

**THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY**

**DEVELOPMENT OF TMF615 STANDARD-BASED
TELECOMMUNICATION OPERATOR USER
MANAGEMENT**

M.S Thesis

MUHAMMET MACİT

İSTANBUL, 2012

**THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES
INFORMATION TECHNOLOGIES**

**DEVELOPMENT OF TMF615 STANDARD-BASED
TELECOMMUNICATION OPERATOR USER
MANAGEMENT**

M.S. Thesis

MUHAMMET MACİT

Supervisor: Asst. Prof. Dr. V. Çağrı Güngör

Co-Advisor: Assoc. Prof. Taşkın Koçak

İSTANBUL, 2012

THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

Title of the Master's Thesis : Development of TMF615 Standard-Based
Telecommunication Operator User Management
Name/Last Name of the Student : Muhammet MACİT
Date of Thesis Defense : 25.01.2012

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

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

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

Examining Committee Members

Signatures

Thesis Supervisor Asst. Prof. Dr., V. Çağrı Güngör	:
Thesis Co-Advisor Assoc. Prof., Taşkın Koçak	:
Member Prof. Dr., Emin Tacer	:
Member Asst. Prof. Dr., Selçuk Baktır	:

ACKNOWLEDGEMENTS

First of all I would like to thank Asst. Prof. Dr. Vehbi Çađrı GÜNGÖR and Assoc. Prof. Dr. Taşkın KOÇAK, who has given me 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.

My special thanks go to my friends and also colleagues for their endless support all through this work, also my personal life and in my master courses.

Last but not least I wish to express my love and gratitude to all my family. I would particularly like to thank my parents for their unlimited support in every stage of my life.

ABSTRACT

DEVELOPMENT OF TMF615 STANDARD-BASED TELECOMMUNICATION OPERATOR USER MANAGEMENT

Muhammet Macit

Computer Engineering Master Program

Thesis Advisor: Asst. Prof. Dr. V. Çağrı Güngör

January 2012, 44 pages

A typical Service Provider (SP) environment is managed by using various user management systems (UMS) that are provided by the Operation Support System (OSS) vendors. For the operators, a standardized provisioning and auditing mechanism is needed absolutely to more secure, consistent and highly automated management of OSS networks. Therefore companies that are play a role in the OSS environment gathering together for deciding to use standardized system at the Tele Management Forum (TMF) meeting and this system is called as TMF615. This thesis presents briefly structure, specification, advantages and disadvantages of the TMF615. Importantly, this thesis contains an integration work, which is supported by a worldwide telecom company.

Keywords: TMF615, TMF615 Implementation, User Management, Operation Support System

ÖZET

DEVELOPMENT OF TMF615 STANDARD-BASED TELECOMMUNICATION OPERATOR USER MANAGEMENT

Muhammet Macit

Bilgisayar Mühendisliği Yüksek Lisans Programı

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

Ocak 2012, 44 sayfa

Telekom operatörleri sahip oldukları ağ ortamları İDS sistemini sunan üreticilerin sağladıkları çeşitli kullanıcı yönetim sistemleri tarafından yönetilmektedir. Operatörler için standartlaşmış provizyon ve denetleme mekanizması daha güvenli, kararlı ve yüksek derecede otomize edilmiş İDS sistemleri yönetimi için mutlak gerekliliktir. Bu sebepten dolayı, TM Forum toplantısında İDS sistemlerinin kullanımında büyük rol oynayan telekom şirketleri standartlaştırılmış bir sistem kullanmaya karar verdiler. Ve bu standart sistem TMF615 olarak anılmaktadır. Bu tez TMF615'in yapısı, özellikleri, avantajları ve dezavantajları hakkında derinlemesine inceleme sunmaktadır. Ayrıca dünya çapındaki bir telekom firmasının desteği ile yapılan bütünleşme çalışması içermektedir.

Anahtar Kelimeler: TMF615, Kullanıcı Yönetimi, Kimlik Yönetim Sistemleri

CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
1 INTRODUCTION.....	1
1.1 Existing challenges	2
1.1.1 Orphaned and dormant accounts	2
1.1.2 Existing accounts and shared accounts.....	2
1.1.3 High number of resources	2
1.1.4 Detailed Audit.....	3
2 LITERATURE	4
2.1 EVOLUTION OF USER MANAGEMENT	4
2.1.1 Existing Protocols and Systems	5
2.1.2 SPML.....	6
2.1.3 TMF 615.....	7
2.1.4 TMF615 Interface	10
3 ANALYSIS OF WORK.....	15
3.1 Requirements	15
3.2 System Architecture	15
3.2.1 Development Platform and Tools	15
3.2.2 Production Environment.....	17
3.2.3 Risks and Solutions	17
4 DESIGN OF WORK.....	18
4.1 Use Case Diagram.....	18
4.2 Business Process Diagram	19
4.3 Sequence Diagrams	20
5 IMPLEMENTATION DETAILS.....	23

5.1	METHODOLOGY	23
5.1.1	Object oriented approach.....	23
5.1.2	WSDL file usage	23
5.1.3	Design Patterns.....	23
5.2	Implementation of the Test Tool.....	24
5.3	Implementation of the Client Adapter	27
5.4	Implementation of the Mapping Layer	32
5.4.1	General mapping between the TMF615 and OSS parameters	32
5.4.2	Mapping between the TMF615 and OSS Operations.....	32
5.4.3	Common Exceptions	36
5.5	Implementation of the Server Adapter.....	36
6	PERFORMANCE EVOLUTION AND TESTING	37
6.1	Unit Tests AND FUNCTIONAL TESTS.....	37
6.2	END TO END Tests	37
6.3	Comparison of the TMF615 with Existing System.....	38
6.4	Performance EvoluAtion	39
7	CONCLUSION.....	41
	REFERENCES.....	43

LIST OF TABLES

Table 5.1: Mapping table of the TMF615 operation names with TMF615 WSDL operation names.	31
Table 5.2: Mapping table of the TMF615 operations with OSS operations	31
Table 5.3: Add User Mapping Table.....	33
Table 5.4: Modify User Mapping Table	34
Table 5.5: Delete User Mapping Table	35
Table 5.6: List User Mapping Table	35
Table 5.7: Expire Password Mapping Table	35
Table 5.8: Reset Password Mapping Table.....	35
Table 5.9: Set Password Mapping Table.....	36

LIST OF FIGURES

Figure 2.1: Traditional User Management in OSS Environment.....	4
Figure 2.2: Centralized User Management in OSS Environment.....	5
Figure 2.3: Authorization Matrix Sample	9
Figure 4.1: Use case diagram.....	19
Figure 4.2: Business Process Diagram.....	20
Figure 4.3: Diagram of successful add user operation.....	21
Figure 4.4: Diagram of unsupported operation.....	22
Figure 4.5: Diagram of operation which has invalid attributes.....	22
Figure 5.1: Remove User Screen	24
Figure 5.2: Reset Account Password Screen.....	24
Figure 5.3: Validate Account Password Screen.....	25
Figure 5.4: Add User Screen.....	25
Figure 5.5: Expire Account Password Screen.....	26
Figure 5.6: Modify User Screen.....	26
Figure 5.7: Set Account Password Screen	27
Figure 5.8: Sample OSS add user request.....	28
Figure 5.9: Sample TMF615 add user request.....	29
Figure 5.10: Sample oss add user response.....	30
Figure 5.11: Sample tmf615 add user response	30
Figure 6.1: Requirements of unit tests	37
Figure 6.2: Comparison of Operation Execution Times	39

LIST OF ABBREVIATIONS

AMS	:	Access Management System
API	:	Application Programming Interface
CORBA	:	Common Object Request Broker Architecture
GUI	:	Graphical User Interface
HTTP	:	Hypertext Transfer Protocol
HTTPS	:	Secure Hypertext Transfer Protocol
JSF	:	Java Server Faces
NBI	:	North Bound Interface
OSS	:	Operation Support System
SOAP	:	Simple Object Access Protocol
SP	:	Service Provider
SPML	:	Service Provisioning Markup Language
SPML	:	Service Provisioning Markup Language
TM	:	Tele Management
TMF	:	Tele Management Forum
UM	:	User Management
UMS-C	:	Central User Management System
UMS-L	:	Local User Management System
WSDL	:	Web Service Description Language
XML	:	Extensible Markup Language

1 INTRODUCTION

Operation support systems (OSS) responsible for maintaining infrastructure at the operation level for telecommunications service providers and their networks. OSS solutions generally cover user (identity) management, inventory management, fault management, performance management and security management.

One of the functions of OSS is user (identity) management. Followings are the responsibilities of user management in OSS:

- a. Creating user and user accounts
- b. Managing user accounts (modify, delete, suspend, and resume user accounts)
- c. User authorization (management of user's roles and access profiles)
- d. User authentication information management (password related operations; set, expire, validate and reset password)
- e. Audit and logging

Generally service providers work with various OSS vendors and they host composition of different OSS solutions at their network infrastructure. But, every OSS has own protocols (SOAP, CORBA etc.) and this makes harder management of different OSSs together. Therefore a new global and standardized user management protocol is needed and main goal of using this new protocol can be explained as automation of manual processes with an improvement concerning efficiency, security and compliance challenges.

