J Netw Syst Manage
DOI 10.1007/s10922-015-9347-7

Operator User Management System Based on the TMF615 Standard

Melike Yigit[1] · Muhammed Macit[1] · V. Cagri Gungor[1,2] · Taskin Kocak[1] · Oguz Ozhan[3]

Received: 6 June 2014 / Revised: 6 March 2015 / Accepted: 15 March 2015

Abstract Multi-vendor telecommunications 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 communications solutions put into place between the global UMS and the local UMS. Nowadays, in service provider environments OSSs exist that use multi-vendor communications' protocols. In the telecommunications sector, the centralized management of all these different OSSs can cause serious problems for 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 that work on these management systems. To address this need and to provide efficient operations among different service provider network components, this paper outlines the design and development of a TMF615 (Tele Management Forum) standard-based, common communication platform. In this respect, the proposed approach includes a common interface to address communication

✉ Melike Yigit
melike.yigit@stu.bahcesehir.edu.tr

Muhammed Macit
muhammed.macit@stu.bahcesehir.edu.tr

V. Cagri Gungor
cagri.gungor@agu.edu.tr; cagri.gungor@bahcesehir.edu.tr

Taskin Kocak
taskin.kocak@bahcesehir.edu.tr

Oguz Ozhan
oguz.ozhan@alcatel-lucent.com

[1] Department of Computer Engineering, Bahcesehir University, Istanbul, Turkey

[2] Department of Computer Engineering, AGU, Kayseri, Turkey

[3] Alcatel-Lucent, Teletas, Istanbul, Turkey

Published online: 22 March 2015

problems in multi-vendor, service provider environments. The interface and performance evaluations developed are some of the first solutions in this field, and the resulting solutions are converted into a commercial product with a high added value. In this regard, our proposed approach makes an important contribution to scientific literature and commercial applications. The realization of the proposed TMF615 standard-based interface enables the efficient and easy integration of existing and new OSSs of the service providers. In this way, a standardized interface is offered, along with a common communications platform adequate for all different systems. The vendors are thereby only responsible for application development based on specifications, and a standardized communications process is introduced for all related systems. This significantly facilitates the management of service providers, system performance is improved, and a massive cost reduction is provided at the same time. Consequently, the efficient management of network components is provided using a common standardized interface. In this respect, we aim to explain the TMF615 specifications; the evolution of UMS, OSSs and TMF615 with centralized UMS, as well as the implementation and performance evaluation of the TMF615 protocol are all explained in this paper.

Keywords Operational support system · Service providers · Standardized provisioning

1 Introduction

Network environments of service providers are managed by different operational support system (OSS) solutions, and by their original UMS. Communications problems between the central user management system (UMS-C) and the local UMS (UMS-L) system must be solved in order to provide service provider network management.

Nowadays, service providers use OSSs offered by different producers in their network environments. Each OSS uses a different communication technology, such as CORBA and XML (as shown in Fig. 4), and this can cause major difficulties for the Telecom operators to deal with integration problems and complex systems. Moreover, each OSS supports a different set of operations. Mapping these operations according to the requirements of the operator is a challenging issue. This is because operators use non-standardized solutions for their network management. Figure 1 illustrates the OSS problem for Telecom operators. As shown in the diagram, the OSS identity management system is the main activation point for the provision and control of the main operations. However, there is no auto-provision and therefore, all the operation requests to the OSSs are made manually. For instance, an operator with the business role of fault manager must deal with thousands of different personal accounts on thousands of different systems. The total time required for the creation of these local accounts, the testing of all these accounts by the operator, and the correction of all errors is very high. Therefore, the management of such local accounts is a big problem for any administration team, in terms of time and complexity. For this reason, a standardized, central specification

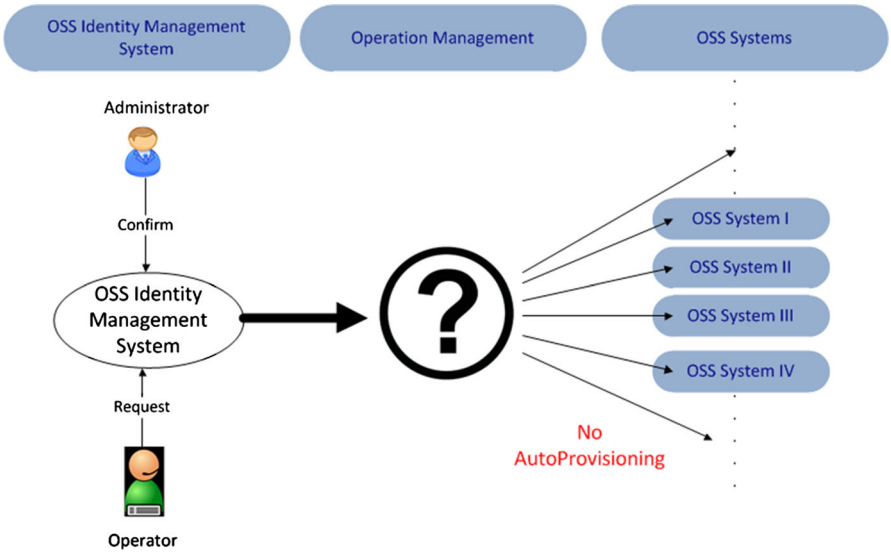

Fig. 1 Telecom operator OSS problem

and auditing mechanism are immediately needed, for the management of OSS network elements in a more secure, consistent, and efficient way. To meet these requirements, one initial group of service providers (Vodafone, T-Mobile, and Telefonica), OSS producers (Ericsson and Nokia-Siemens-Network) and identity management system producers (IBM and Wipro) leagued together to solve the difficulties posed by management systems. These difficulties are caused by using different UMS. This group raised their demands at the Telemanagement World (TMW) meeting. In the same meeting, collaboration was arranged with the TMForum [1] and they determined to solve problems. The first version of the TMF615 operator user management standard was issued at the Telemanagement World (TMW) meeting in 2008.

The purpose of reforming the TMF615 standard was to combine the existing and new OSSs of service providers and their applications with less effort. Service providers do not have to be interested in new and modified interfaces or their unique applications. Therefore, this also covers updated or covered versions of OSS applications. In this way, an operator can connect all the systems consecutively and use the accounts in the manner of the traditional user and audit management.

Multiple proprietary management systems, which are supplied by OSSs vendors, are used by the multi-vendor networks in a typical Service Provider environment. To obtain secure and consistent networks, both uniform and central provisioning operators are needed, which in turn necessitate specific authorizations for service providers. The TMF615 system provides those access rights and authorizations between the service provider and the operator. The TMF615 operator user management system is designed for the following:

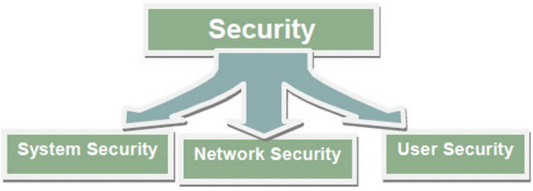

Fig. 2 Classification of the TMF615 security

- Reducing the complexity caused by the use of different protocols
- Providing security
- Increasing the number of people who understand the whole system.

