**T. C.**
**BAHÇEŞEHİR UNIVERSITY**

# DESIGN AND DEVELOPMENT OF COMMON COMMUNICATION PLATFORM FOR NETWORK COMPONENTS

**M.S. Thesis**

**MELİKE YİĞİT**

**ISTANBUL, 2012**

**THE REPUBLIC OF TURKEY**
**BAHÇEŞEHİR UNIVERSITY**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**
**COMPUTER ENGINEERING**

# DESIGN AND DEVELOPMENT OF COMMON COMMUNICATION PLATFORM FOR NETWORK COMPONENTS

**M.S. Thesis**

**MELİKE YİĞİT**

**Supervisor: Asst. Prof. Vehbi Çağrı GÜNGÖR**

**Co-Advisor: Assoc. Prof. Taşkın KOÇAK**

**ISTANBUL, 2012**

**T.C**
**BAHÇEŞEHİR ÜNİVERSİTESİ**
**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**
**COMPUTER ENGINEERING**

Name of the thesis: Design and Development of Common Communication Platform for Network Components
Name/Last Name of the Student: Melike YİĞİT
Date of the Defense of Thesis: 25.01.2012

The thesis has been approved by the Graduate School of Graduate School of Natural and Applied Sciences.

Assoc. Prof. Dr., Tunç BOZBURA
Graduate School Director
Signature

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr., V. Çağrı GÜNGÖR
Program Coordinator
Signature

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Arts.

| Examining Comittee Members | Signature |
|---|---|
| Thesis Supervisor<br>Asst. Prof. Dr., V. Çağrı GÜNGÖR | ---------- |
| Member<br>Prof. Dr., Emin TACER | ---------- |
| Member<br>Asst. Prof. Dr., Selçuk BAKTIR | ---------- |

# ACKNOWLEDGEMENTS

First of all I would like to thank Asst. Prof. Dr. Vehbi Çağrı GÜNGÖR and Assoc. Prof. Dr. Taşkın KOÇAK, who has given me the opportunity to work on this thesis. I'm very grateful for their support, insight, and invaluable help during the preparation of this thesis.

I would also like to thank my lecturers who encouraged me during my master program, and on my master thesis.

My special thanks to my friends and also colleagues, especially to Muhammet MACİT, Ertunç ERDİL, and Mustafa KAPDAN for their endless support all through this work, also in my personal life and master's degree courses.

Last but not the least; I would like to express my love and gratitude to my family.

January 25, 2012 Melike YİĞİT

# ABSTRACT

DESIGN AND DEVELOPMENT OF COMMON COMMUNICATION PLATFORM FOR NETWORK COMPONENTS

Yiğit, Melike

Computer Engineering Master Program

Supervisor: Asst. Prof. Dr. Vehbi Çağrı Güngör

Co-Advisor: Assoc. Prof. Dr. Taşkın Koçak

January 2012, 89 pages

Multi-vendor telecommunication networks in a typical service provider environment are managed using multiple proprietary User Management Systems (UMS) supplied by the Operational Support System (OSS) vendors. The management of a typical service provider includes communication solutions between the Global UMS (UMS-G) and the Local UMS (UMS-L). Nowadays, in service provider environments, there exist OSSs, which use multi-vendor communication protocols.

In telecommunication sector, the centralized management of all these different OSSs causes serious problems in the network operation. In this respect, there is an urgent need for a standardized and centralized provisioning and auditing mechanism for the operators and their entitlements, working on these management systems.

To address this need and to provide efficient operation among different service provider network components, in this thesis a TMF615 standard-based common communication platform is designed and developed. In this respect, the proposed approach includes a common interface to address communication problems in multi-vendor service provider environments. The developed interface and performance evaluations are one of the first solutions in this field and the resulting solutions is converted to a commercial product, which is the Alcatel-Lucent Access Management System (AMS5520), with high added value. In this regard, it is expected that the proposed approach makes important contributions to scientific literature and commercial applications.

The realization of the proposed TMF615 standard-based interface enables efficient and easy integration of existing and new Operational Support Systems of the service providers. In this way, a standardized interface is offered and a common communication platform is adequate for all different systems. Therefore, the vendors are only responsible from application development based on specifications and a standardized communication is introduced for all related systems. This significantly facilitates the management of service providers, a system performance is improved,

and a massive cost reduction is provided at the same time. Consequently, an efficient management of network components is provided using a common standardized interface.

**Keywords:** Operational Support System (OSS), User Management System (UMS), Tele-Management Forum (TMF615)

# ÖZET


AĞ ELEMANLARI İÇİN ORTAK İLETİŞİM PLATFORMU TASARIMI VE GELİŞTİRİLMESİ

Yiğit, Melike

Bilgisayar Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Yrd. Doç. Dr. Vehbi Çağrı Güngör

Ek Danışman: Doç. Dr. Taşkın Koçak

Ocak 2012, 89 sayfa

Telekom operatörlerinin sahip olduğu servis sağlayıcı ağ ortamları, farklı üreticilerin çözümlerinin kullanıldığı İşletim Destek Sistemi (İDS - Operational Support System (OSS)) ve bunların sunduğu özgün kullanıcı yönetim sistemleri tarafından yönetilmektedir. Tipik bir servis sağlayıcı ağ ortamının yönetimi, Merkezi (ve/veya Küresel) Kullanıcı Yönetim Sistemi ile Yerel Kullanıcı Yönetim Sistemleri arasındaki iletişimlerin çözümünden oluşmaktadır. Günümüzde, servis sağlayıcılarının ağ ortamlarında farklı üreticilerin sunduğu ve farklı protokollerle çalışan İDS sistemleri kullanılmaktadır.

Telekom sektöründe bütün bu farklı İDS sistemlerinin merkezi kullanıcı yönetim sistemi tarafından yönetimi ciddi bir sorun teşkil etmektedir. Bu bakımdan, İDS sistemleri ağ elemanlarının daha güvenli, tutarlı ve etkin yönetimi için bu yönetim sistemlerinde çalışan operatörlerin ve onların yetkilendirilmelerinin gerçekleştirileceği "standartlaştırılmış" bir merkezi tanımlama ve denetim mekanizmasına acilen gereksinim duyulmaktadır.

Bu gereksinimi karşılayabilmek ve Telekom operatörlerinin farklı ağ elemanlarının bir arada etkin bir biçimde çalışabilmesi için, bu tezde TMF615 standardı tabanlı ortak iletişim platformu tasarlanmış ve geliştirilmiştir. Genel olarak, bu tezde sunulan yaklaşım, farklı üreticilerin çözümlerinin kullanıldığı bir servis sağlayıcı ağ ortamında kullanıcıya sunulacak iletişim sorunlarını çözmeye yönelik ortak bir ara yüzü içermektedir. Geliştirilen ara yüz ve iletişim protokolü bu alandaki ilk çözümler arasında yer almaktadır ve tezin sonuçları ülkemize katma değer katan bir ürüne dönüşmektedir. Bu bağlamda, sunulan tezin yeni çözümler üreterek bilimsel birikime ve ticari uygulamalara önemli katkılarda bulunacağı öngörülmektedir.

Geliştirilen TMF615 tabanlı ara yüz standardı servis sağlayıcıların mevcut ve yeni işletim destek sistemlerini ve uygulamalarını oldukça kolay ve minimum çaba sarf ederek birleştirilmesine olanak sağlamaktadır. Böylece, TMF615 kullanılarak standartlaştırılmış bir ara yüz sunulmuştur ve tüm değişik sistemler için sadece ortak bir iletişim platformu yeterli olmaktadır. Dolayısıyla, üreticiler sadece uygulamaların gerçekleştirilmesinden ve tanımlamalara uygun olmasından sorumlu olmaktadır ve tüm ilgili sistemler için tavsiye edilen standartlaştırılmış bir iletişim sunulmaktadır.

Bu durum servis sağlayıcılarının sistem yönetimini büyük ölçüde kolaylaştırmıştır, sistemin performansını önemli ölçüde yükselmiştir ve aynı zamanda masrafların da yüksek miktarda düşürülmesini sağlamıştır. Böylece, bütün ağ elemanlarının tek ortak bir ara yüz kullanılarak etkin bir biçimde yönetilmesi sağlanabilmiştir.

**Anahtar Kelimeler:** İşletim Destek Sistemi (İDS), Kullanıcı Yönetim Sistemi (KYS), TMF615

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| AMS | : | Access Management System |
| AOP | : | Aspect Oriented Programming Interface |
| ASF | : | Apache Software Foundation |
| API | : | Application Programming Interface |
| APIs | : | Application Programming Interfaces |
| UMS-C | : | Central User Management Systems |
| CDK | : | Component Development Kit |
| CORBA | : | Common Object Request Broker Architecture |
| DB | : | Database |
| DM | : | Distributed Monitoring Classic |
| EE | : | Enterprise Edition |
| eTOM | : | Ericsson Enhanced Telecom Operations Map |
| XML | : | Extensive Markup Language |
| XSD | : | Extensive Marjup Language Scheme Definition |
| GUI | : | Graphical User Interface |
| HTTP | : | Hypertext Transfer Protocol |
| HTTPS | : | Hypertext Transfer Protocol Secure |
| IRPS | : | Interface Reference Points |
| JAX-RS | : | Java API for Restful Web Services |
| JAX-WS | : | Java API for XML Web Services |
| JAXB | : | Java Architecture for XML Binding |
| JSF | : | Java Server Faces |
| JSP | : | Java Server Pages |
| LDAP | : | Lightweight Directory Access Protocol |
| UMS-L | : | Local User Management System |
| NE | : | Network Element |
| NBI | : | North Bound Interface |
| OS | : | Operational Support |
| OSS | : | Operational Support System |
| OSSs | : | Operational Support Systems |

| | | |
|---|---|---|
| PSO | : | Provisioning Service Object |
| RAM | : | Random Access Memory |
| RPC | : | Remote Procedure Call |
| SSL | : | Secure Sockets Layer |
| SOAP | : | Service Oriented Architecture Protocol |
| SPML | : | Service Provisioning Markup Language |
| SNMP | : | Simple Network Management Protocol |
| TM | : | Tele Management |
| TMF | : | Tele Management Forum |
| TEC | : | Tivoli Enterprise Monitoring Console |
| UAC | : | Unified Access Control |
| UI | : | User Interface |
| UM | : | User Management |
| WSDL | : | Web Service Definition Language |
| WINNT | : | Windows NT |

# 1. INTRODUCTION

This chapter provides information on Operational Support Systems (OSSs), Tele Management Forum (TMF) 615 and general architecture of TMF615 protocol.

## 1.1 OPERATIONAL SUPPORT SYSTEMS

OSS consists of a set of programs, which are used for controlling, monitoring, analyzing, and managing networks by the service providers (Chou 2007). Complexity of the system increases because of the changing of the market dynamics (Nye and Hong 2007). Different OSS is designed by the different vendors to meet the service providers' needing (Harada, Tomita and Sugiyama 2003). User management's complexity increases with the OSS systems.

User management systems' evolution is formed in the following steps (TMF Forum 2010; Wang and Lv 2008),

i. Operational Support (OS) based user management
ii. NE specific user Access
iii. Application specific user Access
iv. Unified user access control
v. Multiple OS user access control
vi. Centralized user management for a specific OSS
vii. User Management in Heterogeneous networks

All of these steps, which constitute evolution of user management systems, are explained in the following subsections.

### 1.1.1 OS Based User Management

UNIX is used to develop OSS systems. In the OSS, there is not implementation of user management. There are users and groups in the UNIX System. And these groups and users do the simple user access control (Aleem 2009).

In this system, user makes the task after logging on the system. Credential of the user is controlled by UNIX. OSS application prepares the response and this response is sent to the user.

Advantage of the system,

i. Provides the basic User Management (UM)

Disadvantage of the system,

i. Flexibility of providing rights cannot be handled.

ii. Applications cannot be accessed specifically.

iii. There is not Network Element (NE) access control.

### 1.1.2 NE Specific User Access

There isn't user access control in most of the network elements (Rader 2011). However, user access control is needed for the security. For this reason, network access control is located in OSS (Wright & Ridgely 2010).

After user logs in the system, user makes the task. Credential of the user is controlled by UNIX. User asks before performing the network task and OSS application sends response to the user. This system is simple, and provides Network Element. However, there is not application specific access in this system (Aleem 2009).

### 1.1.3 Application Specific User Access

Application Specific User Access is used for restricting user to access some applications. After user logins the system, credentials are checked (Aleem 2009; Zhou 2011). If user wants to access a specific application, application access controls it. If the user needs network access to perform task, NE access controls it and gives or does not give permission to this task. System sends the response to the user.

This method is more secure and more flexible. NE access control and application access control are used. These are the advantages of this method. However, user management control is difficult because of the distributed system.

### 1.1.4 Unified User Access Control (UAC)

Multiple access control is difficult for organizations because of the complexity (Richardson 2007). UAC is used to solve complexity. UAC wants to credentials from applications, and processes. In this way, network applications are protected (Hao 2008).

After user logins the system, user's credentials are controlled by the UAC. If the user wants to access a specific application, application access controls it. If the user needs network access to perform task, NE access controls it and gives or does not give permission to this task. System sends the response to the user. In this way, uniform user management is provided (Lin *et al.* 2010; Raymer *et al.* 2006).

In this method, user management is easy. NE access control and application access control are used (Georgalas, Azmoodeh & Ou 2005). These are the advantages of this method. However, if there are multiple OS, provision of the user cannot be done.

### 1.1.5 Multiple OS User Access Control

There are many operating systems. Applications use these different operating systems. For this reason, UAC must involve all operating systems. After user logins the system, user's credentials are controlled by the UAC (Choi and Seo 2006, pp. 4-116). Many OS such as UNIX, Linux, Windows NT (WINNT), and Solaris can be used by UAC. If the user wants to access a specific application, application access controls it via authentication proxies (Kadowaki and Fujita 2009). If the user needs network access to perform task, NE access controls it and gives or does not give permission to this task. System sends the response to the user. This NE based works are tested with many works (Strassner & Raymer 2006).

User management is very extensive. UAC, NE access control and application access control are used (Taniguchi & Ebitani 2010). These are the advantages of this method. However, the UAC is complex in this system.

### 1.1.6 Centralized User Management for a Specific OSS

Telecom networks become bigger each passing day. And management becomes difficult. To handle the easy management, regions are separated (Tselentis *et al.* 2009). Separate clusters manage these regions. And user management is done by the centralized user management (Gupta 2006).

Service provider works in the central place. And user provisioning is done from this place. But, this place is used for only one vendor. This type of work is done by researcher (Ramiro 2009). All the other steps are the same with UAC in this method (Aleem 2009).

Centralized user management provides that the extensive user management (Uchikubo, Mizuno and Gotanda 2000). This is the advantage of this system. However, this system works for only one vendor OSS which is the disadvantage.

### 1.1.7 User Management in Heteregenous Networks

In this method, one service provider has many OSS providers. Different services are combined from OSS vendors by the service provider to have best performance (Uchikubo, Nakamura and Gotanda 2008). User management solutions are different according to the vendor. For this reason, user provisioning and user management must be done regularly. User management is different according to vendor. For this reason, understanding of each vendor is unique.

This method's performance is the best. But, user accesses the system according to vendor. There are many different vendors. For this reason, user has to understand all these vendors. This increases the time and money for the user provisioning operation.

All of these methods are inefficient in some way. To solve this inefficiency, a standard protocol must be used. This protocol is the TMF 615.

## 1.2 TMF 615 PROTOCOL

User Management is difficult in heterogeneous networks. Centralized user management system (UMS-C) TMF615 standard is prepared by TMF to solve this difficulty (TMF Forum 2010). Activities of user provisioning are standardized with this specification (Nakano *et al.* 2009, pp. 272-277).

An interface is specified by TMF615. In this way, service providers can provide access rights and authorities of the operators compatibly with using a UMS-C. UMS-C and Local User Management System (UMS-L) exchange information to make the provision users.

Provision users, authorities of users are provided by UMS-C which is used by Service Providers. UMS-C makes request to UMS-L. Then, UMS-L processes the requests and sends response to these requests.

Some concepts of TMF615 are introduced below.

a. Authorization space

Vendors use different authorization techniques for their users. These authorization techniques are based on the roles. Service providers and the vendors use their own roles as described in research (Meyners, Driss & Feger 2008). For this reason, TMF615 specifies a technique to assign roles. This technique is the authorization space.

Authorization Space has three components. These are Functionality Axis, Activity Axis and Scope Axis (TMF Forum 2010).

OSS user's work area is specified by the Functionality Axis. Activities of the OSS user are defined by Scope Axis. Activity Axis specifies the account of user's scope.

In this case, different vendors give OSS applications to service provider. Figure 10 shows that service provider can only make management of Fault and Configuration. When the service provider wants to make viewing of fault, view's coordinate are handled by the circle. Fault and view are in different axis. For this reason, TMF615 uses the combination of these axes to give roles.

b. Scheduling

Scheduling is used for controlling the user access to OSS resources. User provisioning provides the scheduling. According to TMF615 there are two types of scheduling. One of them is weekly scheduling. UMS-L learns that user accesses the resources which days in a week from the weekly scheduling. Monthly scheduling is second TMF615 scheduling type. In this type, UMS-L learns that the user accesses the resources which days in a month.

c. Audit

User's activities are monitored by using audit. Monitoring is important for security and to prevent abuse. There are two types of auditing in TMF615. One of them is the "Status Audit". In this type, synchronization between the UMS-C and UMS-L is used. UMS-L's data and UMS-C's data are compared. Synchronization is started if the comparison has differences. Second type is the "Audit Trail". In this type, security is the problem because monitoring is done only for a period of time. After this time there is no monitoring and there is no controlling the user activities.

### 1.2.1 TMF615 with Centralized User Management

Every OSS vendor has its own UM technique. UMS-C takes these techniques into one place. TMF615 succeeds it by providing the communication between UMS-C and UMS-L over a standard which is Web Services Definition Language (WSDL). WSDL is an interface to make the communication.

UMS-C and UMS-L's network architecture description is listed below;

i. UMS-C includes the roles, regions. Details of service provider are in the UMS-C.
ii. UMS-L is compatible with the service providers. And service providers are compatible with TMF615.
iii. UMS-C does the user provisioning. Accounts, roles, etc. are assigned the user by UMS-C.
iv. UMS-L makes the authentication to provide user access of an application.
v. Monitoring is done at UMS-C by using auditing.
vi. If the connection loss between the UMS-C and UMS-L, UMS-L does the user provisioning.
vii. Synchronization is provided by UMS-C.

### 1.2.2 General Architecture of TMF615 Protocol

Local vendor OSS includes the UMS-L which is used for the UM solution. Each vendor's UM supports the TMF615. UMS-L provides an interface between the UMS-C and TMF615 specification.

One of the UMS-L components is TMF615 agent. TMF615 specialties are listed below;

i. Complexity of the TMF615 is not seen by using TMF615 agent.
ii. WSDL interfaces are made by the TMF615 agent to communicate with UMS-C.
iii. It uses a Database (DB) for monitoring operations.
iv. Some user provisioning requires the communication between the local UM and TMF615 agent. For this reason, TMF615 agent communicates with local UM to perform the operations.
v. Synchronization is provided after the connection loss between the UMS-C and UMS-L by TMF615 agent.

Another component of the UMS-L is the Local User Management. Local UM is the vendor specific UM solution. Executions of the UMS-C requests are done by using Interface Reference Points (IRPs). Bringing out the IRPs is the task of the Local UM.