To address this need, TMF615 common communication platform for operator user management will be designed and developed by TM (Telemagement) Forum. In this respect, the proposed approach includes a common interface to address communication problems in multi-vendor service provider environments. The realization of the proposed TMF615 standard-based interface will enable efficient and easy integration to existing and future OSS solutions. In this way, a standardized interface is offered and a common communication

platform is adequate for all various OSSs. 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.

1.1 EXISTING CHALLENGES

Existing challenges at the OSS environment are explained briefly below.

1.1.1 Orphaned and dormant accounts

Orphaned and dormant accounts have to be discovered at the earliest possible opportunity and handled these accounts under the existing security requirements. This process may follow these steps respectively; first accounts are locked, then their entitlement richness is reduced and finally they are removed.

1.1.2 Existing accounts and shared accounts

Existing accounts have to be integrated into the specified OSS and shared accounts have to be removed to avoid improper usage.

1.1.3 High number of resources

Different protocols and APIs in OSS environments cause high usage of resources and also cause more performance for user provisioning.

Therefore cost of provisioning can easily be identified and reduced to a minimum by the use of a standardized interface, communication protocol, and functionality.

1.1.4 Detailed Audit

Audit and compliance requirements are hardly to fulfill because of the followings;

- a. Use of shared, non-personalized accounts
- b. High number of resources
- c. Unknown implementation
- d. Missing audit capabilities
- e. Missing knowledge about how to gather and report audit data

2 LITERATURE

2.1 EVOLUTION OF USER MANAGEMENT

Below two figures show the overall view of the system architecture. Traditional user management system provides multiple protocols for service providers and operators to access the OSS (Operation Support System) by using each OSS's client tools as shown on figure 2.1. And these tools use different protocols and different APIs such as SOAP, LDAP, CORBA and etc. On the other hand, offered system architecture provides single protocol and single interface for Service Providers to communicate with OSS environments. In this way, operators can access the OSSs by using their own unique client tool and this reduces the complexity of the user management at service provider side.

Figure 2.1: Traditional User Management in OSS Environment

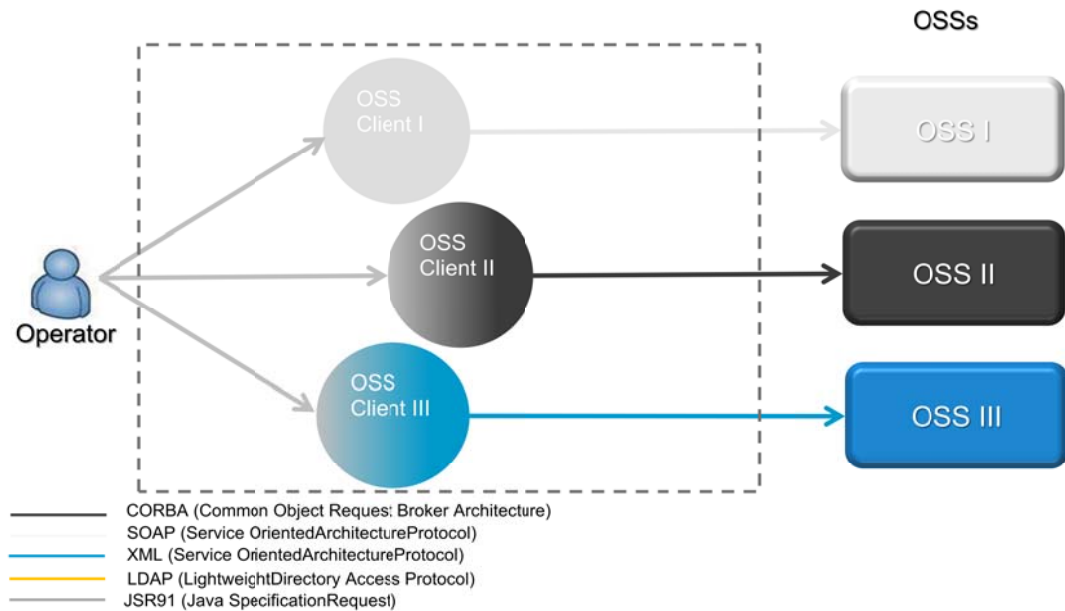
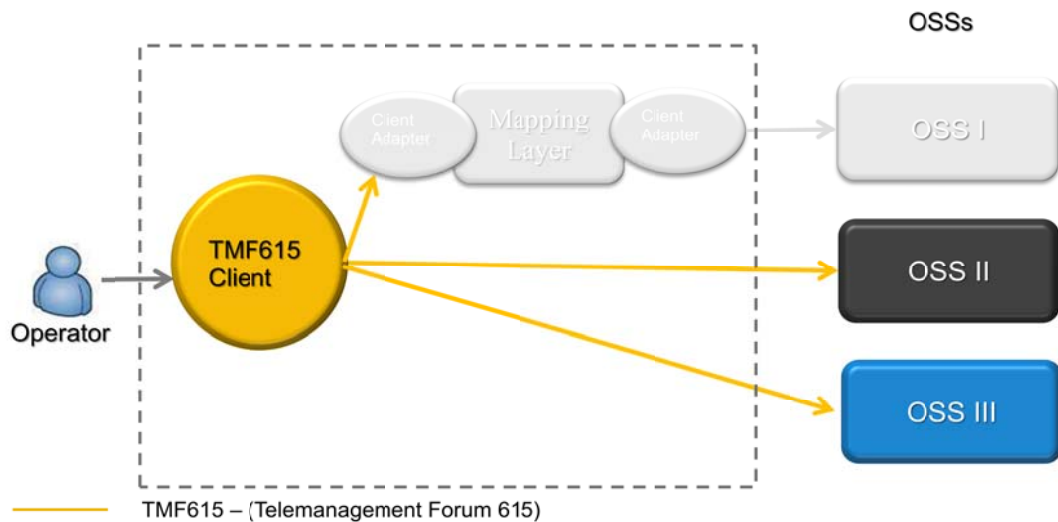


Figure 2.2: Centralized User Management in OSS Environment



2.1.1 Existing Protocols and Systems

OSS systems are produced and supported by various companies. Each of these companies plays a role in the OSS environments. These companies can be divided into three categories;

- a. Service Providers; Vodafone, T-Mobile, Telefonica, etc
- b. Authentication Audit System Providers; IBM, Wipro, etc
- c. OSS Solution Providers; Alcatel-Lucent, Ericsson, Nokia, Siemens, etc

Composition of different OSSs is used by service providers. However, OSSs are produced by various vendors and they use different protocols. Therefore these compositions cause to big problems for the central user management system. These problems can be explained mainly as high complexity and long provisioning time. New approach to operation user management, TMF615, was came out from Telemanagement Forum meetings by above companies. According to the TMF615, some of these companies present their tools and some of them are also presented below.

- a. IBM Tivoli Framework
- b. Ericsson Solution
- c. Nokia Siemens Networks NetAct™

2.1.2 SPML

SPML (Service Provisioning Markup Language) is an XML based language and introduced by OASIS. SPML provides exchanging the provisioning information of user, resource and service between cooperating organizations. SMPL is the open standard for the integration and interoperation of service provisioning requests. The main goal of the SPML allows secure and quick web service and application setup. This can lead to automation of user or system access to electronic, thus customers (corresponds to telecom operators in the thesis) are not locked into proprietary solutions.

Current SPML version provides a lot of capability;

- a. Core capability
- b. Async capability
- c. Batch capability
- d. Bulk capability
- e. Password capability
- f. Search capability
- g. Suspend capability
- h. Updates capability
- i. Custom capability

A provider can define a custom capability that integrates with existing SPML. TMF615 protocol was created by extending some SPML capabilities and implementing custom capabilities. Inherited SPML functionalities are;

- a. *Core capability*: add, modify, delete, listTargets, lookup
- b. *Password capability*: setPassword, expirePassword, resetPassword, validatePassword
- c. *Suspend capability*: suspend, resume, active

Password capabilities are overridden by TM Forum in TMF615 but others are same as SMPL version. Also listUsers operation and audit operations added to TMF615 by TM Forum.

SPML uses PSO (Provisioning Service Object) as a key identifier of the operations and each object has unique identifier PSO ID.

SMPL offers two different profiles; these profiles are XML and DSML (Directory Services Markup Language) profiles. TMF615 uses XML profile.

2.1.3 TMF 615

Service providers provision consistently operator's access rights and authorities across systems using a UMS-C by means of the TMF615 interface which is introduced by TM Forum. TMF615 interface deals with the information exchange between the UMS-C and the UMS-L. This information related to provisioning of access rights, authorities and auditing.

- a. *UMS-C*: Centralized user management application used by service providers to provision users and their authorities across the entire network and for all OSS vendors.
- b. *UMS-L*: User management solution at the respective OSS vendor software. It accepts and processes request coming from UMS-C.

Concepts which are introduced in TMF615 specification are explained at the below section and importance of them to a successful implementation.

