne, a CDMA system.
1.1 Purpose of This Thesis

This thesis aims to accomplish three goals: (1) to explore and identify the characteristics of network innovations, (2) to analyze recent platform innovations in mobile network software, and (3) to suggest how mobile carriers can recognize and avoid potential threats and develop platform network software in future disruptive network innovations.

In the drastic transition of the network architecture, network carriers often fail to develop platform network software. Compared with KDDI, DoCoMo encountered huge network problems in developing 3G mobile systems, which entailed (1) the introduction of 3G systems,
(2) the implementation of global roaming and network dual services, and (3) the integration of the Service Control Point (SCP) function. In deploying 3G networks and services, DoCoMo encountered unexpected problems such as no response to dialed calls dropped calls, disruptions of customer data, and low system performance.

Similar transition will happen in the future network transition (from 3G to 4G, 5G). Telecommunication today is a basic, necessary service for individuals and for corporations. Unsuccessful launches of new technologies can result in critical and very costly failures. New networks require huge volume of initial development (often over 800KLOC\(^4\)\(^5\)) and extremely sophisticated technical expertise of platform design, protocols and traffic management. The pressure to shorten development time and the rapid implementation of the new services and functions can also cause the low software quality. Moreover, the new network must include almost all the functions of the previous network (backward compatibility) and both two networks need intricate interfaces to communicate with each other. However, by addressing the potential architectural and scale problems beforehand, mobile carriers can make smooth network

---

\(^3\) The SCP function provides the advanced intelligent service such as auto-answering, call forwarding and call reject services.

\(^4\) KLOC means "kilo lines of codes." In this thesis KL also means KLOC.

\(^5\) When DoCoMo implemented its 3G systems in 2001, the total development size of the HLR and the switching center (MSC/VLR/SGSN) reached 835KL and over 1.5ML respectively.
transitions, and thus avoid many of migration problems.

1.2 Scope of This Thesis

First, this thesis focuses on developing network software in mobile communications, especially "core network" software (Figure 1-5). Core network provides switching, routing, location management and database functions. All data from handsets is transferred via the core network. Innovation in core networks is essential for the next generation network services. Core networks are slightly different from UMTS (3GPP) and CDMA2000 (3GPP2) networks, but examining these differences is beyond the scope of this thesis. Hardware (infrastructure) and software in handsets are also out of scope.

Second, the analysis is mainly based on the transition from 2G (2.5G) to 3G in Japan. The reasons are: (1) recent transition is more applicable to the coming transition (3G to 4G, 5G), (2) the 3G system in Japan is most advanced and has the longest history worldwide and (3) the discontinuity between PDC and UMTS is the greatest among other network migrations. More than ten years have passed since the first transition occurred (1G to 2G). During this ten-year period, technology and the number of subscribers have completely changed. The recent transition can be considered similar to the next network transition. Since launched in 2001, the
3G system in Japan has experienced continuous implementation of new services without the architectural innovation in network software. Analyzing the network software in Japan, we can identify the latest aspects in platform innovation. It is more challenging to manage huge discontinuity in the network transition. We can also understand how mobile carriers can manage the disruptive network innovations.

Third, this thesis explores the underlying platform concepts and innovations in mobile network software. Meyer and Lehnerd define platform as "a set of subsystems and interfaces that form a common structure from which a stream of related products can be efficiently developed and produced" [1]. Thus, in this thesis "platform" means the core network software that provides fundamental communication services for end users. Due to the limited scope of this study, OS and contents-based services (e.g., web application services) are not included.

1.3 Overview of the Core Network

The core network is divided into circuit switched (CS) and packet switched (PS) domains. Circuit switched elements are Mobile services Switching Center (MSC), Visitor Location Register (VLR), and Gateway MSC (GMSC). Packet switched elements are Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). Both CS/PS domains share
some network elements such as Home Location Register (HLR) and Authentication Center (AuC) (Table 1-1).

Table 1-1: Network Entities in 3G Systems

<table>
<thead>
<tr>
<th>No.</th>
<th>Network Entity</th>
<th>Major Functions</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MSC</td>
<td>Switching services, controlling calls</td>
<td>The VLR/MSC are usually implemented in the same node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mobility management for the subscribers</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>VLR</td>
<td>Temporary subscriber database</td>
<td>The VLR/MSC are usually implemented in the same node.</td>
</tr>
<tr>
<td>3</td>
<td>SGSN</td>
<td>Packet switching</td>
<td>DoCoMo's network integrates the SGSN with the VLR/MSC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mobility management for the subscribers</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>GMSC</td>
<td>Gateway of circuit switching services</td>
<td>Connects to the PSTN and ISDN</td>
</tr>
<tr>
<td>5</td>
<td>GGSN</td>
<td>Gateway of packet switching services</td>
<td>Connect to the Internet</td>
</tr>
<tr>
<td>6</td>
<td>HLR</td>
<td>Management of the subscriber database</td>
<td>The HLR/AuC are usually implemented in the same node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Location management</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Call handling</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>AuC</td>
<td>Authentication of the subscriber</td>
<td>The HLR/AuC are usually implemented in the same node.</td>
</tr>
</tbody>
</table>

Figure 1-5 outlines the core network. The responsibilities of 3GPP core network are the followings:

- Mobility management
- Call connection control between the user equipment and the core network
- Core network signaling among the core network nodes.
- Inter-working functions between the core network and external networks.
- Packet related functions
- Operation and Maintenance (O&M) functions to maintain the network systems [2][3]
This network architecture is based on 3GPP TS 23.002 V6.2.0, 3GPP TS 29.002 V6.3.0 and 3GPP TS 09.02 V7.9.0

Figure 1-5: Core Network in 3G (UMTS) Systems
(3GPP2 systems also have similar architecture)

- **Location Management (from handset A):** 1, 2, 3, 4, 5, 6, 7, 8 (1, 2, 3', 4', 5', 6', 7, 8)

  The main task of location management is to keep track of the subscriber's current location. When mobile handsets connect to a wireless network, the VLR (SGSN) updates
the location information in the HLR and stores the subscriber data sent by the HLR.

- **Authentication (from handset A):** 1, 2, 3, 4, 5, 6, 7, 8 (1, 2, 3', 4', 5', 6', 7, 8)

  3G mobile networks utilize a challenge-response mechanism to ensure that only authorized subscribers can access the network. Authentication information is made in the HLR/AuC and sent back to the VLR (SGSN). The VLR (SGSN) verifies the information and enables the service.

- **Call Handling (circuit switched):** 9,10, 5-8, 1-4, 10', 9'

  After receiving the message from the GMSC, the HLR searches the subscriber location in its database (DB) and requires the VLR to provide a roaming number. This routing information is sent back to the GMSC and the GMSC handles the routing.

- **Packet Handling (to handset A):** 11, 12, 12', 13, 6', 7, 8

  After receiving the message from the GGSN, HLR searches for the subscriber location in its database. This information is sent back to the GGSN and the GGSN sends a confirmation message to the SGSN, which then connects the mobile handset via the RNC.
• **Operation and Maintenance (between VLR/MSC (SGSN) and HLR): 4.5 (4', 5')**

Several operations are used to maintain consistency between network entities. If the VLR (SGSN) loses the subscriber data, it requests the latest subscriber information from the HLR.

### 1.4 Thesis Structure

Chapter 1 describes the general information of mobile communications (the number of subscribers, market share, evolution of mobile systems and network architecture). Chapter 2 illustrates the service and technological transition steps (1) between generations of networks and (2) in the same generation. Chapter 2 also describes the unique characteristics in telecommunication services. Focusing on the network software, Chapter 3 explores the platform principles and provides a framework (mainly that of Meyer and Lehnerd) upon which to analyze platform innovation processes in network software. Chapter 3 examines the innovation style and difficulties that mobile carriers face in platform innovations. Chapter 4 analyzes the network migration from the previous network to a brand-new network. Comparing the successful network migration of KDDI with the troublesome migration of DoCoMo, I would like to point out important findings (the similarities and dissimilarities of these cases), especially
concerning the problematic migration. Building upon Chapter 4, Chapter 5 suggests how mobile carriers manage the software development for future network innovations. Chapter 5 examines platform innovation strategies to reduce migration risk. I present these strategies under four software development phases: (1) initial phase, (2) early development phase, (3) expansion phase and (4) saturation phase.

In this thesis the major data such as the defect rate, development size, development time, reuse rate, number of dynamic steps (PDC and UMTS) and CPU performance, is based on the actual project information.

![Figure 1-6: Thesis Structure](image-url)

---

22
2 Services and Technological Characteristics of Mobile Telecommunications

2.1 Service and Technology Trends

During the past few decades, mobile telecommunications have evolved in both functionality and utility, from first-generation analogue transmission (1G) to second-generation high-speed digital transmission (2G). Demand for data communications, web services, instant availability and packet-based charging, as well as a lack of radio spectrum allocated for 2G has led to the introduction of 2.5G. Today mobile carriers have started to introduce 3G systems, the aim of which is to substantially enhance the mobile subscriber experience by providing instantaneous, ubiquitous access to a large variety of functionality-rich applications at high bit rates and reasonable costs. While 4G systems have not yet entered the market, the approaching 4G mobile communication systems are digital IP-based and projected to provide a wide variety of new services, from high-quality voice to high-definition video, through high-data-rate wireless channels. Mobile technology continues to evolve in the same way.

1G: AMPS, TACS

First-generation systems, designed mainly for voice communication, use analog and Frequency Division Multiple Access (FDMA) technologies.
2G: TDMA (GSM, PDC), CDMA (IS-95A)

Second-generation systems, which use digital technology, are capable of providing voice/data/fax transfer and a range of other value-added services such as call waiting and call forwarding services. Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) are the technologies used in 2G systems.

2.5G: GPRS, PDC packet (i-mode), CDMA (cdmaOne/IS-95B)

Enhanced 2G systems with email and Internet access services are called 2.5G. GPRS and PDC packet services are the 2.5G systems, derived from GSM and PDC respectively. The upgraded version of cdmaOne (IS-95A) is cdmaOne (IS-95B), which adds packet-switched capability and offers data rates up to 115Kbps.\(^6\)

3G: W-CDMA (UMTS), CDMA 2000

Third generation systems promise increased bandwidth of up to 384 Kbps when a device is stationary or moving at pedestrian speed, 128 Kbps in a car, and 2 Mbps in fixed applications. Enhanced multimedia services (picture messaging, video streaming, and remote control),

\(^6\) Kbps means Kilobit per second.
usability on all basic and popular modes (telephony, e-mail, paging, fax, and Web browsing), and global roaming capability (throughout Europe and Japan or North America) are expected in 3G services.