IRP Manager is located at UMS-C and IRP Agent is located at UMS-L. Communication security of this system is provided by adding the super user to the UMS-L. UMS-C assigns administrator role to some users on UMS-L. User provisioning is done by the administrators. Authorization is made for the user provisioning requests by UMS-L.

When the user provisioning is occurred at UMS-L, TMF Agent's DB is updated by Local UM. If the Local UM doesn't update DB, periodically TMF Agent must control the users, who are modified or new. For this reason, user information is updated continuously for the security and monitoring by the Local UM.

**a. More reliable**

Applying security procedures becomes more difficult for the service providers because of the complexity of the system. However, when TMF615 is used, complexity decreases and security procedures are performed easily.

**b. Interoperability**

TMF615 offers standardization to OSS providers. In this way, integration of systems becomes easier. Devices work more interoperable between each other with TMF615 protocol.

**c. High performance**

Performance of OSS increases with the TMF615 protocol because when TMF615 is used in OSS, complexity of system decreases and this provides high performance to the OSS.

**d. Easy management**

Management of OSS is easy with TMF615 because less complex systems are provided by TMF615 and this offers easy management of systems for OSS.

**e. Power saving**

Number of used devices is minimized by TMF615 and this provides to save power and to decrease complexity of the system.

**f. Durable components**

When OSS uses the TMF615 protocol, its components live long and errors, which can be discovered later, are minimized.

**g. Low maintenance cost**

Number of used devices decreases with TMF615 in OSS. Therefore, maintenance cost of OSS becomes lower with this protocol.

# 2. RELEVANT LITERATURE

OSSs are supported by many companies. These companies are Ericson, IBM, Vodafone, T-Mobile, Telefonica, Nokia, Siemens, and Wipro (Frank, Luoma & Tyrväinen (2007), Wong, Ting and Yeh 2007; Qi *et al.* 2006). Each of these companies provides one of the functionalities of the OSS. Vodafone, T-Mobile, and Telefonica are the service providers. IBM and Wipro provide the authentication audit systems. Ericson, Nokia-Siemens-Network presents the OSS solution (Goestl 2006).

Service providers use different OSSs which are presented by different producers and work with different protocols. For this reason, management of OSS, which is worked with different protocols, is the big problem for the central user management system (Faro & Spina 2007). This problem is tried to solving by the companies in Telemanagement meeting. They find a solution for this problem. Solution is the TMF615 which is the standard and it is used in OSSs. According to this standard, some of the companies present their tools. These tools are explained below.

## 2.1 IBM TIVOLI FRAMEWORK

IBM Tivoli Management Framework (TMF) is a platform for the system management. Common Object Request Broker Architecture (CORBA) is used for this framework architecture (IBM 2010). CORBA provides to manage many remote locations and devices (Huang *et al.* 2003, pp. 1236-1242).

Security, storage monitoring and configuration can be provided by the Tivoli products. According to this, Tivoli is divided into some parts (IBM 2010):

i. Service Management
ii. Storage
iii. Security
iv. Performance Automation

v. Business Automation

vi. Network Management

Each of these parts is provided by different IBM Tivoli products. These products are based on TMF. Some of these products and their functions are described in the following subsections.

### 2.1.1 IBM Tivoli Configuration Manager

Hardware and software Inventory and Software Distribution abilities are provided by IBM Tivoli Configuration Manager. Complex applications, which are in multiple locations, can be applied rapidly and efficiently by using this manager according to work of the Ramanathan, Alexander and Kerr (2008, pp. 433).

Hardware and software configuration information is collected by using inventory module of the configuration manager. It changes the system configuration to remove the dependence of the company's policies (IBM 2010).

### 2.1.2 IBM Tivoli Distributed Monitoring Classic (DM)

Substitute based monitoring is provided by IBM Tivoli Distributed Monitoring Classic (Ramanathan, Alexander and Kerr 2008, p. 433). This software is used for management of the operating systems, databases and servers which are in distributed and host environments. Problems of the system resources are detected and recovered automatically by using monitoring tool.

### 2.1.3 IBM Tivoli Enterprise Monitoring Console (TEC)

Class event handling and correlation are provided by Tivoli Enterprise Console tool. Automated problem diagnosis is provided by TEC to increase system performance (IBM 2010).

### 2.1.4 IBM Tivoli Netview

Tivoili NetView is used for network management. It uses Simple Network Management Protocol (SNMP) and TMF. SNMP provides real time monitoring and active testing of servers for this Tivoli program (IBM 2010).

Large networks management is easy with NetView because it provides scalability and flexibility for critic environments.

## 2.2 ERICSSON ENHANCED TELECOM OPERATIONS MAP (eTOM)

Tele Management (TM) Forum publishes the eTom as a guidebook and telecommunication industry uses it as the standard for the business processes (Chou, Seng & Lin 2008; Shangguan, Gao & Zhu 2008; Hannemann 2007). Scope of the business processes, which is used by service providers, is provided by eTOM model (Yahia, Bertin & Crespi 2007; Kelly 2003). Important elements and these elements interaction are described in this scope.

eTOM is divided into four levels. These are Level-0, Level-1, Level-2 and Level-3 processes. There are many processes in each of these levels (Huang & Hsu 2011). eTOM model includes rows and columns. Specific processes are handled from intersections of these rows and columns. Customer activities are shown in the top row (Joshy & Shenoy 2010).

eTOM map shows the interaction between the processes (Matos *et al.* 2008). It shows the interactions by classifying these processes according to sections. These sections are described in many works (Joshy and Shenoy 2010; Liu, Azmoodeh and Georgalas 2007):

i. Strategy, Infrastructure & Product
ii. and Operations.

Strategy, Infrastructure & Product Columns:

i. Strategy and Commit,
ii. Infrastructure Lifecycle Management and Product Lifecycle Management

Strategy, Infrastructure & Product Rows:

i. Marketing & Offer Management,
ii. Service Development & Management,
iii. Resource Development & Management and Supply Chain Development & Management

Operations Columns:

i. Operations Support & Readiness, Fulfillment, Assurance and Billing

Operations Rows:

i. Customer Relationship Management,
ii. Service Management & Operations, Resource Management & Operations
iii. and Supplier/Partner Relationship Management.

## 2.3 COMPARISON OF OSS STUDIES WITH TMF615 PROTOCOL

When all the existing OSSs' properties are examined, TMF615 provides less complex, more reliable, more interoperable, more powerful, easier management, and many other advantages which are all shown in Table 2.1, than the other OSS systems.

**Table 2.1: Comparison of Existing OSS Systems with TMF615**

| OSS Systems | IBM Tivoli | Ericsson eTOM | TMF615 |
|---|---|---|---|
| Complexity | High | High | Low |
| Reliability | Medium | Low | High |
| Interoperability | Low | Low | High |
| Performance | Low | Medium | High |
| Management | Difficult | Difficult | Easy |
| Power Saving | Low | Low | High |
| Durability of Components | Low | High | High |
| Maintenance Cost | High | High | Low |
| Operational Time | Medium | Low | High |

# 3. METHODOLOGY

## 3.1 PROPOSED OPERATIONAL SUPPORT SYSTEM ARCHITECTURE

In traditional OSS systems, operator accesses the OSSs by using each OSS's different client tools as shown in Figure 3.1 via distinct accessing protocols such as Service Oriented Architecture Protocol (SOAP), Extensive Markup Language (XML), Lightweight Directory Access Protocol (LDAP) (Blum, Magedanz and Schreiner 2009a, 2009b). For instance, operator uses the AMS5520 (Access Management System) client tool, which is Alcatel – Lucent's server, for accessing the Alcatel – Lucent AMS5520. However, in our proposed system architecture, operator uses TMF615 Client to access all the OSS systems. In this way, operator doesn't need to have all the client tools of OSS systems and this reduces the complexity of the OSS architecture.

In this respect, thesis' architecture is composed of 4 parts which is shown in figure 3.3. These are:

i. Client Adaptor: Server of Web Service, implementation of TMF615 Interface.
ii. Server Adaptor: Converts TMF615 messages to North Bound Interface (NBI) which is AMS5520 interface.
iii. Processing Engine: Mapping between the client adaptor and server adaptor
iv. Test Program: Sample of OSS Client Tool
v. AMS5520: Access Management System is developed by Alcatel-Lucent and is used as a Service Provider.

The main prupose of the proposed architecture is to make mapping between clients and servers according to TMF615. As shown in Figure 3.2, our system works like that operator makes a request to make one of the TMF615 operations such as addding new user to the OSS systems, deleting a user from OSS systems, and etc. After operator's request, TMF615 Client gets this request and trasnmits it to Client Adaptor. Client adaptor is a web service which is used to communicate two devices via a network.

Client Adaptor sends this request to mapping module which makes conversions between the TMF615 protocol and OSSs. After the mapping, request is taken by Server Adapter and is sent to the OSS system.

**Figure 3.1: Traditional OSS User Management System**

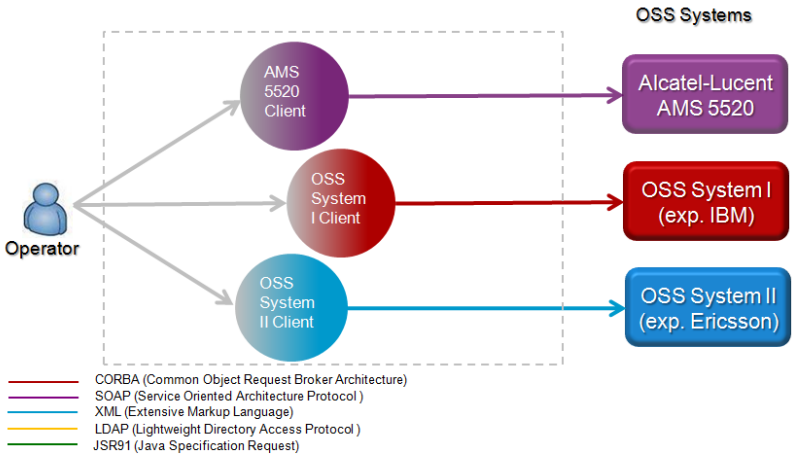
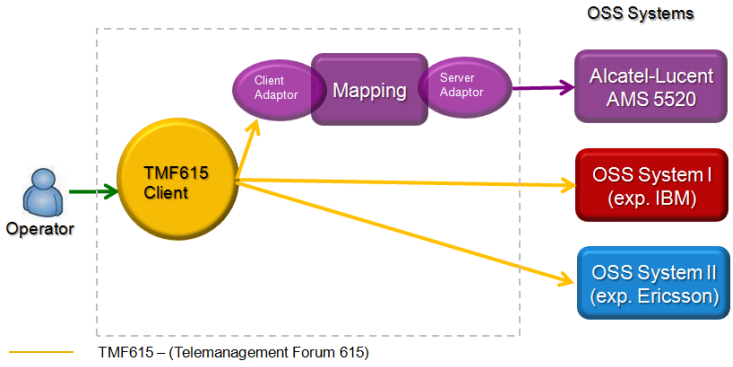



**Figure 3.2: Proposed OSS Architecture**




If an example is given; operator wants to add new user to the Alcatel – Lucent AMS5520. In this respect, it enters new user information from the test tool according to TMF615 protocol. After operator enters the user information, test tool sends these information to the Client Adaptor and Client Adaptor sends the request to Mapping Module. Mapping Module makes conversion of the information from TMF615 to AMS 5520. When the conversion is finished, Server Adapter gets the request for sending it to AMS5520. After AMS5520 makes the request, it sends the response to the Server

Adapter and Server Adapter transmits response to Mapping Module. Mapping Module again makes a translation. At this time, it translates AMS5520 response to accommodate the response TMF615 protocol.When the mapping is finished, Client Adaptor delivers the response to the operator.

Interior architecture of our TMF615 sofwtare is shown in Figure 3.3 and works as explained above with 5 parts. This architecture provides less complex, more reliable, and more standardized OSS systems than the traditional OSS systems.

TMF615 implementation is specific to vendors. Each vendor has vendor specific client tool, which makes service providers' messages compatible with TMF615. Our proposed system's service provider is the Alcatel-Lucent server which is AMS5520. Implementation is done to make AMS5520 compatible with TMF615. System overview of the implementation is shown in Figure 3.4.

**Figure 3.3: Interior architecture of the TMF615 software**

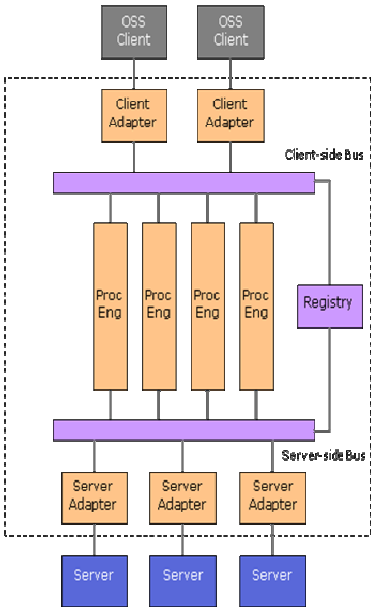
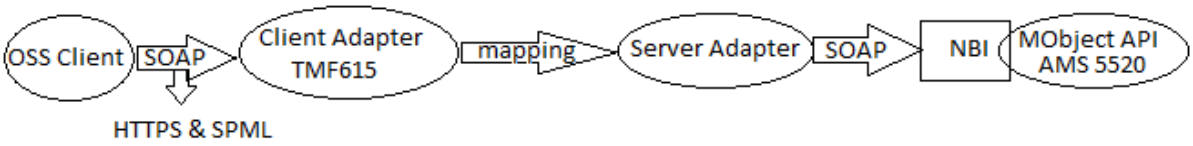


**Figure 3.4: System overview**

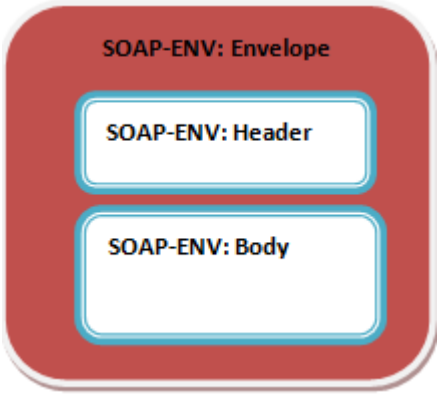



SOAP / Hypertext Transfer Protocol Secure (HTTPS) are used for messaging. Service Provisioning Markup Language (SPML) is used as a standard in messaging which is sent from OSS Client to Client Adapter. Some of the explanations about the system are explained in the following subsections.

### 3.1.1 SOAP and HTTPS

SOAP is a protocol to exchange information in the Web Services (Chen *et al.* 2008, 2010). SOAP message format is based on the XML. Remote Procedure Call (RPC) and Hypertext Transfer Protocol (HTTP) are used for transmission by SOAP (Vrancic *et al.* 2007). SOAP message format is shown in Figure 3.5.

**Figure 3.5: SOAP Structure**




Combination of Secure Sockets Layer (SSL) and HTTP composes the HTTPS. HTTPS is used for the encrypted communication and to make identification more secure.

### 3.1.2 WSDL and Extensive Markup Language Scheme Definition (XSD)

The Web Services Description Language is an XML-based language that provides a model for describing Web services which are used for OSS in many works (Xu *et al.* 2009). Our system gets the WSDL of AMS 5520 from the NBI. In this way, our web service and the AMS5520 can communicate with each other. WSDL has some definitions. One of these definitions is the XSD.

Formal description of elements in an XML document is specified by XSD. In this way, elements' formal descriptions are sent by WSDL. And web services know each other's elements' description.

### 3.1.3 SPML

OSS Client and Client Adapter communicate with each other over the SPML. SPML is used to make standard messaging system (Gallen and Reeve 2005, 2006). SPML is the open standard. Service provisioning requests are integrated and interoperated by using SPML.

## 3.2 EVALUATION

This thesis's operations are implemented by applying Waterfall project methodology. In this respect, thesis is divided into 4 parts which are analysis, designing, coding, and testing and integration of thesis. These parts are realized respectively according to work packages that are independent from each other. Furthermore, all de facto coding and naming standards are all considered, too.

When the TMF615 is realized, web service, which is compatible to TMF615 standard, is offered but, this thesis isn't limited only with implementation of the TMF615 interface, it also includes the integration of TMF615 interface with AMS5520 software. If Alcatel-Lucent's system is paired with TMF615 standard, integrations of AMS5520

with TMF615 interface can be succeeded completely. In addition to these, a test program is implemented. Test program's purpose is that testing the system's functions' behaviors by using the TMF615 WSDL.

Coding the thesis is divided into 4 parts which are defined in some works, too (Wang *et al.* 2011 ):

i. Client Adaptor
ii. Server Adaptor
iii. Processing Engine – Mapping between the client adaptor and server adaptor
iv. Test Program

Locations of these parts inside the software architecture are shown in Figure 3.3.

### 3.2.1 General Overview of the TMF615 Interface

TMF615 offers the WSDL solution set which simplifies implementation of standards. Nowadays, many software tools such as Apache, CXF, and Axis2 can produce interface codes by using WSDL.

If TMF615 interface is simply described;

i. TMF615 interface defines user's basic information into "User" class, user's provision's information into "ProvisionData" class.
ii. User's account information ("AccountData") and user authorization information ("AuthorizationData") are described into "ProvisionData".
iii. User account basis information such as "accountId", "validationInformation", and user account orbit ("targetData") are all defined into "AccountData" class.
iv. User account access profiles ("AccessProfile Value") are described with user account orbit.

This thesis main purpose is that implementing the operations on the Alcatel-Lucent AMS5520 according to TMF615 standard that is formed for minimizing the complexity of the communication between the service provider and user management system which is provided by operation support system manufacturers. In this way, all the services, which are provided by Alcatel-Lucent 5520 AMS, can be used securely and correctly. All the performance and functional tests are done to provide these secure services in this thesis. Complexity of the traditional OSS system is shown in Figure 3.6 and proposed system architecture is shown in Figure 3.7.

All of these, which are mentioned above, are all implemented according to some work packages. These committed packages are all described in the following sections.

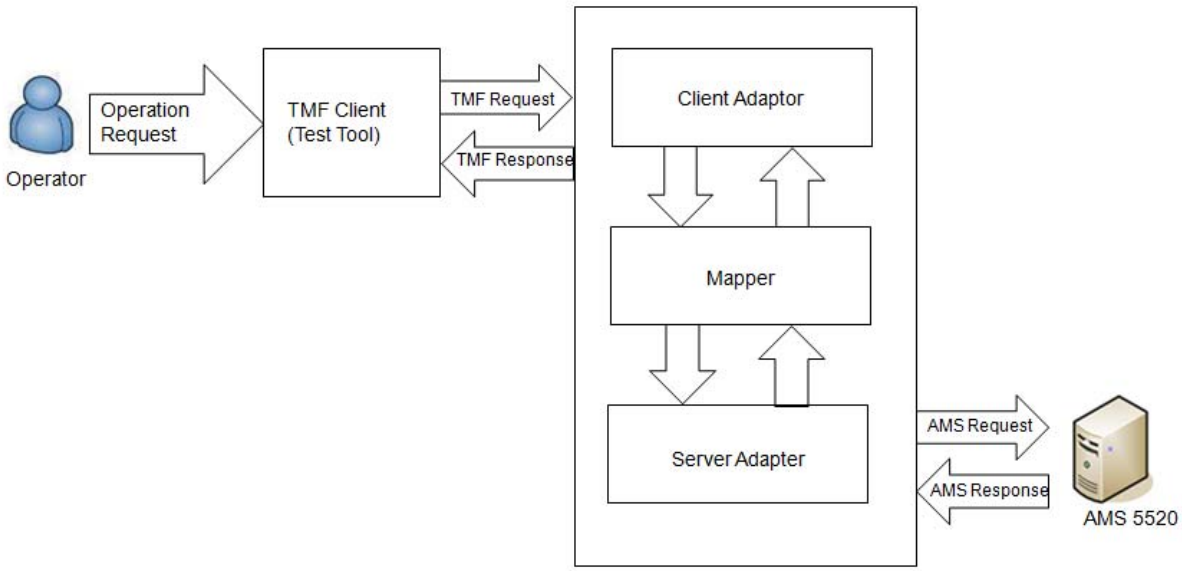
### 3.2.2 Installing AMS5520 Software Inside the Application Server and Investigating the Basis Functions

In this work package, Alcatel – Lucent AMS5520 is installed into application server and all the functions of the AMS5520 are examined. Also, common communication protocols between the AMS5520 and TMF615 are determined; use case and sequence diagrams are realized. TMF615's operations and their descriptions are shown in Table 3.6. General architecture use case diagram is shown in Figure 3.6 and sequence diagrams of add user operation, not supported operation and invalidate parameter are explained in Figure 3.7, in Figure 3.8 and in Figure 3.4 respectively.

