A Distributed-Based Approach for Advanced Telecommunication Services Management

JOSÉ-F. MARTÍNEZ[1], JESÚS VILLAMOR, JUAN-C. YELMO
Department of Telematic Systems Engineering - DIT
Universidad Politécnica de Madrid -UPM
ETSI Telecomunicaión-DIT/UPM, Ciudad Uniersitaria s/n, 28040, Madrid
SPAIN

Abstract: - This paper presents the main contribution of the Service Management Subsystem (SMS) on the *SCARAB* ACTS European Project. Services and users are deployed and managed through a Subscription Management Subsystem (SUB) assisted by the Configuration Management Manager Subsystem (CMM) in order to manage the service and users life cycle. The SUB is a TINA based subsystem that characterises and customises the services according to the stakeholder. The users, both subscribers and end users, have been provided of privileges for making their own business through the on line subscription ancillary service. The CMM Service has been conceived to cover the lack of deployment and withdrawal management of service components in a service provider domain for the best control of the service components life cycle. Applications for off and on line subscription, subscription components and service components for the on line subscription, and finally administration, configuration and monitoring tools of service components have been carried out on a CORBA DPE platform. The SUB was validated and demonstrated through field trials in the VITAL ACTS Project, providing feedback to the TINA specification on the Service Management discipline. Both the SUB and the CMM, as the SMS, have been tested in the SCARAB project with worthy mention results.

Key-Words: - Service Management, CORBA, TINA, Java, Subscription Management, Configuration Management, Software Deployment Life Cycle.

1 Introduction

The modern society demands new telecommunication services to couple more and more its distributed nature:

1) The teleconference, the teletraining, the chat service or the video on demand are more and more outstanding services in the market.

2) In parallel, the great amount of users and the disparity of preferences among them have forced to create large databases and service profiles that personalise the service in order to accommodate the services to the actual needs of the users.

3) In addition, the service providers need a set of software components in order to create, configure and deliver their services.

This is difficult to achieve because the end user demands new and complex services. This triple problematic is part of the so-called *Service Management*, an heterogeneous discipline that undertakes fields so diverse as service provisioning, configuration management, subscription management or billing management. Standardisation forums as ATM Forum, DAVIC, NMF, OMG, X-Open or TINA-C have contributed to them.

TINA-C has designed an architecture for the next generation of telecommunications networks [1]. Its network architecture and service architecture specify software components on a DPE platform as CORBA. Its validation has been commended to auxiliary projects as ReTina, DOLMEN, SCREEN o VITAL [2] [3].

The Software Engineering Group of the Telematic Engineering Department of the Universidad Politécnica of Madrid (UPM) has taken part in the ACTS VITAL (*Validation of Integrated Telecommunications Architecture for the Long term*) project (AC003) and the ACTS SCARAB (*Smart Card and Agent enabled Reliable Access*) project (AC30055). This researching group has undertaken service management activities in the subscription and configuration management fields whose results are

[1] *José Fernán Martínez Ortega*, is a teacher of University of Cauca (Colombia) – UNICAUCA, invited researcher in the DIT-UPM (Spain), his work is sponsored by COLCIENCIAS – Colombia and AECI Spain

worthwhile and have feedback strongly TINA-C and, hence, contributed actively to the Service Management discipline.

The life cycle service is a complex issue, because the service components may be different among them, each component may be used by different services and, more over, a new version of an existent component might be required for creating a new service, whereas its older version may be needed by an existent service [4]. These related life cycles of service components are a management problem and must be taken into account. The Configuration Management Manager (CMM) deals with the deployment and withdrawal phases in the service life cycle. This service configuration management problematic has not been dealt yet in depth, and it is an important part in the introduction and maintenance of advanced telecommunication services.

2 Objectives and Conceptual Framework

The objective of the Service Management Subsystem (SMS) is to assist the service providers in the definition of their service. It is defined in terms of their attributes in a generic way (service type management). To control the user's life cycle and their relationships with the services, there has been necessary to customise the services to the needs of the owners (service customisation). Another important issue is the deployment of the service components, their configuration and monitoring (configuration management) [5]. The SMS conceptual framework will be explained following four ODP-RM viewpoints [6]:

2.1 Business Model

The SMS works in the service provider domain[2]. Two stakeholders are the main subscription actors: the service provider and the user. The user will have two different roles, the subscriber and the end-user. Services and users interlace their life cycles in this way:

Service life cycle: The service provider deploys the service. The service is characterised by service attributes and is offered to the subscriber, who subscribes the service and then customise it (as service profiles) to end users.

[2] A domain is a stakeholder-owned set of computers where related software components (typically, CORBA servers) run for the success of his activity.

