

Service Delivery Platform for Convergence Service Creation and Management

Changwoo Yoon, Hyunwoo Lee

**Electronics & Telecommunications Research Institute, Daejeon, Korea*

cwyoon@etri.re.kr, hwlee@etri.re.kr

Abstract— Service Delivery Platform (SDP) is an enterprise application implementing Service Oriented Architecture (SOA) concepts. SOA is a component model relating services by a contract between services through well defined interface. The SDP, that is a commercial system of SOA concepts, has two main components: Common BUS delivering and routing web services and Enablers. Enablers are a collection of virtualized services. Service virtualization provides an architecture systematically managing services. In this paper, we discuss various aspects of SDP technologies and convergence service implementation using SDP.

Keywords— SDP, SOA, Enabler, Convergence Service

I. INTRODUCTION

Previously, when only wired and wireless telephones existed in the communication environment, network operators or service providers generated and provided services and the number of generated services was small. A collective system for executing a service, such as a service providing system or a service/user management system was constructed for the service whenever needed. [1]

In the current service environment in which circuit-based communication is changing to Internet-centric packet-based communication and wired and wireless communications are integrated, the number of services required to be developed and provided to users geometrically increases and the lifetime of services is shortened to require services to be developed within a short period. Furthermore, with the rapid development of information technology such as web technology, there is a need to integrate information, communication and broadcasting technologies to create new convergence services. Moreover, users want to receive services only for themselves and act not only as service consumers but also service prosumers with the spread of various personal devices. [1]

In the enterprise domain, the SOA (Service Oriented Architecture) has been used for integrating various kinds of business services. In SOA, systems are divided into basic service components. These services are distinct units that can be independently reusable. The Service Delivery Platform (SDP) is one of implementations of the SOA concept in an enterprise domain [2].

A service delivery platform (SDP) is a technique developed to meet this variation in communication and information technology environments, allows common carriers to rapidly create and deliver services to efficiently provide the services and assists third party service providers or personal information providers in efficiently participating in service business. [1,2]

Conventional techniques relating to the SDP mostly define the SDP as a set of enablers corresponding to abstract forms of physical devices of a network or the Internet. These conventional techniques provide general SDP structures in which service common functions such as an operation support system (OSS)/business support system (BSS) are connected with the enablers of the SDP to enable rapid service creation and third party service providers and users can abstract functions to create and provide services even if the third party service providers and the users are not network operators. [3, 4, 5]

This paper describes a service delivery platform structure and method for supporting circumstances in which users can be service providers as a service environment is personalized to meet a demand for service personalization and supporting a service structure in which third party service providers and users are included in a service business model.

II. KEY COMPONENTS OF SDP

A. Common BUS

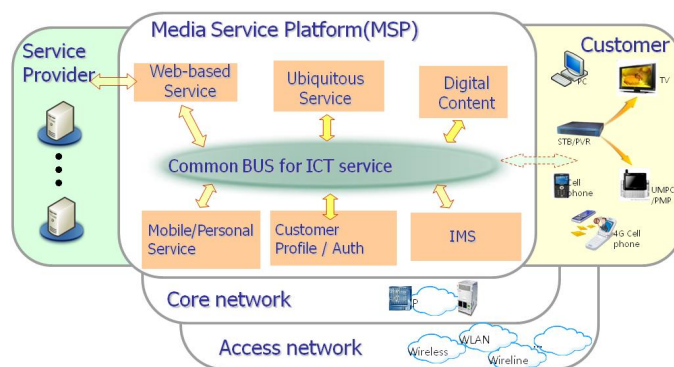


Figure 1. Conceptual Architecture of MSP

Figure 1 is a conceptual architecture of MSP (Media Service Platform) providing media related converged services using SDP. The MSP combines telecommunication services, IT

services, and broadcasting services to make new converged services. The concept of SDP consists of two parts: Common BUS and Enablers. The Common BUS is an execution environment of Web services and web service directories. The common BUS has a similar role with that of the Enterprise Service Bus (ESB). The ESB is a mediating hub at the time of Web Service coordination. It acts as a message broker between services. When an application tries to find Web Service in its routine, ESB finds the appropriate Web Service with application's context. We will explain about enablers on next section.

B. Enablers

The original concept of enablers came from Open Mobile Alliance (OMA). The enablers are the function that abstracts network functions and provides them to third party application developers through application programming interface (API). Enablers interact with each other and create other services. The OMA does not specify a protocol for the interfaces of enablers nor a specific mechanism for combining these enablers. [2]

Enabler is a building block that encapsulates reusable functionality.

We used Web service specification for the implementation of enablers. WSDL is used to describe the interface of services provided by enablers. SOAP protocol is used to communicate between services.