Some TMF615 operations are supported by Alcatel-Lucent AMS5520. However, some of them aren't supported by it. In that situation, as shown in figure 3.8, not supported operation response is sent to the operator by the client adapter. When the operation is supported by AMS5520, which can be seen from Figure 3.7, TMF operation request is firstly sent to the client adapter, and then it transmits TMF request to mapper, that makes mapping and send AMS request to server adapter and then server adaptor sends AMS operation request to AMS5520. Then AMS5520 sends the response reverse order the operator. TMF615 have some parameter criteria such as userId's length must be between the 0-16 characters. Therefore, when the operator doesn't send proper

parameters, as seen from Figure 3.9, mapper sends “TMF invalidate parameter response”.

**Figure 3.6: General architecture use case diagram**




After realizing use case and sequence diagrams, basis TMF operations that are added to application are tested at AMS5520. Interested network components are tested in AMS5520 via application provider and in this way, add user operation is tested and a user is added to AMS5520 successfully.

### 3.2.3 Implementation of Test Tool for Simulating OSSs Operations

A test tool is implemented in this work package for simulating all the operations that are supported TMF615 standard. In this process, all the analysis, design, and coding processes are done for test tool software. This test tool is used as the underpinning program in this thesis.

**Figure 3.7: Add user operation sequence diagram**

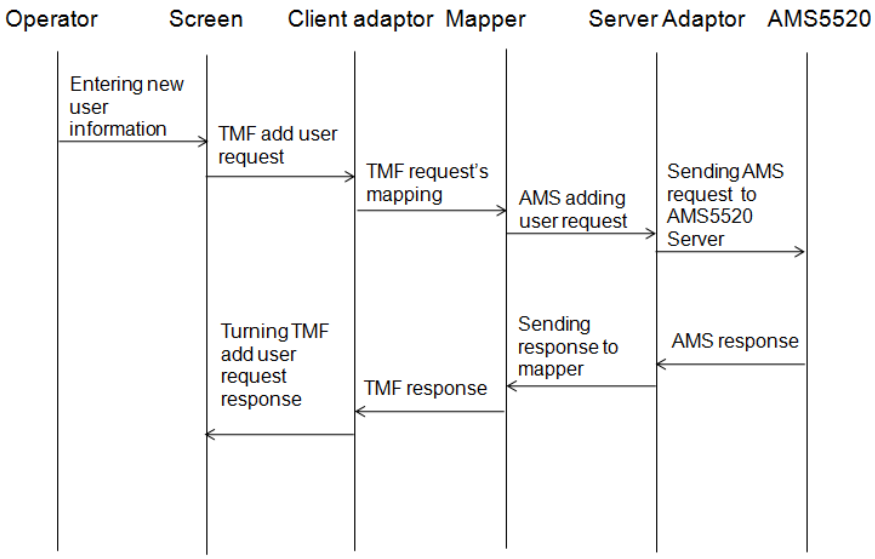



**Figure 3.8: Not supported operation sequence diagram**

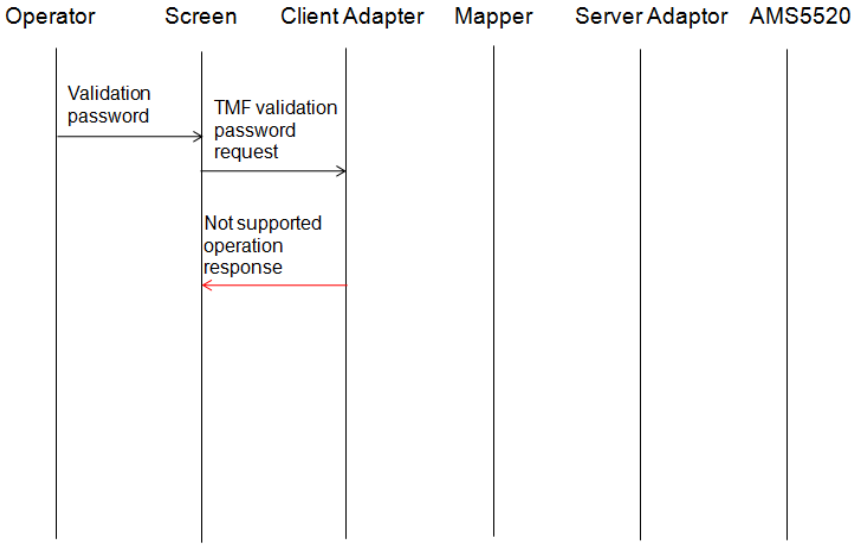
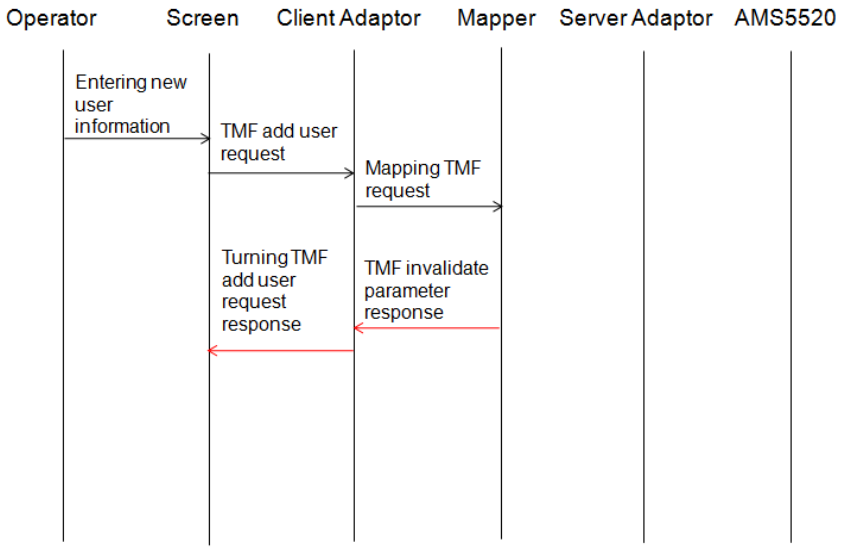


**Figure 3.9: Invalidate parameter sequence diagram**




Java Server Faces (JSF) technology, which is used to develop Java based Web applications, is used for implementing test tool.Test interfaces are realized according to operations. Performed methods in test tool are listed below;

i. Add User Operation
ii. Modify User Operation
iii. Delete User Operation
iv. List User Operations
    a. SPMLLookupRequest – in one go listing only one user
    b. UMListUsers – in one go listing more than one user
v. Reset Password Operation
vi. Suspend User Operation
vii. Questioning User Status Operation
viii. Expire Password Operation
ix. Set Password Operation
x. Resume User Operation
xi. Validate Password Operations
xii. Audit Operations

Test tool interfaces are realized according to these operations, which are listed above. All these interfaces are shown in the following sub sections.

#### 3.2.3.1 Add user interface

Fields are specified according to TMF615 standard for add user operation. Add user interface is shown in Figure 3.10.

#### 3.2.3.2 Modify user interface

Fields are used according to TMF615 standard and modify user interface's fields are generally same with add user interface's fields. Modify user interface is shown in Figure 3.11.

#### 3.2.3.3 Remove user interface

TMF615 standard require only "usrId" for remove user operation. Remove user interface is designed according to TMF615 standard. Remove user interface is shown in Figure 3.12.

#### 3.2.3.4 Suspend user interface

"userId" and "date" are the fields of suspend user interface. These fields are specified according to TMF615 standard. Figure 3.13 shows the suspend user interface.

#### 3.2.3.5 Resume user interface

"userId" and "date" are used as fields of resume user interface. These fields are specified by TMF615 standard for resume user operation. Resume user interface is shown in Figure 3.14.

**Table 3.1: TMF615 operations**

| TMF615 Operations | Description |
|---|---|
| addUser | Adding a user to the system. |
| removeUser | Delete user from the system. |
| modifyUser | Modifying user information |
| suspendUser | Suspending user for making the operation |
| resumeUser | Resuming user in the system. |
| listUsers | Listing the only one user information |
| listUser | Listing the more than one user information |
| expirePassword | Determining user account password validation |
| resetPassword | Reset the user account password |
| isUserActive | Questioning user status |
| setPassword | Assignment a password to user account |
| listTargets | Listing the controlled targets |
| validatePassword | Correcting user account password |
| auditTrialForUsersAdminOperations | Audit user management operations |
| auditTrialForUsersProvisioningOperations | Audit user provision operations |
| auditTargetsForUsersAdminOperations | Audit targets for user management operations |
| auditStatusOfUsersProvisioningInformation | Audit user provision operations' status |
| auditStatusOfTargetsAccountInformation | Audit targets' account information's status |
| auditTrialForTargetsAccountUsage | Audit targets' account usage |
| auditTrialForTargetsAuthorizationUsage | Audit targets' authorization usage |

**Figure 3.10: Add user interface in test tool**

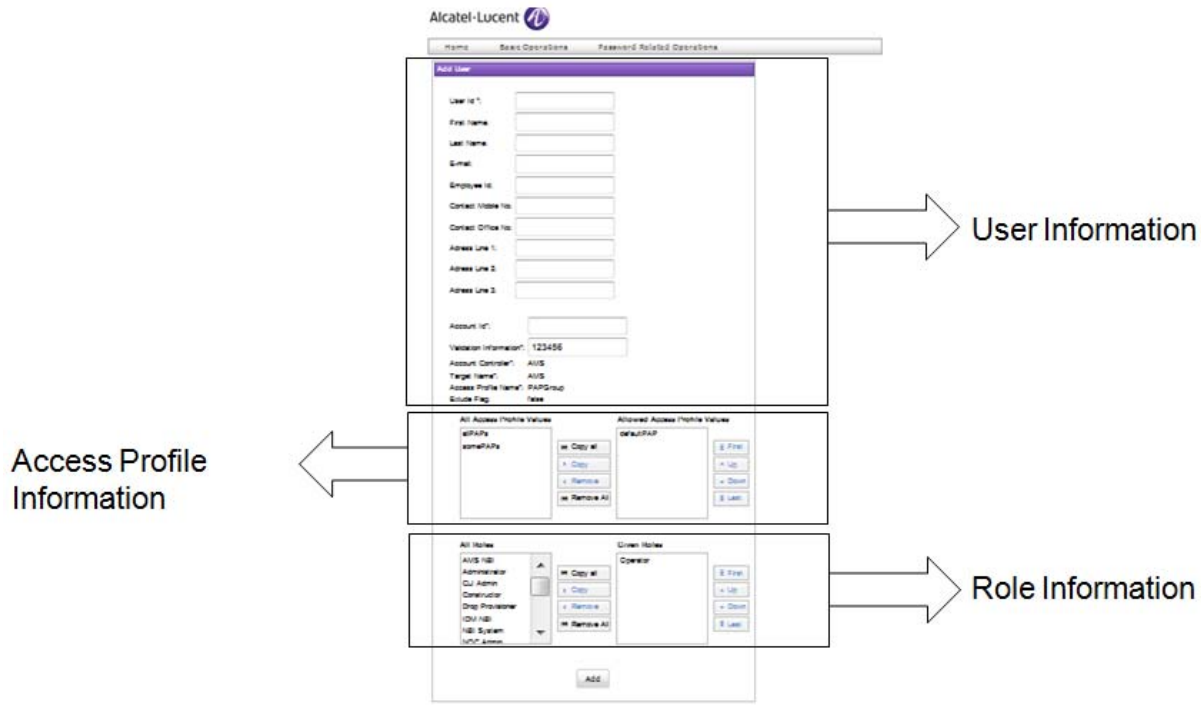



**Figure 3.11: Modify user interface**

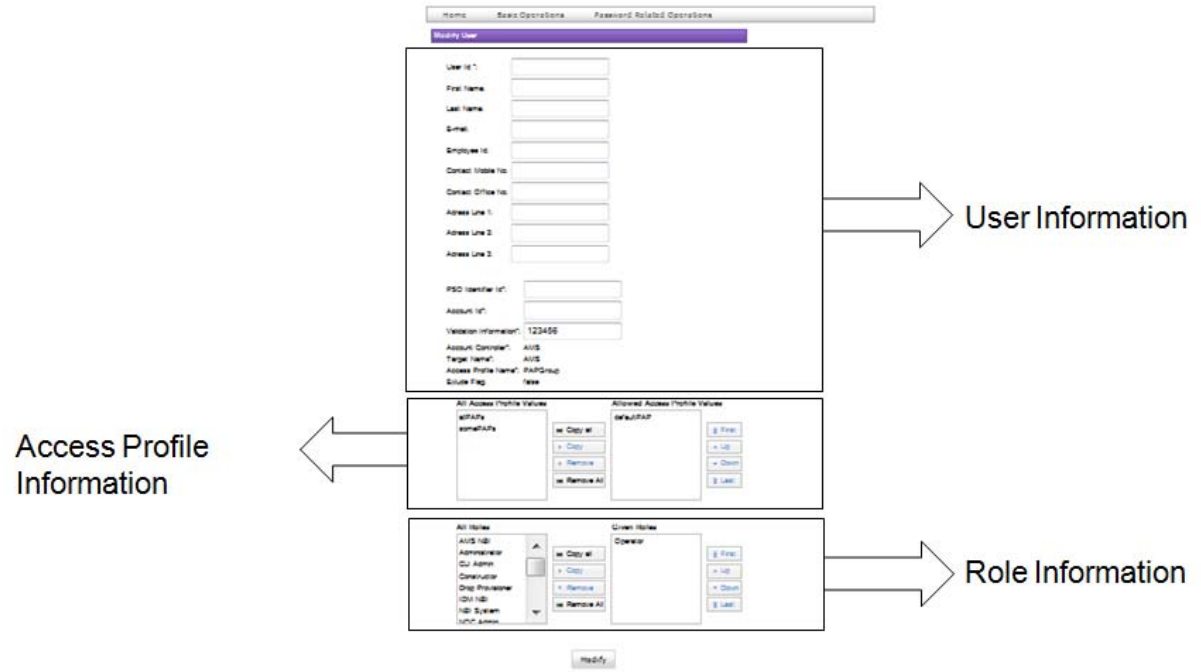


**Figure 3.12: Remove user interface**

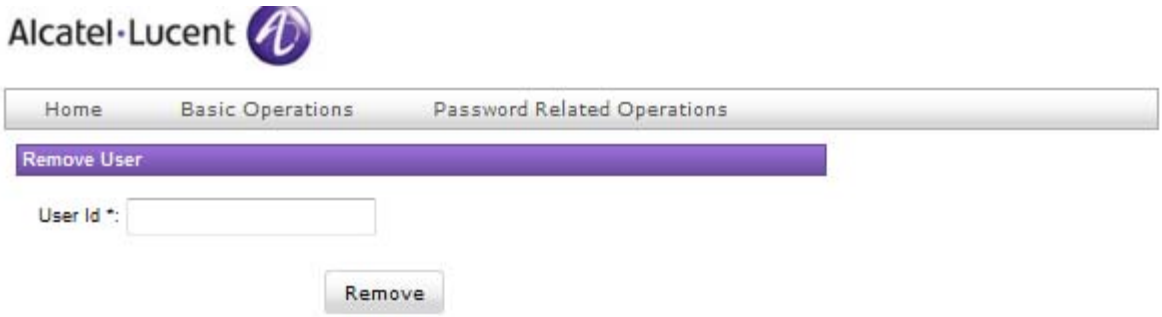



**Figure 3.13: Suspend user Interface**

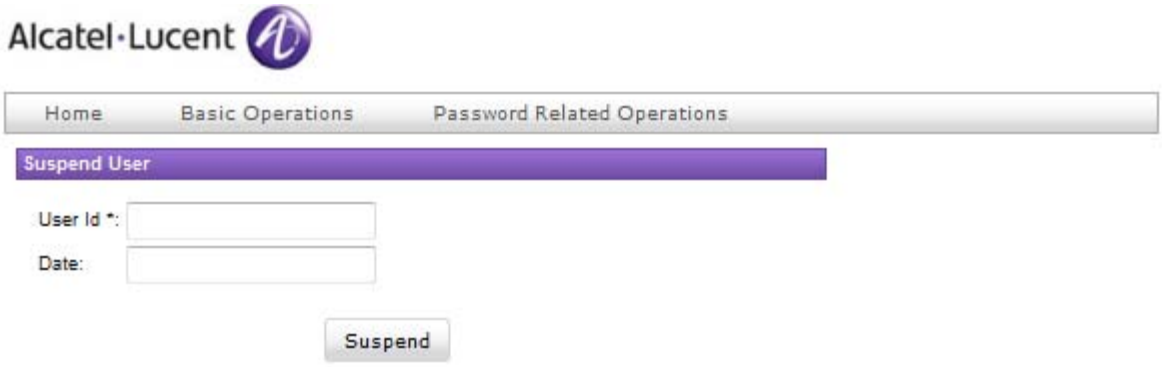
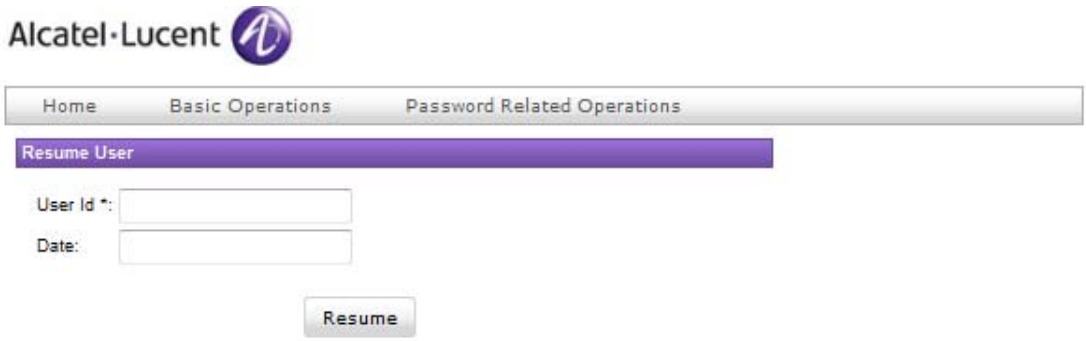



**Figure 3.14: Resume user interface**



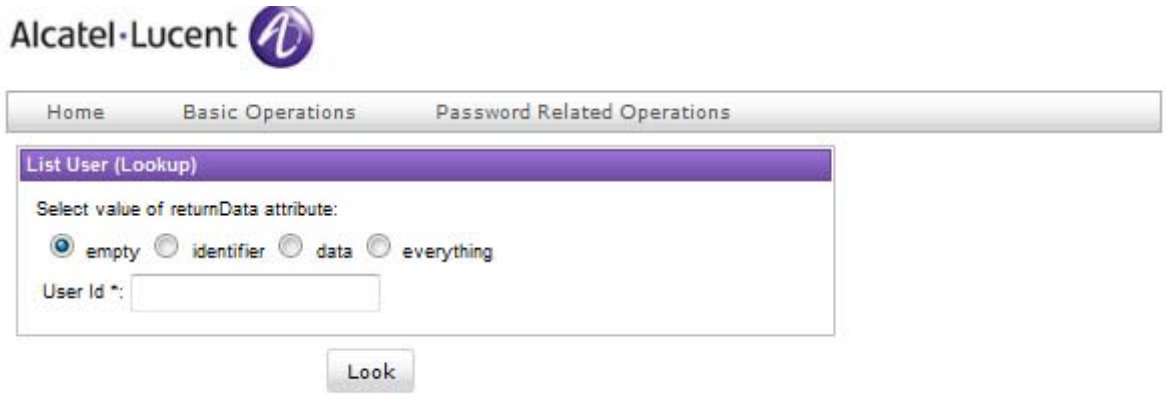
### 3.2.3.6 List user interface

There are two list user operations according to TMF615 standard. These are;

i. SPML List User
ii. UM List User

SPML is a standard which is used for sending message. Messages, which are sent by user, are transmitted to server according to definite format by using SPML standard. Difference between the two list user methods is that SPML List User operation can list only one user information. However, UM List User operation can list more than one user information at the same time. In this thesis, TMF615 SPML List User operation is implemented. SPML List User interface is shown in Figure 3.15.