4G: MC-CDMA (OFDM with CDMA overlay) or OFDM

The fourth generation systems are expected to support interactive multimedia services such as teleconferencing and wireless Internet in addition to wider bandwidths, higher bit rates (up to 100M bps), global mobility and service portability. 4G systems are based on all IP networks.

2.2 Service and Technology Trends as an S-curve (Continuity and Discontinuity)

2.2.1 Technology Trends: Generation to Generation

We can analyze the evolution trends in mobile communications from the viewpoint of the technology "S-curve" theory [4]. An S-curve can be viewed as a plot of time on the x-axis measured against performance of the technology on the y-axis. Performance can be measured by either technological performance or market penetration. In this thesis "performance" means the capacity, functionality (new services, mobility and backward compatibility), and quality of the mobile communications. Mobile network software in core networks allows mobile carriers
to improve their performance.

The interaction of innovations (product innovation and process innovation [4]) tends to enhance performance. As a result, the accumulation of innovations leads to continuous and exponential growth in performance. At the beginning of 2G mobile services in Japan (1993) voice quality was very poor (many dropped calls and much "noise") and the features of network services were quite limited (mainly voice telephony). Over time, voice quality dramatically improved and a wide variety of new services such as answering, call waiting and call forwarding, appeared. Soon in 1999, 2G systems evolved into 2.5G systems that provide packet services (email, and web browsing).

A technological discontinuity occurs when a new technology replaces more mature technology. New technology has the potential to quickly surpass an old technology’s performance. In mobile communications 2G digital systems replaced 1G systems and 3G systems are now substituting the 2G (and 2.5G) systems. Upon the introduction of a new technology, however, the performance almost always appears to be inferior to the performance of the previous (more mature) technology. Current technologies also evolve constantly and expand the gap between generations. For example, after releasing the 3G service (FOMA) in
2001, DoCoMo's 2.5G system (PDC) started to provide photo email services. With high quality telephony communications, email services (text messages and pictures) and web browsing, PDC became quite sophisticated. On the other hand, 3G mobile services began with limited performance on October 1, 2001. Mobility was limited to urban areas and backward compatibility was not perfect. Accompanied by unsophisticated handsets (large, heavy and short battery), DoCoMo's 3G services were obviously inferior to its 2.5G PDC services.

As the performance of the new technology improves, it will ultimately replace an older system. However, a huge discontinuity generally exists between adjoining generations of technologies. In mobile communications, mobile carriers had to cope with a huge gap between 2.5G and 3G systems (Figure 2-1).
2.2.2 Technology Trends: Same Generation

The S-curve represents the accumulation of innovations, thus the curve is continuous in the same generation. However, we can also observe discontinuity in the same generation (Figure 2-2). Upgrading and the lifecycle of network software (six months to one year) often result in discontinuity. With the implementation of new services, the performance of current software
jumps up to the next level of performance. Thus, technology trends in mobile systems advance like stairs.

Software problems in network software can also cause discontinuity in the same generation. Critical software defects prevent mobile carriers from providing continuous network services. In Japan, if more than 30,000 subscribers cannot access their communication services for over two hours, mobile carriers must report the service failure to the government. With repeated such failures, mobile carriers can lose their license to provide network services if they cannot repair the system quickly. These discontinuities (failures) can be difficult to manage, so Operation and Maintenance (O&M) functions are crucial in network software. O&M functions occupy about 20-40\%\textsuperscript{7} of the network software.

\textsuperscript{7} Data is based on DoCoMo's projects.
Figure 2-2: Technology Trends in Mobile Systems (in the Same Generation)

**Vertical Change**

Vertical change represents improvements in performance. Typical examples include: (1) the implementation of new functions for end-users (e.g., new services, improvement of usability and availability), (2) architectural change in platform software (elimination of the unnecessary functions for high performance) and (3) integration of network entities (network carriers sometimes integrate network entities to reduce their fixed costs).

**Horizontal Change**

Horizontal change demonstrates a technology’s unvarying performance. Typical examples include: (1) implementation of O&M functions, (2) replacement of a temporary patch
with a completely new (bug-free) program and (3) resource allocation and interface change among function blocks.

2.3 Service Architectures in Mobile Telecommunications

Service architecture defines a set of concepts and principles for telecommunication services. Mobile telecommunications has three key service categories: (1) basic communication service (basic call), (2) supplementary (contents) service and (3) O&M service [5]. Other common functions such as protocol management, OS interface, and DB management, support these functions.

Basic communication service provides mobility (location updating) and call handling services (originating/terminating calls). This service focuses on the interactions between network entities (DB, switching centers and service control points, etc.) to communicate with each other. Basic communication services are fundamental functions, so the primary requirements of these services are robustness, high availability, and low failure rate.

Supplementary service focuses on the value-added enhancements over basic communication services. Call waiting, call forwarding, calling line identification presentation
and multi-call services represent the supplementary services. Market demands require these services to be highly customized and user-oriented [6]. In many cases, these advanced services operate in different subsystems (function blocks) in the network software architecture.

The O&M service serves as a comprehensive maintenance solution that covers the management of each system. O&M ensures management of network traffic, network configuration, subscriber/message tracing, alarm, and supervision. The evolution of network functions has resulted in the expansion of the O&M service. O&M allows mobile carries to analyze the cause of network problems and to restore the system quickly.

2.4 Unique Characteristics of Telecommunication Systems

A number of characteristics distinguish telecommunication systems from other computer systems [9][11]. The followings are the main characteristics of telecommunications service requirements.

2.4.1 Reliability and Availability

Almost every industry in the world depends upon telecommunications. Emergency services such as calls to police, ambulance and fire fighters, also depend on telecommunication
systems. If the telephone network goes down, even for a short time, most businesses and individuals can experience serious consequences.

Therefore, telecommunication systems must be significantly robust. A switching system is supposed to be available for all but two hours within a 40-year period [9]. New features might destroy the existing functions in telecommunication systems, so the new features are carefully designed and tested for "unfailing reliability" before upgrades are implemented. The fault tolerance system is essential and regression tests of all network elements are routinely performed to maintain network quality.

2.4.2 Compatibility

Telecommunications carriers have to maintain the compatibility (1) between generations in the same carrier, (2) in the same generation in the same carrier, (3) between generations in different carriers, and (4) in the same generation among different carriers (Figure 2-3). Even if the system architecture fundamentally changes (e.g., from PDC to UMTS), the new system must support the previous functions to satisfy customer demands. Basically, the transition between generations in the same carrier takes many years (usually at least five years). This backward compatibility helps subscribers enjoy previous services in the new networks and allows mobile
carriers to achieve smooth migration.

Although (3) and (4) are not mandatory, backward or forward compatibility is a key factor for providing roaming services. In particular, 3G systems require global mobility, so carriers cannot ignore these compatibility requirements.

<table>
<thead>
<tr>
<th>Different generation</th>
<th>Same generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support several protocols to maintain the previous functions E.g., PDC and UMTS E.g., PDC and cdmaOne E.g., PDC and CDMA2000</td>
<td>Support the same protocols and improve functions/services E.g., PDC + new services E.g., UMTS + new services E.g., CDMA2000 + new services</td>
</tr>
<tr>
<td>Support several protocols to keep and expand mobility E.g., GSM and UMTS E.g., cdmaOne and CDMA2000</td>
<td>Support the same protocols to keep and expand mobility E.g., UMTS-UMTS E.g., CDMA2000-CDMA2000</td>
</tr>
</tbody>
</table>

Figure 2-3: Compatibility in Telecommunication Systems

2.4.3 Homogeneity and Heterogeneity

From the viewpoint of customers, telecommunication services are quite similar among telecommunication carriers. Almost all mobile carriers support the basic communication service (originating/terminating calls) and other supplementary services (call waiting, call forwarding, etc). It is extremely difficult for customers to identify differences in network
services.

From the viewpoint of network carriers, however, there is a huge heterogeneity in telecommunication systems. Several manufacturers build hardware for telecommunication systems. In fact, NEC and Fujitsu build almost all the infrastructure for DoCoMo 3G services. Each hardware component of the network comes with its own version of telecommunications software. Different manufacturers organize their software differently. The same feature may be supported by different pieces of software that reside in different hardware components [9].

2.4.4 Scale

The programs that control network nodes consist of about several million lines of code (e.g., the Home Location Register (HLR) in DoCoMo's 3G mobile system has about 2.5 million lines of code). Hundreds of features coexist in a typical mobile system. Furthermore, every six-to-twelve months new features with 200-500KL are added to the network software. When a new feature is added to the network node, it can affect millions of lines of code and tens or even hundreds of other features. Because of heterogeneity, network carriers must multiply this effect by the number of different kinds of switches in the network. Thus, failures in the upgraded software will consequently cause huge problems in the mobile system.
2.4.5 Real-time Response

Telephony is real-time application software that requires the system to respond immediately to its requests even under overloads.\(^8\) The generic requirements for switching systems specify deadlines for various responses such as high-call throughput and low-call setup delays [6][7][8][9]. In DoCoMo's 3G systems, the performance requirement is 750 transactions/second.\(^9\) In order to maximize hardware performance, many carriers use the C/C++ programming language. In developing network software, network engineers must concentrate on both system reliability and performance. If network engineers concentrate only on system reliability, the network software will require so many error-detecting functions that eventually system performance will deteriorate. On the other hand, if network engineers focus on only system performance, the software will lack architectural design and finally lose configurability and reliability. Under the short development time, it is challenging to fulfill both requirements (performance and reliability). The requirement for real-time response makes it difficult to develop the system design and implement new versions.

---

\(^8\) Overload condition means that the usage of CPU exceeds 70% (70-100%).

\(^9\) This is the requirement for HLR.
2.4.6 Geographical Distribution of the System

In today’s switching systems, a call may be distributed among several sites. Heterogeneous clusters of nodes cooperate with each other to set up and complete the call. Network software must not only cover general routing functions but also operate differently in its own environment. Therefore, software engineers must design simple network software configuration (configurability) to manage the network systems.

At the same time, the wide geographical distribution of the system often magnifies the negative effects of software defects. Network software for the HLR is distributed to all of the HLRs in 3G systems and the network software for the MSC, VLR and SGSN10 is also delivered to all of the MSCs, VLRs and SGSNs. For example the number of the HLRs in DoCoMo’s network exceeds 140. The total number of CPUs is over 500. Furthermore, a duplex system is applied in each system. One software defect will create hundreds of defects across the entire network.

---

10 The MSC, VLR and SGSN are often integrated in the same network node. In this case only one network software carries out the MSC/VLR/SGSN functions.
3 Platform Innovation in Network Software

In recent years platform design and platform architecture have been receiving much attention from managers and academics. The market pressure for shorter development time for new services, along with the need for high quality, robust products, and smooth transition/upgrading have prompted technology managers to focus on this field. In the quest to manage the complex need to offer greater product variety, firms in many industries are considering platform-based product development [12]. Key in this approach is the sharing of components (modules\textsuperscript{11}) and other function blocks (subsystems\textsuperscript{12}) across a family of products. Like physical products, software consists of myriad subsystems and modules. These software functions are connected to each other via several interfaces. Thus, the platform-based approach is also quite effective in software development.