Another design approach of the TMF615 is security. As shown in Fig. 2[1], security in the TMF615 is classified into three classes, including system security, network security and user security. However, we only consider the user security class while constructing our system, because we focus on user management systems and these systems require user security. User security includes the following security issues:

- Authentication of the system [1]
- User authorization [1]
- Operations audit [1]

In this paper, a common communications' platform, based on the TMF615 standard, is designed and developed, in order to work together with different network components for Telecom operators. Generally, this paper proposes an approach that includes an interface able to solve the communication problems caused by a service provider's network environment using different products from different producers. The approach presented thereby contributes scientific and commercial applications by producing new solutions. To the best of our knowledge, this study is the first paper focusing on detailed performance evaluations of OSSs using the TMF615 standard. Hence, it is expected that this study will provide a better understanding of OSSs and the potential advantages of the TMF615 standard and provoke interest among the research community to further explore this promising research area.

The remainder of this paper is organized as follows: Section 2 reviews the key literature on the TMF615 and OSSs; Sect. 3 presents an overview of the OSSs and their evaluations; Sect. 4 describes the TMF615 protocol, and the opportunities and challenges of supporting TMF615 and its general architecture; Sect. 5 specifies the proposed system architecture for describing systems' components; Sects. 6, 7, 8, 9, 10 describe the implementation of the proposed system, respectively. Sections 11 and 12 then provide an evaluation of the proposed system via a system performance evaluation, and test the efficiency of the system through further experimental evaluations. Finally, the paper is concluded and future works are described in Sect. 13.

2 Related Works

In this section, relevant research literature is reviewed according to different areas, including next generation operations systems and software (NGOSS), OSS inventions patented by the United States, and TMF-based OSSs already used by companies. The aim of this review is to summarize the status of existing studies and to specify the relationship between these studies and the proposed study.

Telecoms' corporations face many problems when integrating their business process frameworks with heterogeneous platforms [2]. These problems occur because these telecoms' corporations tend to use various business process flows, including trouble management and resource management. Operational support systems (OSSs) have been invented by the telecoms in order to manage all these business process workflows. However, the use of different OSSs can cause bigger problems. In this context, NGOSS was proposed by the TM Forum in 2009 [3] to help solve the problems of telecom corporations.

NGOSS consists of four components: the enhanced Telecom Operations Map (eTOM), SID, TNA, and TAM [4, 5]. The enterprise processes used by the service provider are all described by the eTOM. The eTOM analyzes all the processes at different levels - including Level-0, Level-1, Level-2, Level-3 and Level-4 - and specifies the importance and priority of the business processes for the telecoms' corporation [6]. The eTOM provides a detailed platform for the enterprises used for internal processes and for making working agreements with the other service providers. Suppliers use the eTOM to meet all of the requirements of the customers, because all the necessary functions, inputs, and outputs for the products are specified by the eTOM. The eTOM also provides a comprehensive model by defining the relationships between business processes for all the internal and external entities.

Other approaches have also been proposed to increase the efficiency of the NGOSS methodology. A systematic approach middleware architecture (MA) is one of these approaches [7]. The purpose of the MA study is to migrate the existing legacy systems to the NGOSS-compliant systems [7]. The OSS is used through Java Initiative (OSS/J) Service Provisioning to demonstrate how MA works for the provision of new telecom services. MA consists of three components—an inventory emulator, a client adapter, and a server adapter—that perform the migration of legacy OSSs to an NGOSS-compliant system at low cost.

Several studies have been performed to increase OSS efficiency and these are patented by the United States. One of these studies is the Open Gateway Framework [8]. The efficient modularization, extension, and adaptation of device functionality have been addressed by the Open Gateway Framework. The Open Gateway Framework enables the development of third-party applications on customer electronic devices with a custom Application Programming Interface (API), and this provides the portability between the different devices [8]. In this way, service providers can execute the services without changing device software [9].

Another study has been proposed by [9]. [9] offers an application program interface that allows the execution of software modules inside at different layers by

running a software program to perform tasks [9]. [10] proposes an identity management tool to provide identity management to multiple domains. A single access point is offered by the recommended tool for business support systems (BSSs), OSSs, and third-party systems. A unified user profile is created by the identity management tool in order to make mapping between customer identities and possible several domains [10]. Mapped information is shown by using a virtual directory that becomes accessible through a central data hub for other systems. In this way, various access networks using different channels can access the services with the same experience [10]. All the service and billing information coming from different sources can be shared without needing to integrate schemas into one data storage [10].

TMF-based OSSs exist that were adopted by many companies, including Ericsson and IBM. For instance, the eTOM is used by Ericsson in order to integrate necessary resources into its own system and third-party systems more easily [11]. The eTOM has many benefits for Ericsson since it provides the capacity to integrate and to update the modules independently into a single architecture without requiring compilation [11]. The eTOM also regularizes end-to-end processes for Ericsson and allows for efficient communication within the enterprise as well as with the customers. Although the eTOM has important advantages, it has the disadvantage of high provisioning time. Provisioning time can be reduced about 79.2 %, with the implementation of the TMF615 protocol in the eTOM [11].

IBM is another company that has its own OSS, called Tivoli. Tivoli increases the performance of business services by increasing advanced correlation, minimizing the cost by recovering lost capacity, and identifying the issues proactively to optimize customer satisfaction [12]. However, even with all of these benefits, Tivoli cannot provide an overall solution for IBM since each service provider uses a different OSS solution and cooperation between them is difficult, which in turn increases the provisioning time. Therefore, all of the OSS applications should support the TMF615, which would decrease overall provisioning time by about 98.4 % [11]. In this respect, features of the eTOM, Tivoli, and TMF615 are compared in Table 1, to show their respective efficiency in terms of reliability,

Table 1 Comparison of existing OSSs with TMF615

OSSs	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

complexity, energy efficiency, and manageability. As a result of the comparison between the features of the OSSs, the performance of TMF615 is shown to be better than the other solutions, except on the issue of operational time. The reason why TMF615 has a high operational time is explained in Sect. 12.

3 Operational Support Systems

An OSS consists of a set of programs that are used for controlling, monitoring, analyzing, and managing networks by the service providers [2]. The complexity of the system increases due to changes in market dynamics [13]. Different OSSs are designed by the different vendors to meet the service provider's needs [14]. The complexity of user management increases with the OSSs. The evolution of user management systems can be represented in the following steps [15]:

- Operational Support (OS)-based user management
- Network Element (NE)-specific user access
- Application-specific user access
- Unified user access control
- Multiple OS user access control
- Centralized user management for a specific OSS
- User management in heterogeneous networks

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

3.1 OS Based User Management

UNIX is used to develop OSSs. While in the OSS there is no implementation of user management, there are users and groups in the UNIX system, and these groups and users are responsible for the simple user access control [15].

In this system, the user performs a task after logging onto the system. The credentials of the user are controlled by UNIX. The OSS application prepares the response and this response is sent to the user.

Advantage of the system:

- Provides basic User Management (UM)

Disadvantages of the system:

- Flexibility of providing rights cannot be handled
- Applications cannot be accessed specifically
- Lacking NE access control

3.2 Network Element Specific User Access

Most network elements lack user access control [16]. However, user access control is needed for security. For this reason, network access control is located in the OSSs [17].

After a user logs in to the system, the user performs a task. The credentials of the user are controlled by UNIX. The user makes a request before performing the network task and the OSS application sends a response to the user. This system is simple, and provides a Network Element. However, this system lacks application-specific access [15].

3.3 Application Specific User Access

Application-Specific User Access is used for restricting user access to some applications. After a user logs into the system, their credentials are checked [18]. If a user wants to access a specific application, the application access controls it. If the user needs network access to perform a task, the NE access controls it and gives or does not give permission to this task. The system sends a response to the user.