User life cycle: The service provider manages his own users and can register new subscribers, and they, in turn, can register their own end users. Once the subscriber is registered, he can be subscribed to services. When the end-user exists, he can be subscribed to particularisation of services. Then, the end user can access to the service according his privileges.

2.2 Engineering Model

The standard CORBA [7] has been used as the platform that allows the object distribution and its access through well defined interfaces. It uses IDL language through the IIOP protocol. This guaranties the programming language transparency (C++ and Java, in our case just Java), machine transparency and operating system transparency (Windows or UNIX), and, on the other hand, the JVM ensures the same look and feel. One implementation of CORBA with Java mapping is used. Finally, the configuration management approach uses some CORBA services such as naming, event , properties and life cycle service in order to assist the configuration of the service components [8].

2.3 Information Model

The SCARAB service management information model is defined through an UML diagram [9], The information on the users' roles and the services are stored and managed by the SUB subsystem. The services are particularised through three refinements according the stakeholder[10]. This is shown in the table 1.

Stakeholders	Service attributes
Service Provider	Service Template
Subscriber	Subscription Profile
End User	Service Profile

Table 1 Service refinements

The services are characterised by the service attributes. Additionally, every attribute is constrained by its permissions. This enables/disables a stakeholder to modify his/her service attributes which would be able to be fixed by the service provider, be configured by the subscriber or by the end user.

2.4 Computational Model

Three computational objects[3] compose the SMS Model (STM, SM and SubAg) [10]. In turn, the CMM CO manages all the factories of the service management COs and also monitories all the components of the service and solves activation problems. The *Service Type Manager* (STM) CO is responsible for handling service templates, which defines service settings and attributes that represent service capabilities set by the Service Provider. *The Subscription Manager* (SM) CO is responsible for managing the information of the subscribers and end-users. It manages the service settings of the stakeholders (subscriptions for the subscribers and service profiles for the end-users), portfolios and subscription assignment groups. It is the join of the TINA's *Subscriber Manager* (SubM) and *Subscription Registrar* (SubRgs) [11]. The *Subscription Agent* (SubAg) gives access session related computational objects - *Users Agents* (UA) - subscription information on the users. The CMM CO manages the factories of the STM, SM, SubAg and UA. Both the SubAg and the UA are agents whose activation depends on the user existence. The SM dynamically sets each user's service profile into the SubAg through the CMM. Furthermore, the SM is responsible of managing the subscriber and end-user life cycles.

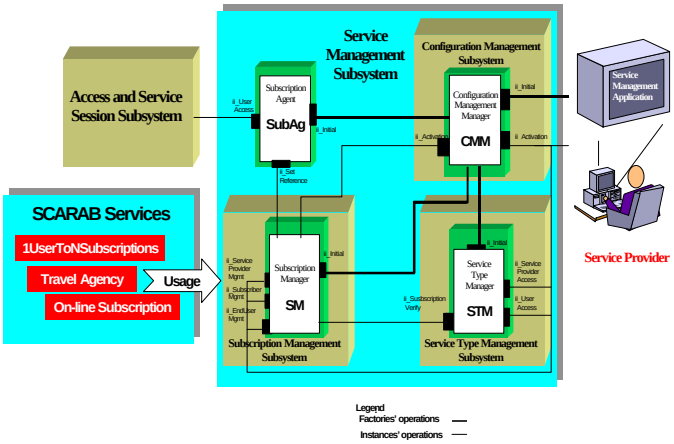

Fig.1 The SCARAB Service Management Subsystem

[3] *A Computational Object* (CO), is a piece of software that may act as server or client on a DPE platform, typically CORBA, interacting with other COs or applications for achieving a common objective.

3 Service Management Subsystem Architecture

A global view of the architecture can be seen in the block diagram depicted in the figure 1.

3.1 The Subscription Agent (SubAg)

SubAg remains as a free agent that serves the access and service session subsystem. This CO knows which services an user is subscribed to. There will be as many SubAg COs as UA COs, and its main job will be to lighten the work to the SM.

3.2 The Service Type Management Subsystem

The STM assists the service provider to the best definition of the service in terms of their attributes. This typing must be generic enough for satisfying the needs of any service provider in the design of the service.

3.3 The Subscription Management Subsystem

The SM manages the user and subscriber life cycle with all their service relationships with the (subscriptions for the subscribers – account holders – and service profiles for the end-users). It contains Service Provider's users and services for their management.

On the other hand, the service management application (*off line subscription*) populates the persistent information objects (STM, SM) with the proper data. In addition, it is possible to populate objects by accessing to the service components indirectly through the access and service components (*On-line subscription ancillary service-OLS*) [14].