**Figure 3.15: SPML list user interface**

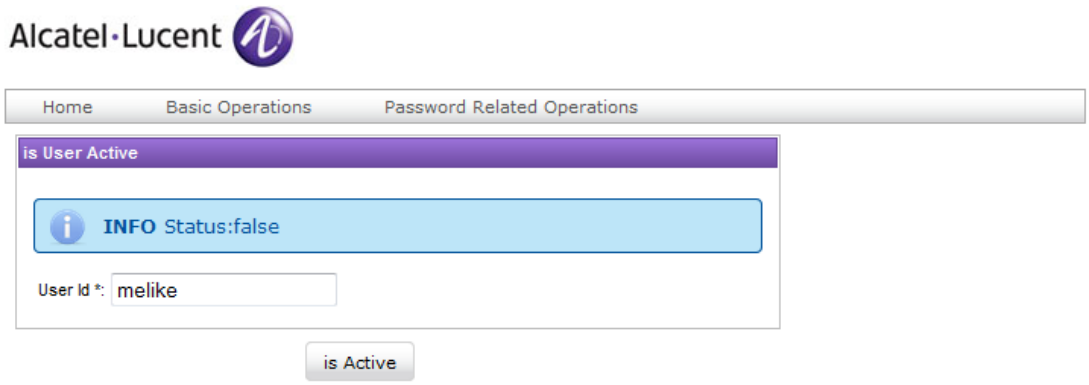



As seen from figure 3.15, “userId” is used for listing the user information. Return data is specified by using “empty”, “identifier”, “data”, and “everything” radio button.

### 3.2.3.7 Is user active interface

“userId” is used for isUserActive operation according to TMF615 standard. User status information is controlled with this interface. Figure 3.16 shows the “isUserActive” operation interface.

**Figure 3.16: Is user active interface**




After all the interfaces are completed, next work package is started to be implemented. Next work package is described in the following section.

#### 3.2.3.8 Implementation of client adapter

In this work package, implementation of the main module is realized. This main module understands the TMF615 commands and responds these commands. Commands and requests, which are sent by client, are achieved according to TMF615 standard with this main module. Client adaptor is implemented to transmit user requests to mapper module, which makes mapping between the TMF615 and AMS5520, and to return mapper's responses to user.

Client adapter's TMF615 standards are specified in TMF615 WSDL solution set. User requests, which come according to TMF615 WSDL standards, must come conveniently SPML and UM standards. Therefore, in this thesis coming requests to client adaptor are realized according to TMF615 WSDL standard. Determination of which request is UM and which request is SPML is done by the TMF615 standard. Which operations are UM, which operations are SPML and naming of operation in TMF615 standard are shown in Table 3.2.

After implementation of the client adaptor, next work package is begun to be implemented. Next work package is explained in the following section.

#### 3.2.3.9 Interface coding between server and server adapter

An interface between the server and server adaptor is implemented on the AMS5520 M-Object API (Application Programming Interface). This module's place is in the interior software architecture.

Server adaptor sends the coming requests from mapper module to AMS5520 and transmits the AMS5520 responses to mapper module. Web service, which is called as NBI, is implemented for server adaptor in this work package. Communication between the AMS5520 and Client Adaptor is provided with this web service.

**Table 3.2: SPML and UM operations**

| TMF615 Operations | TMF615 for WSDL |
|---|---|
| addUser | SPMLAdd |
| removeUser | SPMLDelete |
| modifyUser | SPMLModify |
| suspendUser | SPMLSuspend |
| resumeUser | SPMLResume |
| listUsers | SPMLLookup |
| listUsers | UMListUsers |
| expirePassword | SPMLExpirePassword |
| resetPassword | SPMLResetPassword |
| isUseActive | SPMLActive |
| setPassword | SPMLSetPassword |
| listTargets | SPMLListTargets |
| validatePassword | SPMLValidatePassword |
| auditTrialForUsersAdminOperations | UMUsersAdminOperations |
| auditTrialForUsersProvisioningOperations | UMUsersProvisioningOperations |
| auditTargetsForUsersAdminOperations | UMTargetsAdminOperations |
| auditStatusOfUsersProvisioningInformation | UMUsersProvisioningOperations |
| auditStatusOfTargetsAccountInformation | UMTargetsAccountInformation |
| auditTrialForTargetsAccountUsage | UMTargetsAccountUsage |
| auditTrialForTargetsAuthorizationUsage | UMTargetsAuthorizationUsage |

WSDL is implemented for defining the web service. AMS5520's inputs, outputs, operations, and message types are identified with WSDL and according to this information in WSDL, AMS5520 and TMF615 are mapped.
An example for WSDL is given below:

```
<wsdl:operation name="addUser">
    <soap:operation soapAction="addUser" style="document"/>
    <wsdl:input>
      <soap:header message="aluWS:addUser" part="header" use="literal"/>
      <soap:body parts="body" use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:header message="aluWS:addUserResponse" part="header"
use="literal"/>
      <soap:body parts="body" use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ProcessingFailureException">
      <soap:fault name="ProcessingFailureException" use="literal"/>
    </wsdl:fault>
</wsdl:operation>
```

Above is a WSDL code. As seen from the code, types of the input and output and in which format they are transmitted are all specified by the WSDL.

Connection to AMS5520 is realized through the WSDL. Apache CXF is used for making the connection. CXF is one of the Apache frameworks which are used in web service application. Apache CXF has a code generator tool. By using Apache CXF's code generator tool WSDL codes are transformed into Java codes. TMF615 and AMS5520's fields are mapped with using these transformed codes.

Spring framework is used in the thesis. Spring framework is an open source framework for Java platform. There are multiple properties of Spring Framework and therefore, it is used in the thesis. Some of the Spring Framework's features are summarized below;

i. Bean Management
ii. Simplify management of Apache CXF
iii. Transaction Management
iv. Provide Aspect Oriented Programming (AOP) support

### 3.2.4 Implementation of Processing Engine (Mapper) Module

According to TMF615 standard, mapping is realized between the client adaptor and server adaptor. This interface is built on the AMS5520 M-Object API.

Mapping is done between the TMF615 and AMS5520 as follows;

i. Mapping of the Operations between the TMF615 and AMS5520;
    a. TMF615 is identified with SPML and UM, which is developed by TMF Forum, standards.
    b. AMS5520 doesn't support all the TMF615 operations. Although some of the operations are supported by AMS5520, some parameters of the TMF615 operations aren't equivalence to AMS5520 operations' parameters. However, this doesn't affect mapping the fields. At the following table, mappings are identified.

#### 3.2.4.1 Implementation of general mapping

There are multiple provision data for one user in TMF615 standard. Each of the provision data has more than one account information. However, in AMS5520 users have only one account and therefore, there is only one provision data for one user and only one account is specified for provision data.

Some parameter names are given to TMF615 standard parameters. For instance; "AMS" is given as a name to "accountController" variable. This thesis is integration of AMS5520 and TMF615 because of this situation; different account information and another "accountController" variable name except "AMS" aren't required. User and

account are the same concept when AMS5520 are taken into account. Therefore, “userId” and “accountId” get the same values as specified in TMF65 specifications and they can be sometimes substituted each other by the TMF615. In addition to this, as defined in SPML specifications “PSOIdentifier.Id” variable can be used as “userId” variable when it is required. Main purpose in here is that realizing SPML and TMF615 standards completely and mapping them with AMS5520.

#### 3.2.4.2 Implementation of add user operation mapping

Some variables are included by TMF615. However, these aren’t included by AMS5520 and vice versa. For instance, AMS5520 has Tl1Commands variable, but TMF615 doesn’t include Tl1Commands variable. Table 3.4 shows how add user operation’s fields are mapped according to TMF615 protocol.

#### 3.2.4.3 Implementation of modify user operation mapping

Difference between the add user operation and modify user operation is that modify user operation uses “PSOIdentifier” object which has the “Id” variable. In this operation, “Id” variable is defined to modify “userId” variable. However, AMS5520 doesn’t give permission to change the user name and therefore, three id variables must get the same id value. Mapping of the modify user operation’s fields are shown in Table 3.5.

#### 3.2.4.4 Implementation of remove user operation mapping

Id, which comes from TMF615, is mapped with name, which is included in AMS5520, for remove user operation. Remove user operation mapping is described in Table 3.6.

#### 3.2.4.5 Implementation of suspend user operation mapping

Id from TMF615 is mapped with name from AMS5520 for suspend user operation. Duration date is defined in TMF615 for the suspend user operation. However, AMS5520 doesn't define duration date and for this reason, duration date in TMF615 isn't mapped with any variables from AMS5520. Mapping of the suspend user operation's fields are shown in Table 3.7.

**Table 3.3: Operation mapping table**

| TMF615 Standart | AMS 5520 | TMF615 WSDL |
|---|---|---|
| addUser | addUser | SPMLAddRequest |
| removeUser | deleteUser | SPMLDeleteRequest |
| modifyUser | modifyUser | SPMLModifyRequest |
| suspendUser | suspendUser | SPMLSuspendRequest |
| resumeUser | resumeUser | SPMLResumeRequest |
| listUsers | listUser | SPMLLookupRequest |
| listUsers | listUser | UMListUsers |
| expirePassword | expirePassword | SPMLExpirePasswordRequest |
| resetPassword | resetPassword | SPMLResetPasswordRequest |
| isUseActive | listUser | SPMLActiveRequest |
| setPassword | resetPassword | SPMLSetPasswordRequest |
| listTargets | - | SPMLListTargets |
| validatePassword | - | UMValidatePassword |
| auditTrialForUsersAdminOperations | - | UMUsersAdminOperations |
| auditTrialForUsersProvisioningOperations | - | UMUsersProvisioningOperations |
| auditTargetsForUsersAdminOperations | - | UMTargetsAdminOperations |
| auditStatusOfUsersProvisioningInformation | - | UMUsersProvisioningInformation |
| auditStatusOfTargetsAccountInformation | - | UMTargetsAccountInformation |
| auditTrialForTargetsAccountUsage | - | UMTargetsAccountUsage |
| auditTrialForTargetsAuthorizationUsage | - | UMTargetsAuthorizationUsage |

#### 3.2.4.6 Implementation of resume user operation mapping

In AMS5520, user cannot define any other account other than oneself and resume user operation only needs the user name and therefore, “accountResumeInfoList” is neglected. According to TMF615 document, using of the “accountResumeInfoList” isn’t already mandatory. However, just in case controlling of “accountId” and “accountController” variables are done. Controlling of the “validationInformation” variable is impossible because “validatePassword” operation isn’t supported by AMS5520. Some fileds of resume user operation are mapped according to TMF615 protocol. These are explained in Table 3.8.

#### 3.2.4.7 Implementation of SPML list user operation mapping

There are two list user operations. These are UM List User Operation and SPML List User Operation. All of these operations have returnData variable and this variable can take three different values which are “EVERYTHING”, “DATA” and “IDENTIFIER”. These three different values specify sizes of the user information which are returned as a result by list user operation.

Below which value returns which information is explained;

i. EVERYTHING: All user information is returned. userData (user personal information), userAccountData (user basic account information) and provisioningData (user detailed account and authorization information) are all turned objects of the EVERYTHING.
ii. DATA: Personal and personal account information, which are associated with user, are returned by DATA. userData and userAccountData are the turned objects.
iii. IDENTIFIER: Only user name is turned. Turned object is PSOIdentifier (user name is defined as PSOIdentifier obejct’s “Id” variable).

**Table 3.4: Add user operation mapping**

| TMF615 | AMS5520 | Description |
|---|---|---|
| userData | | |
| **\|- userId** | **name** | |
| \|- firstname | - | |
| \|- lastName | - | |
| \|- email | - | |
| \|- contactOfficeNo | - | |
| \|- employeeId | - | |
| \|- addressLine1 | - | |
| \|- addressLine2 | - | |
| \|- addressLine3 | - | |
| \|- contactMobileNo | - | |
| provioningDataList[] | | Sending only one. |
| \|- accountData[] | | Sending only one. |
| **\|- accountId** | **name** | |
| **\|- validationInformation** | **password** | |
| **\|- accountController** | **"AMS"** | |
| \|- validationCategory | - | |
| \|- accountCategory | - | |
| \|- accountUsageCategory | - | |
| \|- accountAdminStatus | - | |
| \|- targetData | | |
| **\|- targetName** | **"AMS"** | |
| \|- accessProfileData[] | | Sending only one. |
| **\|- accessProfileName** | **"PAPGroup"** | |
| \|- accessProfileInfo[] | | |
| **\|- accessProfileValue** | **papgroup** | |
| \|- accessProfileSubValue[] | - | |
| **\|- excludeFlag** | **false** | |
| \|- authorizationData | | |
| **\|- roleName[]** | **role** | |
| \|- assignedAuthorization | - | |
| \|- scheduleData | **-** | |
| \|- provisioningDataId | - | |

**Table 3.5: Modify user operation mapping**

| TMF615 | AMS | Description |
|---|---|---|
| PSOIdentifier | | |
| **\|- Id** | **name** | |
| userData | | |
| **\|- userId** | **name** | |
| \|- firstname | - | |
| \|- lastName | - | |
| \|- email | - | |
| \|- contactOfficeNo | - | |
| \|- employeeId | - | |
| \|- addressLine1 | - | |
| \|- addressLine2 | - | |
| \|- addressLine3 | - | |
| \|- contactMobileNo | - | |
| provioningDataList[] | | Sending only one. |
| \|- accountData[] | | Sending only one. |
| **\|- accountId** | **name** | |
| **\|- validationInformation** | **password** | |
| **\|- accountController** | **"AMS"** | |
| \|- validationCategory | - | |
| \|- accountCategory | - | |
| \|- accountUsageCategory | - | |
| \|- accountAdminStatus | - | |
| \|- targetData | | |
| **\|- targetName** | **"AMS"** | |
| \|- accessProfileData[] | | Sending only one. |
| **\|- accessProfileName** | **"PAPGroup"** | |
| \|- accessProfileInfo[] | | |
| **\|- accessProfileValue** | **papgroup** | |
| \|- accessProfileSubValue[] | - | |
| **\|- excludeFlag** | **false** | |
| \|- authorizationData | | |
| **\|- roleName[]** | **role** | |
| \|- assignedAuthorization | - | |
| \|- scheduleData | **-** | |
| \|- provisioningDataId | - | |

List user operations can be applied in two different manners;

a) SPMLLookupRequest
b) UMListUsers

**Table 3.6: Remove user operation mapping**

| **TMF615** | **AMS** | **Description** |
|---|---|---|
| PSOIdentifier | | |
| **\|- id** | **name** | |

**Table 3.7: Suspend user operation mapping**

| **TMF615** | **AMS** | **Description** |
|---|---|---|
| PSOIdentifier | | |
| **\|- id** | **name** | |
| effectiveDate | - | |

**Table 3.8: Resume user operation mapping**

| **TMF615** | **AMS** | **Description** |
|---|---|---|
| PSOIdentifier | | |
| **\|- id** | **name** | |
| accountResumeInfoList[] | | Optional |
| \|- accountResumeInfo | | |
| \|- accountId | name | |
| \|- accountController | "AMS" | |
| \|- validationInformation | password | |

In this thesis, SPMLLookupRequest is implemented. This list user operation gets only one user name at a time and returns this user information according to “EVERYTHING”, “DATA” and “IDENTIFIER” objects. Table 3.9 shows SPML list user operation mapping.

**Table 3.9: SPML list user operation mapping**

| TMF615 | AMS | Description |
|---|---|---|
| PSOIdentifier | | |
| **\|- id** | **name** | |
| returnData | - | |

### 3.2.4.8 Implementation of is user active operation mapping

Although, there is no corresponding operation for questioning the user status in AMS5520, "listUser" operation returns the user status information and therefore, user status information is gotten from "listUser" operation and in this way, is user active operation is implemented.

**Table 3.10: Is user active operation mapping**

| TMF615 | AMS | Description |
|---|---|---|
| PSOIdentifier | | |
| **\|- id** | **name** | |

## 3.3 SYSTEM TEST AND INTEGRATION

Integration and testing of all reformed modules are realized. All the modules results, which come from the tests, come convenient to TMF615 standard. Before making the tests, firstly some testing plans are done and tests & system integration are done according to these plans. All the committed plans, tests and their results are shown in the following subsections.

### 3.3.1 Add User OperationTesting Plan

Purpose of the test is adding new user to AMS. Test steps and test results are shown below;

a. Test Requirements:

i. Network configuration.

ii. AMS (Repository) server is up and load is ready under its location.

b. Actions:

a. Part I:

i. Start with test tool.

ii. Define specific userId.

a. Regular id.

b. Regular id with spaces.

iii. accountId must be same with userId.

iv. Define validationInformation regularly.

v. accountController must be “AMS”.

vi. targetName must be “AMS”.

vii. accessProfileInfoName must be “PAPGroup”.

viii. excludeFlag must be “FALSE”.

ix. Add new user from the AddUser page in the test took with define userId.

b. Part II:

x. Try to add same user from the AddUser page in the test tool with defined userId again.

c. Part III

xi. Remove user with same userId from the removeUser page in the test tool.

xii. Add same user from the AddUser page in the test tool with defined userId by giving roleNameList and accessProgileInfoList.

c. Expected Results:

a. Part I:

i. User id added in the AMS.

ii. Response status must be “SUCCESS”.

b. Part II:

i. Expected exception is “userAlreadyExists”.

ii. Response status must be “FAILURE”.

c. Part III:

i. User is added in the AMS.

ii. Response status must be “SUCCESS”.

#### 3.3.1.1 Add user with exceptions testing plan

There are also some TMF615 exceptions if invalid parameters are entered by the user. These situations’ testing is also done. Add user with exceptions testing plan is shown below. Purpose of this testing plan is to test exceptions while adding user to AMS.

a. Test requirements:

i. Network configuration.

ii. AMS (Repository) server is up and load is ready under its location.

b. Actions:

i. Start with test tool.