2.1.3.1 Users and accounts

Users of the system are the operators working on management systems in the SP. A unique identity is assigned to the each User and it is generally used between the management systems to identify user. Also accounts can be assigned to User in order to access resources. Different accounts and their identifiers are used to access different resources for each user. Also account identifiers can be different from identifier of user.

2.1.3.2 Account controller

Basically every account in the system is associated with a single Account Controller and also each Account Controller controls multiple accounts.

2.1.3.3 Authentication

Authentication is the routine to verify the identity of the sender of a communication. An example for a communication is request to add user. In addition to this, sender may be a person using a computer, a computer itself or a computer program.

2.1.3.4 Authorization

Authorization is the routine after the entity authentication; it is explained with granting certain privileges and access rights on the target system to the authenticated entity. The authorizations which are granted to an entity are exactly related to the ability of target system. TMF615 interface offers “Authorization Space” concept for authorization.

2.1.3.5 Authorization space

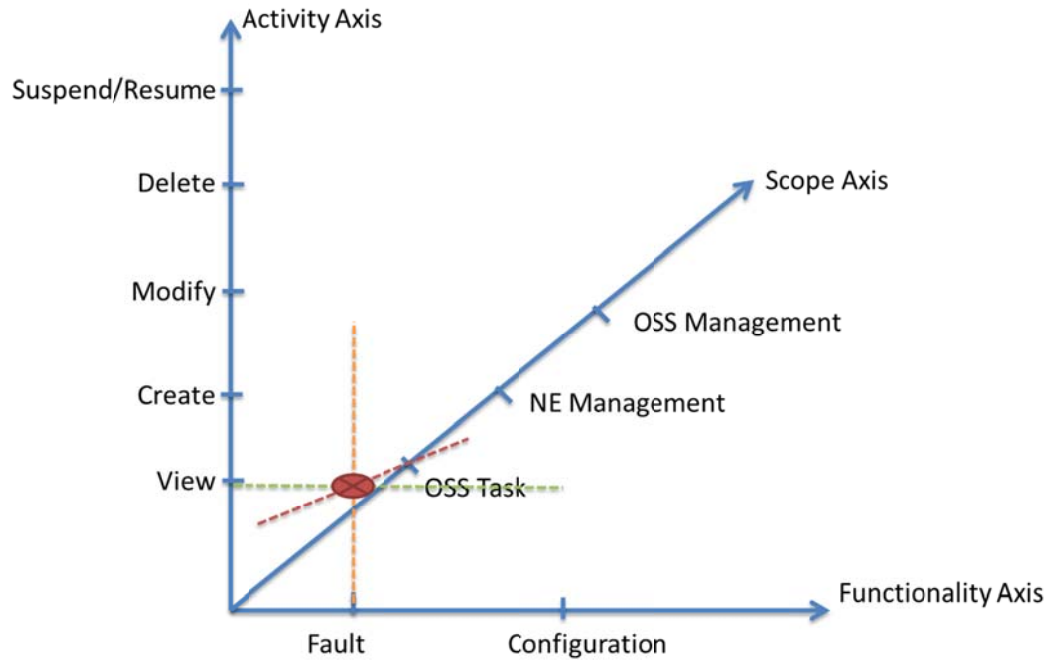
Different ways are used to handle User Management for every vendor; therefore privileges are specific to the vendors. However privileges are managed by roles generally. TMF615 specification presents the authorization concept in order to create generic role management system. Also this generic system is not vendor specific. Authorization space consists of three axes;

- a. *Functionality Axis*: This axis defines the working area of the OSS user. It can be like Fault Management, Configuration, Accounting, Performance and Security.
- b. *Activity Axis*: This axis defines the scope. It can be like view, create, delete, modify, suspend and resume for the user account.
- c. *Scope Axis*: This axis defines the possible various user activities. They can be Network Element Management, OSS Management etc.

Roles which are side of Service Provider can be defined using the coordinates in the authorization space. An example for authorization space;

Assume that the SP is using OSS applications from various vendors and it has the license only for Fault Management and Configuration Management. If “Fault Monitor” role is desired to any User, then coordinates of the above point define this role. This shows that, “Fault Monitor” role described by Fault from Functionality (x) axis, View from Activity (y) axis and OSS Task from Scope (z) axis.

Figure 2.3: Authorization Matrix Sample



On the other hand, service provider can define many various roles with using coordinates and one or more coordinates can be used from this three dimensional matrix. Description of roles has differences between the operators and also they depend on authorization space which is already offered in the TMF615 specification. Also TMF615 specification does not mandate any specific authorization matrix but shows the matrix as a possible solution.

2.1.3.6 Scheduling

Service providers need to execute different tasks in different time. Therefore scheduling provides restricting authorities of user with specific time period to execute tasks. In this way, right access is granted to the appropriate resource, at the defined time. TMF615 supports two types of scheduling, weekly and monthly.

- a. *Weekly*: This informs the respective UMS-L during which access should be granted for each day of the week. This set of time durations can be repeated weekly between the two calendar days.

- b. *Monthly*: This informs the respective UMS-L during which access should be granted for each day of the month. This set of time durations can be repeated monthly between the two calendar days.

2.1.3.7 Provisioning

The term provisioning refers to task of information exchange between UMS-C and UMS-L. This is carried out by sending information about a user, his/her authorities, accounts that need to be created and time during which she/he should have the necessary access.

2.1.3.8 Audit

Auditing is essential for User Management System because activities of users and any improper usage have to be monitored. In addition to this auditing can be used by UMS-C to synchronize with UMS-L due to the independent modifications that are made at the UMS-L in emergency. TMF615 offers two different kind of Audit.

- a. *Status Audit*: Essentially gives the current condition of user accounts and their authorities. This is useful for synchronization between the UMS-C and UMS-L.
- b. *Audit Trail*: Audit trail informs about all the provisioning actions that have been happened in a period of time.

2.1.4 TMF615 Interface

TMF615 interface based on SPML standards, but to meet further necessities some improvements and extensions were made by TMForum and these extensions are called UM (User Management) standard. Both of standards are rely on XML and they are prepared via XSD files. TMF615 WSDL solution set contains a WSDL which is Web Service Description Language consists of TMF615 interface with SPML and UM protocols.

TMF615 is designed to perform the following;

- a. Expose the WSDL to the UMS-C for communication

- b. Communicate with the local UM for performing the different actions in order to complete the user provisioning.
- c. Ensure there is sync after connection is lost between UMS-L and UMS-C

2.1.4.1 Basic Operations

Add User:

This operation is used for creating new user. If user is created once, other requests will be rejected during the created user existence in the UMS-L. Changing user attributes can be made by modify user operation. Add User operation also creates privileges of the user in two different ways such as old way string list or new way authorization matrix.

Modify User:

This operation is used for changing privileges or restrictions already assigned to any existing users. Modifying user attributes can be accomplish without changing user password by using ‘\$\$\$UNUSED\$\$\$’ keyword as a validation information attribute. Password operations are encouraged to use for changing password of an existing user. Also modify user operation supports multiple modification at a time.

Remove User:

This operation is used to delete an existing user. When a user is deleted, its accounts are also deleted.

Suspend User:

This operation is used to suspend an existing user. A user can be suspended only if it has been not in the suspended state.

Both suspend and resume operations are idempotent. Any requestor should be able to suspend (or to resume) the same account multiple times without error.

Resume User:

This operation is used to resume a suspended account.

Is User Active:

This operation is used query resume status of account.

List Users:

This operation is used to list users and their associated account information and provisioning data. This operation returns details of an individual specified user or set of user. Details of user are returned even if it is suspended state.

Requests can be specified by returnData attribute to change returned information of users. Also there is exception cases listed below;

If any user id is not defined in the request, then all existing users are listed. Some userids are requested to list are not correspond to any record then just details of available users are listed without any error.

List Targets:

This operation returns available target at the OSS side.

2.1.5 Password Related Operations

The main actor of password related operations is account. Because users can have more than one account according to the TMF615 specifications and these accounts can have different validation information. Target account is identified by account controller and if target account is not defined in the request then request will be performed for all user accounts.

Set Password:

This operation is used to change password of an existing account.

Expire Password:

This operation is used to expire password of an existing account expired.

Reset Password:

This operation is used to reset an existing account password.

Validate Password:

This operation is used to validate account password.

2.1.6 Audit Operations

Audit Trail for Users Admin Operations:

This operation is used to audit one or more users for a set of specified administrative operation for a specified time interval. The possible administrative operations can be Add, Modify, Delete, Suspend, and Resume.

Audit Trail for Users Provisioning Operations:

This operation is performed between a specified time intervals for auditing one or more users for a specific set of provisioning operations. The possible provisioning operations could be account provision, authorization provision, schedule provision.

Audit Targets for Users Admin Operations:

This operation is performed in a specified time range for auditing one or more targets for a specific administrative operation.

Audit Status of Users Provisioning Information:

This operation is performed on one or more users corresponding to various provisioning operations. The possible provisioning operations could be account provision, authorization provision, schedule provision.

Using this operation UMS-L data can be synchronized with UMS-C.

Audit Status of Targets Account Information:

This operation supports audit one or more targets for account information through targets. It basically gives back information about the existing accounts corresponding to the targets specified.