3.4 The On-line Subscription Management Subsystem

Allows the users to subscribe some service offered by his/her progenitor. The users can change the attributes of the subscribed service. This subsystem is indeed an ancillary service, hence it has a Service Session Manager (OLS-SSM) and Service Session User Application (OLS-ssUAP) as any TINA service. The SSM has three interfaces for the three different usages (Service Provider, Subscriber and End-User usage). The OLS-ssUAP, in turn, has been launched using the SCARAB Terminal Manager subsystem that allows a reliable access in a Service Provider

domain.

4 The Configuration Management Subsystem

It manages the service components for the best control of its life cycle for deployment and withdrawal. STM, SM and the access session related UAF (UA Factory) are some of the components profited by this functionality, this includes configuration and assembling of service components (installation/uninstallation), activation/deactivation, updating/reconfiguration of components and adaptation to service components changes [12].

The figure 2 shows the general Configuration Management Manager architecture. The integration between all the subsystem components is by means of interfaces IDL using IIOP in a CORBA bus through an event channel complaint with the CORBA event service.

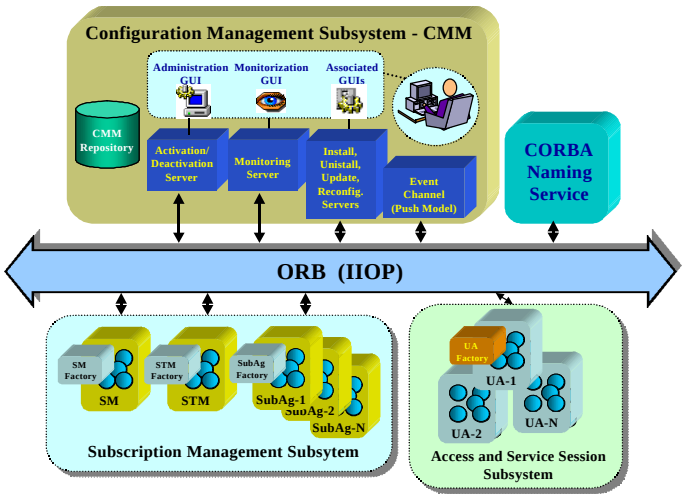

Fig.2 The Configuration Management Manager Architecture

4.1 Creation and Deletion Management

The creation and deletion management is carried out by the CMM activation/deactivation (A/D) server. The A/D server is based in the CORBA life cycle service using a generic factory model. It defines generic and open interfaces which allow to create (activate) and delete (deactivate) any kind of COs in a service provider domain. The components that use the CMM follow the OSI System Management model, which defines both Managed and Manager Objects:

A managed object is activated or deactivated and must inherit of the interfaces defined by the CMM A/D server.

A manager object (CMM A/D) has client/server roles and is in charge of create/delete the objects under its administration. The server role is taken when a server wants to create a instance of a CO in the system by means of requesting to CMM. The client role is taken when the CMM on behalf of a server tries to activate or deactivate a CO and it changes its server role by returning back the reference of the CO created.

It is important to mention here that the CMM uses CORBA DII for creation/deletion of COs. In addition, it allows servers to define activation/deactivation policies of COs in runtime (for instance, a server can have several COs, depending on who makes the request to activate them). In the other hand, the A/D server has the additional functionality of allowing the reactivation of COs associated to servers under requesting, when an exception in the system is arisen.

Finally, the A/D server has an administration tool, which knows about servers (factories) under the CMM control (i.e. localisation, name of servers, security information about invocation and launching and associated interfaces). This tool allows to make bootstrap of the COs, register/unregister the servers, and, summarising, all the management capabilities on all the servers (create/delete subsystem, add/remove factories of servers, etc) in a service provider domain.

4.2 Monitoring

The Monitoring server is in charge of monitoring the CO under the CMM control in a provider domain. When a new object is created or deleted, the monitor manages the status of the COs. In addition, the monitoring server is be able to test and show the status of the COs created in the system. For example, if a CO dead or had a problem, the monitoring server would report the problem and, in collaboration with the A/D server, would try to solve the problem. In the other hand, the monitoring server has a monitoring tool with a GUI in charge of depicting the system status. Some of the functionality of the tool is to show the status, to ping COs in order to know whether a CO was alive or had a problem or to look for a particular object in the system. Finally, the monitoring GUI shows a static and dynamical view of the whole of COs running in the system: the static view displays the object factories that creates COs for each kind of server and the dynamic view exhibits the COs instantiated for each server running in the service provider domain.