Historical success stories such as the Sony Walkman [15][16], and Microsoft’s Windows NT [17] have shown both the benefits and the logic behind this platform concept. In terms of lifecycle, thinking about platforms for families of products rather than individual products is one of the key drivers behind the success of short-cycle-time companies. In fact, the software

\textsuperscript{11} Module is the smallest unit in the software (10-1000 lines).
\textsuperscript{12} In this thesis subsystem means "function block." A subsystem consists of several modules.
development cycle in telecommunications has shortened from two years to a half year. Today, the development time is decreasing (mobile carriers are trying to shorten the software lifecycle from six months to three months). A clear gap between platform concept and practical issues in literature, however, still exists when it comes to designing, testing, implementing and managing product families and their successive platforms [18].

As a first step to fill this gap between platform concept and practical issues, it is essential to analyze how mobile carriers (3G mobile network carriers, e.g., DoCoMo and KDDI) adopted the concept of platform thinking in their software development processes. It is also important to discuss the grounds behind thinking in terms of platforms and product families by reviewing the relevant literature related to these concepts.

3.1 Platform Principles and Perspectives from the Literature

A position of leadership is supposed to get what powerful companies want. Firms that made the main or core product continue to have a strong influence on the market. Gawer and Cusumano define "platform leaders" as "the firms that can exert a strong influence over the direction of innovation in their industries and produce/use complements [19]." These authors emphasize that more and more firms want their products to become foundation on which other
companies build their products or offer their services. At the same time, they note that no single company can usually create all the complementary innovations needed to develop a market [19].

Offering a wide variety of new services to meet customers' needs, companies can more effectively compete in the global market. In order to attract customers, 3G mobile systems provide not only basic call functions but also more attractive supplementary services such as high-resolution photos, movie email and global roaming.

When designing and implementing services, firms are challenged to efficiently develop a wide variety of new services. In order to manage the quality, cost and delivery, firms are considering product/software development approaches that reduce complexity and better leverage investments in product design and manufacturing [12][13][14]. Platform thinking, the process of identifying and exploiting commonalities among services, target markets, and the processes for creating and delivering offerings appears to be a successful strategy to create new services at low costs [20][21].

Before moving forward, I will clarify the concepts related to platform thinking: component
standardization, product architecture, product platform, process platform, global platform and product family [21][23][24][25][26][27][28][29].

3.1.1 Component Standardization

Component standardization of software enables the use of the same module/subsystem in multiple subsystems. The use of standard components can lower the complexity, cost and lead-time of product development. Standardization can occur only when (1) a component contains commonly useful functions and (2) the interface of the component is identical across more than one product [21]. Henderson and Clark also observed that innovation is associated with investments on the level of product components as well as on the level of their integration into product architecture [22]. In developing its network software, DoCoMo standardized several common functions, including database access, encoding/decoding and resource management functions. Standardizing these functions, DoCoMo enhanced software resource efficiency, accelerated the development speed and made its software robust (Figure 3-1).

Focusing on the process of developing and defining the required interfaces has proven to facilitate the often-challenging shift from a single product development process to a product family approach [23].
3.1.2 Product Architecture

Ulrich [21] has defined the physical product architecture as (1) the arrangement of functional elements, (2) the mapping from the functional elements to physical components and (3) the specification of the interfaces among interacting physical components. This concept is also applicable to software. A key issue in the design of the product architecture is whether the technical design of the architecture allows changes to be made to a product easily.

Small functional elements have several interfaces among the elements and form the integral software architecture. In order to implement a new product's function one modification often requires other modifications in other modules. On the other hand, for products with a
modular architecture, desired changes to a functional element can be localized to one component. A modular product design, therefore, increases the likelihood to use standard components and also enables component interfaces to be identical across several products.

### 3.1.3 Product Platform

Meyer [24] defines a product platform as "a set of subsystems and interfaces that form a common structure from which a stream of related products can be efficiently developed and produced." A product platform is often defined in terms of physical components, but the product platform can be defined in software as well. The product platform is the basis for developing new product variants. Several variants can be found in network software. For example, the supplementary service function is a variant of a basic call function in DoCoMo's network software. We can find the significant difference between these two functions in terms of real coding, but the basic architecture of the subsystem is quite similar (Figure 3-2). The basic call function is a product platform, so the specifications of basic call functions are reused to design supplementary service functions. The major variants in the supplementary service function include the message checking and message handling function, message translation function, subscriber data access function, error detection function, and O&M function.
DoCoMo also applied the product platform concept to design the Service Control Point (SCP\textsuperscript{13}) functions in the HLR. DoCoMo integrated the SCP with the HLR in 2003 to reduce the fixed cost of the infrastructure. DoCoMo's HLR network software now performs three major call handling functions: basic call, supplementary service, and the SCP. The SCP function also has a similar structure and functions to the basic call function. The major differences among these three functions are required speed, quality and resources (Table 3-1).

\textbf{Figure 3-2: Basic Structure of Call Handling Subsystems}

(Basic Call, Supplementary Service and the SCP Functions)

\textsuperscript{13} The SCP provides advanced intelligent services such as auto-answering, call forwarding and call reject services.
Table 3-1: Major Differences among Subsystems (Call Functions)

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Required speed(^{14})</th>
<th>Required quality</th>
<th>Required resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic call function</td>
<td>High (max 750 TPS(^{15}))</td>
<td>Extremely high</td>
<td>Large</td>
</tr>
<tr>
<td>Supplementary service function</td>
<td>Low (max 200TPS)</td>
<td>High</td>
<td>Small</td>
</tr>
<tr>
<td>SCP function</td>
<td>Low (max 200TPS)</td>
<td>High</td>
<td>Medium</td>
</tr>
</tbody>
</table>

3.1.4 Process Platform

The process platform, the standard process in software development, refers to the specific setup of the production system to readily produce the desired variety of products. A well-developed production system includes flexible equipment and systems such as automatic error detection, auto tests and computerized scheduling [15][16]. The process platform helps project managers free their projects from isolated, chaotic, or ad-hoc development methods with industry-side best practices and a configurable architecture. Process platforms can reduce process complexity and improve process performance.

Conradi and Fuggetta examine the SPI, Software Process Improvement [31]. Software process improvement encompasses process assessment, process refinements and process

\(^{14}\) The value of TPS (Transactions Per Second) is based on DoCoMo's system.  
\(^{15}\) One transaction means a pair of messages: receiving and sending.
innovation [31]. Capability Maturity Model/Capability Maturity Model Integration (CMM/CMMI) is a typical example of process platform to maintain and improve software development process.

The process platform concept is essential to improve software productivity and quality, but it is difficult for software firms to apply this concept to new projects. In investigating the projects of DoCoMo, I found that it took more than one year to educate network engineers on process management and apply it to software development. In software development DoCoMo collaborates with other manufacturers such as NEC, NTT Comware and Fujitsu. In the early stage of the network software development (first and second versions) each manufacturer performed poorly because the process platform was not yet established. Required information for the documents, the management process of quality control, the tool version (e.g., Microsoft Office 98 or 2000), and criteria for "improvement" were entirely different in each manufacturer. After establishing the baseline standards, the manufacturers shared the software development information and cooperated with each other.

3.1.5 Global Platform

Global platform is the core standardized foundation. However, in many cases
country-specific conditions and customer preferences are also adapted. In order to reduce the unnecessary conflicts, the International Telecommunications Union (ITU), defines the global standardized specifications. For instance, basic network protocols, SCCP, TC and ISUP, are stipulated in the ITU standards. The goal of the global platform is to position the application support in different locales without modifying the original source code [32]. The 3G mobile technologies are also based on global standards. Even if the method of air access is different (WCDMA or CDMA2000), service entities and basic network interfaces are almost same in 3G networks.

Several problems in global platform are time and modification. It often takes more than one year to fix global specifications and the specifications often continue to change before the final release. Network engineers are required to catch up with the latest version and continually modify the network software.

3.1.6 Product Family and Implementation of Platforms

Utterback, Meyer and Lehnerd [1][26][27] define "family" as individual products that share common technology and address related market applications. According to Simpson [28], a product family is a group of related products that share common features, components, and
subsystems, and satisfy a variety of market niches. A product family comprises a set of
variables, features or components that remain constant from product to product (product
platform) and others that vary from product to product [28]. The distinctive aspects between
individual product variants are the difference in their structure.

Meyer and Lehnerd propose a general framework for product family development (Figure
3-3) that represents a single product family starting with the initial development of a product
platform. This platform is followed by successive major enhancements to the core product and
process technology of that platform, with derivative product development within each generation.
New generations of the product family can be based on either an extension of the product
platform or on an entirely new product platform. In case of an extension, the group of
subsystems and interfaces remains constant.

However, one or more subsystems sometimes must undergo major revision to reduce costs
or to add new features. An entirely new platform emerges only when its basic architecture
changes and aims at value cost leadership and new market applications (Figure 3-4). Systems
and interfaces from prior generations may be carried forward into the new design but are joined
by entirely new subsystems and interfaces [1][29].

Figure 3-3: Product Family Evolution
**Platform Architecture:** Common Subsystems and Interfaces for Multiple Products

![Diagram](image)

**Platform Extensions:** A new generation where number and types of subsystems and interfaces remain constant, but where subsystems and interfaces are enhanced.

![Diagram](image)

**Platform Renewal:** A New Architecture, where subsystems and interfaces from prior generations may be carried forward and combined with new subsystems and interfaces in the new design.

![Diagram](image)


**Figure 3-4: Typology of Platform Change in Product Family Evolution**

3.2 **Platform Innovation Style of Network Software**

**Original platform architecture**

Applying the framework of Meyer and Lehnerd, we can analyze the platform innovation of
network software. If we pick the starting point from the S-Curve, the original platform architecture is described as Figure 3-5. Subsystems S1 and S1' manage the basic call or supplementary service functions respectively. Subsystems S2 and S3 are the common functions in the network software (component standardized). S2 encodes/decodes the message protocols and S3 controls the OS interface. In order to simplify the model, other major functions (e.g., database access, O&M) are not written. P1, P2, and P3 show the services provided by the network software (P here means service (intangible "Product").)

Performance

Figure 3-5: Original Platform of Network Software

Platform extension

Network software evolves as new functions added to the architecture. Maintaining the backward compatibility (P1, P2 and P3), the subsystems are enhanced (S1*, S1'* and S2*). P4,
P5, P6 are the new services provided by S1* and S1**. The number and types of subsystems and major interfaces still remain constant (Figure 3-6). Upgrading the existing services (e.g., an increase in the number of parameters in authentication) and reducing unnecessary functions in the subsystems represent a typical platform extension.