Audit Trail for Targets Account Usage:

This operation supports audit one or more targets for account usage information through targets. It basically gives back information about the usage (login logout) details of accounts by the users, between the specified time ranges.

Audit Trail for Targets Authorization Usage:

This operation supports audit one or more targets for authorization usage information through targets. It basically gives back information about the usage details of authorization by the users, between the specified time ranges for a given list of targets.

3 ANALYSIS OF WORK

This work was implemented by using the waterfall methodology which is one of the software development methodologies. Therefore analysis, design, implementation and test phases are defined properly for this work. In addition to this, coding standards was considered in design and implementation phase. Class names and class variables are properly named according to the java coding standards. All project phases are explained below respectively.

3.1 REQUIREMENTS

Functional and non-functional requirements of project are listed below.

- a. Test tool should simulate the TMF615 operations.
- b. Client adapter must reply each coming request anyway.
- c. Mapping layer must implement both TMF615 specifications and OSS restrictions.
- d. Communication should be synchronous.
- e. Module based project. Four module; client adapter, mapping layer, server adapter and test tool should be developed.
- f. Software is integrated to the NBI (North Bound Interface) of the specific OSS element.
- g. Free and open source tools should be used.

3.2 SYSTEM ARCHITECTURE

3.2.1 Development Platform and Tools

Eclipse:

Eclipse is an open source software development environment. It is written in Java and can be used to develop applications in Java and other programming languages including. Most powerful specification of the Eclipse is extensible plug-in system. Therefore there are many plug-ins available for eclipse, such as; revision control

systems and build automation tools. This richness makes eclipse valuable for enterprise application development.

Java Enterprise Edition (EE):

Java EE is widely used platform for enterprise application development within the Java programming language. The main advantage of the Java platform is JVM (Java Virtual Machine) that makes easy observing and scaling enterprise application. In addition to Java, the Java EE provides web and application server specifications and that gives high level, distributed, multi-tier and modular application development and deployment.

In our project we use web technologies, Servlet and JSF, of the Java EE.

Maven:

Maven is a build automation and software comprehension tool. Generally used for Java enterprise application development. Maven dynamically downloads Java libraries and Maven plug-ins from one or more pre-defined repositories to use within the project.

Maven is used as build automation tool of our project. Dependency management and WSDL code generation is controlled by maven while compiling the project and building the package.

Subversion (SVN):

SVN is a software versioning and a revision control system and it is distributed under a free license. Enterprise corporations, big development teams and developers use SVN to maintain current and historical versions of files such as source code, web pages, and documentation. Development teams can work together on the same project by using SVN.

Jetty:

Jetty is a pure Java-based HTTP client/server and servlet container. It is developed as a free and open source project as a part of the Eclipse Foundation. We prefer jetty because it makes easy hot deployment for web applications. Hot deployment provides context updating without restart the web container when code has been changed.

SOAP UI:

SoapUI is an open source web service testing tool. We use the soapUI for web service invoking, development, simulation and mocking.

3.2.2 Production Environment**Apache Tomcat:**

Apache Tomcat is an open source web container. Tomcat provides a pure Java HTTP web server environment for Java code to run. Tomcat is preferred for production for its stability. Test tool module is deployed into the Tomcat context.

JBoss:

JBoss Application Server is an open-source Java EE based application container. An important distinction for this class of software is that it not only implements a server container, it implements the Java Enterprise specification totally. Client adapter, mapping layer and server adapter are deployed into JBoss context. But actually JBoss is not necessary because in our project we don't use EJB (Enterprise Java Beans). Hence these modules can run on any web container.

3.2.3 Risks and Solutions**Integration problems:**

Successful implementation of project modules is not same as the successful implementation of whole project. The possible risk is end to end problems after the integration of successfully implemented modules. To ensure about these possible problems only add user and delete user operations are implemented once and they are tested before starting implementation of project modules.

4 DESIGN OF WORK

Project is divided into 4 different modules at this phase.

Vendor specific application AMS 5520 was installed to the application server and functions of this application were inspected. Operations were defined.

- a. add user
- b. modify user
- c. delete user
- d. list users
- e. set password
- f. reset password
- g. expire password
- h. validate password
- i. audit trial for users admin operations
- j. audit trial for users provisioning operations
- k. audit targets for users admin operations

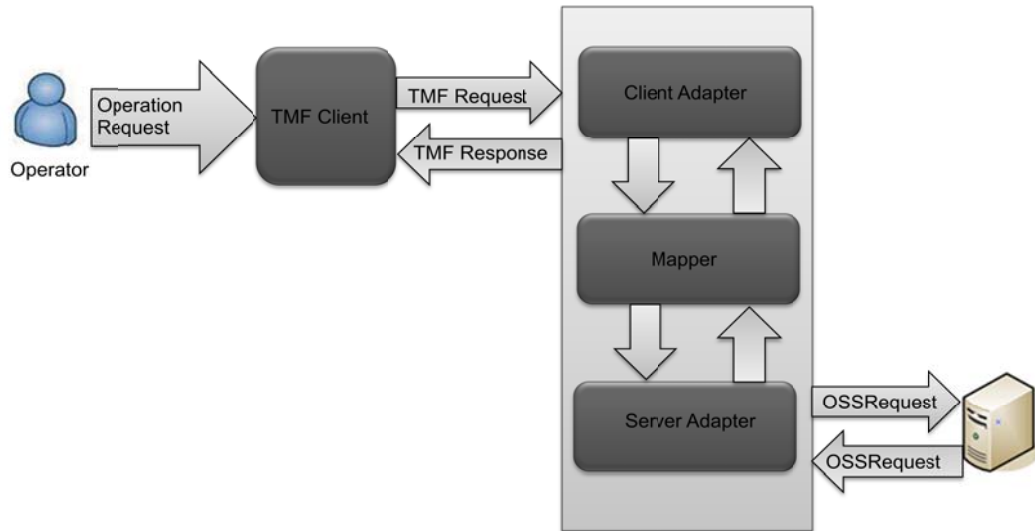
Working mechanism of project can be explained as follow. TMF615 request comes to client adapter. Client adapter sends incoming request to the mapping layer. Mapping layer converts TMF615 request message to the Alcatel AMS5520 request message and sends to server adapter. Server adapter waits reply from AMS 5520 and sends back reply to the mapping layer. Mapping layer converts AMS 5520 response to the TMF615 response and sends back to the client adapter. Client adapter sends back to the response message. This structure is explained below by using business process diagrams, use case diagrams and sequence diagrams. Some unsupported TMF615 operations are also explained in sequence diagrams.

4.1 USE CASE DIAGRAM

Below use case diagrams shows general overview of the project at module level. Operator requests deliver to the OSS environment by using TMF615 client and a proxy which is located front of the OSS environment. This proxy consists of client adapter,

mapping layer and server adapter. The main role of the proxy is serving TMF615 compatible communication infrastructure to the operators. And operators can use TMF615 client to reach all kind of OSS environments.

Figure 4.1: Use case diagram



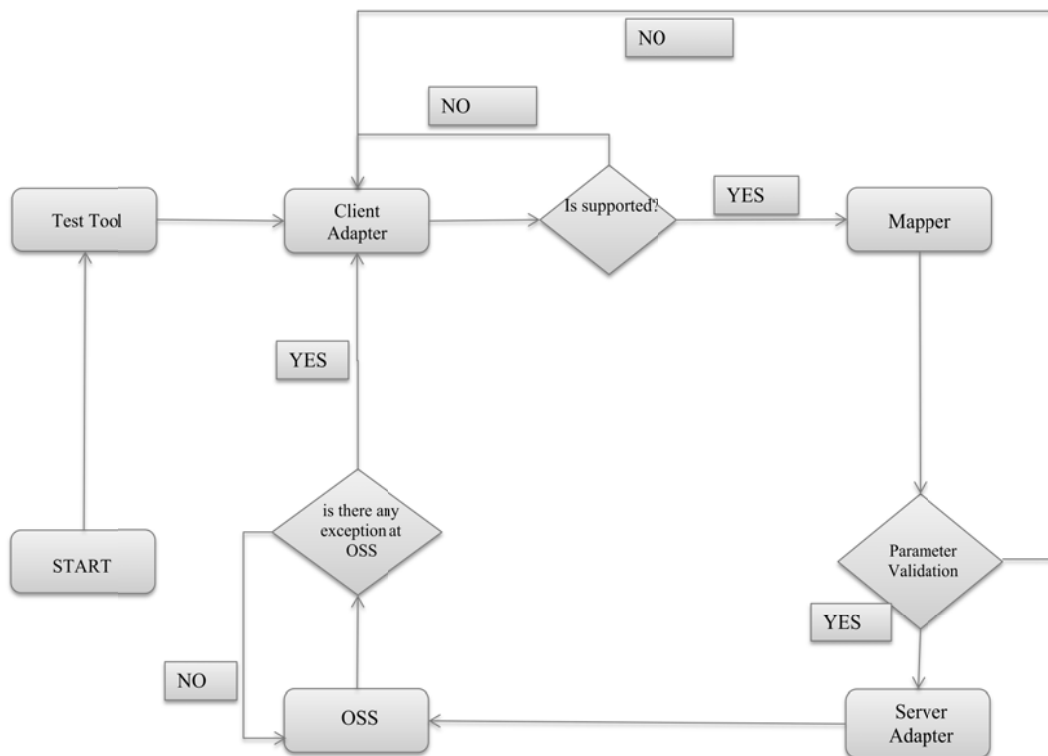
4.2 BUSINESS PROCESS DIAGRAM

Below business process diagram shows work flow between the modules and end to end flow between the operator and OSS environment. Milestones can be explained with below questions;

- a. Is this operation supported?
- b. Are the input parameters validated successfully?
- c. Is there any exception at the OSS side?