4.3 Installation/Uninstall and Updating Management

The installing/uninstalling/updating is another part of the configuration and deployment life cycle [13]. This process follows a packaging approach. The configuration of the system is defined through its properties using the CORBA property service. These properties are used by the current servers to generate a package with all the necessary components to install a given subsystem (i.e. SUB). This functionality is working progress and is being tested and redesigned. However, it is independent of the whole CMM because the current architecture is composed by lightly coupled components that are being and are going to be defined shortly. An event channel couples these components, as can be described in the next section.

4.4 Event Channel for Configuration Management and Deployment

The CMM event server is the means of integration between the servers that takes part in the Configuration Management Subsystem. This event channel is compliant with the CORBA event service, using *push model* and *type events* in order to couple the servers that configure and deploy the components that deliver the advanced telecommunications services. For instance, the A/D server and the monitoring server register their intentions of receiving and delivering events (new activation/deactivation, a problem in a CO creation, alive/dead CO status) in order to manage the entire COs in the system. The activation server uses DII for activating the instances of a CO and sends an event for updating the static or dynamical view in the Monitor GUI. When a exception arises, the event channel is in charge of reporting the problem to the monitor, which will take the suited decisions (i.e. pinging a server and reporting its status), then the A/D server uses the event channel to reactivate/delete COs.

4.5 Repositories of Configuration

The CMM Repository is a collection of databases that stores different kind of information. In the case of the A/D server, the CMM activation database stores the information associated to the server description itself (i.e. interfaces, server name, security information, etc). Another kind of information is related to configuration properties of COs and relationships between them by means of the CORBA properties and relationship service.

4.6 References Management

The Naming service is in charge of managing the references of the COs running in a service provider. In this case, STM, SM, SubAg and UA (the two last COs are created for each user) and so on. The CMM has a naming manager in order to use the naming service interfaces and stores the references of the servers (COs) running in the service provider.

5 Conclusions

The implemented SMS has validated the TINA specification. It has demonstrated to be efficient enough for the management of the users and services life cycles for an advanced service architecture as the proposed by TINA. The first general comment of the SCARAB project audit report was *"It is in general a good project with came to an excellent demonstrator"*.

The CMM subsystem it is a good starting point for covering the lack of configuration management of advanced telecommunication services, CMM has been proved efficient to solve the management problem between the life cycles of services components. Furthermore, the CMM is an open subsystem that allows managing the needs of any distributed environment.

The SMS has been integrated with the SCARAB Access and Service Session subsystem, which uses the CORBA and agent technology in a federated service provider domain.

The OLS has extended the subscription functionality to the users, who are the real profiteers of the subscribed services.

These subsystems have been successfully tested in the VITAL and SCARAB ACTS European projects. The integration between the CORBA platform and the relational technology has served to evaluate a possible way of making the distribution of process components, achieving localisation transparency and through JVM to guarantee operating system transparency and same look and feel independent of a specific environment.

References:

[1] TINA-C, Overall Concepts and Principles of TINA, *TINA Deliverable*, Version 1.0., 1995.

[2] VITAL Project, Deliverable D13: ODTA Validation 3rd Trial, *VITAL Deliverable*, 1998.

[3] VITAL Project, Deliverable D15: User Application Validation 3rd Trial, *VITAL Deliverable*, 1998.

[4] TINA-C, Service Architecture, *TINA*

Deliverable, Version 5.0, 1997.
[5] TINA-C, Service Architecture, *TINA Deliverable*, Version 1.0, 1997.
[6] ITU-T, ODP Standard: Reference Model for Open Distribute Processing. Part I. Overview and Part 2 Foundations, *Recommendation ITU-T X.901 and X.092 ISO/IEC 10746-1 (DIS) and 10746-2 (IS)*, 1997 and 1996.
[7] Object Management Group, The Common Object Request Broker: Architecture and Specification, *OMG Standard*, 1995.
[8] Object Management Group. CORBA Services: Common Object Services Specification, *OMG Standard (1995)*, update, 1997.
[9] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley Longman Inc., 1999.
[10] TINA-C, Service Component Specification, *TINA Deliverable*, Version 1.0b, 1998.
[11] TINA-C, Service Component Specification, *TINA Deliverable*, Version 1.0, 1995.
[12] R. Conradi, B. Westfechtl, Version Models for Software Configuration Management, *ACM Computing Surveys*, 1998.
[13] Van der Hoek André, Hall Richard, Carzaniga Antonio, Heimbigner Dennis and Wolf Alexander, Software Deployment: Extending Configuration Management Support into the Field, *University of Colorado*, http://www.stsc.hill.af.mil/crosstalk/1998/feb/deployment.asp. February 1998.
[14] P. Hellemans, H Vanderstraeten, F. Wellesplein, J.C. Yelmo, J. Villamor, P. Lago and G. Canal, TINA Service Architecture: From Specification to Implementation", *Proceedings TINA'97 Conference*, 1997, pp. 17-20.