**Exceptions occur because of these cases;**

ii. Null userId,

iii. UserId is empty string,

iv. UserId is empty string with spaces,

v. UserId length is unbounded. Length must be between 1 and 32.

vi. accountId is different from userId.

vii. validationInformation length is not between 1 and 16.

viii. accountController is not “AMS”.

ix. targetName is not “AMS”.

x. targetData is null.

xi. accessProfileInfoName is not “PAPGroup”.

xii. excludeFlag is not “FALSE”.

xiii. provisioningData size different from 1.

xiv. accountData size different from 1.

xv. accessProfileData size is bigger than 1.
xvi. If accessProfileData size is 1, accessProfileInfo size must be bigger than 0.

c. Part II:
xvii. ProfileAccessValue does not exist in AMS.
xviii. RoleName does not exist in AMS.

d. Expected Results:
i. Operation must be failed for each case.
ii. Expected exception is “operation_failed_invalid_input_parameter”. Except for targetName is not equal to “AMS”. Its exception is “noValidTargetData”.
iii. Response status must be “FAILURE”.

e. Part II Results:
i. User does not add in the AMS.
ii. Response status must be “FAILURE”.

#### 3.3.1.2 Modify user operation testing plan

Purpose of the test is modifying a user successfully. Table 2.13 shows the testing plan of modify user without caring of modify user exceptions which are defined in TMF615. Therefore, another testing plans are done for testing modify user according to TMF615 exceptions. These are;

i. Modifying non-existing user to AMS.
ii. Modify a user in the AMS with inappropriate variables.

Modify user testing plan is explained below;

a. Test requirements:

i. Hardware and software network configuration done.
ii. AMS server is up and ready.

b. Actions:
i. Use test tool and and “modifyUser.jsf” page.
ii. Select appropriate userId which is **already** in use at the AMS.
iii. Use accountId and psoId same as userId.
iv. Other options can be default for integration issues but roles and papGroups must be different than existing one to understand is user modified or not.
v. Modify a user by clicking “Modify” button.

c. Expected results:
i. Response status should be “SUCCESS”.
ii. Also check that is user added to AMS successfully.

#### 3.3.1.3 Modifying non-existing user in AMS testing plan

Purpose of the test is checking behavior of program while trying to modifying non-existing user.

a. Test Requirements:
i. Hardware and software network configurations are done.
ii. AMS server is up and ready.

b. Actions:
i. Use test tool and “modifyUser.jsf” page.
ii. Select appropriate userId which is **not** in use at the AMS.
iii. Use accountId and psoId same as userId.
iv. Other options can be default for integration issues
v. Modify a user by clicking “Modify” button.

c. Expected results:

i. Response status should be “FAILURE”.

ii. Also exception message must equals to “userDoesNotExist”

#### 3.3.1.4 Modifying a user in AMS with inappropriate variables

Purpose of the test is checking behavior of program while trying to modifying a user with inappropriate variables.

a. Test Requirements:

i. Hardware and software network configuration done.

ii. AMS server is up and ready.

b. Actions:

i. Use test tool and “modfiyUser.jsf” page. This test has many cases and some cases have same result so, cases are grouped by exception name.

Case 1 : “operation_failed_invalid_input_parameter”

a. userId equals empty string.

b. userId equals empty string with left and right spaces.

c. userId equals null.

d. userId does not equal to accountId or/and psoId

e. account controller is different than “AMS”

f. modificationMode equals to “add” or “delete”

Case 2 : “noValidTargetData”

g. targetName is different than “AMS”

h. accessProfileName is different than “PAPGroup”

Case 3 :

i. “operation_failed_unsupported_optional_input_parameter” excludeFlag equals to “true”

c. Expected results:

i. Response status should be “FAILURE”.

ii. Also exception message must equals to case exception.

iii. Also check that is user not added to AMS.

### 3.3.2 Lookup User (SPML List User) OperationTesting Plan

Purpose of the test is listing one user at a time. There are two other tests for Lookup User Method. These tests are;

i. Lookup a user which does not exist in AMS.

ii. Lookup a user from AMS with inappropriate variables.

SPML list user testing steps are listed below;

a. Test requirements:

i. Hardware and software network configuration done.

ii. AMS server is up and ready.

b. Actions:

i. Use test tool and and “lookUpUser.jsf” page.

ii. Select appropriate userId which is already in use at the AMS.

iii. Four case available

a. Case 1 : returnData is “default”

b. Case 2 : returnData is “everthing”

c. Case 3 : returnData is “data”

d. Case 4 : returnData is “identifier”

iv. List a user by clicking “lookUp” button.

c. Expected results:

i. Response status should be “SUCCESS”.

ii. User information should be listed.

iii. Also check that is user variables in the response are matches with AMS.

iv. Case results :

a. Case 1 and case 2 results should be same and they return userData, userAccountData and provisioningData but not contain identifier.

b. Case 3 returns userData and userAccountData but not contains identifier and provisoningData.

c. Case 4 returns only identifier but not contains userData, userAccountData and provisioningData.

#### 3.3.2.1 Lookup a non-exiting user in AMS testing plan

Purpose of the test is echecking behavior of program while trying to list non-existing user.

a. Tets requirements:

i. Hardware and software network configuration done.

ii. AMS server is up and ready.

b. Actions:

i. Use test tool and “lookUpUser.jsf” page.

ii. Exception: “operation_failed_invalid_input_parameter”

a. userId equals empty string.

b. userId equals empty string with left and right spaces.

c. userId equals null.

c. Expected results:

i. Response status should be “FAILURE”.

ii. Also response exception message must equals to exception.

#### 3.3.2.2 Lookup a user with inappropriate variable testing plan

Purpose of the test is listing many ysers at a time.

a. Test requirements:
   i. Hardware and software network configuration done.
   ii. AMS server is up and ready.

b. Actions:
   i. Use test tool and and “lookUpUser.jsf” page.
   ii. Select appropriate userIds which are already in use at the AMS.
   iii. Enter userIds with “comma”
   iv. Four case available;
      a. Case 1 : returnData is “default”
      b. Case 2 : returnData is “everthing”
      c. Case 3 : returnData is “data”
      d. Case 4 : returnData is “identifier”
   v. List a user by clicking “list” button.

c. Expected results:
   i. Response status should be “SUCCESS”.
   ii. Users’ information should be listed.
   iii. Also check that is user variables in the response are matches with AMS.
   iv. Case results :
      a. Case 1 and case 2 results should be same and they return userData, userAccountData and provisioningData but not contain identifier.
      b. Case 3 returns userData and userAccountData but not contains identifier and provisoningData.
      c. Case 4 returns only identifier but not contains userData, userAccountData and provisioningData.

### 3.3.3 Delete User Testing Plan

Purpose of the test is deleting user from AMS. Steps of testing plan are listed below;

a. Test requirements:
   i. Network configuration
   ii. AMS (Repository) server is up and load is ready under its location.

b. Actions:
   a. Part I:
      ii. Start with test tool.
      iii. Define a specific userId.
         a. Regular id.
         b. Regular id with spaces.
      iv. Add new user from the AddUser page in the test tool with defined userId.
      v. Remove user from the removeUser page with the defined userId.

   b. Part II:
      v. Try to remove user with the same userId again.

c. Expected results:
   a. Part I:
      i. User is removed from the AMS.
      ii. Response status must be “SUCCESS”.

   b. Part II:
      i. Expected exception is “userDoesNotExist”.
      ii. Response status must be “FAILURE”.

An addition testing plan is done for Delete User Method for testing TMF615 required exceptions. This testing plan is explained in the following subsection.

#### 3.3.3.1 Delete user from AMS with exceptions testing plan

Purpose of the test is to testing exceptions while deleting user from AMS. Steps of this testing plan are listed below;

a. Test requirements:
   i. Network configuration
   ii. AMS (Repository) server is up and load is ready under its location.

b. Actions:
   i. Start with test tool.
   ii. Define a specific userId. According to these conditions,
      a. Null,
      b. Empty string,
      c. Empty string with spaces,
      d. Null PSOIdentifier.
   iii. Remove user from the removeUser page with the defined userId.

c. Expected results:
   i. Operation must be failed because of the invalid userId.
   ii. Expected exception is “operation_failed_invalid_input_parameter”.
   iii. Response status must be “FAILURE”.

### 3.3.4 Suspend User Testing Plan

Purpose of the test is to suspend a user from AMS. This testing plan is explained below;

a. Test requirements:
   i. Network configuration
   ii. AMS (Repository) server is up and load is ready under its location.

b. Actions:
   a. PART I
      i. Start with test tool.

ii. Define a specific userId.
    a. Regular id.
    b. Regular id with spaces.
iii. Add new user from the AddUser page in the test tool with defined userId.
iv. Suspend user from the suspendUser page with the defined userId.
v. User status is checked by using isUserActive page with the defined userId.

b. PART II
vi. Delete user with the defined userId.
vii. Try to suspend user with the same userId again.

c. Expected results:
a. PART I
i. User is suspended from the AMS.
ii. Response status must be “SUCCESS”.

b. PART II
i. Expected exception is “userDoesNotExist”.
ii. Response status must be “FAILURE”.

#### 3.3.4.1 Suspend user with exceptions testing plan

Suspend User method is tested according to TMF615 exceptions. Purpose of the test is to testing exceptions while suspending user. Steps of testing plan;

a. Test requirements:
i. Network configuration
ii. AMS (Repository) server is up and load is ready under its location.

b. Actions:
i. Start with test tool.
ii. Define a specific userId. According to these conditions,

a. Null,

b. Empty string,

c. Empty string with spaces,

d. Null PSOIdentifier.

iii. Suspend user from the suspendUser page with the defined userId.

c. Expected results:

vi. Operation must be failed because of the invalid userId.

vii. Expected exception is “operation_failed_invalid_input_parameter”.

viii. Response status must be “FAILURE”.

### 3.3.5 Resume User Testing Plan

Purpose of the test is to resume a user from AMS. Testing steps are below;

a. Test requirements:

i. Network configuration

ii. AMS (Repository) server is up and load is ready under its location.

b. Actions:

a. Part I:

i. Start with test tool.

ii. Define a specific userId.

a. Regular id.

b. Regular id with spaces.

iii. Add new user from the AddUser page in the test tool with defined userId.

iv. Suspend user by using suspendUser page.

v. Resume user from the resumeUser page with the defined userId.

vi. User status is checked by using isUserActive page with the defined userId.

b. Part II:

v. Delete user with the defined userId.

vi. Try to resume user with the same userId again.

c. Expected results:
   a. Part I:
      i. User is resumed from the AMS.
      ii. Response status must be “SUCCESS”.
   b. Part II:
      i. Expected exception is “userOrAccountDoesNotExist”.
      ii. Response status must be “FAILURE”.

Another testing plan is done for testing Resume User method with exceptions. It is explained in the following subsection.

#### 3.3.5.1 Resume user with exceptions testing plan

Purpose of the test is to testing exceptions while resuming user. Testing plan is explained below;

a. Test requirements:
   i. Network configuration
   ii. AMS (Repository) server is up and load is ready under its location.
b. Actions:
   i. Start with test tool.
   ii. Define a specific userId. According to these conditions,
      a. Null,
      b. Empty string,
      c. Empty string with spaces,
      d. Null PSOIdentifier.
   iii. Resume user from the resumeUser page with the defined userId.
c. Expected results:
   i. Operation must be failed because of the invalid userId.
   ii. Expected exception is “operation_failed_invalid_input_parameter”.

iii. Response status must be “FAILURE”.

### 3.3.6 Is User Active Method Testing Plan

Purpose of the test is to control status of the user from AMS. Testin plan and its results are shown below;

a. Test requirements:
   i. Network configuration
   ii. AMS (Repository) server is up and load is ready under its location.

b. Actions
   a. Part I:
      i. Start with test tool.
      ii. Define a specific userId.
         a. Regular id.
         b. Regular id with spaces.
      iii. Add new user from the AddUser page in the test tool with defined userId.

   b. Part II:
      iv. Suspend user from the suspendUser page in the test tool.
      v. Control the user status by using isUserActive page.
   c. Part III:
      vi. Delete user with the defined userId.
      vii. Try to control status of user with the same userId again from the isUserActive page in the test tool.

c. Expected results:
   a. Part I:
      i. Response status must be “SUCCESS”.
      ii. Active variable must be “TRUE” in the response.

   b. Part II:
      i. Response status must be “SUCCESS”.

ii. Active variable must be “FALSE” in the response.

c. Part III:

i. Expected exception is “userDoesNotExist”.

ii. Response status must be “FAILURE”.

A testing plan is done for Is User Active method with exceptions which are defined in TMF615 (Tele Management Forum). It is explained in the following subsection.

#### 3.3.6.1 Is user active with exceptions testing plan

Purpose of the test is to testing exceptions while controlling status of the user. This testing plan is described as follows;

a. Test requirements:

i. Network configuration

ii. AMS (Repository) server is up and load is ready under its location.

b. Actions:

i. Start with test tool.

ii. Define a specific userId. According to these conditions,

a. Null,

b. Empty string,

c. Empty string with spaces,

d. Null PSOIdentifier.

iii. Control status of the user from the isUserActive page with the defined userId.

c. Expected results:

i. Operation must be failed because of the invalid userId.

ii. Expected exception is “operation_failed_invalid_input_parameter”.

iii. Response status must be “FAILURE”.

## 3.4 DEVELOPMENT AND PRODUCTION ENVIRONMENT

### 3.4.1 Development Environment

Many technologies are used in development environment. These are explained in the following subsections.

#### 3.4.1.1 Java

Java programming language is used for realizing this thesis implementation. Java is an object oriented programming language. Our integrated development platform (IDE) is Eclipse which is one of the best for programming in Java. Eclipse works with java so development platform is independent from operating systems. The development platform will be on Microsoft Windows 7, Microsoft Windows XP and Linux. Eclipse needs 256 MB of Random Access Memory (RAM) at least.

#### 3.4.1.2 Maven

Maven is used as a software project management tool in this project. Management of project's build, declaration and documentation can be done with Maven.

#### 3.4.1.3 Apache CXF

Web Service implementation has two methods;

i. Top to bottom: first code then WSDL (code 2wsdl)
ii. Bottom to top: first WSDL then code (wsdl2code)

Our implementation method is the wsdl2code. WSDL is an interface to make the communication. WSDLs are ready in the system. From these WSDLs, codes are generated by using code generation tools. There are many code generators work with Maven. We choose the Apache CXF among them.

Apache CXF is used to develop services. Services are developed with using frontend programming APIs (Application Programming Interfaces) such as Java API for XML Web Services (JAX-WS) and Java API for Restful Web Services (JAX-RS). Different protocols such as SOAP are used by these services to communicate with each other. Apache CXF has code generator tool for WSDL2JAVA. We use XML to Object binding, so Apache CXF supports many binding tool. However, we choose JAXB because it comes with jdk. And CXF is created by using spring framework. So, it can easily adapt to spring projects.

#### 3.4.1.4 Spring

Implementation of thesis works with Spring Framework. Open source application framework is offered by Spring Framework for the Java platform. The core features of the Spring Framework can be used by any Java application, but there are extensions for building web applications on top of the Java Enterprise Edition (EE) platform.

#### 3.4.1.5 Jax-ws

JAX-WS refers to Java API for XML for creating Web Services. Java EE platform has the JAX-WS. JAX-WS uses annotations to simplify the development and deployment of web service clients and endpoints. In this thesis, web services are created by using JAX-WS.

#### 3.4.1.6 Java server faces

JSF is used for making Java-based Web applications in this project. Test tool is implemented with JSF.

#### 3.4.1.7 Facelets

Facelets are used for constructing the Web template system in JSF. Valid input XML documents are used for the language. Facelets supports all of The JSF User Interface

(UI) components are supported by Facelets. JSF component tree is built by Facelets. View is reflected for a JSF application by the Facelts.

#### 3.4.1.8 Richfaces

RichFaces is a library which enables Ajax for JSF. Easy integration of Ajax capabilities into enterprise application development is provided by RichFaces. RichFaces library is used in this thesis to implement test tool.

#### 3.4.1.9 Primefaces

PrimeFaces is an Ajax enabled component library for Java Server Faces. It allows easy integration of Ajax capabilities into enterprise application development.

PrimeFaces is used in this thesis because it is more than just a component library for Java Server Faces. It adds;

i. Skin ability which means that application can be changed and updated easily,
ii. Component Development Kit (CDK) is used to help for constructing Java Server Faces components,
iii. Dynamic Resource Framework,
iv. Both page wide, and component based Ajax control components.

#### 3.4.1.10 Jaxb

Java Architecture for XML Binding (JAXB) provides to convert Java classes to XML representations (Keeney *et al.* 2005).
JAXB provides two main features to this project. These are;

i. Marshaling Java objects into XML.
ii. And the inverse, i.e. Un-marshalling XML back into Java objects.

Storing and retrieving data in memory in any XML format is done by JAXB.

#### 3.4.1.11 Jetty

Jetty is a basic Java-based HTTP server and servlet container (Application server). It is integrated with Maven. We use Jetty to make development speedily.

#### 3.4.1.12 Soapui

SoapUI is a Web Service Testing Tool which is used for Service Oriented Architectures. Java platform is used for building and Java Swing as Graphical User Interface (GUI) is used by SoapUI. SoapUI is a cross-platform. SoapUI supports Eclipse. We use SoapUI for testing the Web Services in this thesis.

### 3.4.2 Production Environment

Two tools are used for the production environment in this thesis. These are explained in the following subsections.

#### 3.4.2.1 Apache tompcat

Apache Tomcat is a servlet container. Apache Software Foundation (ASF) develops Apache Tomcat. Java Servlet and Java Server Pages (JSP) can be implemented by Tomcat. A pure Java HTTP web server environment for Java code to run is provided by Tomcat. We use the Tomcat for the production of the test tool.

#### 3.4.2.2 Jboss

JBoss Application Server is a Java EE-based application server. JBoss doesn't only the server it runs the Java, too. Especially, Java EE is implemented by JBoss. JBoss is Java-Based. For this reason, it is cross platform which means that any operating system, which support java, can use it. We use JBoss for the production of the main implementation part.

# 4. DISCUSSION AND EXPERIMENTAL RESULTS

In this section, we discuss the importance features of TMF615 protocol. We also provide our experimental results about the performance of our proposed system.

## 4.1 TMF615 PROTOCOL

TMF615 protocol initiates a new structure for Operational Support Systems by providing management of network elements via common communication platform. As a result of performed TMF615 protocol, service providers combine the new OSS systems and their applications quite easily and with minimum effort. This situation provides easy system management, high system performance and low system costs for the service providers.

Products of service providers, which use TMF615 interface, become more competitive and more favored in the market because of economical benefits of TMF615 interface in telecom sector. Today, many service providers use Operational Support Systems, which are produced by different producers and work with different protocols, in their network systems. This situation causes consistency and control problems for service providers, which work multinationally and intercontinentally.

When the literature is reviewed, it is seen that management of Operational Support Systems is rather difficult and number of people, who use this complex structure, is few before the TMF615 standard is formed. OSSs have to have high level security and with this direction, doing operations of user management fully convenient to secrutiy procedures makes more difficult the management of OSSs. Also, untapped accounts must be found and necessary procedures have to be implemented without they are exposed. In this situation, extra security precautions must be taken and new work layers should be included in general operations.

TMF615 provides solutions all the OSSs' problems, which are explained above. Account control becomes easier and user account problems are solved faster with our system. Also, it reduces energy usage and system complexity by minimizing number of using devices for realizing operations. In this way, lives of components, which are used in OSSs, become longer and discoverable errors are minimized. Companies have safer structures when devices, which are convenient to TMF615, are prepared according to company's security politicies and company's Operational Support Systems.