As shown from the diagram, every module can raise exception according to their roles in the work flow.

Figure 4.2: Business Process Diagram



4.3 SEQUENCE DIAGRAMS

Figure 4.3 shows successful add user operation. Operation triggered by operator from any GUI (in this project we use test tool as a GUI) to the client adapter. Client adapter validates operation mode and checks that operation is supported, after that sends coming TMF request to the Mapping Layer. Mapping Layer converts TMF request to the OSS compatible request and validates attributes of the TMF request together. After successful conversion and validation, Mapping Layer sends mapped message to the Server Adapter. Server adapter sends coming request to the OSS via web service and waits response from OSS. Finally OSS response is converted to the TMF response and sends back to the test tool successfully.

Figure 4.4 shows successful unsupported operation. Operation triggered by operator from test tool to the client adapter. Client adapter validates operation mode and checks that operation is supported, and sends created response to the test tool which is web fault and this fault contains “unsupportedOperation” exception.

Figure 4.3: Diagram of successful add user operation

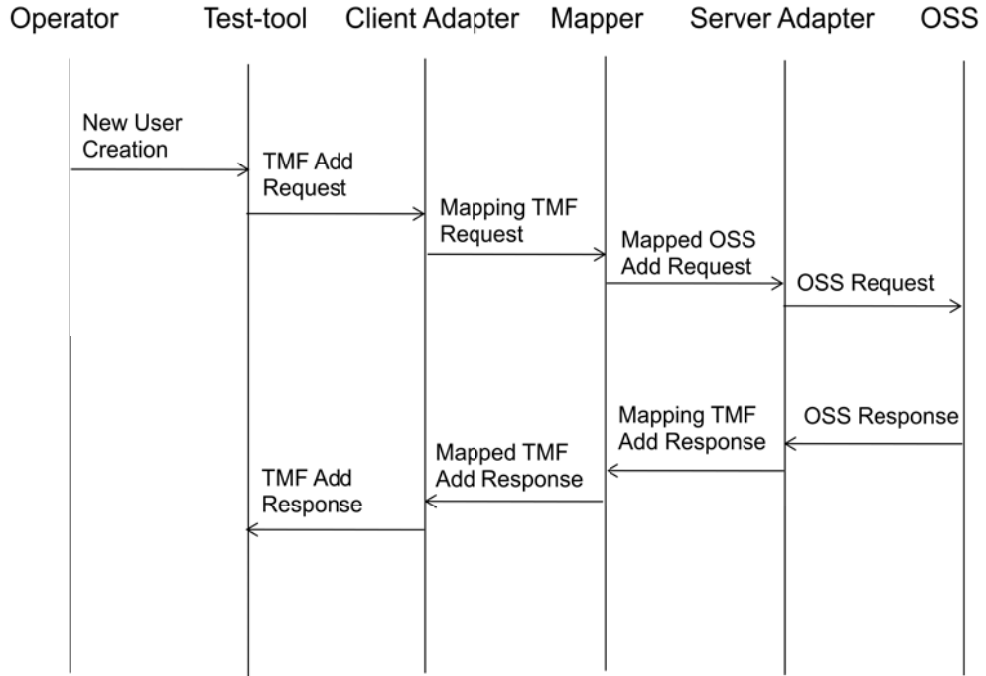


Figure 4.5 diagram shows validation of invalid attributes. Operation triggered by operator from test tool to the client adapter. Client adapter validates operation mode and checks that operation is supported, after that sends coming TMF request to the Mapping Layer. Mapping Layer converts TMF request to the OSS compatible request and validates attributes of the TMF request together. While validation at the Mapping Layer, if any invalid attributes detected, then Mapping Layer sends response back to the client adapter predefined exceptions of TMF, such as exception “operation_failed_invalid_input_parameter”

Figure 4.4: Diagram of unsupported operation

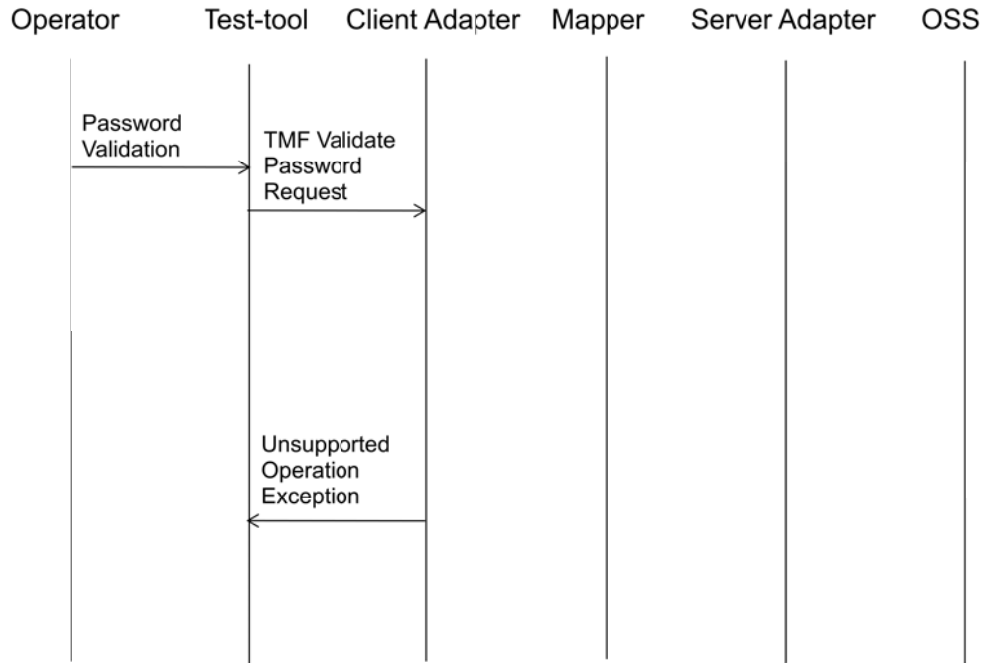
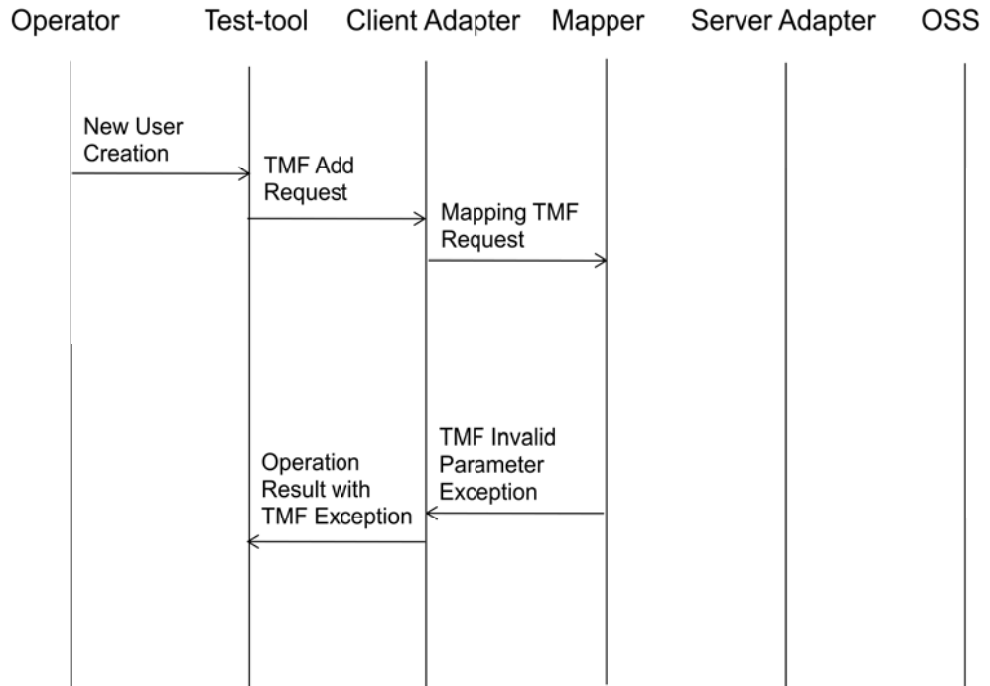


Figure 4.5: Diagram of operation which has invalid attributes



5 IMPLEMENTATION DETAILS

Implementation of TMF615 is specific to the vendors.