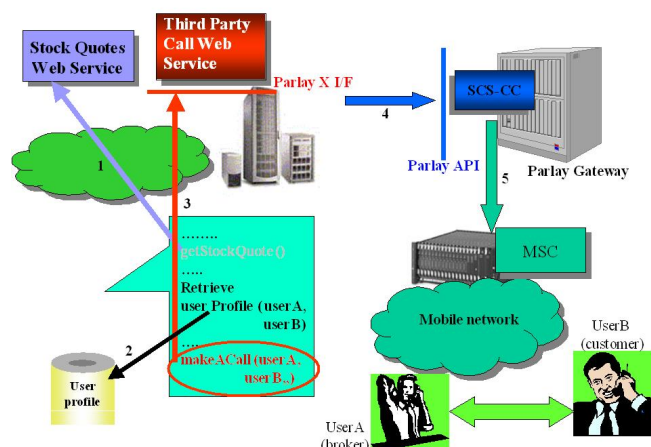


Figure 2. Parlay X TPC Scenario

In our implementation of telecommunication services enabler, we used Parlay X. Figure 2 shows a Third Party Call (TPC) scenario proposed by Parlay X specification. We implemented TPC and Call Notification (CN) enablers. The TPC and TN function is executed in IP Multimedia-subsystem (IMS) enablers. The IP Multimedia Subsystem (IMS) is an architectural framework for delivering Internet Protocol (IP) multimedia services. IMS enablers abstracts underlying telecommunication network capabilities.

C. Convergence Service Creation

Mashup and Business Process Execution Language (BPEL) are both service creation issue in convergence service. [5]

For the convergence of services, mashup is key issue in the Web 2.0 area. A mashup is originated from the web. It combines content from several sources into an integrated experience. [4] In telecommunication, Parlay/OSA and Parlay X is a technology implementing the mashup concept. But the Parlay/OSA is too complicated to be used because of its overly fine granularity. On the other hand, Parlay X isn't complicated enough to meet the needs of an IT service designer.

Service coordination such as a mashup is difficult. Even if the concept of SOA supports loosely coupled service integration, the programmer must do a fair amount of programming to make an executable application. BPEL is a specifying interaction with Web Services to make service coordination easily.

Recently, Representational state transfer (REST) based Really Simple Syndication (RSS)/Atom technology are commonly used because of its easy way of integrating into new services. REST based APIs can be a good alternative for SOAP based web services because of some difficulties in processing SOAP messages due to the verbose XML format. [10]

III. SERVICE VIRTUALIZATION OF SDP

Service virtualization is a recent trend of SOA focusing on providing common infrastructure to create and manage complex service eco-system. By using service virtualization, service developer can focus on service feature developing without worrying on how functions are provided, consumed, and managed.

Service virtualization does not modify general service code, rather makes virtual service providing the API functions before executing service. The key idea of service virtualization is service brokering residing between service client and service implementation.

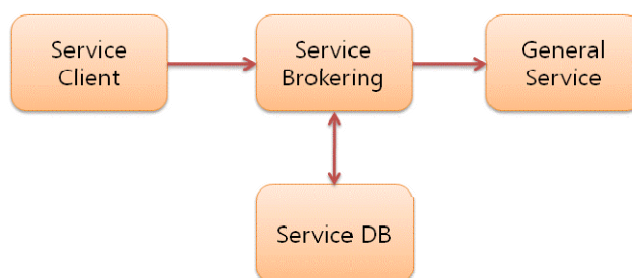


Figure 3. Concept of Service Virtualization

Service brokering is needed for a separation of client and general service. In this architecture, client cannot connect directly with general service. It can only communicate through brokering service. At the position of service brokering, virtualized service is hosted and exported to service user.

Service broker reads information for DB and decides how to host virtualized service. This approach exports several

virtualized service for one general service, we can adopt easily to various customer scenarios.

All communications are transmitted through service broker, we can provide various kinds of brokering services: version control, protocol transfer, monitoring, and real time policy decision. Service virtualization does not change client and general service code.

Service virtualization has several advantages. It provides independent ownership of SOA's each layer. For example, service provider develops service interface and logic. Operator describes endpoint policy and message standards. Architect professionals defines service level policy and execution pattern. Business owner describes business policy and requirement. All of above can cooperate at the centre points.

Eventually, service virtualization reduces time-to-market for a new service and provides more flexible and substantial methods for managing service environment.

IV. IMPLEMENTING CONVERGENCE SERVICE

Figure 4 shows a system configuration for SDP-based integration of telecom-services and web-based services.