Generally, existing OSSs have problems because of complexity, robustness and security. It is concluded that all these problems are solved by using common communication platform, which is based on TMF615 protocol. This means that our realized system works very well in terms of easier management, less complex, more robust and more security. Therefore, our work remedies the weakness of the related works.

## 4.2 EXPERIMENTAL EVALUATIONS

Role of TMF615 can be defined as agent that holds one of frontend of OSS. Hence implementation of TMF615 doesn't have any database operation. As already shown from the Figure 4.2, most of the operation execution time is spent for OSS part of general system and this is caused by database operation at OSS. Furthermore, an index which is most probably used at the user table causes longer insertion time for records. Therefore adduser execution operation takes longer time among the operations.

In addition to the avoiding file operations, object to xml and xml to object transformations are used with specific frameworks (Apache CXF, JAXB) to make implementation of TMF615 is more stable and speedy. These frameworks are used for handling SOAP messages and generation web services.

As shown in Figure 4.2, there is difference between the bar charts with TMF and without TMF. The big difference is seen in the addUser operation. When TMF615

protocol is implemented, addUser operation takes about 7000 milli seconds delay. However, when TMF615 isn't implemented, this operation's delay decrrease to about 4000 milli seconds. The remaining operations' delays don't differ much with TMF and without TMF because adding a new user includes more information, which need to be mapped according to TMF615 protocol, than the other operations and getting response from the AMS5520 for addUser operation takes longer time than the other operations' response times. All of these show that, extra mapper layer adds extra delay when TMF is implemented.

On the other hand, although TMF615 agent operations take short execution time, TMF615 agent causes performance lost for entire OSS solution. Instead of TMF615 agent, native implementation of TMF615 at the OSS is the best choice according to the performance issues. But reimplementation of related OSS modules based on TMF615 and migration of whole systems are hard to perform.

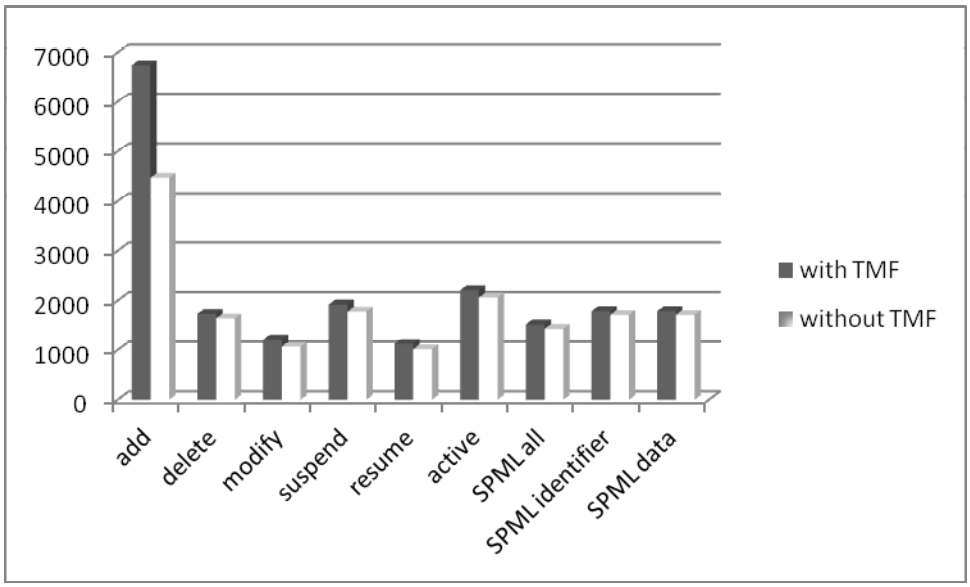
### 4.2.1 Delays of Operations' Layers with TMF615 Protocol

Operations pass through 5 layers until they reach AMS5520 and vice versa. These layers are;

a. Test tool
b. Client adapter
c. Mapper
d. Server adapter
e. AMS5520

Each operation's delays are measured from test tool to client adapter, from client adapter to mapper and from server adapter to AMS5520 when operation request comes from user via test tool. These delays are shown following subsections.

**Figure 4.2: Operations' delays with and without TMF**




#### 4.2.1.1 Add user operation layer delays

Add user operation request comes from test tool for adding a new user. After test tool gets request, it sends this request to client adapter, then client adapter sends it to mapper for mapping fields according to TMF615 protocol. When the mapping process is finished, request is sends to server adapter and server adapter transmits request to AMS5520. AMS5520 sends response when the add user operation is finished by using the same way with the request. Table 4.2 shows the delays of layers while add user operation is running.

**Table 4.3: Add user operation's layer delays**

| Add user request delays (milliseconds) | | | |
|---|---|---|---|
| Test tool to client adapter | Client adapter to mapper | Mapper to server adapter | Server adapter to AMS5520 |
| 1228 | 81 | 25 | 778 |
| Add user response delays (milliseconds) | | | |
| AMS5520 to server adapter | Server adapter to mapper | Mapper to client adapter | Client adapter to test tool |
| 3688 | 1 | 3 | 654 |

As seen from Table 4.2, the most time is taken by AMS5520 to server adapter way because AMS5520 isn't local and therefore, it takes longer time than the other ways.

Appendix A - Figure A.1 shows the add user request and response delays in the pie chart. In these pie charts, big slices are gotten by between server adapter and AMS5520 and between test tool and client adapter because getting response from AMS takes the longest time for realizing add user operation and transmitting request from test tool to client adapter causes big delay because of the parameters.

#### 4.2.1.2 Modify user operation layer delays

Modify user request goes to AMS5520 through 5 layers same as add user operation. Its request and response times are measured and delays are shown in Table 4.3 for making comparision with other operations' delay times.

When modify user operation's delays are compared with add user operations delays, modify user takes less time than add user because add user has many fileds, which is added to AMS5520, and adding these files take longer time than modifying these fields. In modify user operation; delay from AMS5520 to server adapter is the longest delay same as add user operation. This occurs because of the remote server, which is AMS5520.

**Table 4.4: Modify user operation's layer delays**

| Modify user request delays (milliseconds) | | | |
|---|---|---|---|
| Test tool to client adapter | Client adapter to mapper | Mapper to server adapter | Server adapter to AMS5520 |
| 137 | 0 | 3 | 72 |
| Modify user response delays (milliseconds) | | | |
| AMS5520 to server adapter | Server adapter to mapper | Mapper to client adapter | Client adapter to test tool |
| 999 | 1 | 1 | 519 |

Appendix A - Figure A.2 shows the modify user operation delays in pie chart. Getting response from AMS5520 to server adapter takes the big part in Appendix A - Figure A.2b because modifying user information in AMS is difficult. AMS firstly controls the information, which information is modified, and then it changes the modified information. Therefore, getting response's slice is 66% in Appendix A - Figure A.2b.

Modify request from test tool to client adapter takes the longest time as shown in Appendix A - Figure A.2a because user changes some parameters from test tool and test tool sends the entire user information parameters to client adapter. Sending all these information causes to be prepared many packets in the network. For this reason, network traffic occurs and delays increase.

When modify user response and request delays are compared. In request part, big delay occurs between test tool and client adapter. However, in response part, big delay occurs between the server adapter and AMS5520.

#### 4.2.1.3 Suspend user operation layer delays

Suspend user operation layer delays are shown in Table 4.4. As shown in Table 4.4, the most time is taken from AMS5520 to server adapter. AMS5520 gives response to server adapter in 1717 milliseconds which means that AMS5520's response for suspending a user is longer than AMS5520's reponse for modifying a user which is shown in Table 3.3.

Appendix A – Figure A.3 shows the suspend user request and response operation modules' delays. As shown in Appendix A – Figure A.3a, most time is taken by sending request from test tool to client adapter because getting parameter from the user and transmitting it to client adaptor takes more time than the other modules while making suspend user request. In suspend user response big slice of the pie chart is AMS5520 module delay for transmitting response from AMS5520 to server adapter. These delays are shown in Appendix A – Figure A.3b.

**Table 4.5: Suspend user operation's layer delays**

| Suspend user request delays (milliseconds) | | | |
|---|---|---|---|
| Test tool to client adapter | Client adapter to mapper | Mapper to server adapter | Server adapter to AMS5520 |
| 145 | 1 | 1 | 60 |
| Suspend user response delays (milliseconds) | | | |
| AMS5520 to server adapter | Server adapter to mapper | Mapper to client adapter | Client adapter to test tool |
| 1717 | 1 | 1 | 523 |

#### 4.2.1.4 Resume user operation layer delays

Realizing resume user operation in AMS5520 takes less time than suspending a user in AMS5520. This is understood when Table 4.4 and Table 4.5 are compared. Getting response from AMS for suspend user operation is 1717 milliseconds; however, time for resume user operation in AMS is 963 milliseconds.

Appendix A – Figure A.4 shows the resume user request and response delays in the pie chart. As seen in Appendix A – Figure A.4a most delay occurs between the test tool and client adapter while sending resume user request. In addition to Appendix A – Figure A.4b shows that resume user modules' response delays. As the same of suspend user response, big part in the pie chart is taken by AMS5520 to server adapter module because getting response from AMS5520 takes long time.

**Table 4.6: Resume user operation's layer delays**

| Resume user request delays (milliseconds) | | | |
|---|---|---|---|
| Test tool to client adapter | Client adapter to mapper | Mapper to server adapter | Server adapter to AMS5520 |
| 112 | 1 | 1 | 52 |
| Resume user response delays (milliseconds) | | | |
| AMS5520 to server adapter | Server adapter to mapper | Mapper to client adapter | Client adapter to test tool |
| 963 | 0 | 2 | 518 |

#### 4.2.1.5 Is user active operation layer delays

Is user active operation isn't defined in AMS5520. However, in this thesis, this operation is done with using list user operation. List user operation holds a parameter to determine the user status. Therefore, is user active operation firstly does the list user operation and looks the user status and then it gives response. As shown in Table 4.6, getting response from AMS5520 to server adapter takes long time because of realizing two operations, which is controlling user status by list user operation, at the same time.

Pie charts of is user active operation modules' delays are shown in Appendix A – Figure A.5. The results are the same with other operations which are add user, modify user, suspend user, and etc. As shown in Appendix A – Figure A.5a and Appendix A – Figure A.5b the most slice is taken by test tool to client adapter module and AMS5520 to server adapter module, respectively.

**Table 4.7: Is user active operation's layer delays**

| Is user active request delays (milliseconds) | | | |
|---|---|---|---|
| Test tool to client adapter | Client adapter to mapper | Mapper to server adapter | Server adapter to AMS5520 |
| 145 | 1 | 1 | 62 |
| Is user active response delays (milliseconds) | | | |
| AMS5520 to server adapter | Server adapter to mapper | Mapper to client adapter | Client adapter to test tool |
| 1999 | 1 | 1 | 525 |

#### 4.2.1.6 Lookup user operation layer delays

SPML Lookup user operation has three conditions. These are listed below:

i. Empty: All user information is returned.
ii. Identifier: Only username is listed.
iii. Data: Personal information of the user is returned.

Delays of these conditions are shown in Table 4.7, in Table 4.8 and in Table 4.9.

As seen from Table 4.7 and Table 4.8, getting response from AMS for SPML Lookup operation for empty condition, which returns all the user information, take the longer time than getting response form AMS for identifier condition because empty condition lists all the user information and getting all user information from AMS is more difficult than getting only the user name from AMS in identifier condition.

SPML Lookup user for data condition takes less time than the empty condition when Table 4.7 and Table 4.9 are compared becaue data condition lists only the user personal information and therefore, getting response from AMS for user information is faster than getting response from AMS for all the user information.

**Table 4.8: SPML Lookup user operation's delays for empty condition**

| Lookup user request delays for empty condition (milliseconds) | | | |
|---|---|---|---|
| Test tool to client adapter | Client adapter to mapper | Mapper to server adapter | Server adapter to AMS5520 |
| 76 | 0 | 1 | 43 |
| Lookup user response delays for empty condition (milliseconds) | | | |
| AMS5520 to server adapter | Server adapter to mapper | Mapper to client adapter | Client adapter to test tool |
| 1885 | 1 | 1 | 413 |

As seen in Appendix A – Figure A.6, Appendix A – Figure A.7 and Appendix A – Figure A.8, big delays occur while getting response from AMS5520 to server adapter. Getting response from AMS5520 for lookup user for empty condition as shown in Appendix A – Figure A.6b causes more delay than the other responses for lookup user identifier and data conditions which are shown in Appendix A – Figure A.7b and Appendix A – Figure A.8b. Lookup user request delays mostly occur while sending

request from test tool to client adapter as shown in Appendix A – Figure A.6a, Appendix A – Figure A.7a and Appendix A – Figure A.8a.

**Table 4.9: Operation's layer delays for identifier condition**

| Lookup user request delays for identifier condition (milliseconds) | | | |
|---|---|---|---|
| Test tool to client adapter | Client adapter to mapper | Mapper to server adapter | Server adapter to AMS5520 |
| 88 | 0 | 1 | 44 |
| Lookup user response delays for identifier condition (milliseconds) | | | |
| AMS5520 to server adapter | Server adapter to mapper | Mapper to client adapter | Client adapter to test tool |
| 1389 | 0 | 1 | 524 |

**Table 4.10: Operation's layer delays for data condition**

| Lookup user request delays for data condition (milliseconds) | | | |
|---|---|---|---|
| Test tool to client adapter | Client adapter to mapper | Mapper to server adapter | Server adapter to AMS5520 |
| 69 | 0 | 1 | 217 |
| Lookup user response delays for data condition (milliseconds) | | | |
| AMS5520 to server adapter | Server adapter to mapper | Mapper to client adapter | Client adapter to test tool |
| 1498 | 1 | 1 | 520 |

As a result, empty condition takes the longest time for listing all the user information and identifier condition takes the least time for listing user name. There are differences between these conditions because of the different sizes of information.

# 5. CONCLUSION AND SUGGESTIONS

In this thesis, we introduced a novel method for designing and developing common communication platform with TMF615 protocol for network components. TMF615 base interface standard enables to combine service providers' existing and their new OSSs and applications quite easily and with minimum effort. This simplifies the service providers' system management and increases the system performance and decreases the costs.

In our system, we implement all the supported operations of TMF615 protocol for realizing User Management Solutions in AMS5520. Some TMF615 operations aren't supported by the AMS5520. In that situation, we send unsupported operation response for these operations. Therefore, if these unsupported TMF615 operations are implemented to AMS5520 in the future, they can be mapped according to TMF615 protocol further on. As a result of implemented operations, all the operations are worked seamlessly.

System test and integration results show that all the parameters control and operations' messages work well. This means that operations are fully implemented according to TMF615 protocol.

Through the experimental evaluations carried out, delays of operations are calculated for measuring the our system performance according to TMF615 protocol and without TMF615 protocol. When the results are compared, operations' delays are higher in our system because of the extra layers which are test tool, client adaptor and processing engine. After then, we compute our layers' delays to see which layer causes the more delay and we see that sending request and getting response between test tool and client adapter cause more delay than the other delays. It occurs because of the processing engine layer which makes the mapping according to TMF615 protocol because client adaptor waits the mapping operation for sending and getting message from the processing engine. However, these delays aren't very important when the our system

complexity, security, interoperability, easy management, power saving, components lives and power savings advantages are considered.

## 5.1 SUGGESTIONS

UMS-C and UMS-L communicate by using WSDL/SOAP messages. HTTPS provides the security of the messages. However, some improvements can be achieved to get more secure systems. Some of the improvement suggestions are listed below;

i. Certification of security: Each vendor OSS can have specific security certificate. UMS-C and UMS-L communicate with this certificate. And authentication between the source and the destination is provided according to certificate. In this way, security can be increase.
ii. Informing service: UMS-C starts the communication between the UMS-C and UMS-L. UMS-C produces Provisioning ID which is used for tracking completion of requests. Provisioning ID is not used for only tracking; it is used for checking status of provisioning, too. For this reason, complexity of asynchronous requests increase. To reduce the complexity, a notification service can be used.
iii. Location change of a user: Service provider cannot migrate to TMF615 specification easily because existing user information must migrate to UMS-C. TMF615 specification must have a standard mechanism to make the migration.

# REFERENCES

***Books***


Harada, Y., Tomita, Y. & Sugiyama, S., 2003. *Operation support system*. Tokyo: Hitachi, Ltd.

Chou, J., 2007. *Operation support system for wireless networks.* Minneapolis: United States Patent Application Publication.

Nye, B. & Hong S., 2007. *High level operational support system*. United States of America: TeleCommunication Systems, Inc.

Richardson, T., 2007. Next generation network management. *The cable and telecommunications professionals' reference,* 3rd edition, Burlington: Elsevier, pp. 207-213.

Taniguchi, S. & Ebitani, K., 2010. *Operation support system.* Tokyo: United States Patent

Tselentis, G., Domingue, J., Galis, A., Gavras, A., Hausheer, D., Krco, S., Lotz, V. & Zahariadis, T., 2009. *Towards the future internet.* Amsterdam: IOS Press BV.

Uchikubo, A., Nakamura, T. & Gotanda, M., 2008. *Operaton support system.* Tokyo: Olympus Corporation.

Uchikubo, A., Mizuno, H. & Gotanda, M., 2000. *Remote operation support system and method.* Tokyo: Olympus Optical Co., Ltd.

Wright, S. L. & Ridgely C. H., 2010. *Method, computer program product, and apparatus for providing a universal high speed link for operation support system data.* United States of America: Verizon Services Corp.

Zhou, Y., Dong, Y., Huang, X. & Yoshikawa H., 2011. *A human interface toolkit for developing operation support system of complex industrial systems with IVI-COM technology.* Berlin: Springer-Verlag.

***Periodicals***


Aleem, M. İ., 2009. *Centralized user management and TMF615 report.* North America: Wipro Technologies.

Blum, N., Magedanz, T. & Schreiner, F., 2009. Management of SOA based NGN service exposure, service discovery and service composition. *Integrated Network Management, 2009. IM '09. IFIP/IEEE International Symposium.*pp. 430-437.

Blum, N., Magedanz, T., Schreiner, F. & Wahle, S., 2009. From IMS management to SOA based NGN management. *Springer Journal of Network and Systems Management*. **17** (1-2). pp. 33-52.

Chen, I. Y., Ni, G. K., Kuo, C. H. & Lin C. Y., 2010. A service-oriented management framework for telecom operation support systems. *Servive Systems and Service Management (ICSSSM), 2010 7th International Conference on, IEEE.* pp. 1-5.

Chen, I. Y., Ni, G. K. & Lin, C. Y., 2008. A service-oriented fault-tolerant Environment for telecom operation support systems. *2008 IEEE International Symposium on Service-Oriented System Engineering.* pp. 208-214.

Choi, B. C. & Seo, D. I., 2006. On the study of service security model for privacy using global user management framework. *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference.* pp. 4-116.

Chou, T. H., Seng, J. L. & Lin, B., 2008. eTOM and e-services based trouble-management operations: a large scale telecom case study. *International Journal of Technology Management*. **43** (4), pp. 383-403.

Faro, L. & Spina, E., 2007. The need for a customer-centric approach for services development and operation in the Brazilian telecommunications market. *Proceedings of the First International Conference on the Digital Society (ICDS'07).* pp. 23.

Frank, L, Luoma, E. & Tyrväinen, P., 2007. Market scope of vendors in the OSS software market. *Industrial Engineering and Engineering Management, 2007 IEEE International Conference.* pp. 2096-2100.