Software architecture consists of four different modules;

1. Test Tool
2. Client Adapter
3. Server Adapter
4. Mapping Layer

5.1 METHODOLOGY

5.1.1 Object oriented approach

Web service messages generally based on XML structure. There is challenging problem for parsing and building XML messages. To overcome this problem, WSDL (Web Service Definition Language) files turn into Java objects by using WSDL code generators and Java XML-object binding tools. In our project we use Apache CXF Code Generator Maven Plug-in to generate stub and skeleton codes of web services.

5.1.2 WSDL file usage

Maven becomes very efficient tool with its plug-ins. Just adding the a few lines into the Maven POM (Project Object Modeling), software can generate the WSDL codes at the compile time and any changes at the WSDL file reflected to the codes immediately. This kind of usage makes project more maintainable.

5.1.3 Design Patterns

In our project we use architectural design patterns. The mainly used patterns are SOA (Service Oriented Architecture) and MVC (Model-View-Controller).

From SOA perspective, our implemented modules have well-defined business functionalities that are built as software components that can be reused.

As mentioned before, we use JSF (Java Server Faces) as web framework for view layer of test tool. JSF implements natively MVC. Means of that is web application is already

implements MVC if created within JSF in high level design. But in developer perspective, the view layer turn into MVC application by using tiered applications. These tiers are web tier and service tier and also data layer optionally. So actually our test tool implements the M(MVC)C pattern. This kind of usage provides differentiate the presentation and data layer both overall view and developers view.

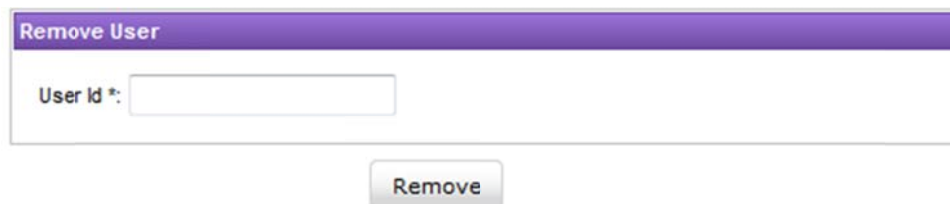
We also use factory pattern, builder pattern and singleton pattern in this work in addition to the architectural patterns.

5.2 IMPLEMENTATION OF THE TEST TOOL

One of the modules of the project is test tool; this tool was prepared for simulate service provider side and used to create requests for testing the developed software and implemented protocol. Therefore test tool module is distinguished from the other modules according to the usage purpose.

Test tool was developed by considering the enterprise software development architectures and architectural patterns. Snapshots of the implemented test pages are listed below.

Figure 5.1: Remove User Screen

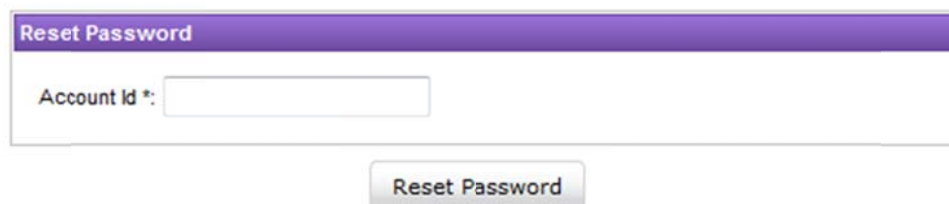


Remove User

User Id *:

Remove

Figure 5.2: Reset Account Password Screen



Reset Password

Account Id *:

Reset Password

Figure 5.3: Validate Account Password Screen

Validate Password

User Id *:

Password *:

Figure 5.4: Add User Screen

Add User

User Id *:

Account Id*:

Validation Information*:

Account Controller*: AMS

Target Name*: AMS

Access Profile Name*: PAPGroup

Exclude Flag: false

erver

All Access Profile Values

- allPAPs
- somePAPs

» Copy all
» Copy
» Remove
» Remove All

Allowed Access Profile Values

- defaultPAP

» First
» Up
» Down
» Last

All Roles

- AMS NBI
- Administrator
- CLI Admin
- Constructor
- Drop Provisioner
- IDM NBI
- NBI System
- NOC Admin

» Copy all
» Copy
» Remove
» Remove All

Given Roles

- Operator

» First
» Up
» Down
» Last

Figure 5.5: Expire Account Password Screen

Expire User

Account Id *:

Expire

Figure 5.6: Modify User Screen

Modify User

User Id *:

PSO Identifier Id*:

Account Id*:

Validation Information*: 123456

Account Controller*: AMS

Target Name*: AMS

Access Profile Name*: PAPGroup

Exclude Flag: false

All Access Profile Values

allPAPs
somePAPs

Copy all
Copy
Remove
Remove All

Allowed Access Profile Values

defaultPAP

First
Up
Down
Last

All Roles

AMS NBI
Administrator
CLI Admin
Constructor
Drop Provisioner
IDM NBI
NBI System
NOC Admin

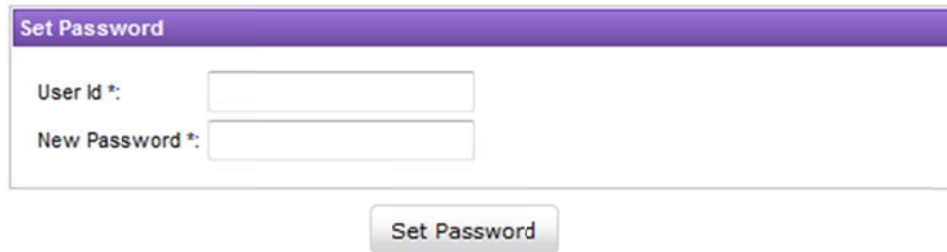
Given Roles

Operator

First
Up
Down
Last

Modify

Figure 5.7: Set Account Password Screen



The screenshot shows a web form titled "Set Password". It features a purple header bar with the text "Set Password". Below the header, there are two input fields: "User Id *:" and "New Password *:". Below the input fields is a button labeled "Set Password".

5.3 IMPLEMENTATION OF THE CLIENT ADAPTER

Client adapter module hosts TMF web service. This service accepts coming TMF SOAP requests and sends back TMF SOAP responses according to the requests.

Most powerful open source web service framework Apache CXF is used to host web service and its code generation plug-in is used to generate stub and skeleton codes of service from WSDL. Server side implementation of TMF615 web service is performed to apply TMF615 specifications. Client adapter validates operation mode which can be synchronous or asynchronous but our implementation just supports synchronous due to the restriction of the OSS environment. Before the checking operation mode client adapter controls the operation support status and throws an inherited java exception "UnsupportedOperation" to the requester. This exception is converted to the web fault by Apache CXF framework. After validations, client adapter sends coming request to the Mapping Layer for conversion of the request to the OSS compatible request. Figure 5.8, figure 5.9, figure 5.10 and figure 5.11 show basic add user operation samples.

Figure 5.8: Sample OSS add user request

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <header xmlns:ns2="alu.v1" xmlns="tmf854.v1">
      <activityName>test</activityName>
      <msgName>addUser</msgName>
      <msgType>REQUEST</msgType>
      <senderURI>tmf854.header.senderURI</senderURI>
      <destinationURI>tmf854.header.destinationURI</destinationURI>
      <communicationPattern>SimpleResponse</communicationPattern>
      <communicationStyle>RPC</communicationStyle>
    </header>
  </soap:Header>
  <soap:Body>
    <ns2:addUser xmlns="tmf854.v1" xmlns:ns2="alu.v1">
      <ns2:userCreateData>
        <ns2:name>test</ns2:name>
        <ns2:password>123456</ns2:password>
        <ns2:roles>
          <ns2:role>Operator</ns2:role>
        </ns2:roles>
        <ns2:allowedPapGroups>
          <ns2:papGroup>allPAPs</ns2:papGroup>
        </ns2:allowedPapGroups>
      </ns2:userCreateData>
    </ns2:addUser>
  </soap:Body>
</soap:Envelope>
```

Figure 5.9: Sample TMF615 add user request

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:addRequest xmlns="urn:oasis:names:tc:SPML:2:0:audit"
      xmlns:ns2="urn:oasis:names:tc:SPML:2:0:provisioning"
      xmlns:ns3="urn:oasis:names:tc:SPML:2:0"
      xmlns:ns4="urn:oasis:names:tc:SPML:2:0:suspend"
      xmlns:ns5="urn:oasis:names:tc:SPML:2:0:umpassword">
      <ns3:data>
        <ns2:User>
          <ns2:userId>test</ns2:userId>
        </ns2:User>
        <ns2:provisioningData>
          <ns2:accountData>
            <ns2:accountId>test</ns2:accountId>
            <ns2:validationInformation>123456</ns2:validationInformation>
            <ns2:accountController>AMS</ns2:accountController>
            <ns2:targetData>
              <ns2:targetName>AMS</ns2:targetName>
              <ns2:accessProfileData>
                <ns2:accessProfileName>PAPGroup</ns2:accessProfileName>
                <ns2:accessProfileInfo>
                  <ns2:accessProfileValue>allPAPs</ns2:accessProfileValue>
                </ns2:accessProfileInfo>
                <ns2:excludeFlag>false</ns2:excludeFlag>
              </ns2:accessProfileData>
            </ns2:targetData>
          </ns2:accountData>
          <ns2:authorizationData>
            <ns2:roleName>Operator</ns2:roleName>
          </ns2:authorizationData>
        </ns2:provisioningData>
      </ns3:data>
    </ns3:addRequest>
  </soap:Body>
</soap:Envelope>
```