This method is more secure and more flexible; both the NE access control and application access control are used. However, user management control is made difficult by the distributed system.

3.4 Unified User Access Control

Multiple access control is difficult for organizations because of its complexity. Unified access control (UAC) is used to solve this complexity. UAC wants to make user credentials from applications and processes. In this way, network applications are protected.

After a user logs into the system, the 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 a task, NE access controls it and gives or does not give permission to perform this task. The system sends a response to the user.

In this method, user management is easy; both NE access control and application access control are used. However, if there are multiple OSSs, user provision cannot be achieved.

3.5 Multiple Operational Support User Access Control

There are many different operating systems in existence, and applications use these different operating systems. For this reason, UAC must involve all operating systems. After a user logs into the system, the user's credentials are controlled by the UAC [19]. Many OSSs—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 [20]. If the user needs network access to perform a task, NE access controls it and gives or does not give

permission to perform this task. The system sends a response to the user. This NE-based system has been tested with many different tasks [21].

User management is very extensive; here, UAC, NE access control and application access control are all used [22]. However, in this system, the UAC brings a lot of complexity.

3.6 Centralized User Management for a Specific OSS

Telecom networks become bigger with each passing day, and their management subsequently becomes increasingly difficult. In order to handle management more easily, regions are separated [23].Separate clusters manage these different regions, while user management is done by the centralized user management [24].

The service provider works from a central location, and the user provisioning is done from this location. But this location is only used by one vendor; all the other steps in this method are identical with UAC [15].

Centralized user management enables an extensive user management approach; this is the advantage of this system. However, this system works for only one vendor OSS, which is the disadvantage.

3.7 User Management in Heterogonous Networks

In this method, one service provider has many OSS providers. Different services are combined from among the OSS vendors by the service provider to obtain the best performance [25]. User management solutions differ according to the vendor. For this reason, user provisioning and user management must be performed regularly. User management differs according to the vendor. For this reason, the insight of each vendor is unique.

This method affords the best performance, but users access the system according to the vendor. There are many different vendors and, for this reason, the user has to understand all these vendors. This increases the time and money needed 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 TMF615. The next section introduces the TMF615 protocol.

4 TMF615 Protocol

User Management is difficult in heterogeneous networks; the UMS-C TMF615 standard was prepared by the TMF to solve this difficulty [2]. User provisioning activities are standardized with this specification. In this way, service providers can provide compatible access rights and authorizations for the operators, using a UMS-C. The UMS-C and UMS-L exchange information to facilitate user provisioning.

User provisions and authorizations are provided by the UMS-C, which is used by the Service Providers. The UMS-C makes requests to the UMS-L. Then, the UMS-L processes the requests and sends a response to these requests. TMF615 also provides

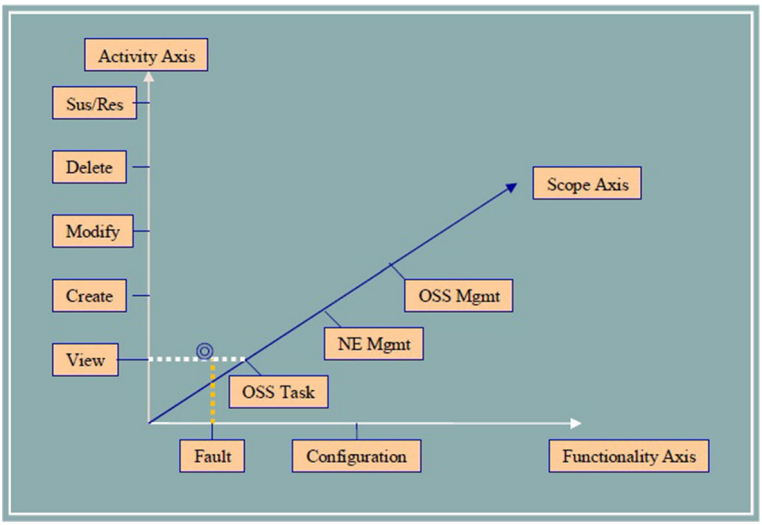

Fig. 3 Authorization space data flow

the centralized user management for bringing every OSS vendor's user management technique into one place. TMF615 succeeds in this by enabling communication between a UMS-C and UMS-L over a standard: namely, the Web Services Definition Language (WSDL). WSDL is an interface for enabling communication between two remote hosts: the UMS-C and UMS-L. Some of the concepts of TMF615 are introduced as follows:

Authorization Space Vendors use different authorization techniques for their users. Each authorization technique is designed based on the specific role of the vendor. Service providers and vendors specify their own roles as described in research [26]. For this reason, TMF615 provides a specific technique to assign roles. This technique is the authorization space.

As shown in Fig. 3 [1], the authorization space has three components. These are: the Functionality Axis, the Activity Axis and the Scope Axis [2]. The Functionality Axis specifies the work area of an OSS user. The Scope Axis defines the activities of the OSS user. The Activity Axis specifies the scope of a user account.

In this case, different vendors give different OSS applications to service providers. Service providers can only facilitate the management of Fault and Configuration. "Fault" and "View" are on different axes. For this reason, TMF615 uses the combination of these axes to assign roles.

Scheduling Scheduling is used to control the user access to the OSS resources. User provisioning provides the scheduling. According to TMF615, there are two types of scheduling. The first of these is weekly scheduling. The UMS-L learns what days of the week that the user accesses the resources from the weekly scheduling. Monthly scheduling is the second TMF615 scheduling type. In this type, the UMS-L learns what days of the month the user accesses the resources.

Audit Users' activities are monitored through the use of an audit. Monitoring is important for security and to prevent abuse. There are two types of auditing in TMF615; the first of these is the Status Audit. In this type, a synchronization between the UMS-C and UMS-L is used, and the data from the UMS-L and the UMS-C are compared. Synchronization is initiated if the comparison indicates differences. The second type is the Audit Trail. Security becomes a problem for this type, because monitoring is only undertaken for a certain period of time. After this time, there is no monitoring and there is subsequently no control of user activities.

4.1 General Architecture of TMF615 Protocol

The local vendor OSS includes the UMS-L, which is used for the UM solution; the UM of each vendor supports the TMF615. The UMS-L provides an interface between the UMS-C and the TMF615 specification.

One of the UMS-L components is the TMF615 agent. The TMF615 specifications are as follows:

- The TMF615 agent does not see the complexity of TMF615;
- WSDL interfaces are made by the TMF615 agent to communicate with the UMS-C;
- The TMF615 agent uses a database for monitoring operations;
- Some user provisioning requires communication between the Local UM and the TMF615 agent. For this reason, the TMF615 agent communicates with the Local UM in order to perform the operations;
- Synchronization is provided after a connection loss between the UMS-C and UMS-L, by the TMF615 agent.

Another component of the UMS-L is the Local UM. The Local UM is the vendor-specific UM solution. UMS-C requests are executed using Interface Reference Points (IRPs). Bringing out the IRPs is the task of the Local UM.

The IRP Manager is located at the UMS-C, and the IRP Agent is located at the UMS-L. This system's communication security is provided by adding the super user to the UMS-L. The UMS-C assigns an administrator role to some of the users on the UMS-L. User provisioning is performed by these administrators. Authorization is granted for the user provisioning requests by the UMS-L.