![Diagram of network software architecture]

**Figure 3-6: Platform Extension of Network Software**

**Platform renewal**

Platform renewal occurs when completely new services are added to current platform. New subsystems and new interfaces are created in the architecture. The introduction of 3G services in network software (HLR) illustrates this type of platform renewal. The HLR manages the location management, call handling and the subscriber data function in both 2.5G and 3G networks. In short, the same network software provides different network services (e.g.,
PDC and UMTS services). Keeping the backward compatibility (P1-P6), the software in the HLR integrates the previous platform with the new platform. Compatibility between 2.5G and 3G is necessary, however, this backward compatibility significantly complicates the software architecture (Figure 3-7).

Figure 3-7: Platform Renewal of Network Software
### 3.3 Difficulties in Platform Innovation

A wide variety of the requirements prevents unproblematic platform innovation of network software. Linden and Müller [33][34] illustrate the architectural requirements for software (Figure 3-8). Quality (high quality), cost (low development effort) and delivery (short lead time) are the basic software requirements and these significantly affect other architectural requirements: extensibility, reusability, configurability and so on. Market demands for flexibility, accountability, and robustness also greatly complicate the software architecture. These software requirements indicate that network software development has potential difficulties.

![Diagram of architectural requirements](source)

**Figure 3-8: Requirements for Network Software**

Source: Frank J. van der Linden, Jürgen K. Müller, 1995, Creating Architectures with Building Blocks, IEEE Software 12(6), p.52
3.3.1 Less Extensibility

The existence of a current platform limits the extensibility of the new platform. In the new generation a new platform emerges, but the software architecture is greatly affected by existing architecture. Even if the network architecture, service types, subscriber data and protocols are completely different, the new software architecture forms the similar structural design to previous one (Figure 3-7). In other words, the old standards continue to be supported in the new platform and the best architecture for the new generation services is not created in the new platform.

At the same time, the previous platform limits the resource management of the new platform. The new platform has to be designed to perform within the limited resources (memory capacity and the number of tasks). Since the new architecture is developed based on the old architecture, the new architecture loses both its degree of the freedom and extensibility. In DoCoMo's case, the architecture of the PDC platform (2.5G) significantly affected the UMTS platform (3G). The UMTS network requires higher HLR performance\(^{16}\) (especially in the protocol translation and authentication procedures), but no significant architectural change was conducted.

\(^{16}\) UMTS requires 13.5% to 28.7% higher CPU performance than PDC.
3.3.2 Complex Architecture

As discussed in Chapter 2, network software has to maintain compatibility (1) between the previous and new services and (2) between the various mobile carriers. Backward compatibility is one of the most primary requirements that must be supported in network software. In order to support the previous services/functions and communicate with previous platforms, the interfaces between platforms over time become quite complex. For example, the UMTS protocol must be translated into PDC protocol to transfer the message. The related functions have to hold the interfaces with other functions (Figure 3-9). As the number of supported services increases, the interfaces get more complicated. As a result, the extensibility and testability become lost and platform innovation in network software gets very hard.

![Diagram showing complicated interfaces between different protocols and services.]

Figure 3-9: Complicated Interfaces

56
3.3.3 Scalability (Scaling Problems Caused by Excessive Functions)

Scalability prevents architectural change in network software. Each software development needs at least 100K lines of codes to implement new services (which can take six months to one year) and the accumulated source codes exceed one million of lines. Architectural change will cause potential software defects and requires exhaustive tests (over 10,000 tests in each subsystem). Many network engineers, therefore, hesitate to avoid the radical change in platform.

The total size of the network software always increases because of the new functions (services) and requirements for the compatibility and robustness. For example, if we add a new operation (e.g., location management or call handling), we have to consider numerous semi-normal routes. According to the 3GPP specification [5], the function has to manage the following potential errors: (1) generic error (unexpected value and data missing), (2) identification or numbering problem (unknown subscriber), (3) subscription problem (roaming not allowed), (4) handover problem, (5) call set-up problem (busy subscriber), (6) supplementary services problem (call barred), (7) short message problem (memory capacity exceeded), (8) location services problem (unauthorized requesting network). New services inevitably require huge software development and testing related to the functions.
Furthermore, when the number of services increases, O&M services become even more essential to maintain consistency in the network. New functions must be added to both basic call (or supplementary services) and O&M services. Consequently, new services will enlarge the software size and make it difficult to complete software development in a short period and change the architectural innovation.

3.3.4 Requirements for High Reliability

Network software operates in the real-time telecommunication systems. Critical errors immediately result in disastrous outcomes worldwide. Therefore, reliability is crucial in any new platform. However, compared with the existing platform, the new platform is less stable. Before innovating a platform, network engineers must fix all software problems and improve the software quality. These engineers do not have time to change the platform architecture. Ironically, after fixing these troubles, network engineers also cannot readily innovate the platform architecture. The bug-fixed platform is considered more stable than brand-new platform. Requirement for high reliability limits the architectural platform innovation and results in the low extensibility of the software.
3.3.5 Insufficient Development Time

Market demands for new services and severe competition in the mobile telecommunication market make the development time shorter and shorter. The lead time for platform upgrading is one year in KDDI's system and a half year in DoCoMo's system. In today's market, mobile carriers are trying to shorten the software life cycle from six months to three months. New platform architecture needs sufficient testing to confirm the performance and reliability. Under the traditional development process (i.e., the rigid waterfall model) it is quite challenging to alter the software architecture in such a short time. Insufficient development time prevents platform innovation and renders the platform architecture less flexible.

3.3.6 Imperfect Technical Specifications

3G mobile systems are based on the global specifications issued by 3GPP/3GPP2. However, generally it takes a very long time (up to one year) to fix the specifications. In particular, specifications of 3GPP have about 30 versions/revisions and they constantly change during software development. If there is a modification in the major specifications (such as a list of ASN.1, an operation timer, new parameters), network engineers have to focus on the modification of the network software. In fact, DoCoMo had started the software design based on GSM specifications released in 1998, but the release version was based on UMTS
specifications (June 2000). Network engineers had to catch up with the new requirements and modify the data format and interfaces between modules.

Imperfect specifications prevent software innovation. Network engineers exhaust themselves catching up with the latest specification and consequently they have no time establish the optimized platform for the new version.
4 Migration from Previous Network to New Network

In Chapter 4 I discuss two types of migrations: (1) successful migration (KDDI) and (2) troublesome migration (DoCoMo) from the previous network. KDDI migrated from PDC/cdmaOne to CDMA2000 (KDDI quit PDC services on March 31, 2003) and DoCoMo is making the transition from PDC to UMTS (W-CDMA). PDC subscribers are dominant in DoCoMo's network (93.4% of the DoCoMo's subscribers used PDC as of March 2004). Launching "cdmaOne" and "CDMA2000 1x" in 1998 and 2002 respectively, KDDI succeeded in migrating from the previous network to a brand new network. KDDI closed down PDC services at the end of March 2003 as scheduled. By the same date KDDI achieved the target of seven million CDMA2000 subscribers. During this transition, on the other hand, DoCoMo experienced huge network problems in the development of 3G mobile systems: (1) the introduction of 3G systems, (2) the implementation of global roaming / network dual mode services, and (3) the integration of the SCP function. DoCoMo has been in the process of migrating from PDC to UMTS in its network since 2001 and is still struggling with the migration.

Generally speaking, the easiest transition from 2.5G to 3G is from cdmaOne to
CDMA2000. While CDMA2000 has backward compatibility with cdmaOne because these two systems share the same air interface, the same is not true for PDC/GSM and W-CDMA. It will be necessary to replace all base station equipment when the shift from PDC/GSM to W-CDMA occurs. The more important thing is, however, whether or not mobile carriers have a clear platform-based approach. Discontinuity always exists in the network migration. A clear platform-based approach helps reduce the migration risk and achieve a smooth transition. DoCoMo did not have a clear platform-based approach for its software architecture, while KDDI applied an obvious platform-based approach. In this chapter I analyze the approaches of DoCoMo and KDDI. Especially I focus on the important learning in the problematic migration to glean important lessons. In essence, I found that:

- DoCoMo’s insufficient platform-based approach caused a difficult migration
- KDDI’s success was based on its clear platform-based approach.

Table 4-1: Comparison of Mobile Systems

<table>
<thead>
<tr>
<th></th>
<th>PDC (+ PDC-P)</th>
<th>W-CDMA (UMTS)</th>
<th>cdmaOne (IS-95B)</th>
<th>CDMA2000 (+ EV-DO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation</td>
<td>2.5G</td>
<td>3G</td>
<td>2.5G</td>
<td>3G</td>
</tr>
<tr>
<td>Mobile carriers</td>
<td>DoCoMo/KDDI</td>
<td>DoCoMo</td>
<td>KDDI</td>
<td>KDDI</td>
</tr>
<tr>
<td>Uplink Speed</td>
<td>9.6kbps</td>
<td>64kbps</td>
<td>14.4kbps</td>
<td>64kbps</td>
</tr>
<tr>
<td>Downlink Speed</td>
<td>9.6kbps</td>
<td>384k/Max 2Mbps</td>
<td>64kbps</td>
<td>144k/Max 2.4Mbps</td>
</tr>
<tr>
<td>Major Protocol</td>
<td>PDC-MAP</td>
<td>GSM-MAP(^\text{17})</td>
<td>ANSI-41</td>
<td>ANSI-41</td>
</tr>
</tbody>
</table>

\(^{17}\) In addition, DoCoMo implemented its own specific network operations in the extension area of GSM-MAP.
4.1 Observation of the Discontinuity

During the transition of PDC/UMTS and cdmaOne/CDMA2000 we can observe the discontinuity between 2.5G and 3G. Both transitions need platform innovation in network software to provide new services. Compared with the transition from cdmaOne to CDMA2000, the transition from PDC to W-CDMA (UMTS) is more challenging. Unlike GSM, PDC has completely different network architecture, originating/terminating methods, authentication, and protocols from UMTS's (Table 4-2). The discontinuity between PDC and W-CDMA (UMTS) is larger than that (1) between GSM and UMTS, and much larger than that (2) between cdmaOne and CDMA2000 (Figure 4-1).

The basic concept of the previous network platform cannot be ignored to maintain backward compatibility, but there is no guarantee that the previous platform will perform well in the new network. For example, message sequences and required performance in a new network entirely differ from those in previous network. "Platform renewal" should be considered in order to compromise huge discontinuity. In the network development, DoCoMo intuitively noticed a huge discontinuity, but did not understand how difficult it was to develop the UMTS systems quantitatively. DoCoMo had not experienced to develop GSM systems before. In
addition, global specifications were changing when DoCoMo started its projects. It was quite hard to evaluate the UMTS system quantitatively. Without a detailed investigation, DoCoMo applied the PDC platform to the UMTS platform to reduce the development time (insufficient platform renewal). The insufficient platform-based approach made DoCoMo postpone the original schedule and develop low quality and low extensible network software.