Gallen, C. R. & Reeve J. S., 2006. Using open source to realize an NGOSS proff of concept. *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP.* pp. 1-12.s

Gallen, C. R. & Reeve J. S.,2005. Investigating the feasibility of open development of operations support solutions. *Integrated Network Management, 2005. IM 2005. 2005 9th IFIP/IEEE International Symposium.* pp. 689-702.

Georgalas, N., Azmoodeh, M. & Ou, S., 2005. Model driven integration of standard based OSS components. *EURESCOM Summit 2005-Ubiquitous Services and Applications-Exploiting the Potential.* pp. 8.

Goestl, H., 2006. Using NGOSS principles in todays OSS/BSS projects NGOSS meets IMS/SDP. *Telecommunications Network Strategy and Planning Symposium, 2006. Networks 2006. 12th IEE International.* pp. 1-8.

Gupta, A., 2006. Network management: current trends and future perspectives. *Springer Journal of network and systems management.* **14** (4), pp. 483-491.

Hanemann, A., 2007. Refining ITIL/eTOM processes for automation in service fault management. *Business-Driven IT Management, 2007. BDIM'07. 2nd IEEE/IFIP International Workshop.* pp. 106-107.

Hao, Q. M., 2008. Toward a unified service delivery process for next-generation services. *Bell Labs Technical Journal.* **12** (4), pp. 5-20.

Huang, C. I. & Hsu, C.C., 2011. eTom/SOA for the digital convergence operation support system and process implementation. *Journal of Beijing University of Posts and Telecommunications.* **34** (2), pp. 39-44.

Huang, T. S., Lee, C. C., Chen, Y. M., Chen, C. W., Tsai, J. F. & Wu, H. J., 2003. A new architecture to operation support systems. *IEEE Tele*communications, *2003. ICT 2003. 10th International Conference.* pp. 1236-1242.

Kadowaki, K. & Fujita, S., 2009. A dynamic user management in networked consumer electronics via authentication proxies. *2009 International Conference on Parallel and Distributed Computing, Applications and Technologies.* pp. 195-200.

Keeney, J., Carey, K., Lewis, D., O'Sullivan, D. & Wade, V., 2005. Ontology-based semantics for composable autonomic elements. *Computer Science (Scholarly Publications).* pp. 1-8.

Kelly, M. B., 2003. Report: the telemanagement forum's enhanced telecom operations map (eTOM). *Springer Journal of Network and Systems Management.* **11** (1), pp. 109-119.

Lin, J., Lu, X., Yu, L., Zou, Y. & Zha, L., 2010. VegaWarden: A uniform user management system for cloud applications. *2010 Fifth IEEE International Conference on Networking, Architecture, and Storage.* pp. 457-464.

Liu, X., Azmoodeh, M. & Georgalas, N., 2007. Specification of non-functional requirements for contract specification in the NGOSS framework for quality management and product evaluation. *IEEE Fifth International Workshop on Software Quality (WoSQ'07).*pp. 7-7.

Matos, A., Matos, F., Simoes, P. & Monteiro, E., 2008. A framework for the establishment of inter-domain, on-demand VPNs. *Network Operations and Management Symposium, 2008. NOMS 2008, IEEE.* pp. 232-239.

Qi, J., Xu, L. D., Shu, H. & Li, H., 2006. Knowledge management in OSS-an enterprise information system for the telecommunications industry. *Special Issue: Knowledge Management in the ERP Era.* **23** (2), pp: 177-190.

Rader, R., 2011. Operations support systems for mission critical public safety communication networks. *Bell Labs Technical Journal.* **16** (3). pp. 151-162.

Ramanathan, S., Alexander & M, Kerr, G., 2008. The IBM telecommunications service delivery platform. *IBM Systems Journal.* **47**(3), pp. 433.

Raymer, D., Strassner, J., Lehtihet, E. & Meer, S., 2006. End-to-end model driven policy based network management. *Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'06).* pp. 4-70.

Shangguan, Z., Gao, Z. & Zhu, K., 2008. Ontology-based process modeling using eTOM and ITIL. *IFIP International Federation for Information Processing.* **255**, pp. 1001-1010.

Strassner, J. & Raymer, D., 2006. Implementing next generation services using policy-based management and autonomic computing principles. *Network Operations and Management Symprosium, 2006. NOMS 2006. 10th IEEE/IFIP.* pp. 1-15.

TMF Forum. 2010.*Telecom OSS operator user management report*. January.

Vrancic, A., Jurasovic, K., Kusek, M., Jesic, G. & Trzec, K., 2007. Service provisioning in telecommunication networks using software agents and rule-based approach.*Proceedings of the 9th International Conference on Telecommunications ConTEL 2007, Zabreb, Crotia.* pp. 1-5.

Wang, C. Y., Lin, P., Shih, C. S., Fu, H. L. & Jeng, J. Y., 2011. A middleware approach for migration of legacy telecom operational support systems into NGOSS-compliant. *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific, IEEE.* pp. 1-7.

Wang, L. & Lv, T., 2008. The NG-OSS evolution of telecom service providers: from network focused to customer-focused. *IFIP International Federation for Information Processing.* **255**, pp. 1207-1213.

Wong, D., Ting, C. & Yeh, C., 2007. From network management to service management-a challenge to telecom service providers. *Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on IEEE.* pp. 280.

Xu, K., Zhang, X., Song, M. & Song, J., 2009. Research on SLA management model in service operation support system. *Wireless Communications, Networking and*

*Mobile Computing, 2009. WiCom '09. 5th International Conference on, IEEE.* pp. 224-227.

Yahia, I. G. B., Bertin, E. & Crespi N., 2007. Ontology-based management systems for the next generation services: state-of-the-art. *Thir International Conference on Networling and Services (ICNS'07), IEEE.* pp.40.

***Other Publications***

IBM, Tivoli Software Solutions, 2010, http://www-01.ibm.com/software/tivoli/solutions/ [accessed 9 August 2011]

Joshy N. & Shenoy N., (2010). Integrating the eTOM Business Process Framework into the Ericsson Seamless Modeling. *Thesis for the M.A. Degree.* Eindhoven: Technische Universiteit Eindoven Mathematics and Computer Science.

Meyners, M., Driss, B. & Feger, U., 2008, TMF615 OSS identity management [online], TMF Forum, www.tmforum.org [accessed 1 August 2011]

Ramiro, J. M. D. A., (2009). Contribution to Operation Support Systems and Service Management Architectures for User-Centric Telecommunications Service Over Next Generation Networks. *Thesis for PhD. Degree.* Madrid: Universidad Politécnica de Madrid Departamento de Ingeniería de Sistemas Telemáticos