When user provisioning occurs at the UMS-L, the TMF agent's database is updated by the Local UM. If the Local UM does not update the database, the TMF Agent must periodically control the users, who are modified or new. For this reason, user information is updated continuously for security and monitoring by the Local UM. The general architecture of the centralized user management with the TMF615 protocol is explained above. Our proposed architecture is combined with this TMF615 protocol's architecture to make a solution for the Operation Support Systems (OSSs).

4.2 Opportunities and Challenges of Supporting TMF615 Protocol

Although the TMF615 protocol presents many opportunities, some real challenges also exist in supporting TMF615, especially when existing UMSs are considered. These opportunities and challenges presented by the TMF615 protocol can be outlined as follows.

4.2.1 Opportunities of Supporting TMF615 Protocol

TMF615 provides many advantages to the service providers, OSS providers and identity management system manufacturers. These advantages are listed below:

- *Less complex* TMF615 offers less complex systems to service providers (such as Vodafone, T-Mobile, Telefonica), to OSS providers (such as Ericsson, Nokia-Siemens-Network), and to identity management systems manufacturers (such as IBM and Wipro). The use of different protocols can result in a complex structure. Therefore, the providers mentioned above opt to be unified under a single protocol, i.e., the TMF615 protocol, in order to reduce complexity.
- *More reliable* Applying security procedures can become more difficult for service providers because of the complexity of the system. However, when TMF615 is used, the complexity decreases and security procedures are performed more easily.
- *Interoperability* TMF615 offers standardization to the OSS providers. In this way, the integration of systems becomes easier. Devices have a greater interoperability with the TMF615 protocol.
- *High performance* The performance of an OSS increases with the TMF615 protocol because when TMF615 is used in an OSS, the complexity of the system decreases.
- *Easy management* The management of an OSS is easy with TMF615, because TMF615 provides less complex systems.
- *Power saving* The number of devices used is minimized by TMF615, which saves power and decreases complexity within the system.
- *Durability of components* When an OSS uses the TMF615 protocol, its components live longer, and errors, which can be discovered later are minimized.
- *Low maintenance cost* The number of devices used decreases with TMF615 in an OSS. Therefore, the maintenance cost of the OSS becomes lower with this protocol.

4.2.2 Challenges of Supporting TMF615 Protocol

There are real challenges in supporting TMF615, when considering the existing UMSs. These are described as follows:

- *Operational delay* The implementation of the TMF615 protocol adds extra layers to the system architecture in order to enable mapping between the servers and the protocol. Therefore, the TMF615 protocol can cause extra operational delays when compared with existing UMSs.
- *Lack of coverage* The TMF615 protocol does not include some sets of attributes that are covered in the systems' operations. Therefore, the mapping of these attributes cannot be achieved, and operations cannot be implemented with full coverage to TMF615.
- *Unnecessary object creation* The TMF615 protocol has lots of attributes inside different objects. Each object consists of multiple objects and accessing a required attribute can be achieved by creating some of these unnecessary objects. This causes relatively cumbersome coding compared to existing UMSs.
- *Naming conflicts* Mapping is performed between the TMF615 protocol and the server, according to the attribute names and explanations of these attributes. However, some of the explanations of the TMF615 attributes are not coherent with the server attributes, even though the names of the attributes match each other. For this reason, the mapping of these kinds of attributes is a challenging issue in the implementation of the TMF615 protocol.

5 Proposed OSS Architecture

Figure 4 shows the traditional OSSs, in that a systems' operator accesses the various OSSs by using each OSS's client's tool, via distinct accessing protocols such as Service Oriented Architecture Protocol (SOAP), Extensive Markup Language (XML), Lightweight Directory Access Protocol (LDAP), etc. For instance, an

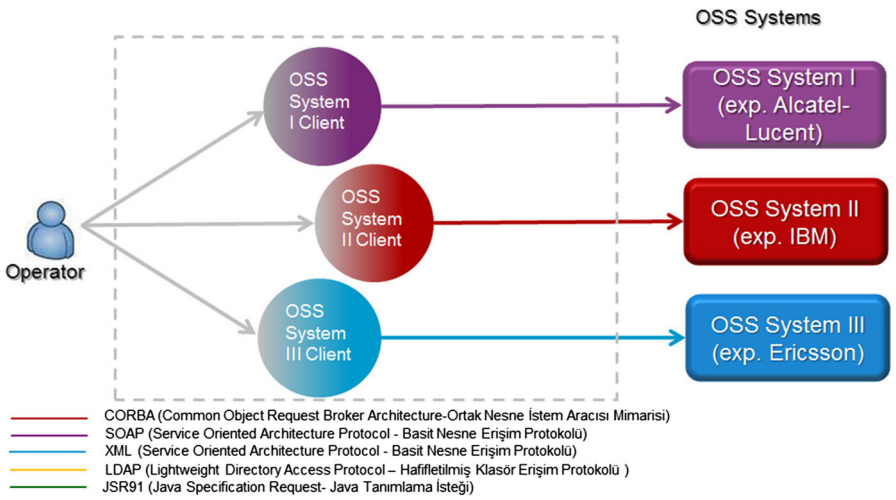

Fig. 4 Traditional OSS UMS

operator uses the client tool of the vendor OSS to access the OSS product. However, in our proposed system architecture, an operator uses the TMF615 client tool to access all the OSSs. In this way, the operator does not need to have all the client tools of the OSSs, and this reduces the complexity of the OSS architecture.

In this respect, our system architecture is composed of five parts, which are shown in Fig. 8. These are

- *Client Adapter* Server for the Web Service, responsible for the implementation of the TMF615 interface.
- *Server Adapter* Converts TMF615 messages to NBI, which is the Vendor OSS interface.
- *Processing Engine* Maps between the client adapter and server adapter.
- *Test Program* Sample of the OSS Client Tool.
- *Vendor OSS* OSS developed by the vendor.

The main purpose of the proposed architecture is to enable mapping between clients and servers, according to TMF615 that shows the proposed OSS architecture. As shown in Fig. 5, the work flow of the proposed system starts with a request from the operator to perform one of TMF615's operations, which include adding a new user to the OSSs, deleting a user from the OSSs, etc. After the operator's request is made, the TMF615 Client receives this request and transmits it to the Client Adapter. The client adapter is a web service used to communicate between two devices via a network. The Client Adapter sends this request to a mapping module, which makes conversions between the TMF615 protocol and OSSs. After the mapping, the request is transferred to the Server Adapter and is sent to the OSS. To give an example: an operator wants to add a new user to the vendor OSS. It enters the new user's information from the test tool, according to the TMF615 protocol. After the operator has entered the user information, the test tool sends this information to the Client Adapter, and the Client Adapter sends the request to the Mapping Module. The Mapping Module converts the information from TMF615 to the vendor OSS.