Figure 5.10: Sample oss add user response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:tmf854="tmf854.v1" xmlns:alu="alu.v1">
  <soapenv:Header>
    <tmf854:header>
      <tmf854:activityName>test</tmf854:activityName>
      <tmf854:msgName>addUserResponse</tmf854:msgName>
      <tmf854:msgType>RESPONSE</tmf854:msgType>
      <tmf854:senderURI>tmf854.header.destinationURI</tmf854:senderURI>
      <tmf854:destinationURI>tmf854.header.senderURI
      </tmf854:destinationURI>
      <tmf854:activityStatus>SUCCESS</tmf854:activityStatus>
      <tmf854:communicationPattern>SimpleResponse
      </tmf854:communicationPattern>
      <tmf854:communicationStyle>RPC</tmf854:communicationStyle>
      <tmf854:timestamp>20120320145429.82+0530</tmf854:timestamp>
    </tmf854:header>
  </soapenv:Header>
  <soapenv:Body>
    <addUserResponse xmlns="alu.v1">
      <status>Success</status>
      <name>
        <usrNm>test</usrNm>
      </name>
    </addUserResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Figure 5.11: Sample tmf615 add user response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns3:addResponse xmlns="urn:oasis:names:tc:SPML:2:0:audit"
      xmlns:ns2="urn:oasis:names:tc:SPML:2:0:provisioning"
      xmlns:ns3="urn:oasis:names:tc:SPML:2:0"
      xmlns:ns4="urn:oasis:names:tc:SPML:2:0:umpassword"
      xmlns:ns5="urn:oasis:names:tc:SPML:2:0:suspend"
      status="success"/>
  </soap:Body>
</soap:Envelope>
```

Table 5.1: Mapping table of the TMF615 operation names with TMF615 WSDL operation names.

TMF615 Operations	Equivalents at the TMF615 WSDL
addUser	SPMLAdd
modifyUser	SPMLModify
removeUser	SPMLDelete
listUsers	UMListUser
setPassword	UMSetPassword
resetPassword	UMResetPassword
expirePassword	UMExpirePassword
auditTrialForUsersAdminOperations	UMUsersAdminOperations
auditTrialForUsersProvisioningOperations	UMUsersProvisioningOperations
auditTargetsForUsersAdminOperations	UMTargetsAdminOperations

Table 5.2: Mapping table of the TMF615 operations with OSS operations

TMF615 Standard	Supported OSS Operation
addUser	addUser
modifyUser	modifyUser
removeUser	deleteUser
listUsers	listUser
setPassword	resetPassword
resetPassword	resetPassword
expirePassword	expirePassword
validatePassword	-
auditTrialForUsersAdminOperations	-
auditTrialForUsersProvisioningOperations	-
auditTargetsForUsersAdminOperations	-

5.4 IMPLEMENTATION OF THE MAPPING LAYER

This module implements real business logics and check every constraint provided by TMF and OSS vendor.

5.4.1 General mapping between the TMF615 and OSS parameters

There are many differences between the TMF615 and supported OSS. One of these problems is different definition of user provisioning data and user's accounts.

In TMF615, there can be exist more than one provision data per user and for each provision information there can be exist more than one account. On the other hand, specific OSS supports one provision data per user and one account per provision data.

Therefore one type account data is enough for TMF615 to map successfully to the OSS. User and account concepts are the same according to the OSS. Because of this equality, 'userId' and 'accountId' parameters takes same values in TMF615 and sometimes they can be alternatively used for each other by TMF. In addition to this, parameter 'PSOIdentifier.Id' can also be used instead of them as defined in SPML specifications. 'accountController' which is parameter of account data takes OSS specific value because this project aims integration of OSS and TMF615.

The main concept is implementing TMF615 and SPML standards exactly and mapping them to the OSS reasonable way.

5.4.2 Mapping between the TMF615 and OSS Operations

Some TMF615 operations are not supported by OSS. Some of them are supported but doesn't match exactly however they can be mapped to OSS operations.

Add User Mapping Table:

Table 5.3: Add User Mapping Table

TMF615	OSS	Occurrence	Mandatory/ Optional
userData		1	M
- userId	name	1	M
- firstName	-		
- lastName	-		
- email	-		
- contactOfficeNo	-		
- employeeId	-		
- addressLine1	-		
- addressLine2	-		
- addressLine3	-		
- contactMobileNo	-		
provisioningDataList[]		1	M
- accountData[]		1	M
- accountId	name	1	M
- validationInformation	password	1	M
- accountController	“AMS”	1	M
- validationCategory	-		
- accountCategory	-		
- accountUsageCategory	-		
- accountAdminStatus	-		
- targetData			
- targetName	“AMS”	1	M
- accessProfileData[]		1	M
- accessProfileName	“PAPGroup”	1	M
- accessProfileInfo[]		1..N	M
- accessProfileValue	papgroup	1	M
- accessProfileSubValue[]	-		
- excludeFlag	false	1	M
- authorizationData			
- roleName[]	role	1...N	M
- assignedAuthorization	-		
- scheduleData	-		
- provisioningDataId	-		

Modify User Mapping Table:

Table 5.4: Modify User Mapping Table

TMF615	OSS	Occurrence	Mandatory/ Optional
PSOIdentifier		1	M
- Id	name	1	M
modification[]		1	M
userData		1	M
- userId	name	1	M
- firstname	-		
- lastName	-		
- email	-		
- contactOfficeNo	-		
- employeeId	-		
- addressLine1	-		
- addressLine2	-		
- addressLine3	-		
- contactMobileNo	-		
provisioningDataList[]		1	M
- accountData[]		1	M
- accountId	name	1	M
- validationInformation	password	1	M
- accountController	"AMS"	1	M
- validationCategory	-		
- accountCategory	-		
- accountUsageCategory	-		
- accountAdminStatus	-		
- targetData			
- targetName	"AMS"	1	M
- accessProfileData[]		1	
- accessProfileName	"PAPGroup"	1	M
- accessProfileInfo[]		1..N	M
- accessProfileValue	papgroup	1	M
- accessProfileSubValue[]	-		
- excludeFlag	false	1	M
authorizationData		1	M
- roleName[]	role	1..N	M
- assignedAuthorization	-		
- scheduleData	-		
- provisioningDataId	-		

Delete User Mapping Table:

Table 5.5: Delete User Mapping Table

TMF615	OSS	Occurrence	Mandatory/ Optional
PSOIdentifier		1	M
- Id	name	1	M

List User Mapping Table (UM List Users):

Table 5.6: List User Mapping Table

TMF615	OSS	Occurrence	Mandatory/ Optional
PSOIdList[]		1..N	O
- PSOIdentifier		1	M
- Id	name	1	M
returnData	“EVERYTHING” ”DATA” ”IDENTIFIER”	1	O

Expire Password Mapping Table:

Table 5.7: Expire Password Mapping Table

TMF615	OSS	Occurrence	Mandatory/ Optional
PSOIdentifier		1	M
- Id	name	1	M

Reset Password Mapping Table:

Table 5.8: Reset Password Mapping Table

TMF615	OSS	Occurrence	Mandatory/ Optional
PSOIdentifier		1	M
- Id	name	1	M

Set Password Mapping Table:

Table 5.9: Set Password Mapping Table

TMF615	OSS	Occurrence	Mandatory/ Optional
PSOIdentifier		1	M
- Id	name	1	M
newPassword	password	1	M
oldPassword	-		

5.4.3 Common Exceptions

Common exception rules of all operations are listed below.

- a. If a mandatory input parameter of any operation is invalid, then exception “operation_failed_invalid_input_parameter” should be raised. And operation should be ended within this exception.
- b. If an optional input parameter of any operation is invalid, then exception “supported_optional_input_parameter_x” should be raised. Means of “x” is name of optional input parameter. And operation should be ended within this exception.
- c. Each operation should support “operation_failed_internal_problem” and this exception should be raised when internal problem occurs and operation cannot be completed. And operation should be ended within this exception.

5.5 IMPLEMENTATION OF THE SERVER ADAPTER

This module behaves a client side of SOAP protocol of OSS environment. Sends mapped OSS requests to the OSS and waits response. This module is created by Apache CXF framework. Codes are generated by WSDL by using CXF code generation plug-in. If any exceptions rise from OSS, they are converted to the TMF specific exceptions and sent back to the test tool.

6 PERFORMANCE EVOLUTION AND TESTING

6.1 UNIT TESTS AND FUNCTIONAL TESTS

Unit tests and functional are written by using JUnit framework. They can be classified into two classes. Because they are two types of specification, one of them comes from TMF615 side, other comes from OSS side. Hence, first class is requirements of TMF615 specification and other class is capability of OSS specifications. We can model requirements like below diagram.

Figure 6.1: Requirements of unit tests



6.2 END TO END TESTS

All end to end test cases pass the tests successfully. All test cases are controlled by third party tool to confirm result of test case and requirements of the test case.