On the other hand, KDDI clearly understood that cdmaOne and CDMA2000 came from a similar product family and that the previous platform could perform well in the new system. Figuring out the differences (Table 4-2), KDDI applied the platform extension approach for new services. The sufficient platform-based approach helped KDDI keep on its original schedule\(^{18}\) and release high quality services acquiring perfect backward compatibility.

- DoCoMo observed the potential huge discontinuity but its investigation of UMTS systems was not enough (insufficient platform renewal).
- KDDI figured out the potential discontinuities and understood that only small modifications were needed (sufficient platform extension).

---

\(^{18}\) Actually KDDI changed the original schedule from fall 2002 to April 2003. However, this change was not based on technological problems. The largest competitor, DoCoMo, changed its original schedule from May to Oct 2003, so KDDI changed its strategy to maximize its profits.
Table 4-2: Required Changes in Network Software

<table>
<thead>
<tr>
<th></th>
<th>PDC to UMTS(^{19})</th>
<th>cdmaOne to CDMA2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network entity (HLR/VLR etc)</td>
<td>Completely new</td>
<td>No change</td>
</tr>
<tr>
<td>Network Protocol</td>
<td>Completely new</td>
<td>Small modification</td>
</tr>
<tr>
<td>Mobility management</td>
<td>Completely new</td>
<td>Small modification</td>
</tr>
<tr>
<td>Call handling</td>
<td>Completely new</td>
<td>Small modification</td>
</tr>
<tr>
<td>Authentication</td>
<td>Completely new</td>
<td>Small modification</td>
</tr>
<tr>
<td>Packet routing</td>
<td>Completely new</td>
<td>Small modification</td>
</tr>
<tr>
<td>O&amp;M</td>
<td>New + Modification</td>
<td>Small modification</td>
</tr>
<tr>
<td>Subscriber data management</td>
<td>New + Modification</td>
<td>Small modification</td>
</tr>
</tbody>
</table>

Figure 4-1: The Discontinuity of Generation in DoCoMo and KDDI Networks

4.2 Transition Paths

Chapter 1 showed the transition paths of DoCoMo and KDDI. While KDDI evolved its network from PDC to CDMA2000 via cdmaOne, DoCoMo drastically established its new

\(^{19}\) DoCoMo also added its own specific network operations to maintain backward compatibility.
network (W-CDMA (UMTS)). While KDDI initiated new services (cdmaOne) in rural areas of Japan, DoCoMo launched its 3G services in urban areas. KDDI’s strategy was based on incremental approach and minimized the initial troubles. As a result, DoCoMo struggled with deploying the 3G mobile services (FOMA), but KDDI changed the network structure smoothly and introduced the new services effectively.

KDDI’s expenditures to keep both PDC and CDMA systems (cdmaOne and CDMA2000) were huge (about $1.5 billion), but KDDI avoided the migration risk of the 3G systems. One important finding is that KDDI spent about four years to introduce CDMA2000 (DoCoMo spent only one year on the field tests). The transition from cdmaOne to CDMA2000 is much easier than the transition from PDC to W-CDMA, but KDDI carefully tested the new system.

Consequently KDDI succeeded in the migration from PDC to CDMA systems in March/April 2002. Over thirteen million KDDI subscribers are now enjoying 3G systems while DoCoMo has three million 3G subscribers (Figure 4-2, Figure 4-3). This success has allowed KDDI to decrease the investment of PDC networks and focus on the next generation data communications (movie, mp3 and GPS). In November 2003, KDDI began to provide the new CDMA2000 method, EV-DO (EVolution-Data Only) technology. EV-DO allows mobile
subscribers to enjoy six times faster data communication than DoCoMo's 3G service (downlink 384 kbps) and 17 times faster communication than previous CDMA2000 (uplink: maximum 144kbps, downlink: maximum 2.4Mbps). At present KDDI is ahead of DoCoMo in terms of data transmission speed.\textsuperscript{20}

The precise transition paths of DoCoMo and KDDI are shown in (Figure 4-4, Figure 4-5). The similarities and differences are as follows.

**Similarities**
- Both systems experienced PDC network.
- The lead-time for 3G systems was about 2.5 years.

**Dissimilarities**
- KDDI had three types of network systems (PDC, cdmaOne and CDMA 2000 1X)
- While DoCoMo took a long time to develop a grand design (investigation of the 3GPP specifications) and coding, KDDI mainly focused on field tests.
- While DoCoMo started the new service in large cities (Tokyo, Osaka etc), KDDI started a new service (cdmaOne) in small cities.
- KDDI quit providing PDC services on March 31, 2003, but DoCoMo continued to provide PDC services. PDC subscribers are still dominant in DoCoMo’s network (93.4%).
- While UMTS is a completely new system for DoCoMo, KDDI accumulated the knowledge of CDMA technology for over four years.

\textsuperscript{20} DoCoMo is now trying to release High Speed Downlink Packet Access (HSDPA) service in 2005. This technology will improve data transmission capability up to 14.2Mbps.
Figure 4-2: Number of Subscribers (DoCoMo)

<table>
<thead>
<tr>
<th>Thousand</th>
<th>1998</th>
<th>1999</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoCoMo</td>
<td>W-CDMA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>89</td>
<td>330</td>
</tr>
<tr>
<td></td>
<td>PDC</td>
<td>17984</td>
<td>23898</td>
<td>29356</td>
<td>36026</td>
<td>40694</td>
<td>43531</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>17984</td>
<td>23898</td>
<td>29356</td>
<td>36026</td>
<td>40783</td>
<td>43861</td>
</tr>
</tbody>
</table>

Figure 4-3: Number of Subscribers (KDDI)

<table>
<thead>
<tr>
<th>Thousand</th>
<th>1998</th>
<th>1999</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDDI</td>
<td>CDMA2000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6806</td>
</tr>
<tr>
<td></td>
<td>cdmaOne</td>
<td>0</td>
<td>541</td>
<td>4838</td>
<td>8277</td>
<td>10822</td>
<td>7208</td>
</tr>
<tr>
<td></td>
<td>PDC</td>
<td>7097</td>
<td>8022</td>
<td>5287</td>
<td>2709</td>
<td>1392</td>
<td>35</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>7097</td>
<td>8563</td>
<td>10125</td>
<td>10986</td>
<td>12214</td>
<td>14049</td>
</tr>
</tbody>
</table>

68
Figure 4-4: Roadmap of New Mobile Systems (DoCoMo)

Figure 4-5: Roadmap of New Mobile Systems (KDDI)

(Data is based on the firms’ Annual Reports and personal interviews. [35][36][37][38])
4.3 Software Development Strategy

Cooperating with other manufacturers such as NEC, NTT Comware and Fujitsu, DoCoMo develops its network software in-house (Figure 4-6). Network software is one of the core competencies of DoCoMo. Designing and building in-house are expected to develop highly flexible software and deliver new software on time. In addition, DoCoMo can better safeguard the confidential information and educate its network engineers by keeping its network development an in-house operation.

On the other hand, KDDI outsources much of its network software development. Software development is a highly technical job. To develop software in-house requires time, skilled network engineers, and expensive programs and equipment. Outsourcing is a cost effective alternative to developing software in-house. Collaborating with Motorola, KDDI outsources its own network software (Figure 4-7).

DoCoMo and KDDI have different software life cycles. The life cycle depends on the development size, but almost every six months DoCoMo upgrades its network software. On the other hand, KDDI develops software once a year. KDDI outsources its software design,
construction, testing (validation) and installation to Motorola, so the life cycle is fixed. The major role of KDDI is to establish technical requirements and conduct field tests, operations, maintenance, and retirement.

![Diagram of Network Software Development of DoCoMo]

**Figure 4-6: Network Software Development of DoCoMo**

![Diagram of Network Software Development of KDDI]

**Figure 4-7: Network Software Development of KDDI**

Basically, both DoCoMo and KDDI have applied the waterfall model in developing their network software. However, DoCoMo's waterfall model is more conservative and rigid. KDDI allows Motorola to apply Motorola's own style: advanced coding techniques, a rigid
process for defining business, technical, and architectural requirements for all development activities [39][40][41].

The waterfall model shown in Figure 4-8 is one of the most popular models for software processes. In the waterfall model, a number of phases are traversed in a linear fashion, i.e., one phase must be completed before the next phase is started [44]. The model is document-driven in the sense that some documents must result from each phase. Based on a review of the documents from each phase, the phase is evaluated and the next phase initiated. It is normally considered easier to control the waterfall model because of its document-driven nature. At the same time, it may be difficult to convince the management to use other dynamic models. In the development of 3G mobile systems, most telecommunications carriers applied this waterfall model.

However, over the past decades, the traditional waterfall life cycle model for software development project planning and management has been challenged [44][45]. The model is criticized for its rigidity and poor ability to deal with unstable and/or uncertain requirements, the late identification and resolution of technical risks, and the accumulation of delays and quality problems until the final test and integration phase. In fact, in 2001, DoCoMo postponed the
commercial release of the 3G systems from May 30 to October 1. One of the main reasons was the modification of 3GPP specifications. Unchanging requirements do not exist in the real world. In other words, requirements often change and evolve, which is known as the "moving target problem". In order to accommodate requirement changes while executing the project based on the waterfall model, organizations are forced to break a software development process. Under the moving target problems the rigid waterfall model is quite difficult to operate (project flexibility is needed).

Table 4-3: Differences in Software Development Strategy

<table>
<thead>
<tr>
<th></th>
<th>DoCoMo</th>
<th>KDDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-house/outsource</td>
<td>Mainly in-house</td>
<td>Mainly outsource</td>
</tr>
<tr>
<td>Partners</td>
<td>Several manufactures Multi-vendors</td>
<td>Motorola</td>
</tr>
<tr>
<td>Product cycle</td>
<td>6-18 months</td>
<td>1 year</td>
</tr>
<tr>
<td>Methodology</td>
<td>Rigid waterfall</td>
<td>Not rigid waterfall</td>
</tr>
<tr>
<td>Flexibility in Development</td>
<td>Extremely low</td>
<td>Middle</td>
</tr>
<tr>
<td>Programming Language</td>
<td>C/C++</td>
<td>C</td>
</tr>
</tbody>
</table>
4.4 Platform Innovation Strategy

Implementing the platform concept can significantly increase the speed of a new product launch and also assist in developing high quality software. By using standardized and tested
components, the accumulated learning and experience also result in higher productivity.