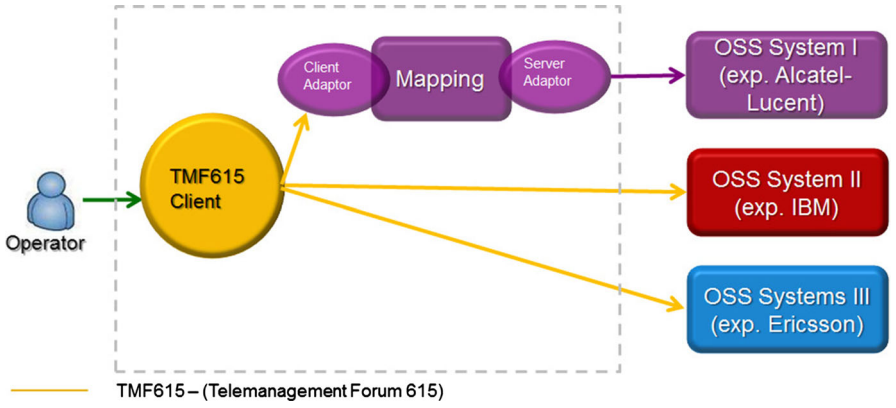

Fig. 5 Proposed OSS architecture

When the conversion is finished, the Server Adapter receives the request to send it to the vendor OSS. After the vendor OSS receives the request, it sends a response to the Server Adapter, and the Server Adapter transmits the response to the Mapping Module. The Mapping Module again makes a translation. This time, it translates the vendor OSS response to accommodate the response to the TMF615 protocol. When the mapping is finished, the Client Adapter delivers the response to the operator. This architecture allows for less complex, more reliable, and more standardized OSSs than the traditional OSSs.

6 Implementation of TMF615 Operations

The TMF615 interface consists of three types of operations: basic operations, password-related operations, and audit operations. Basic operations generally focus on user information and user provisioning data. TMF615 offers the WSDL solution set, which simplifies the implementation of standards. Nowadays, many software tools (such as Apache, CXF and Axis2) can produce interface codes using WSDL.

To describe the TMF615 interface simply:

- The TMF615 interface defines a users' basic information into a User class and users' provision information into a ProvisionData class;
- User account information (AccountData) and user authorization information (AuthorizationData) is included in ProvisionData;
- User account basis information, such as accountId, validationInformation, and user account orbit (targetData), are all defined into an AccountData class;
- User account access profiles (AccessProfile Value) are defined within the user account orbit.

This main purpose of this study is to implement the operations on the Vendor OSS according to the TMF615 standard. This standard was established to minimize the complexity of communication between the service provider, and the user management system provided by the OSS manufacturers. In this way, all the services provided by the service provider can be made to use of securely and correctly. All of the performance and functionality tests are therefore done to assure these secure services in this study.

6.1 "Add User" Operation

This operation is used to create a new user instance within the OSS. If a user has been created once, any other created requests will be rejected by the OSS. The "add user" operation is also used only for user creation; it is not used for user modification.

The "add user" operation requires user information, user provisioning information, and user authorization information. All these data generally cover the user id, accounts, roles, access profiles, and target system. A user can be associated with many accounts.

Authorizations can be set in the old way, by string list, or in the new way by authorization matrix, which is also recommended by the TM Forum. In our implementation, we use one account for one user because of integration issues. The OSS that is integrated with TMF615 does not support multiple accounts for a user. Hence, our implementation uses the same user id and account id to express different users or accounts

6.2 "Modify User" Operation

This operation is used to change, set, or remove the user privileges of an existing user. The "modify user" operation can also be used to change user account validation information that corresponds to the account password. On the other hand, the TM Forum recommends the use of password-related operations for this purpose, rather than the "modify user" operation. This operation supports multiple user modifications at a time. However, our implementation does not support multiple modifications at a time, except in it involves the "replace" mode. The modes of the operation can be defined as: "add", "replace" and "remove". For instance, assuming that a user has two roles - Software Admin and Security Admin - if we want to apply an Administrator role to that user, these three modes must be able to act differently.

6.3 "Remove User" Operation

This operation is used to remove an existing user. When a user is deleted, its accounts are also deleted. Only the user id is needed to perform this operation. Our implementation satisfies the TMF615 specifications exactly, regarding the "remove user" operation.

6.4 "Resume/Suspend User" Operations

The "suspend user" operation is used to suspend an existing user, and the "resume user" operation is used to resume an existing. Both operations do not produce any results, if the user is in the same state as the state being requested. The operations return a success message only if the users are in a different state rather than the requested state. Both operations may require a date to be specified in order to perform timed tasks.

Our implementation does not support timed tasks due to OSS limitations, but both operations can be still performed at the requested time.

6.5 User Activation Status ("isUserActive") Operation

This operation is used to query a user's activation status. User id is sufficient as an input for this operation.

In our implementation, the OSS returns a user activation status only if the OSS's "list user" operation is used. Therefore, the OSS response is converted from a "list user" operation to a TMF615 response. As should be understood from the

implementation of this operation, some of the operations are not directly mapped by their names. The important factor for operation mapping is the similarity of operation attributes, rather than the operation name.

6.6 "List User" Operations

There are two types of operations employed to list user(s). The first is the SPML List User operation, as defined in the SPML specification. The SPML List User operation only supports the listing of one user at a time; therefore, the TM Forum specifies a second "list user" operation—called UM (User Management) List Users—in order to increase the usability of user listing. The UM List User operation can take an unlimited user id to list their information; if a user id is not specified with the request, all users will be listed. If a user's id already exists in the request list, but does not correspond to any of the records at the OSS, then only the available users information will be listed without any error.

6.7 Password Related Operations

All password-related operations take an account as a target. Hence, an account id is mandatory, but a user id is not. The account controller is used to unify the target account, by using the account id together. If any accounts are not defined, then all that user's accounts are affected by the operation.

6.8 "Set Password" Operation

This operation is used to change the password of an existing user's account. This operation requires a user id, account id, account controller, and new password information, and it returns the success or failure status of the operation.

In our implementation, this operation is mapped with the "reset" operation at the OSS. Only the user id is considered when using this operation, because of the equivalence of the user and account at the OSS.

6.9 "Reset Password" Operation

This operation is used to reset the password for an existing user account. This operation requires a user id, account id, and account controller information, and it returns the status of the operation and the new password assigned to the user account.

6.10 "Expire Password" Operation

This operation is used to expire the password of an existing user account. This operation requires a user id, account id, account controller and new password information, and it returns the status of the operation's success or failure.

6.11 "Validate Password" Operation

This operation is used to validate the existing password of an existing user account. This operation requires a user id, account id, account controller and password information. The "validate password" operation was not implemented in our study, because the OSS does not support "validate" operations in any form.

7 Implementation of a Test Tool for Simulating Operational Support Systems Operations

A test tool is implemented in this step to simulate all the operations supported by the TMF615 standard. In this process, all the analysis, design, and coding processes are performed by the test tool software. This test tool is used as the supported program in this system.

Java Server Faces (JSF) technology, which is used to develop Java-based Web applications, is used for implementing the test tool. Test interfaces are realized according to each operation. The operations performed by the test tool are listed below:

- Add User Operation
- Modify User Operation
- Delete User Operation
- List User Operations
- SPMLLookupRequest (in one go, listing only one user)
- UMListUsers (in one go, listing more than one user)
- Reset Password Operation
- Suspend User Operation
- User Activation Status Operation
- Expire Password Operation
- Set Password Operation
- Resume User Operation
- Validate Password Operations
- Audit Operations

The test tool interfaces are realized according to the above-listed operations. For instance, the addUser operation test tool interface is shown in Fig. 6. As indicated in Fig. 6, the addUser operation includes three information parts: user information, to retrieve information from the user; access profile information, to define which operations can be accessed by the user; and role information, to specify the user role in the user management system. A general use case diagram of the operations is shown in Fig. 7. In the diagram, the TMF operation request is first sent to the client adapter, then it transmits the TMF request to the mapper that maps it and sends the request to the server adapter. The server adapter then sends the OSS operation request to the vendor OSS. Then the vendor OSS sends the response in the reverse order, back to the operator.