New user creation:

New user creation was tested according to the TMF specifications. TMF add operation message was used to create a new user with provisioning information.

Assigning an account to the created user:

User provisioning was tested with assigning an account to the existing user.

New user creation with defined roles:

New user creation was tested with assigning roles to the user. Provisioning completed successfully.

User information listing:

Listing user information of requested users was tested.

User deletion:

Deleting user with accounts was tested.

User modification:

Modification of the user information was tested.

User suspension:

Suspension of unsuspended user was tested.

User resumption:

Resumption of suspended user was tested.

User account password expiration:

Expiration of the user account password was tested.

User account password reset:

Resetting the user password was tested.

User account password set up:

Setting a new account password was tested.

Querying user account status:

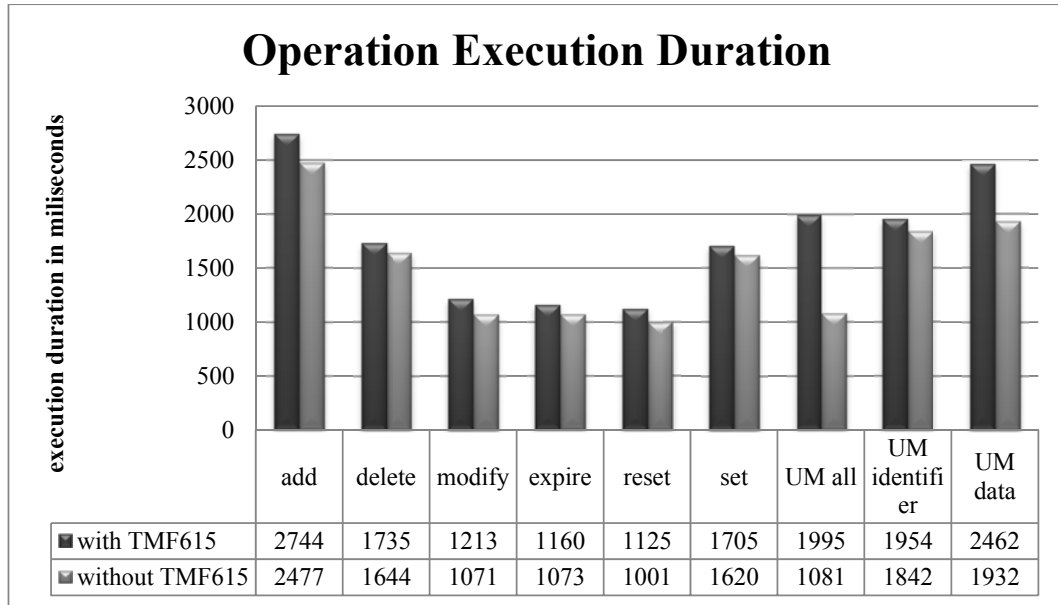
Querying the suspend status of user account was tested.

6.3 COMPARISON OF THE TMF615 WITH EXISTING SYSTEM

No way to compare TMF615 with other implementations because all other implementations are commercial and confidential. And also scope of this work is dependent to one OSS vendor. Therefore we can compare TMF615 implementation with legacy system. That means we compare our implemented operations with TMF615 support and without TMF615 support regarding to the operation execution time. As shown from figure 6.2 duration of operation is higher with TMF615 support. This comes from putting extra layer between the Service Provider and OSS. Hence, if TMF615 is implemented natively to the OSS, then execution time difference becomes lost. On the other hand, calculated times can be tricky because these times depend on server specification, network load and running processes both client and server

machines. Therefore, this evaluation shows cost of TMF615 layer between the OSS and Operator.

Figure 6.2: Comparison of Operation Execution Times



6.4 PERFORMANCE EVOLUTION

Reliability:

TMF615 uses synchronous communication, authorization and auditing. These features of the TMF615 provide the security of the system. And in this way, reliable systems are provided. Also, TMF615 based on SOAP communication protocol; therefore it is easy to make use HTTPS (Secure Hypertext Transfer Protocol) with TMF615.

Complexity:

Central user management with TMF615 decreases the complexity at the service provider side and increases the complexity at the legacy OSS environments. Because legacy OSSs already support some other protocols and TMF615 adds complexity to the OSS environments within them. But future versions of the OSSs only support TMF615 as a user management protocol and complexity is going to be decrease at the OSS environments.

Management:

User provisioning, user auditing are centralized. In this way, management is easy with the TMF615.

Usability:

Roles can be defined by service providers. This provides the easy usage of the system. TMF615 supports authorization matrix and legacy role mechanism.

Cost:

Clients don't have to have vendor specific client tool to access the service providers. They only use TMF615 interface to access the OSS. And this provides the money savings.

Performance:

Performance decreases with the TMF615 because of adding new web service which is located between the OSS operator and the vendor.

Compatibility:

Each vendor UM must compatible with the TMF615.

7 CONCLUSION

In telecommunication sector, there are many kinds of protocols and also number of these protocols has been growing day by day. Development of the protocols raises integration problem between the network elements and software. This challenging integration problem cause more cost and complexity and performance lost. After integration of a new protocol into the existing systems, management of the existing and new protocols together becomes harder. Network and system administrators have to learn different kinds of protocols and their tools to achieve efficient management of the systems and perform their daily responsibilities. A typical example of this issue is operator user management problems at the service provider side. With today's economics dynamics service providers generally work with different vendors for their OSS (Operation Support System) infrastructure. Vendors cause usage of many protocols and tools at the service provider side by their supports for different protocols. Therefore, an operator at the service provider uses these vendor-based protocols by vendor-specific tools. This causes performance lost regarding to the time domain and more complexity. TM (Tele management) Forum has been developing a new protocol which is named TMF615 to address these issues and this protocol makes standardization at the both service provider side and OSS vendor side.

This thesis gives information about TMF615 operator user management protocol briefly and also presents a work which is real-world implementation and integration of TMF615 protocol. The core subject of this thesis is implementation of the TMF615 as a proxy between the operator and OSS. In addition, implementation and integration of the TMF615 protocol to the existing network element have been conducted using as a proxy causes one more layer between the operator and OSS, and this extra layer causes extra delay for each unit operation. On the other hand, the operator can use one client tool and one standard protocol to manage the users by means of TMF615 standard. Operator performs many operations like CRUD (Create-Remove-Update-Delete) operations and audit operations, these operations can be managed from a one central user management system instead of different systems. In fact, it is best to implement TMF615 natively into the OSS system instead of extra layer; because this kind of implementation also

does not cause extra overhead as execution time duration per each unit operation. Therefore, OSS vendors are responsible for implementing TMF615 protocol into their OSS environments natively for their future OSS releases.

Future work includes, increasing security by using OSS specific security certificates between source and destination, and migrating TMF615 specification into existing OSS systems. Migration is difficult because existing user information must be reorganized according to the TMF615 specification.

REFERENCES

Books

Casey J., Fox B., Moser M., O'Brien T. & Redmond E., Shatzer L., Zyl J. V., *Maven: The Definitive Guide*, 2008 August, O'Reilly Media

Freeman E., Robson E. & Sierra K., *Head First Design Patterns*, 2004, O'Reilly Media

Martin R. C., *Agile Software Development, Principles, Patterns, and Practices*, October 2002, Prentice Hall

Liang Y. D., *Introduction to Java Programming Comprehensive Version*, Sixth Edition, December 2012, Prentice Hall

Periodicals

Aleem, M. I., *Centralized user management and TMF615*, Wipro Technologies, 2009

TM Forum, *Telecom OSS operator user management Information Agreement Release 1.1*, January, 2010

TM Forum Case Study, *TMF615 "OSS Identity Management"*, February 2008

TM Forum Webinar, *Successful Security for OSS environments using the TM Forum Standards*, 5 August 2010

Other Publications

Apache Tomcat Documentation, <http://tomcat.apache.org/tomcat-6.0-doc/index.html> [accessed 22 April 2012]

CXF User's Guide, <http://cxf.apache.org/docs/index.html> [accessed 22 April 2012]

Jetty Wiki, <http://jetty.codehaus.org/jetty/> [accessed 22 April 2012]

OASIS, *Service Provisioning Markup Language (SPML) Version 2*, April, 2006, <http://www.oasis-open.org/committees/download.php/18720/pstc-spml2-os.pdf> [accessed 22 April 2012]

Oracle, *Your First Cup: An Introduction to the Java EE Platform*, June 2010, <http://docs.oracle.com/javaee/5/firstcup/doc/firstcup.pdf> [accessed 22 April 2012]

W3C, *SOAP Version 1.2 Part 1: Messaging Framework*, 27 April 2007, <http://www.w3.org/TR/soap12-part1/> [accessed 22 April 2012]

W3C, *WSDL 1.1*, 15 March 2001, <http://www.w3.org/TR/wsdl> [accessed 22 April 2012]

<http://stackoverflow.com/questions/5104094/what-components-are-mvc-in-jsf-mvc-framework> [accessed 22 April 2012]