# APPENDICES

## APPENDIX A. FIGURES

**Figure A.1: Add user operation modules' request and response delays**

**Figure A.1a** Add user operation modules' request delays

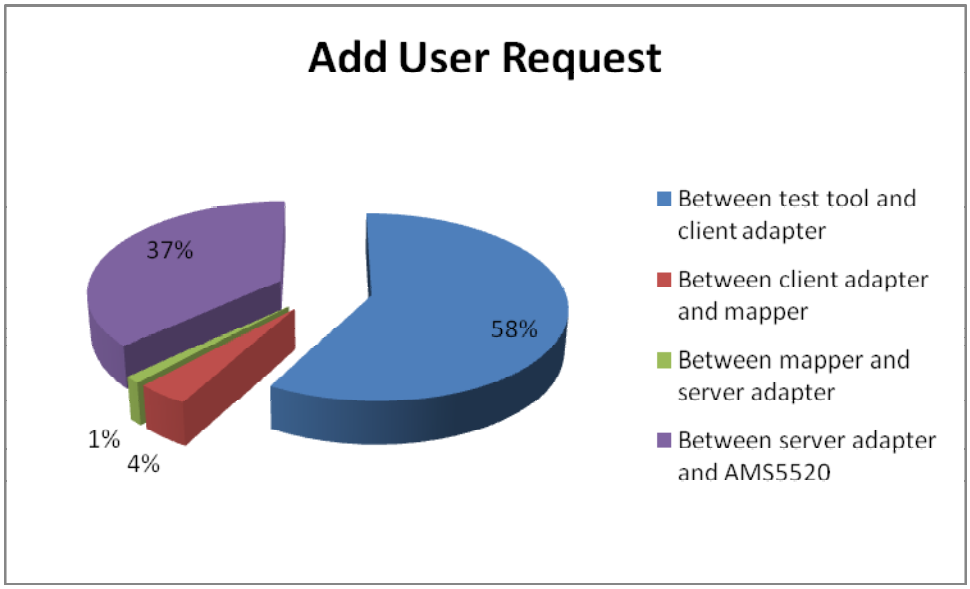



**Figure A.1b** Add user operation modules' response delays

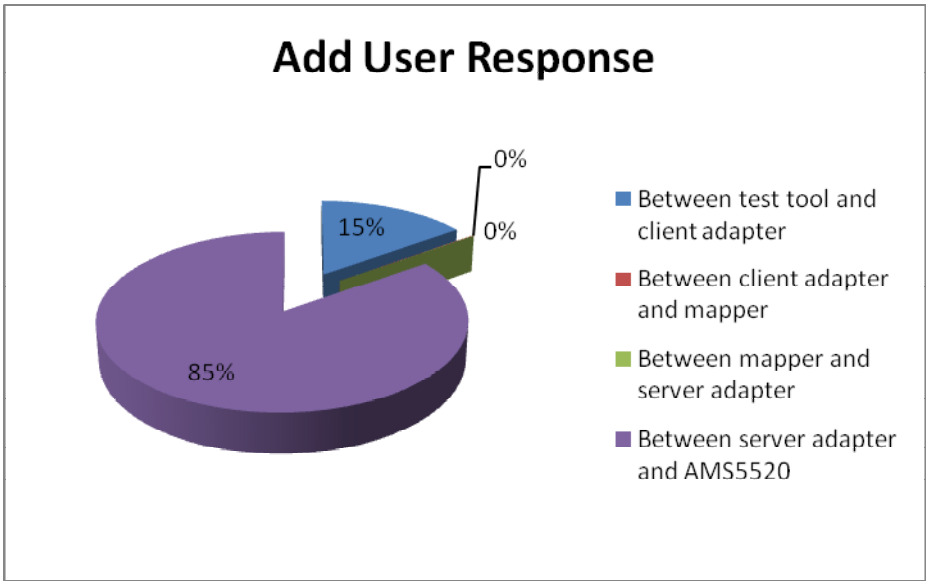


**Figure A.2: Modify user operation modules' request and response delays**

**Figure A.2a** Modify user operation modules' request delays

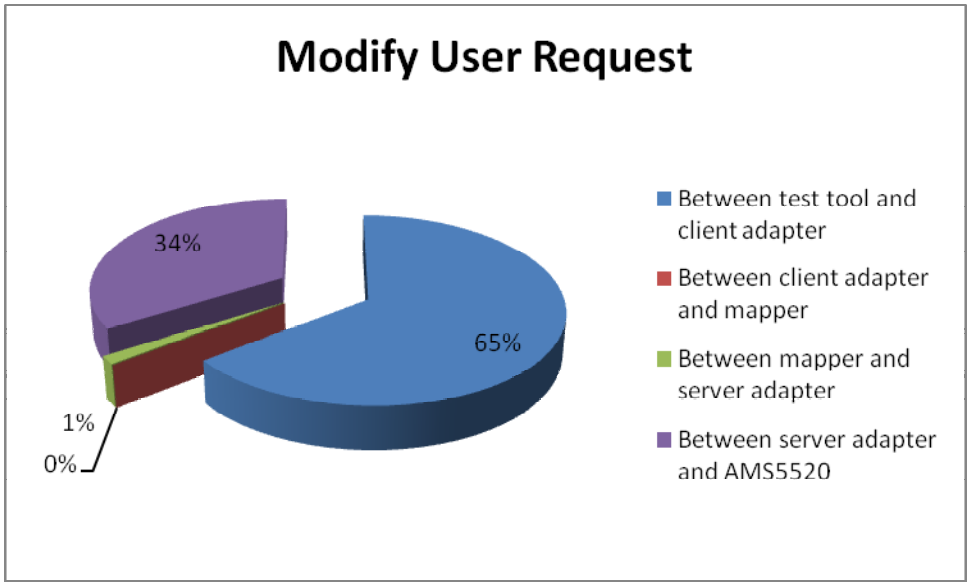



**Figure A.2b** Modify user operation modules' response delays

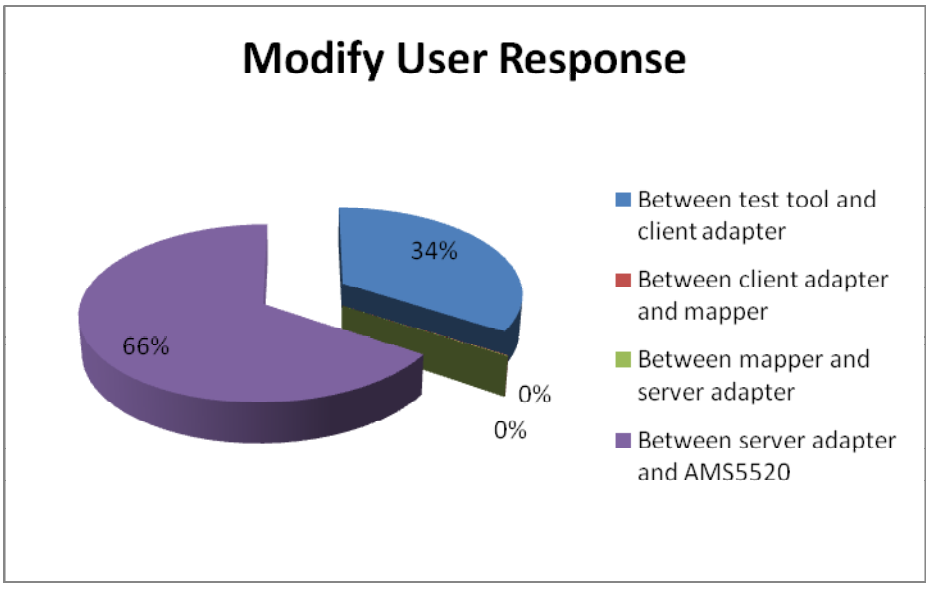


**Figure A.3: Suspend user operation modules' request and response delays**

**Figure A.3.a** Suspend user operation modules' request delays

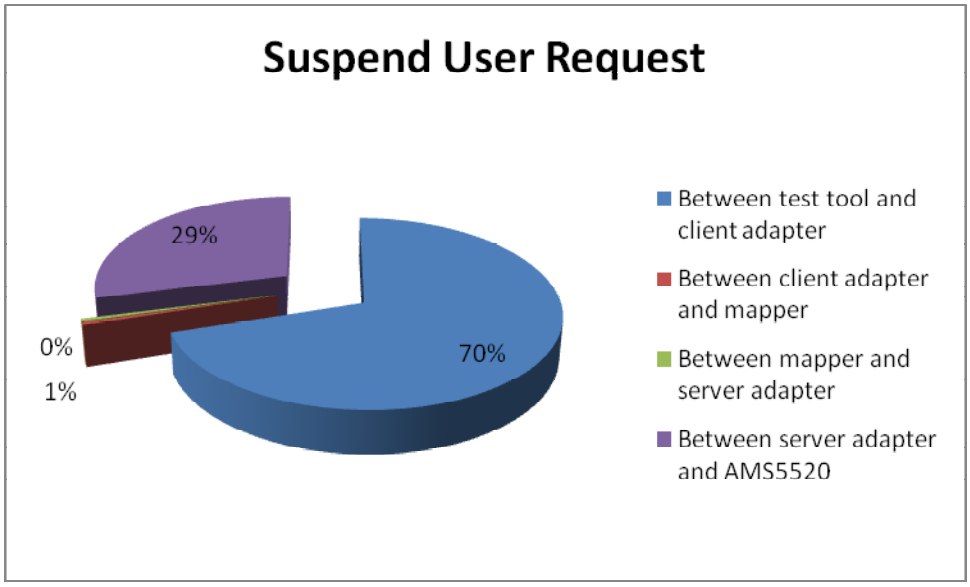



**Figure A.3b** Suspend user operation modules' response delays

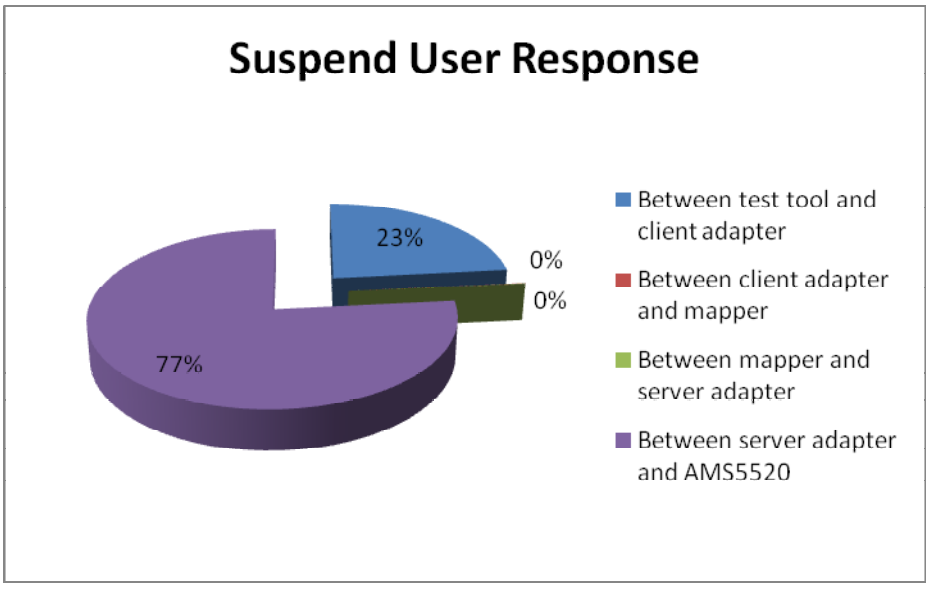


**Figure A.4: Resume user operation modules' request and response delays**

**Figure A.4a** Resume user operation modules' request delays

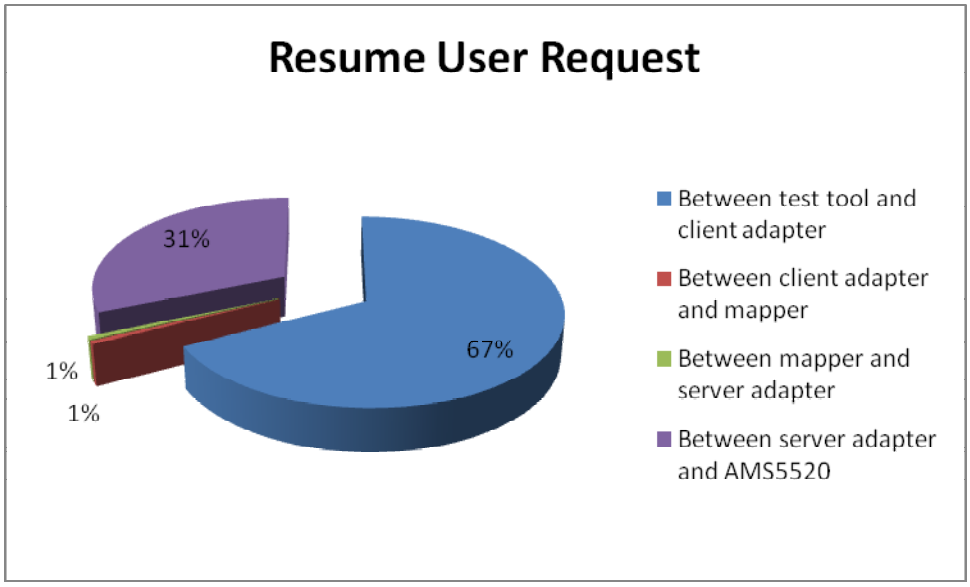



**Figure A.4b** Resume user operation modules' response delays

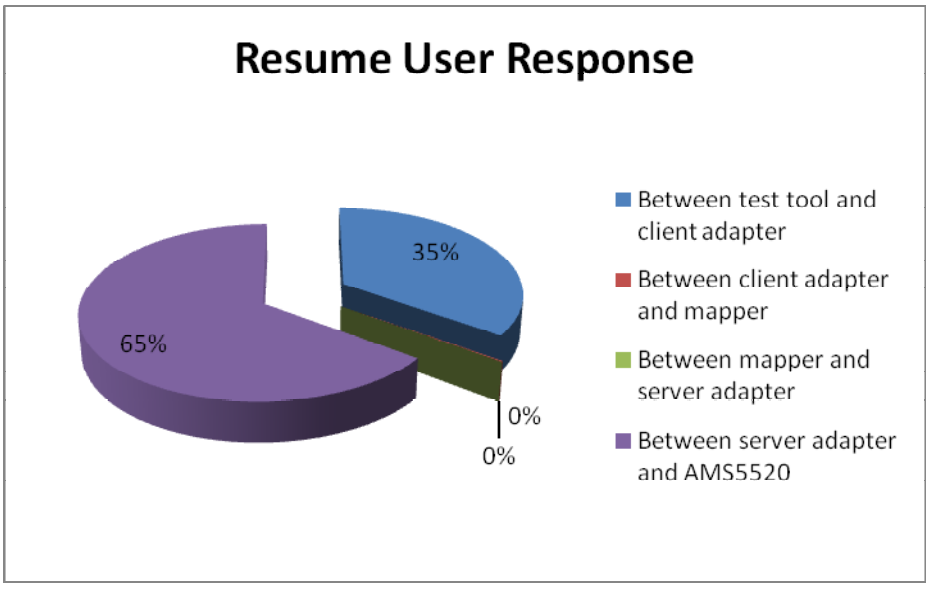


**Figure A.5: Is user active operation modules' request and response delays**

**Figure A.5a** Is user active operation modules' request delays

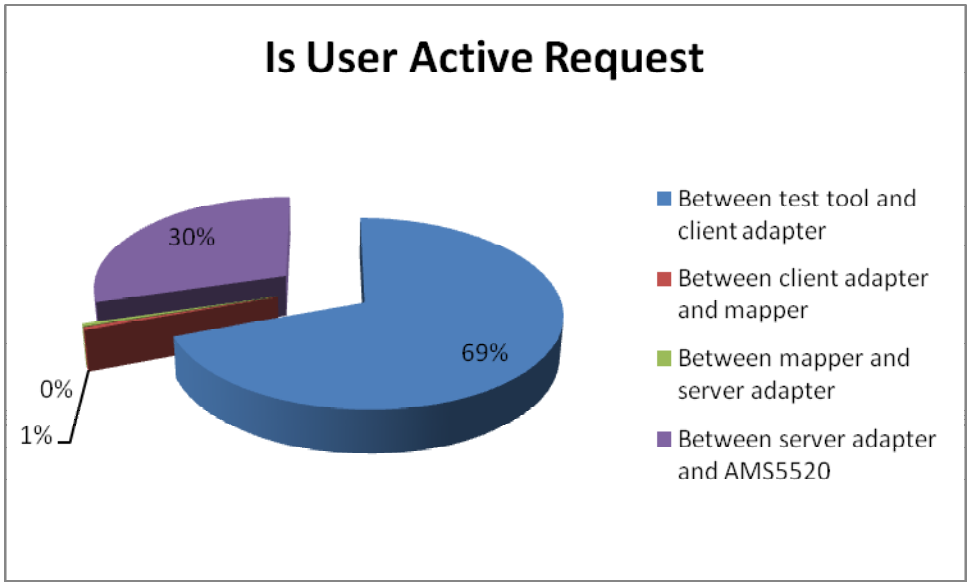



**Figure A.5b** Is user active operation modules' response delays

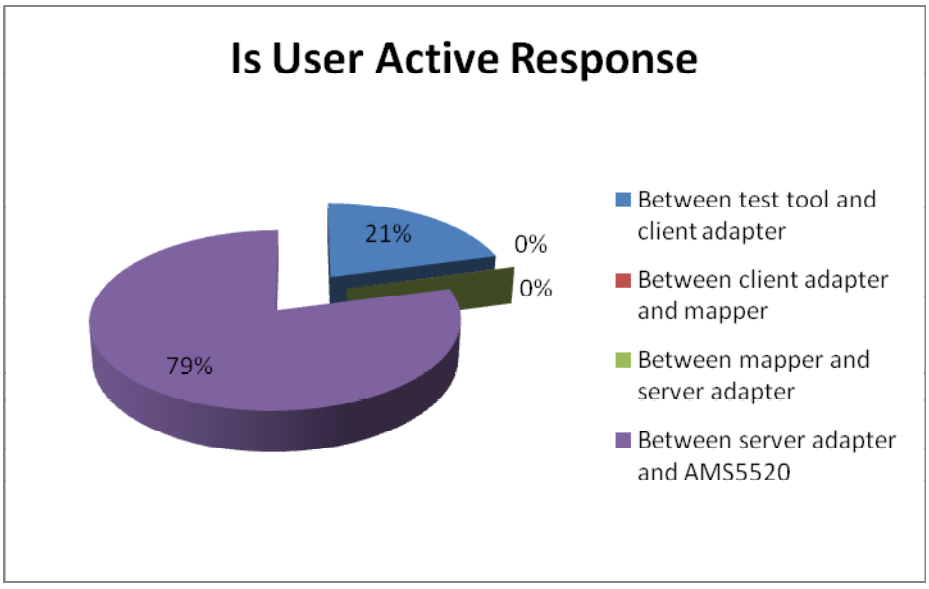


**Figure A.6: Lookup user for empty condition operation's request and response delays**

**Figure A.6a** Lookup user for empty condition operation modules' request delays

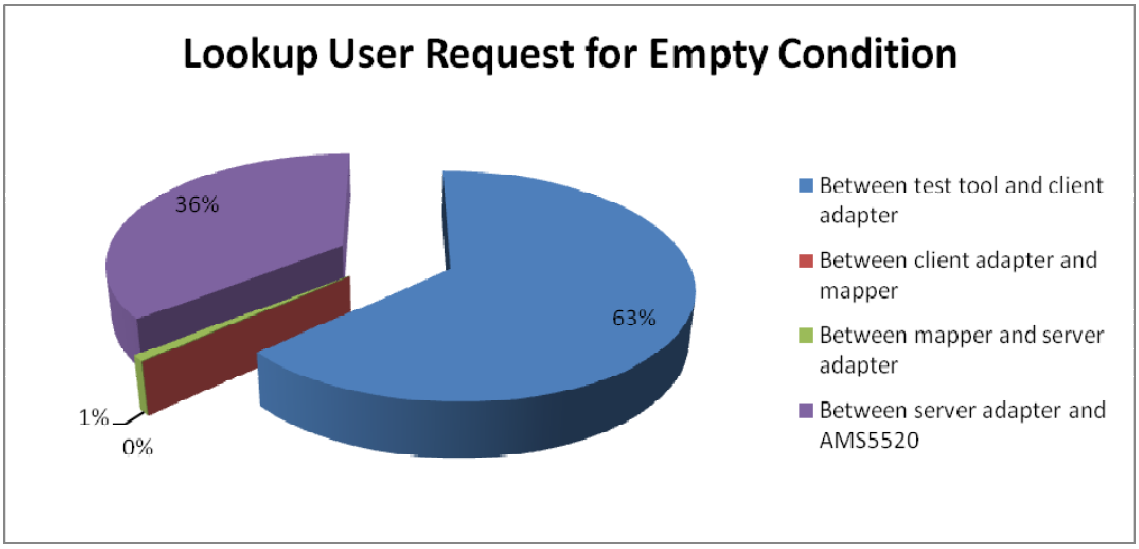



**Figure A.6b** Lookup user for empty condition operation modules' response delays

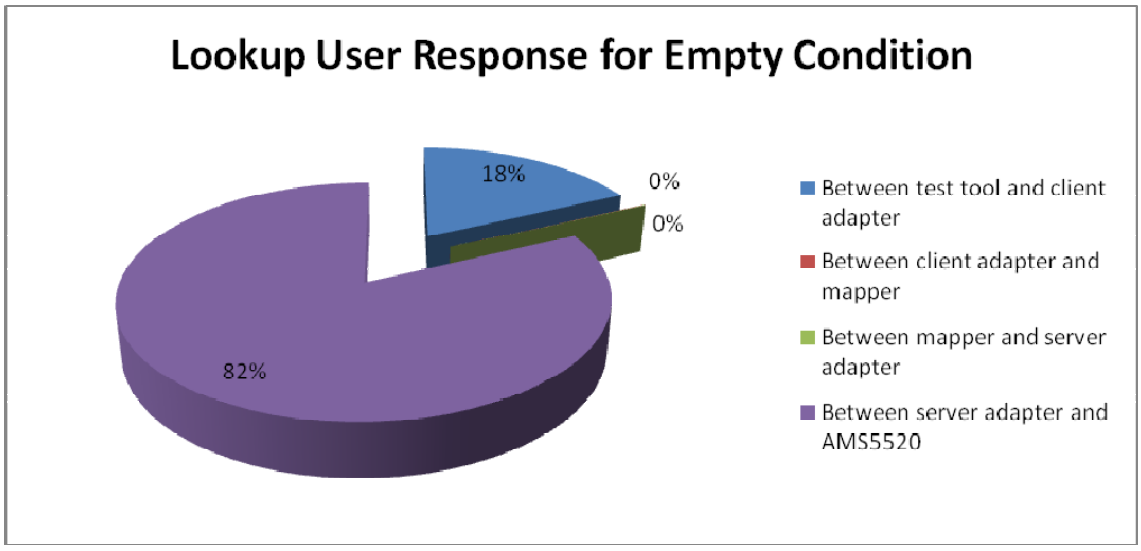


**Figure A.7: Lookup user for identifier condition operation's request and response delays**

**Figure A.7a** Lookup user for identifier condition operation modules' request delays

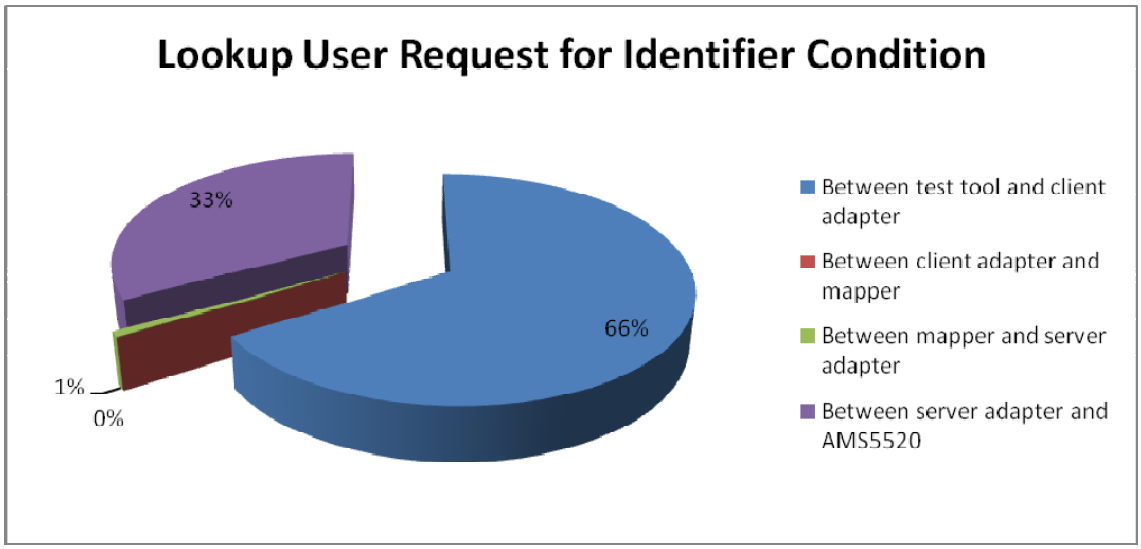



**Figure A.7b** Lookup user for identifier condition operation modules' request delays

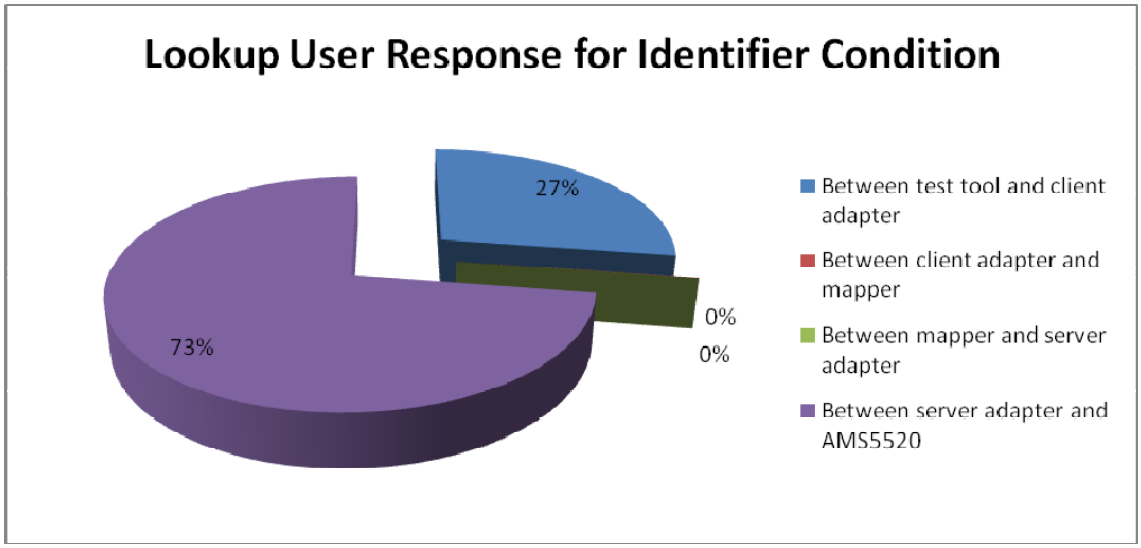


**Figure A.8: Lookup user for data condition operation's request and response delays**

**Figure A.8a** Lookup user for data condition operation modules' request delays

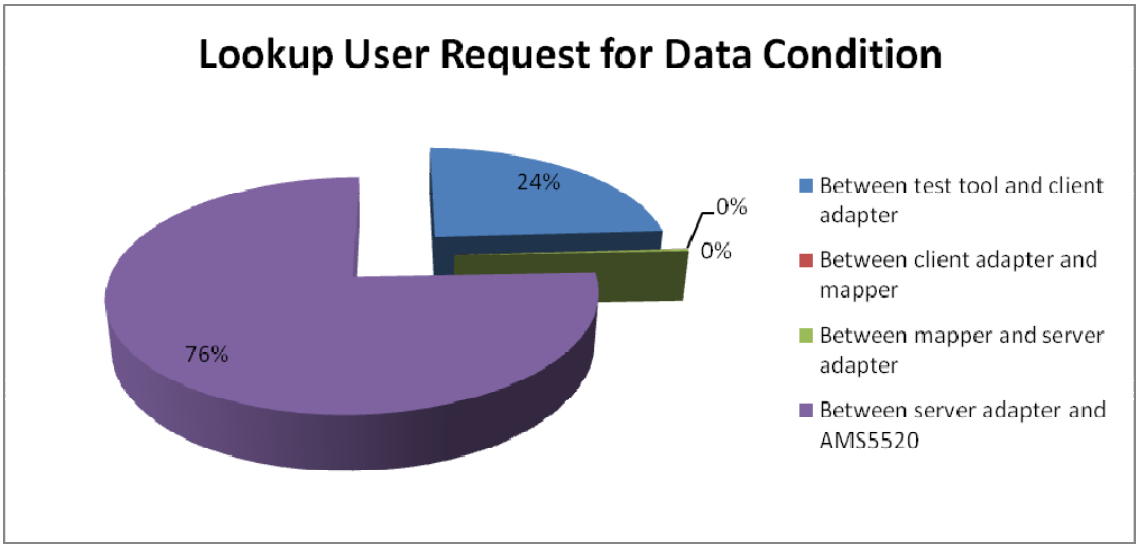
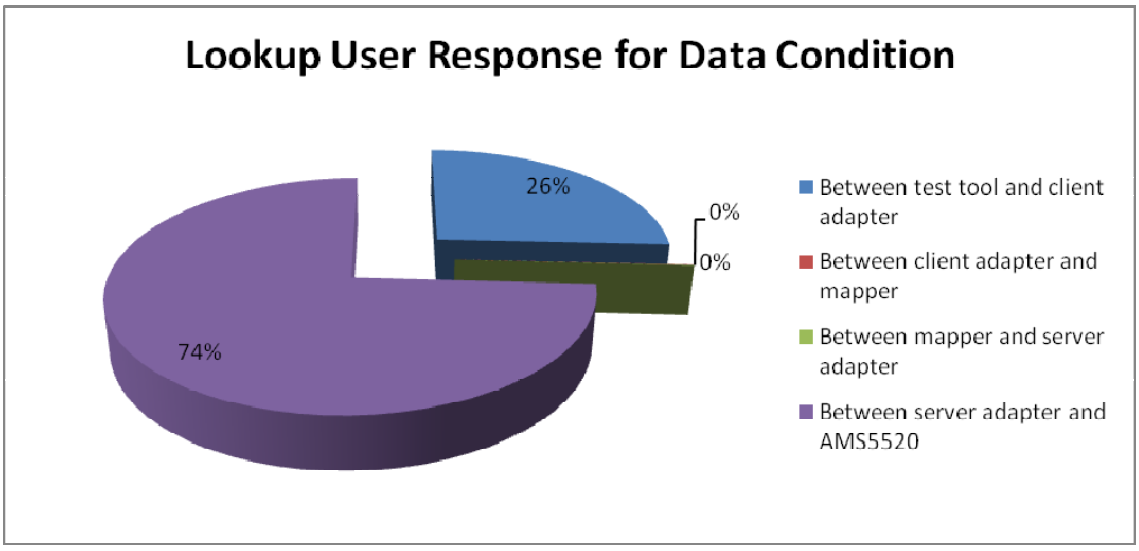



**Figure A.8b** Lookup user for data condition operation modules' response delays



# CURRICULUM VITAE

| | |
|---|---|
| **Name Surname :** | Melike YİĞİT |
| **Address :** | Bahçeşehir Üniversitesi Mühendislik Fakültesi Çırağan Caddesi 342353 Beşiktaş / İSTANBUL |
| **Date and Place of Birth :** | 04.04.1987 TOKAT |
| **Languages :** | Turkish (native), English (fluent) |
| **Elementary Education:** | Tokat Gazi Osman Paşa Elementary School 2001 |
| **Secondary Education:** | Tokat Gazi Osman Paşa Secondary School, 2005 |
| **B. S. :** | Bahçeşehir University, 2010 |
| **M.S. :** | Bahçeşehir University, 2012 |
| **Institute :** | The Graduate School of Natural and Applied Sciences |
| **Program :** | Computer Engineering |
| **Publications :** | - |
| **Work Experience :** | Bahcesehir University Computer Engineering Department *Research Assistant* (Istanbul, 2010 – 2012) |