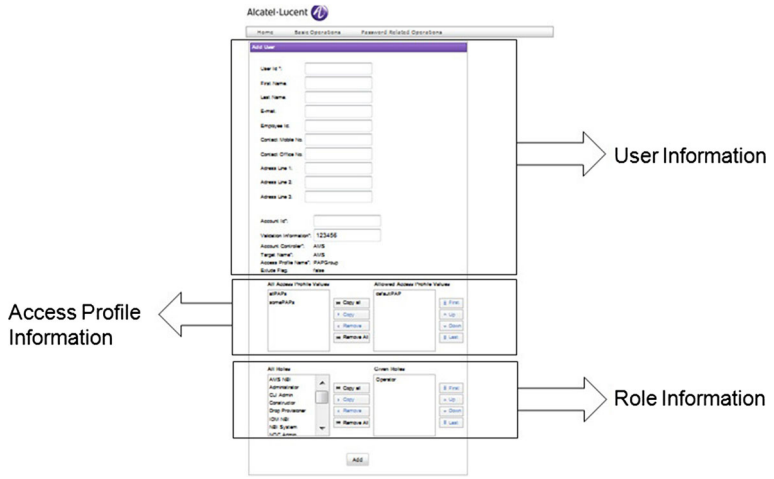

Fig. 6 Add user interface

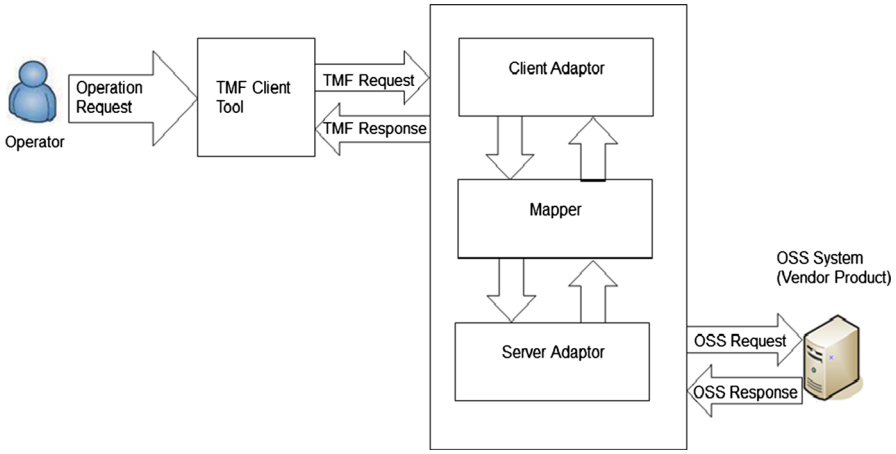

Fig. 7 General use case diagram

8 Implementation of the Client Adaptor

The implementation of the main module has been realized. This main module, which is the Client Adaptor, understands the TMF615 commands and responds to these commands. This main module allows the commands and requests, which are sent by the client, to be implemented according to the TMF615 standard. The client adapter is implemented to transmit user requests to the mapper module, which maps between the TMF615 and the vendor OSS, and returns the mapper's responses to the user.

The TMF615-standard client adaptors are specified in the TMF615 WSDL solution set. User requests, which come according to the TMF615 WSDL standards, must come conveniently in SPML, which is a framework based on XML and UM standards. Therefore, in our system, incoming requests to the client adapter are realized according to the TMF615 WSDL standard. The task of determining which requests are UM and which requests are SPML is undertaken by the TMF615 standard.

9 Interface Coding Between Server and Server Adaptor

An interface between the server and server adapter—called the Vendor Product Interface—is implemented on the vendor OSS API, as shown in Fig. 8.

The server adapter sends the incoming requests from the mapper module to the vendor OSS, and transmits the vendor's OSS responses to the mapper module. A web service, called the vendor OSS API, is implemented for the server adapter in this step. Communication between the vendor OSS and the Client Adaptor is provided by this web service. The inputs, outputs, operations and message types from the vendor OSS are identified with the WSDL, and according to this information in WSDL, the vendor OSS and TMF615 are mapped. An example for a 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, the types of input and output and the format in which they are transmitted are all specified by the WSDL. Connection

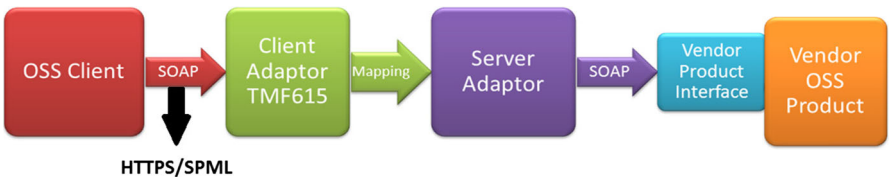

Fig. 8 System overview

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

10 Implementation of Processing An Engine (Mapper) Module

According to the TMF615 standard, mapping is realized between the client adapter and server adapter. This interface is built on the vendor's OSS API.

The process of mapping between the TMF615 and the vendor's OSS is described in the following subsections.

10.1 Mapping of the Operations between the TMF615 and the Vendor's OSS

TMF615 is identified with SPML and UM, which are developed by the TMF Forum as standards.

The vendor's OSS does not support all the TMF615 operations. Although some of the operations are supported by the vendor's OSS, some parameters of the TMF615 operations are not equivalent to the parameters of the OSS operations. However, this does not affect the field mapping. In Table 2, operations mappings are identified. As shown in Table 2, the names of TMF615 operations are mapped to the operations of the vendor's OSSs according to attributes included in the TMF615 WSDLs. Not all of the TMF615 operations are supported by the vendor's OSS, and therefore some of the TMF615 operations, such as auditTrialForTar-getsAuthorizationUsage, are not mapped.

10.2 Implementation of General Mapping Between the TMF615 and the Vendor's OSS

There are multiple provision data for every one user in the TMF615 standard. Each of the provision data has more than one account's information. However, in the vendor's OSS, users have only one account and therefore, there is only one provision data for every user, and only one account specified for each provision data. Some parameter names are given to TMF615 standard parameters. For instance, "OSS" is given as a name to the accountController variable. This study examines the integration of the vendor's OSS and TMF615 because of this situation; different account information and another accountController variable name (other than "OSS") are not required. User and account are the same concept when the vendor's OSS is considered. Therefore, a userId and accountId get the same values as specified in the TMF615 specifications, and they can sometimes be substituted for each other by the TMF615. In addition to this, as defined in the SPML specifications, the PSOIdentifier.Id variable can be used as a userId variable when it is required.

Table 2 Operations mapping

TMF615 Standard	Vendor OSS	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

The main purpose of this process is realize SPML and TMF615 standards completely and map them onto the vendor's OSS. In this respect, all TMF615 operations that are also supported by the vendor's OSS are mapped according to the TMF615 standard.

11 Performance Evaluations

The TMF615 protocol initiates a new structure for OSSs, by providing management of network elements via a common communication platform. As a result of the performed TMF615 protocol, service providers can combine new OSSs with their applications with relative ease and minimal effort. This situation provides easy system management, high system performance, and low system costs for the service providers.

The service providers' products, which use the TMF615 interface, have become more competitive and more favored in the market due to the economic benefits of the TMF615 interface in the telecoms' sector. Today, many service providers use OSSs, which are produced by different producers and work with different protocols, in their network systems. This situation can cause problems with consistency and

control for service providers that operate on a multinational and intercontinental scale.