DoCoMo and KDDI experienced different platform innovations in network software. The network concept of the 3G was novel, but KDDI definitely considered 3G systems as a product family of 2G systems (platform extension, Figure 3-6). This platform extension strategy allowed KDDI to operate robust telecommunication systems and maintain backward compatibility. Unlike DoCoMo, KDDI has supported fully backward compatibility between 2G and 3G services since KDDI launched its 3G services.

DoCoMo, on the other hand, tried to apply the platform renewal concept to provide 3G services but did not have a clear platform strategy. It was a critical mission for DoCoMo to launch the world's first 3G mobile services, and consequently DoCoMo overlooked long-term platform thinking. In order to reduce the development time, DoCoMo employed the similar idea of PDC's architecture and reused previous modules. However, UMTS has completely different network entities, message formats, message sequences and required performance. The number of the dynamic steps in PDC and UMTS clearly shows the difference in the required performance (Table 4-4). In order to provide the basic call service, the UMTS system requires from 2.7 to 6 times as high performance as the PDC system.
Table 4-4: The Number of Dynamic Steps in PDC and UMTS (HLR)

<table>
<thead>
<tr>
<th></th>
<th>PDC</th>
<th>UMTS</th>
<th>Difference (UMTS/PDC)</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location management</td>
<td>15404</td>
<td>91618</td>
<td>5.95</td>
<td>UMTS needs several message sequences in location management.</td>
</tr>
<tr>
<td>Authentication</td>
<td>32704</td>
<td>171125</td>
<td>5.23</td>
<td>PDC includes originating call function.</td>
</tr>
<tr>
<td>Terminating call</td>
<td>24794</td>
<td>66992</td>
<td>2.70</td>
<td>UMTS needs several message sequences in terminating call.</td>
</tr>
</tbody>
</table>

The data is the average number of dynamic steps in 2nd and 3rd versions (DoCoMo).

New platform architecture should have been applied to the UMTS platform, but DoCoMo's new platform formed a combination of platform extension and platform renewal. Platform innovation was insufficient. The software architecture/interfaces thus became complex and the new platform lost the extensibility (Figure 4-9).

- The platform extension strategy allowed KDDI to cover perfect backward compatibility.
- The short development time prevented DoCoMo from concentrating on the platform-based approach (insufficient platform renewal).
- DoCoMo's platform architecture did not fully bridge the differences between PDC and UMTS.
- DoCoMo's platform architecture became cumbersome and complex.

4.5 Analysis of Major Problems in DoCoMo's Platform Innovation

Analyzing a problematic case in network software development, we can discover important findings. Insufficient platform-based approach led DoCoMo to experience a troublesome migration. Migration from PDC to UMTS has been in progress since 2001 and
DoCoMo is still struggling with the platform innovation to expand the functionality and system performance (speed). Major problems in DoCoMo's platform innovation have resulted from (1) complex platform architecture (original platform and after introducing dual mode services), (2) reuse of the low quality platform software, and (3) scalability/feature-interaction problems.

4.5.1 Complex Platform Architecture

Investigating the software platform architecture of DoCoMo, we can find various interfaces among subsystems, especially between PDC and UMTS. The subsystems are not independent in the new platform and often interfere with each other (Figure 4-9).

During the migration from 2G to 3G, the functions of both PDC and UMTS improved. One change in a certain subsystem will affect all other subsystems in another platform (PDC or UMTS). Network engineers have to consider the influence of the modifications and mitigate the potential problems. A combination of the complex architecture and the new platform that lacked a degree of freedom, resulted in low-quality software and prompted DoCoMo to postpone the official release date from March 30 to October 1, 2001.
Figure 4-9: The New Platform of the DoCoMo's Network Software (HLR)

Exploring the details of the new UMTS platform (Figure 4-10), we can discover other problems. In the new platform (UMTS) we can find many unnecessary and overlapping functions. First, there are two protocol functions (Protocol 1 and 2) in the UMTS platform, but these two functions provide similar functions. The major difference is that S5 provides encode/decode functions (ASN.1 converter) while S5' offers a data distribution function to basic call and supplementary services. Second, we can find similar temporary data tables (T1, T1' and T1''). The message sequences of 3G have more patterns (message continuation, message
segmentation, etc.) than those of PDC. The temporary data table is essential to store the message status. However, each T1, T1' and T1" has quite similar data items (mobile subscriber number, message status and timer) and they operate similarly. In addition, T1" has two access interfaces (S5'-T1" and S5-T1"). Management of the data consistency in T1" became quite challenging. These data tables need an audit function (data checking function) to maintain data consistency. Audit functions (A1, A1' and A1") also perform similarly. These functions can be integrated into the new platform. Unnecessary functions required DoCoMo to design various modules, develop large programming codes, and test all routes. Consequently the complex architecture made it difficult to maintain quality and the original schedule.\(^{21}\)

\[
\text{Figure 4-10: The Complex Platform Architecture of the UMTS Platform}
\]

\(^{21}\) This inefficient structure was modified in the 3rd version of the network software. It took over one year to integrate T1, T1' and T1", and A1, A1' and A".

79
4.5.2 Dual Mode Service between PDC and UMTS

After introducing 3G services in October, 2001, DoCoMo suffered from deploying UMTS networks. Subscribers could not use the new handsets in rural area. The number of 3G subscribers was still low (89,000 in March 2002 and 114,500 in June, 2002). At the same period, KDDI increased the number of subscribers to 1,151,300. In order to improve the coverage, on July 1, 2002, DoCoMo introduced a network dual service\textsuperscript{22}, which was expected to offer both PDC and UMTS services and maintain fully backward compatibility throughout the whole region. DoCoMo anticipated that the dual service would help increase the number of 3G subscribers. However, this new service did not contribute to the rapid increase in subscribers\textsuperscript{23} (Figure 4-11), but left the more complex platform architecture.

\textsuperscript{22} Subscribers need two handsets for PDC and UMTS. No integrated handset is provided.
\textsuperscript{23} The number of 3G subscribers of DoCoMo increased after June 2003 mainly due to improvements in the handsets (long time battery, smaller size and lighter weight).
The dual mode service requires more interactions between PDC and UMTS platform. Activation process is generated from either the PDC or UMTS network and the service needs new message sequences and subscriber data. Furthermore, there is no global specification about the dual mode service. DoCoMo had to develop the service from the scratch.

In order to implement network dual service, and especially manage the interactions between PDC and UMTS, DoCoMo decided to add one subsystem to the platform (S8). However, DoCoMo did not change original platform architecture to limit the impact. The
platform architecture became much more complex (Figure 4-12). Complex architecture itself is not so terrible if the services and required performance remain unchanged. Unlike PDC, UMTS supports the ASN.1 compiler to encode/decode the message, authentication process (authentication vector creation), and message segmentation procedures. As the number of UMTS subscribers increases, system performance decreases and it will prove difficult to maintain the required performance (processing speed). DoCoMo's platform thus has a critical bottle neck in future services.

![Diagram of the More Complex Platform Architecture of the UMTS Platform](image)

**Figure 4-12: The More Complex Platform Architecture of the UMTS Platform**

*(Dual Mode)*
4.5.3 Reuse of Low-Quality Platform Software

The new platform must be developed on a continuous platform and developers should reuse previously established modules to help shorten the development time. Although software reuse has possible architectural mismatch [41][42], software reuse clearly has the potentials to improve productivity and hence reduce costs. It also improves the quality of software systems. According to NATO's standard for software reuse procedures [46], software reuse has five major advantages: (1) productivity improvement, (2) reduced maintenance cost, (3) improved interoperability, (4) support for rapid prototyping, and (5) reduced training costs. Especially DoCoMo expected the productivity improvement in software reuse. Basically, systems built from reusable parts have the potential for improved performance and reliability because the reusable parts can be greatly optimized and be more stable.

DoCoMo reused its software on a large scale (over 100 KL) to develop new platform. A certain subsystem in the new platform was made from 80% of previous software modules (Table 4-5). Reuse of large-scale components, whether the architectures only or the implementation of those architectures in code, can greatly benefit the project. However, if not done carefully, it can also greatly increase risk. The problem is not the reuse of the modules, but the reuse of the
low quality software. Ironically the reused modules had more software defects than brand-new modules. The 80% of Protocol B was reused part and this subsystem has highest defect rate (0.751/KL). This unexpected situation caused the projects to fall behind schedule. The important learning is that new platform should be established from a stable, "well-debugged" platform.

Table 4-5: The Relationship between Reuse Rate and Defect Rate (VLR/MSC/HLR)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Reuse rate %</th>
<th>Defect rate (Number/KL)</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol A</td>
<td>50.4%</td>
<td>0.478</td>
<td>System test</td>
</tr>
<tr>
<td>Protocol B</td>
<td>81.4%</td>
<td>0.751</td>
<td>System test</td>
</tr>
<tr>
<td>Protocol C</td>
<td>51.3%</td>
<td>0.406</td>
<td>System test</td>
</tr>
<tr>
<td>Protocol D</td>
<td>0%</td>
<td>0.245</td>
<td>System test</td>
</tr>
<tr>
<td>Location Management A</td>
<td>0%</td>
<td>0.103</td>
<td>System test</td>
</tr>
<tr>
<td>Basic call A</td>
<td>0%</td>
<td>0.349</td>
<td>System test</td>
</tr>
<tr>
<td>Operation &amp; maintenance A</td>
<td>0%</td>
<td>0.213</td>
<td>System test</td>
</tr>
</tbody>
</table>

4.5.4 Scalability/Feature-Interaction Problems

Many features are simple and it is not so difficult to detect interactions between a particular pair of them. The problem becomes difficult when network carriers want to accelerate rapid addition of many new features [9][17][47][48]. Every new feature can change the behavior of pre-existing features or even "break" these features, crashing the system. In the telecommunications industry, Griffeth and Lin [9] call this "the feature-interaction problem."
When we introduce new functions that require platform innovation, this feature-interaction problem inevitably occurs. The implementation of global roaming service and the integration of the SCP is a typical case of the feature-interaction problem. The implementation of these new services caused dropped calls, no answers, and disconnections in the DoCoMo's system.

Global roaming service allows subscribers traveling to foreign countries to enjoy mobile communications services with the same device (handset or small card). The SCP provides value-added services such as auto-answering, call forwarding and call reject services. Usually the SCP is separated from the HLR (Figure 1-5), but the network entity can be integrated with the HLR (Network carriers often integrate these entities. For example, the Authentication Center (AuC) is often integrated with the HLR). The functions are not so difficult because global roaming service is based on existing functions and the value-added services provided by the SCP are well known (i.e., specifications are clear). However, rapid service creation led to low quality software and lower performance system.