We implemented the IMS enabler described in chapter 3. The IMS adapter in figure 3 converts SIP protocol in Parlay into SOAP protocol in Web service. Current IMS adapter has components of TPC and CN. The common profile adapter connects to the home subscriber server (HSS) database containing unified subscriber management data. The messenger adapter connects to the instant messenger (IM). Through this adapter we can use IM in the developed application

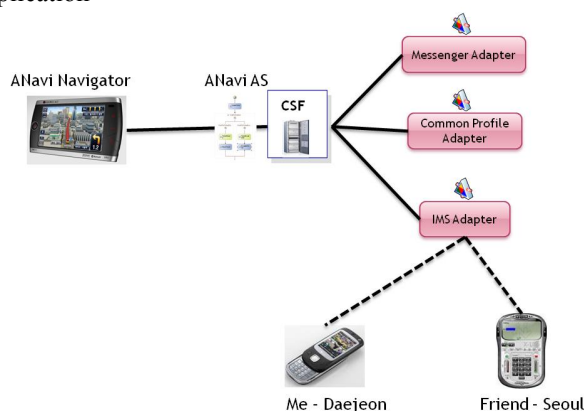


Figure 4. System Configuration

The developed demo application is advanced navigation (ANavi). The scenario of service is described below:

- Mr. A wants to visit friend B to Seoul from Daejeon.
- He opens instant messenger (IM adapter) on the ANavi screen, finds friend B, and gets B's phone number.
- When A selects B on the IM screen, ANavi asks the Common profile adapter about B's current location.

- At the same time, ANavi calls B using IMS adapter's TPC.
- When the appointment is made with B, ANavi starts navigation.
- When the car approaches to within 1 km of B, ANavi sets up a call with B again to prepare him.

The ANavi application is a IT-telecommunication convergence service integrating telephone calls, database and Instant messaging.

V. CONCLUSIONS

Service Delivery Platform (SDP) used for a creation and management of convergence service. Service virtualization is a key concept of SDP for a service convergence because it provides service feature independence among unit services (components) of convergence services.

Service provider can create service components using standard interface without worrying how to integrate each unit services.

This will brought a business market chance: creating various business model such as Prosumer, service provider, hosting provider, MVNO, and Telco.

We will describe more details of service virtualization such as service version control, policy control and event handling methods in full paper.

IPTV service is an example of convergence service between telecommunication and broadcasting services. SDP will have a essential role on convergence service creation and management because of its SOA concepts.

ACKNOWLEDGMENT

This work was supported by the IT R&D program of KCC/MKE/KEIT. [2009-S-018-01, Development of Open-IPTV Platform Technologies for IPTV Convergence Service and Content Sharing]

REFERENCES

- [1] Changwoo Yoon, Shinmo Kim, Hyunwoo Lee, "Convergence Service Implementation based on Service Delivery Platform and Research Issues," *International Technical Conference on Circuits/Systems, Computers and Communications*, pp. 1080-1083, July 2009.
- [2] H. Ohnishi, Y. Yamato, M. Kaneko, T. Moriya, M. Hirano, and H. Sunaga, "Service delivery platform for teleco-enterprise-Internet combined services", *IEEE GLOBECOM*, pp.108-112, 2007.
- [3] Sun Young Lee, Jong Yun Lee, and Byung Il Lee, "Service Composition Techniques Using Data Mining for Ubiquitous Computing Environments", *International Journal of Computer Science and Network Security*, Vol 6., No 9B, September 2006.
- [4] Nilanjan Banerjee, et. Al., "Telecom Telecom mashups: enabling web 2.0 for telecom services", *International Conference on Ubiquitous Information Management and Communication*, pp.146-150, 2008.
- [5] Moriya, T.; Ohnishi, H.; Yoshida, M.; Hirano, M., "Dataflow Generation for Service Composition to Incorporate Web and Telecommunication", *GLOBECOM 2007*, pp.26-30, Nov. 2007.
- [6] Stefan Arbanowski, Pieter Ballon, Klaus David, Olaf Droegehorn, Henk Eertink, Wolfgang Kellerer, Herma Van Kranenburg, Kimmo Raatikainen, and Radu Popescu-zeletin, "I-centric Communications: Personalization, Ambient Awareness, and Adaptability for Future Mobile Services," *IEEE Communications Magazine*, pp.63-69, September 2004.
- [7] Marc Steen, Ronald van Eijk, Nicole de Koning and Erik Reitsema, "A We-Centric Telecom Service for Police Officers to Support

- Spontaneous Communication,” *Lecture Notes in Business Information Processing*, Vol.12, pp.357-365, Oct. 2008.
- [8] S. Arbanowski, S. van der Meer, S. Steglich and R. Popescu-Zeletin, “The Human Communication Space: Towards I-centric Communications,” *Personal and Ubiquitous Computing*, pp.34-37, 2001.
- [9] Stephan Steglich , Raju N. Vaidya, Olga Gimpeliovskaja, Stefan Arbanowski, R. Popescu-Zeletin, Shigetoshi Sameshima and Katsumi Kawano, “I-centric services based on super distributed objects,” *Proceedings of the The Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03)*, pp.232-239, 2003.
- [10] Soonchul Jung Mi-Kyoung Kang Dae-Woo Choi, “Call/Messaging Open API for Telecommunication Services”, *10th International Conference on Advanced Communication Technology*, pp.1139-1143, Feb. 2008.