When the existing systems are reviewed, it can be seen that the management of the OSSs is rather difficult, and the number of people who used this complex structure was relatively few before the TMF615 standard was formed. OSSs need to have high-level security, yet the operation of user management in a way that fits fully with the necessary security procedures makes the management of OSSs more difficult. Untapped accounts must also be found and necessary procedures implemented to prevent them being exposed. In this situation, extra security precautions must be taken and new work layers should be included in the general operations.

TMF615 provides solutions to all problems of the OSSs, which are explained above. Account control becomes easier and user account problems are solved faster with our system. It also reduces energy usage and system complexity by minimizing the number of devices used that are necessary for realizing operations. In this way, the service lives of the components used in the OSSs become longer and discoverable errors are minimized. Companies have safer structures when devices, which are compatible with TMF615, are prepared according to the companies' security policies and companies' OSSs. Generally, existing OSSs have problems relating to complexity, robustness and security. It can be concluded that all these problems are solvable by using a common communication platform, based on the TMF615 protocol. For this reason, our realized system works very well in terms of easier management; it is less complex, more robust and more secure. Therefore, our work remedies the weaknesses in related studies.

12 Experimental Evaluations

The role of TMF615 can be defined as that of an agent that holds one of the frontends of OSS. Hence, the implementation of TMF615 does not involve any database operation. Most of the operation execution time is spent on the OSS part of the general system, and this is caused by the database operation in the OSS. In addition, delays caused by file operations, such as object-to-XML and XML-to-object transformations, can be avoided by using specific frameworks (e.g. Apache CXF and JAXB) while implementing the proposed TMF615-based system. In this way, the proposed OSS becomes more stable and faster.

Both Fig. 9 and Table 3 show the delay of operations with and without the TMF615 implementation. The difference to the operational delays when the TMF615-based OSS is used can be clearly seen from Fig. 9 and Table 3. The delay of each operation is different. As shown in Fig. 9, the addUser operation has the biggest delay when compared to other operations. When the TMF615 protocol is implemented, the addUser operation involves a 6744 ms delay. However, when TMF615 is not implemented, this operation delay decreases to about 4477 ms. The delays to the remaining operations do not differ so much with and without TMF, since adding a new user includes more information than the other operations, which need to be mapped according to the TMF615 protocol. Getting a response from the

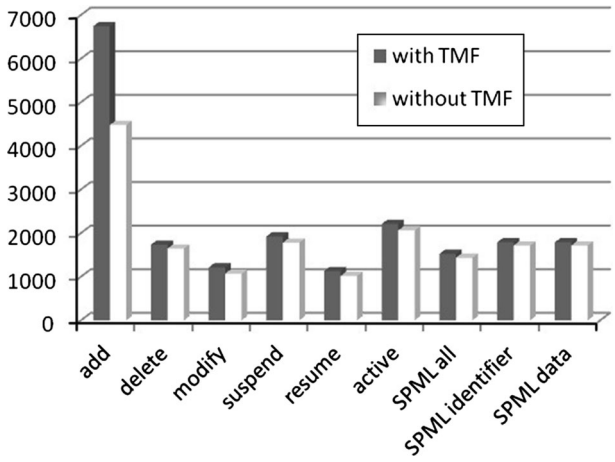

Fig. 9 Delays of operations with and without TMF615

Table 3 Delays of TMF615 operations with TMF615 and without TMF615 in milliseconds

Operations	Delays of operations with TMF615	Delays of operations without TMF615
Add User	6744	4477
Delete User	1735	1644
Modify User	1213	1071
Suspend User	1923	1780
Resume User	1129	1015
Is User Active	2210	2061
Expire Password	1160	1073
Reset Password	1125	1001
Set Password	1705	1620
Validate Password	79	70
UM List User—All	1995	1081
UM List User—Identifier	1954	1842
UM List User—Data	2462	19s32
SPML List User—All	1523	1433
SPML List User—Identifier	1788	1715
SPML List User—Data	1788	1715

vendor OSS for the addUser operation also takes a longer time than the other operations' response times. All of this demonstrates that the extra mapper layer adds an extra delay when the TMF is implemented.

Figures 10 and 11 show the proportion of overall delays caused by each TMF615 operation.These pie charts also show that the addUser operation takes the biggest

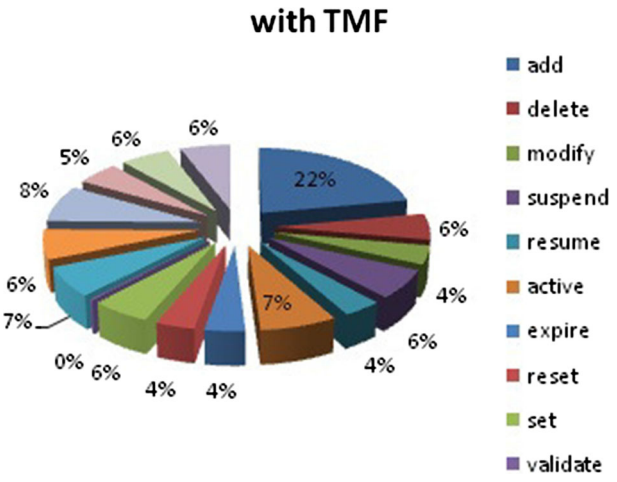

Fig. 10 Proportions of total delay by operation when TMF615 is implemented

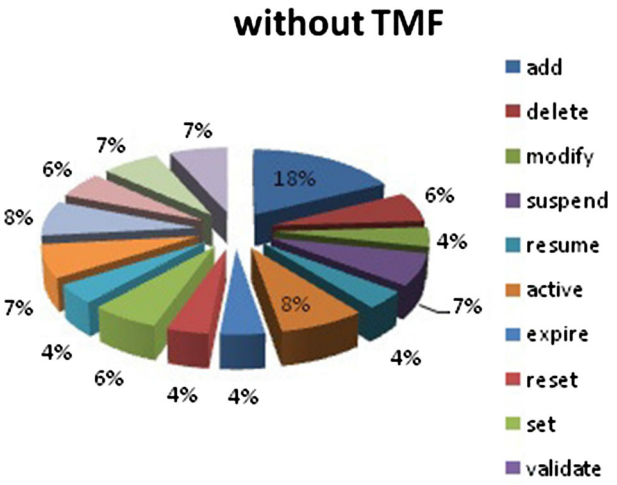

Fig. 11 Proportions of total delay by operation when TMF615 is not implemented

percentage of overall delays (22 %), as opposed to the other operations when the TMF is implemented.When the TMF is not implemented, the addUser operation is responsible for 18 % of the delays, as demonstrated in Fig. 11. This operational delay increases to 22 % when the TMF is implemented because of the extra layers of our implementation.

Although single operation execution time increases at the OSS environment, overall execution time of whole operations is decreased at the Service Provider environment. This can be explained by the fact that the daily tasks of the user management systems' administrator take less time when using a single interface, rather than various different interfaces and tools.

On the other hand, although the TMF615 agent operations require a short execution time, the TMF615 agent causes performance loss for entire OSS solution. Therefore, instead of a TMF615 agent, the native implementation of TMF615 in the OSS is the best possible choice where performance issues are concerned. However, the reimplementation of related OSS modules based on TMF615, and the migration of whole systems, are hard to perform.

13 Conclusion

This study introduces a novel method for designing and developing a common communication platform based on the TMF615 protocol for network components. The TMF615-based interface standard allows service providers to combine new and existing OSSs and applications quite easily and with minimal effort. This simplifies the service providers' system management, increases the system performance, and decreases the costs.