The following architecture (Figure 4-13) shows the new platform after DoCoMo implemented global roaming and the SCP functions. New services were implemented as a platform extension (improvement of the basic call function, S4*) and platform renewal (addition
to subsystem S4", S5', S6') respectively. The new subsystems, S4", S5' and S6', seem brand-new, but the architecture is quite similar to S4* (basic call functions), S5 (protocol) and S6 (OS interface). These new subsystems are considered variants of an existing platform (product platform). On the basis of the existing subsystems it was not very difficult to implement new functions, but every new feature has the potential to change the behavior of pre-existing features.

The following architecture suggests that one critical defect in the subsystem widely influences other subsystems. For example, the critical defect in S4* terminates not only S4* function but also many related functions. A critical defect might cause the new system to shut down.

Figure 4-13: New Platform Architecture: Global Roaming and Integration of the SCP
After crossing the huge discontinuity (releasing the first version of the software), DoCoMo implemented these new features every one year. Here is the example of one project of DoCoMo (Figure 4-14, Figure 4-15). Term 1 represents the initial development and terms 3,5 represent the development of global roaming and the SCP functions respectively. The development size reached about 50-60 % of the initial development. During this period there was no significant reconstruction in software platform architecture. Without fixing the platform architecture, DoCoMo continued to introduce new functions. Rapid creation of new services resulted in underestimation of the development size and low software quality (Table 4-6). The original (estimated) development size of the SCP function A was 46.7 KL, but the actual development reached 104.5 KL. The defect rate (2.3/KL) is much higher than that of Location Management A (0.8/KL). Rapid implementation of these new services also caused the system performance problem. The UMTS system requires high CPU performance. Under the high-load conditions (600TPS) UMTS needs 13.5% to 28.7% higher CPU performance (Table 4-7). Although DoCoMo improved the redundant modules in subsystems in version 3, architectural change in platform has not been carried out. Since CPU usage is growing 1% per year, if DoCoMo continues to implement new services, the UMTS system will reach the system limitation (CPU: over 70%) within five years.
Figure 4-14: The Development Size of Software (Example of One Project)

Figure 4-15: The Accumulated Development Size of Software (Example of One Project)
Table 4-6: Low Quality Due to Rapid Creation

<table>
<thead>
<tr>
<th></th>
<th>Plan (KL)</th>
<th>Actual development time</th>
<th>Defect rate (Number/KL)</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCP function A(^{24})</td>
<td>46.7</td>
<td>6 months</td>
<td>2.3</td>
<td>Integration test(^{25}) (product verification)</td>
</tr>
<tr>
<td></td>
<td>104.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location management A(^{26})</td>
<td>3.5</td>
<td>6 months</td>
<td>0.8</td>
<td>Integration test (product verification)</td>
</tr>
<tr>
<td></td>
<td>4.8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4-7: CPU Usage under the High-Load Conditions (%)

<table>
<thead>
<tr>
<th>Version</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMTS600TPS</td>
<td>70.7</td>
<td>67.7</td>
<td>62.0</td>
<td>62.1</td>
<td>64.0</td>
<td>65.3</td>
</tr>
<tr>
<td>PDC600TPS</td>
<td>n.a.</td>
<td>39</td>
<td>44.3</td>
<td>48.6</td>
<td>43.7</td>
<td>43.9</td>
</tr>
<tr>
<td>Difference</td>
<td>n.a.</td>
<td>28.7</td>
<td>17.7</td>
<td>13.5</td>
<td>20.3</td>
<td>21.4</td>
</tr>
</tbody>
</table>

DoCoMo improved the redundant modules in subsystems in version 3.

---

\(^{24}\) This is a new function.

\(^{25}\) The "integration test" runs between the unit test and system test phases.

\(^{26}\) This is the existing function (modifications were applied to implement new services).
5 Management of the Future Telecommunication Software Development

Although 3G networks have yet to become commercial, the industry is already laying the groundwork for future wireless technology currently known as 4G/5G. The goal of 4G/5G wireless technology is to replace the current proliferation of core networks with a single worldwide cellular core network standard based on IP for control, video, packet data, and VoIP. The 4G/5G wireless technology contains an all-IP based access and core with effective management of QoS (Quality of Service) over IP. IPv6 will likely replace IPv4 since it is better adapted to mobile networks in addressing capacity, multicast management, security mechanisms, QoS mechanism and mobility management. The concepts of the new network are not defined clearly, but the network architecture will be completely different from the existing 3G networks (Table 5-1). Platform network software will play a significant role in providing 4G/5G services.

Based on the lessons learned from the disruptive innovations in 3G mobile systems, Chapter 5 suggests how incumbent mobile carriers can avoid the potential threats and develop platform network software in future disruptive network innovations.
Table 5-1: Required Changes in Network Software to 4G

<table>
<thead>
<tr>
<th></th>
<th>UMTS to 4G network</th>
<th>CDMA2000 to 4G network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network entity (HLR/VLR etc)</td>
<td>Almost same</td>
<td>Almost same</td>
</tr>
<tr>
<td>Network Protocol</td>
<td>New</td>
<td>New</td>
</tr>
<tr>
<td>Mobility management</td>
<td>Modification</td>
<td>Modification</td>
</tr>
<tr>
<td>Call handling</td>
<td>Modification</td>
<td>Modification</td>
</tr>
<tr>
<td>Authentication</td>
<td>Modification</td>
<td>Modification</td>
</tr>
<tr>
<td>Packet routing</td>
<td>New + Modification</td>
<td>New + Modification</td>
</tr>
<tr>
<td>O&amp;M</td>
<td>New + Modification</td>
<td>New + Modification</td>
</tr>
<tr>
<td>Subscriber data management</td>
<td>Modification</td>
<td>Modification</td>
</tr>
</tbody>
</table>

5.1 Important Learning from the Migration (2.5G/3G)

DoCoMo and KDDI's cases provide important lessons: (1) incremental innovation is effective (small change, rural area to urban area), (2) new projects should be flexible (requirements often change), (3) platform architecture should be simple (less interfaces), (4) unnecessary interactions between previous and new platforms should be reduced, (5) only stable (fully debugged) software should be reused, (6) reconstruction of the software platform is quite necessary.

Incremental Innovation

Migration from previous network inevitably causes the discontinuity. However, applying the incremental innovations in existing network, mobile carriers can minimize the discontinuity.
Incremental innovation has two dimensions: (1) experience the intermediate generation (e.g., 2.5G, 3.5G and 4.5G) and (2) implement the system little by little (trial and error in small towns/rural areas). It will need additional investment to implement intermediate generation, but the strategy helps mobile carriers avoid migration risk. Field tests in rural area also allow mobile carriers to detect unexpected errors and minimize the influence.

**Flexible Development**

It takes long time for ITU-T to release the global specifications. In addition, these specifications are often modified after published. The rigid waterfall methodology cannot follow the new changes in global specifications. In order to maintain high software quality, waterfall model is effective, but the project should allow a degree of freedom in the development processes. Unlike outsourcing, in-house development has more potential to be flexible. In-house development of the network software is effective to enhance the flexibility.

**Simple Platform**

In order to reduce the software defects, increase productivity and improve reliability, the new platform should be simple. It is essential to minimize the interfaces between the previous and new platforms. In addition, similar and unnecessary functions should be integrated. This
integration will help increase the memory capacity of the system and improve the system performance.

Minimize Interaction between Previous and New Platforms

Managing the interaction between previous and new platforms is quite challenging. The interaction unites the two platforms firmly and unnecessary functions degrade the performance of the system. If the new service requirements and message protocols are completely different from the previous ones, interactions between the previous and new platforms should be limited. This indicates that a network dual mode can become a potential threat in future network software. When introducing a network dual service, mobile carriers should apply platform extension (not platform renewal) to limit the influence. At least mobile carriers should avoid implementing huge discontinuity caused by the dual modes such as PDC/UMTS.

Reuse of the Stable Software Platform

When utilizing the previous software, mobile carriers have to reuse stable software. The reuse of unreliable software will increase the risk and render the project chaotic. If mobile carriers apply software reuse on a large scale, software specifications and design as well as software codes are essential. The results of testing process and other users/developers manuals
also help software reuse.

**Reconstruction of the Platform Software**

Complex architecture has less flexibility and prevents platform innovation. The gap between expectation and reality will become larger in the complex architecture (Figure 5-1). Mobile carriers sometimes need to review the software architecture and reconstruct the platform. The main purpose of the restructuring is to make the architecture robust, maximize the performance and improve the flexibility.

![Graph showing expectation and reality over time](image)

**Figure 5-1: The Gap Created by Complex Architecture**

5.2 **Strategies in Each Phase**

In order to develop new services in a short time, keep software quality, overcome
performance limitation, and avoid the migration risk, network carriers should concentrate on the following platform software strategies.

- The architecture should be simple (avoid complex architecture).
- Reduce the number of the interfaces among subsystems
- Reduce the interactions between previous and new platform software.
- Reconstruct the platform architecture in the earlier phases (In the later phase it becomes extremely difficult to change the platform architecture).

The S-curve theory helps us determine the platform strategies in each phase. In this S-curve we can consider four phases of the network software innovation: (1) initial phase, (2) early development phase, (3) expansion phase and (4) saturation phase (Figure 5-2).

![Figure 5-2: Phases of Platform Innovation](image)
Basically, the first version and second version can be considered in the initial phase. The third version and fourth version can be considered in the early development phase. After the fifth version, it depends on the situation. If the development size of the network software continues to shrink, we can consider saturation phase has started.

**Initial Phase (First Version and Second Version)**

In the initial phase mobile carriers should pursue two actions: (1) fix the current software defects and (2) investigate the architectural problems in the current network software.

Platform architecture in the initial phase has a wide variety of defects. Sometimes these defects cause serious problems such as dropped calls and disruptions of customer data. Therefore, first, mobile carriers have to fix the software defects. At the same time, mobile carriers should investigate the current platform architecture and identify unnecessary functions, interfaces and subsystems. Bug fixing requires much time and energy, so it becomes quite challenging to simultaneously balance bug fixing efforts and investigation of the current platform. However, these two actions are essential to avoid future problems (resource limitation, low performance and migration problems). Identifying the architectural problems, mobile carriers
should decide on a future platform strategy and this should be based on whether or not the current platform can survive in the future. What it must do is:

- Fix the current software defects.
- Investigate the architectural problems in current network software.

**Early Development Phase (Third Version and Fourth Version)**

The early development phase is quite important because this phase will direct the potentials of the network software. Crucial steps are (1) check the software quality in the current platform, (2) select the new services (avoid introducing so many functions), and (3) apply the platform renewal.