In the proposed system, all the supported operations of the TMF615 protocol are implemented in order to realize the User Management Solutions in the vendor OSS. Some TMF615 operations are not supported by the vendor OSS. In these situations, unsupported operation responses are sent for these operations. In this way, if these unsupported TMF615 operations are implemented to the vendor OSS in the future, they can later be mapped according to the TMF615 protocol. As a result of the implemented operations, all the operations worked seamlessly.

Through the experimental evaluations carried out, operational delays were calculated in order to measure the proposed system's performance according to the TMF615 protocol, and without the protocol. When the results are compared, operational delays are higher in the proposed system because of the extra layers, which include the test tool, client adapter, and processing engine. However, these delays are not very important when the system complexity, security, interoperability, easy management, component life, and power-saving advantages of the TMF615 standard are considered.

To the best of our knowledge, this study is the first paper to focus on a detailed performance evaluation of OSSs using the TMF615 standard. Hence, it is expected that this study will provide a better understanding of OSS and the potential advantages of the TMF615 standard, and provoke interest among the research community to further explore this promising research area.

Acknowledgments This work was supported by Alcatel-Lucent Teletaş and Turkish Ministry of Science, Industry and Technology's SanTez Program under Grant # 00632.STZ.2010-1.

References

1. TMF615: Telecom oss operator user management information agreement. TM Forum Release 1.1, 1.5 (2008)
2. Chou, T.-H., Seng, J.-L., Lin, B.: eTOM and e-services based trouble-management operations: a large scale telecom case study. Int. J. Technol. Manag. **43**(4), 383–403 (2008)

3. TMForum.org. Ngoss release 7.5 solution suite release notes. http://www.tmforum.org/DocumentLibrary/RN303NGOSSRelease/35614/article.html (2009a). 2 March 2009
4. TMForum.org. The ngoss technology-neutral architecture. http://www.tmforum.org/TechnicalSpecifications/28760/article.html?linkID=28760 (2006b). 2 March 2009
5. TMForum.org. Shared information/data (sid) model. http://www.tmforum.org/DocumentLibrary/NGOSSSIDSystemView/28818/article.html (2004). 11 Feb 2009
6. Milham, D., Ronco, E.: How can the eTOM/sup/spl reg//framework help service providers in today's marketplace? In: Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP, vol. 1, pp. 59–71. IEEE (2004)
7. Wang, C.-Y., Lin, P., Shih, C.-S., Fu, H.-L., Jeng, J.-Y.: A middleware approach for migration of legacy telecom operational support systems into ngoss-compliant. In: Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific, pp. 1–7. IEEE (2011)
8. Borghini, M., Capuozzo, G., D'angelo, G.: Open gateway framework. US Patent 8,392,933 (2013)
9. Eslambolchi, H., McCanuel, J., Vasa, P.: Tiered and modular approach to operational support systems. US Patent 7,882,209 (2011)
10. Maes, S.: Shared view of customers across business support systems (bss) and a service delivery platform (sdp). US Patent 8,073,810 (2011)
11. Driss, B., Hilbert, F., Wallis, K.: Ericsson solution framework based on tmf standards. www.ericsson.com (2010)
12. Shi, F.S., Zhou, Y., Shi, P.: The design of operation and maintenance management system about it platform based on tivoli. In: Sung, W.-P., Kao, J.C.-M., Chen, R. (eds.) Applied Mechanics and Materials, vol. 599, pp. 2215–2219. Trans Tech Publications, Switzerland (2014)
13. Nye, B., Hong, D.S.: High level operational support system. US Patent 7,302,612 (2007)
14. Harada, Y., Tomita, Y., Sugiyama, S.: Operation support system and method. US Patent App. 10/278,886, 24 Oct 2002
15. Aleem, M.I.: Auditing in TMF615 and its benefits. In: Wipro Council for Industry Research (2011)
16. Rader, Reinhard: Operations support systems for mission critical public safety communication networks. Bell Labs Tech. J. **16**(3), 151–162 (2011)
17. Ridgely, C.H., Wright III, S.L.: Method, computer program product, and apparatus for providing a universal high speed link for operation support system data. US Patent 7,796,641 (2010)
18. Zhou, Y., Dong, Y., Huang, X, Yoshikawa, H.: A human interface toolkit for developing operation support system of complex industrial systems with IVI-COM technology. In: Human Interface and the Management of Information. Interacting with Information, pp. 82–89. Springer (2011)
19. Choi, C.-S., Seo, D.-I.: On the study of service security model for privacy using global user management framework. In: Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference, vol. 1, pp. 4-pp. IEEE (2006)
20. Kadowaki, K., Fujita, S.: A dynamic user management in networked consumer electronics via authentication proxies. In: Parallel and Distributed Computing, Applications and Technologies, 2009 International Conference on, pp. 195–200. IEEE (2009)
21. Strassner, J., Raymer, D.: Implementing next generation services using policy-based management and autonomic computing principles. In: Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP, pp. 1–15. IEEE (2006)
22. Kasuya, H., Muneshima, R.: Operation support system. US Patent App. 13/957,599 (2013)
23. Tselentis, G.: Towards the Future Internet: A European Research Perspective. IOS press, Amsterdam (2009)
24. Gupta, A.: Network management: current trends and future perspectives. J. Netw. Syst. Manag. **14**(4), 483–491 (2006)
25. Gotanda, M., Nakamura, T., Uchikubo, A.: Operation support system. US Patent App. 12/114,525 (2008)
26. Meyners, M., Driss, B., Feger, U.: TMF615 oss identity management. TM Forum Case Study, pp. 1–11 (2008)

Melike Yigit received her B.S. and M.S. degrees in computer engineering from Bahcesehir University, Istanbul, Turkey, in 2010 and 2012, respectively. Currently, she is a Ph.D. student at Bahcesehir University, Istanbul, Turkey and works at Turkish Airlines (THY), which is an international airlines in Turkey, as a Business Analyst.

Muhammed Macit received his B.S. degree in Computer Engineering from Marmara University, Istanbul, Turkey in 2010 and his M.S. degree in Computer Engineering from Bahcesehir University, Istanbul, Turkey in 2012. He is currently working as a Senior Software Engineer and pursuing to his Ph.D. degree in Computer Engineering from Bahcesehir University, Istanbul, Turkey.

V. Cagri Gungor received his B.S. and M.S. degrees in Electrical and Electronics Engineering from Middle East Technical University, Ankara, Turkey, in 2001 and 2003, respectively. He received his Ph.D. degree in electrical and computer engineering from the Broadband and Wireless Networking Laboratory, Georgia Institute of Technology, Atlanta, GA, USA, in 2007. Currently, he is an Associate Professor and Chair of Computer Engineering Department, Abdullah Gul University (AGU), Kayseri, Turkey.

Taskin Kocak received B.S. degrees in Electrical and Electronic Engineering, and in Physics (as a double major) from Bogazici University, Istanbul, Turkey in 1996 as well as M.S. and Ph.D. degrees in Electrical and Computer Engineering from Duke University, Durham, NC, USA in 1998 and 2001, respectively. He is currently a full-professor and chairman of the Computer Engineering Department at Bahcesehir University, Istanbul, Turkey.

Oguz Ozhan received his B.S. degree in Food Engineering from Hacettepe University, Ankara, Turkey and received his M.S. degree in computer engineering from Ege University, Izmir, Turkey, in 1999 and 2007, respectively. Currently, he has been working as a software engineer at Alcatel-Lucent Teletas A.S. since 2011.