Mobile carriers should pay attention to the quality of the current platform software. If the software architecture has critical problems in initial phase, mobile carriers should avoid implementing too many new services, especially new services that require interaction between platforms (e.g., dual mode service). In the imperfect platform, new functions inevitably cause the chaotic architecture and limit the extensibility. The interactions between previous and current platforms also cause software defects and lower the system performance. After fixing the architectural problems, mobile carriers should introduce new services (value-added service, dual mode service). Platform renewal should be done in this phase and should include these
steps:

- Check the software quality in the current platform.
- Avoid introducing too many new functions.
- Apply the platform renewal strategy.

**Expansion Phase (Depending on the Situation)**

In the expansion phase, mobile carriers should concentrate on small modifications of the platform: (1) improve the current subsystems and (2) try to avoid creating interfaces. If mobile carriers would like to change the platform architecture not radically but moderately, they should apply the concept of "product platform." The existing platform, providing services over one year, allows new platform to maintain certain quality. Thus, the new platform should be based on the existing platform.

It becomes quite difficult to change the software architecture in the expansion phase. Even if the current platform has architectural problems, the platform has been working since implemented. In many cases software architecture is already complex, so the architectural change has huge risk to cause the serious problems. Therefore, making small modifications is the most effective way to implement changes in this phase. Complex platform architecture does not have extensibility and flexibility. Furthermore, this architecture often leads a small trouble
to trigger serious problems in other subsystems. Under the complex platform architecture, mobile carriers should pay careful attention to feature-interaction problems because every new feature can change the behavior of pre-existing features or even "break" these features, crashing the system. Platform extension strategy is effective in the expansion phase.

At the same time, we can expect the process innovation ("process platform") in this phase. Accumulated knowledge helps mobile carriers manage the software development well: higher productivity and better quality. Mobile carriers can make software development process significantly sophisticated in this phase.

- Small modifications are better to maintain platform consistency.
- If architectural change is needed, the idea of "product platform" can be applied to the new platform.
- Under the complex platform architecture mobile carriers should pay careful attention to feature-interaction problems.
- We can expect process innovation in this phase.

**Saturation Phase (Software Development Size Continues to Shrink)**

Platform innovation is quite limited in this phase. Thus, mobile carriers should prepare for the next migration and focus on these two steps: (1) complete the migration from the previous network (i.e., mobile carriers should avoid maintaining three network platforms simultaneously.), (2) modularize the functions and improve the software quality of the current subsystems.
Generally speaking, network migration often takes over five years. If the speed of network innovations is faster than migration speed, mobile carriers will implement new networks before finishing previous migration. However, this strategy is quite risky because network software has to cover two discontinuities (Figure 5-3). Migration risk caused by backward compatibility and complexity will double.

![Graph showing discontinuities among three network generations](image)

**Figure 5-3: Discontinuities among Three Network Generations**

In the saturation phase mobile carriers should focus on software modularization and improvement of the software quality. Platform architecture may be renewal in the next generation, but the new platform will reuse numerous existing modules to increase productivity. Reliable modules are quite essential. Lessons learned:
- Complete the migration from previous network.
- Modularize the functions and improve the software quality of the current subsystems.
6 Conclusion

Network migration inevitably occurs in the near future. Disruptive innovations in network systems are forcing mobile carriers to manage difficult network software development. Within the limited development time, mobile carriers are required to develop high quality and robust network software on a large scale, and accomplish smooth transition/upgrading. In addition, mobile carriers have to maintain backward compatibility. It is becoming more essential to overcome the discontinuities not only between generations but also in the same generation.\footnote{As discussed section 2.2, discontinuity exists in the same generation.} Network migration from previous network to new network takes at least five years. Mobile carriers have to manage the network innovations (discontinuities) for many years.

The ultimate goal of this thesis is to discover how incumbent mobile carriers can avoid the potential threats and develop the platform network software in future disruptive network innovations. Telecommunications is a basic, necessary service for human activity today, for businesses, organizations and for individuals. Unsuccessful launch and troublesome upgrading mean not only a critical failure in telecommunication business but also a loss of social benefits.
The cases of DoCoMo and KDDI provide very important lessons. The most essential learning is how the platform-based approach is crucial for successful migration. We found that (1) insufficient platform-based approach led DoCoMo to difficult migration, and (2) clear platform-based approach allowed KDDI to enjoy successful (relatively problem free) migration. DoCoMo observed the huge discontinuity between generations but the investigation of UMTS systems was not enough. Consequently insufficient platform renewal was applied. KDDI figured out the discontinuities and understood that only small modifications were needed (i.e., sufficient platform extension). The sufficient platform-based approach allowed KDDI to maintain the original schedule and release the high quality services with perfect backward compatibility.

In order to launch the first 3G services worldwide, DoCoMo concentrated simply on productivity. Platform renewal was not enough and complex platform architecture emerged. In addition, the reuse of low quality platform software made the new platform unreliable. Eventually project management became chaotic and the original schedule was postponed. After the implementation of 3G services, DoCoMo still has not reconstructed the basic platform architecture. If we consider additional new services and the rapid increase in 3G mobile subscribers, DoCoMo’s platform will reach the performance limitation in the near future.
Summarizing the DoCoMo and KDDI's migration cases, we can point out six important findings: (1) incremental innovation is effective (small change, rural area to urban area), (2) new projects should be flexible (requirements often change), (3) platform architecture should be simple (less interfaces), (4) unnecessary interaction between previous and new platform should be reduced, (5) only stable (fully debugged) software should be reused, (6) reconstruction of the software platform is quite necessary. Disruptive innovations in network systems will create huge discontinuities, but these approaches will help reduce the gaps and implement new services/functions without critical defects.

Mobile carriers can apply platform innovations in each development phase: (1) initial phase, (2) early development phase, (3) expansion phase and (4) saturation phase. Basically "platform renewal" and "platform extension" should be applied before the early phase and after the expansion phase respectively.

In the initial phase mobile carriers have to fix the current software defects of the new platform. At the same time it is necessary to investigate the architectural problems in the current network software. Identifying the architectural problems, mobile carriers should decide
the future platform strategy: whether or not the current platform can "survive" in the future. The early development phase is quite important because this phase will direct the potentials of the network software. In the early stage mobile carriers should apply the "platform renewal" strategy if the current platform architecture is not appropriate. These mobile carriers should limit the implementation of new functions to complete the sufficient platform renewal.

In the expansion phase, small modifications ("platform extension") are better to maintain platform consistency. However, if architectural change is required, the concept of "product platform" can be applied to the new platform (e.g., the basic call function can be applied to the development of the SCP function). In this phase even small modifications often break the pre-existing features, so mobile carriers have to pay careful attention to feature-interaction problems. In the saturation phase mobile carriers have to complete the migration from the previous network and prepare for the future migration. Modularization of the functions and improvement of the current platform quality are essential. Mobile carriers can continue to innovate the platform based on the outcomes of each phase.

History repeats itself. Similar drastic network innovations to 2.5G/3G's will occur in the future. Management of telecommunication systems design and development will face more
challenging and more complicated migration. However, the learning from DoCoMo and KDDI (the six key points) suggests how mobile carriers can react under the tough situations. The platform-based approach definitely will help mobile carriers develop successful migration strategies and reduce the potential threats in current and future network systems. Telecommunication is a fundamental service for human life. The successful platform-based approach will deliver not only economic value to mobile carriers but also social benefits to society.
Appendix: Abbreviations and Terminologies

1G: the First Generation Mobile Technologies
2G: the Second Generation Mobile Technologies
3G: the Third Generation Mobile Technologies
3GPP: The Third Generation Partnership Project (UMTS Network)
3GPP2: The Third Generation Partnership Project 2 (CDMA2000 Network)
4G: the Forth Generation Mobile Technologies
5G: the Fifth Generation Mobile Technologies
AMPS: Advanced Mobile Phone System
AuC: Authentication Center
bps: bits per second
CDMA: Code Division Multiple Access
CN: Core Network
DB: Database
EDGE: Enhanced Data GSM Environment
EV-DO: Evolution Data Only
EV-DV: Evolution Data Voice
FDMA: Frequency Division Multiple Access
FOMA: Freedom Of Mobile multimedia Access
GGSN: Gateway GPRS Support Node
GMSC: Gateway Mobile Switching Center
GPRS: General Packet Radio Service
GSM: Global System for Mobile Communications
GSM-MAP: Global System for Mobile Communications–Mobile Application Part
HLR: Home Location Register
HSDPA: High Speed Downlink Packet Access
IMT-2000: International Mobile Telecommunications 2000
IS-95: Interim Standard 95
ISDN: Integrated Services Digital Network
ISUP: Integrated Services Digital Network User Part
ITU: International Telecommunication Union
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITU-T</td>
<td>The ITU Telecommunication Standardization Sector</td>
</tr>
<tr>
<td>Kbps</td>
<td>Kilobits per second</td>
</tr>
<tr>
<td>MAP</td>
<td>Mobile Application Part</td>
</tr>
<tr>
<td>MC-CDMA</td>
<td>MultiCarrier-Code Division Multiple Access</td>
</tr>
<tr>
<td>MSC</td>
<td>Mobile services Switching Center</td>
</tr>
<tr>
<td>O&amp;M</td>
<td>Operation and Maintenance</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>PDC</td>
<td>Personal Digital Cellular</td>
</tr>
<tr>
<td>PDC-P</td>
<td>Personal Digital Cellular Packet</td>
</tr>
<tr>
<td>PLMN</td>
<td>Public Land Mobile Network</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public Switched Telephone Network</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RNC</td>
<td>Radio Network Controller</td>
</tr>
<tr>
<td>SCCP</td>
<td>Signaling Connection Control Part</td>
</tr>
<tr>
<td>SCP</td>
<td>Service Control Point</td>
</tr>
<tr>
<td>SGSN</td>
<td>Serving GPRS Support Node</td>
</tr>
<tr>
<td>TACS</td>
<td>Total Access Communication System</td>
</tr>
<tr>
<td>TC</td>
<td>Transaction Capabilities</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>VLR</td>
<td>Visitor Location Register</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>W-CDMA</td>
<td>Wideband CDMA</td>
</tr>
</tbody>
</table>
Bibliography


109


[32] Wim van Vuuren and Johannes I.M. Halman, 2001, Platform driven development of


[38] KDDI Corporation Annual Report 2000-2003


[40] Z. Xu, N.D. Samaan, 2001, An Adaptive Neural Network Based Technique to Enhance Software System Assurance for Regression Testing, Motorola


[43] Nancy S. Eickelmann and Allan L. Willey, 2002, Traceability and Reproducability in Integrated System Test Environments, Motorola


[46] NATO COMMUNICATIONS AND INFORMATION SYSTEMS AGENCY, NATO STANDARD FOR SOFTWARE REUSE PROCEDURES, Volume 3, pp 2-5


[50] Sarah Disch and Robert Nicodemus, 2002, 3G or 4G: That is the Question.,


