Presence-based Call Reservation Service using Open API on the Web-service architecture in Broadband Converged Network

Yoo-mi Park
BcN Service Research Group, ETRI
parkym@etri.re.kr

Young-il Choi
BcN Service Research Group, ETRI
yichoi@etri.re.kr

Sang-ha Kim
Chungnam National University
shkim@cnu.ac.kr

••

••

***Abstract* — Open API in telecommunication is a service control technologies and is a set of abstract standard operations which can provide capabilities to access telecom network for the 3rd party service provider in a secure manner. This paper describes presence-based call reservation service which is a case study of converged service with Open API on the Web-service architecture in Broadband converged Network. The presence-based call reservation service could provide the enterprise with combining capabilities of accessing network resources and manipulating enterprise data resources. We also propose a simple control mechanism for realization of this service with Open API on the Web-service architecture.**

••

***Keywords* — Open API, Open Service Access, BcN, BcN Service, NGN**

••

1. Introduction

Applications and services in telecom are a part of the network operator's domain and are usually built using Intelligent Network technology in the current network. This approach is excellent for simple telecom applications, but it requires highly experienced programmers and spends a couple of years to deploy of new services. With the appearance of mobility and IP, easy and fast creation of innovative applications that combine different features and critical enterprise data becomes a challenge. In order to meet this challenge in 3rd generation networks, the network architecture is based on horizontal layering principles where applications are logically found in the upper layer as shown in Figure 1, called application layer or Service Network [1-2].

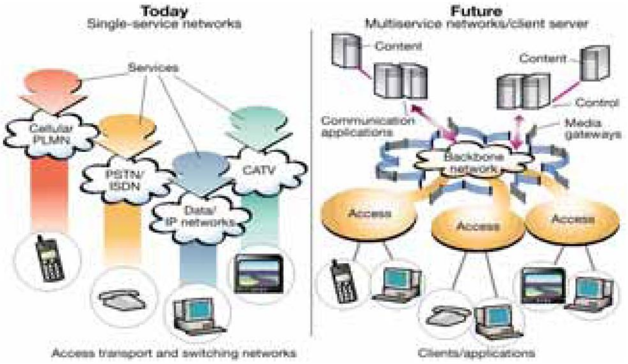


Figure 1. Evolution of Telecommunication Network

Service Network is used to distinguish it from the Core Network located in the lower layers. The Service Network is based on open distributed technology and applications are able to access core network functionality by means of open and standardized APIs. This is also referred to as opening up of the network and will make the telecom functionality accessible for a large developer community such as Internet Service Providers (ISP).

Open Service Architecture (OSA) is a typical Service Network for the next generation network. It has been a result of enormous effort in opening up and convergence of network functionality for application development. Applications will be able to access core network functionality using open standardized APIs. The goal of APIs is the applicability for different core network technologies. It means that applications can run any other core network once it would be written. Combined with the fact that applications can be built with standard IT technology new innovative applications could be hit the market with shorter time to market (TTM) [1].

Open API in telecom is a standardized API for Open Service Architecture. This paper describes a case study of service using Open API on the Web-service architecture, presence-based call reservation service, which the calling number is added to a call waiting list during a called party is busy. It shows how easy and how fast integrated telecom service with internet application could be developed using Open API on the fixed-mobile and IP converged Network.

This paper is composed as follows: Section 2 introduces the Open Service Architecture and Open API. Section 3 explains Web-service based Open API. In Section 4, we outline standardization of Open API. In Section 5, presence-based call reservation service based on Web-service architecture is described in detail. Finally, we will conclude this paper in Section 6.

2. Open Service Architecture and Open API

The traditional telecommunication environments have some limitations in the areas of service portability and fast service deployment. To overcome these problems in the traditional telecommunication environments, OSA is aimed to provide a set of open, standardised interface that enables applications implementing the services to make use of network functionality. The interface is the Open API that enables network operator and 3rd party service provider to make use of network functionality in a secure manner [2].

Utilizing network features and capabilities is necessary for both enterprise and consumer markets to enable the creation of vast numbers of new applications in and around the network and to enable the worldwide software development community to exploit the features of current and future networks. In other words, it allows operators to use 3rd party developers to develop new applications and services and enables operators to explore new business models for working with Independent Software Vendors (ISV) and enterprises as shown in Figure 2.

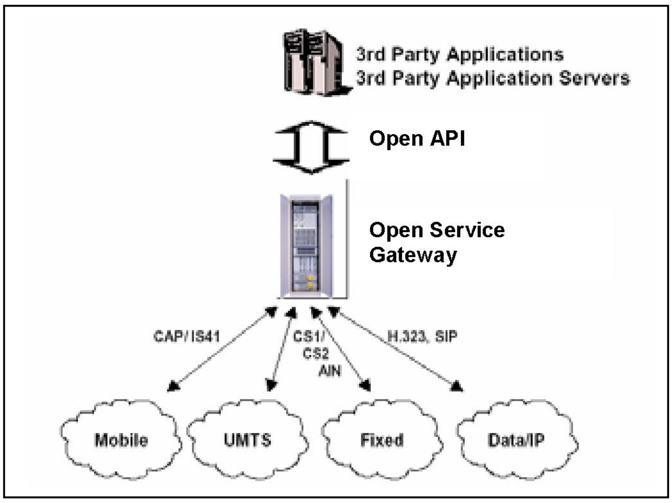


Figure 2. Open Service Architecture and Open API

In Figure 2, to realize the service using Open API, Open service gateway should be built in the operator's domain. Open service gateway is to transfer requests from applications invoking Open API to the underlying network by mapping the appropriate network protocols such as SIP for IP network.

3. Web-services based Open API

Open API is classified two control mechanisms by remote object access technology: CORBA-based API and Web-services based API. The former is designed to enable creation of telephony applications as well as to "telecom-enable" IT applications. IT developers, who develop and deploy applications outside the traditional telecommunications network, are viewed as crucial for creating a dramatic market growth in next generation applications, services and networks. The latter is intended to stimulate the development of next generation network applications by IT developers who are not necessarily experts in telephony or telecommunications [3-4].

Web-services based Open API is utilized under the Web-service architecture as an implementation of Service Oriented Architecture (SOA) that has a triangle architecture consisting of service provider, service requester, and service broker. To exploit them, open service gateway plays a role of service provider and application invoking Open API acts as a service requester as show in Figure 3.

Web-services based Open APIs are powerful yet simple, highly abstracted, imaginative, building blocks of telecommunications capabilities that developers and the IT community can both quickly comprehend and use to generate new and innovative applications. This empowers service providers, independent software vendors and other developers in the IT and telecommunication industries to generate a new range of applications that benefit from functionality resident in public or private communications networks.

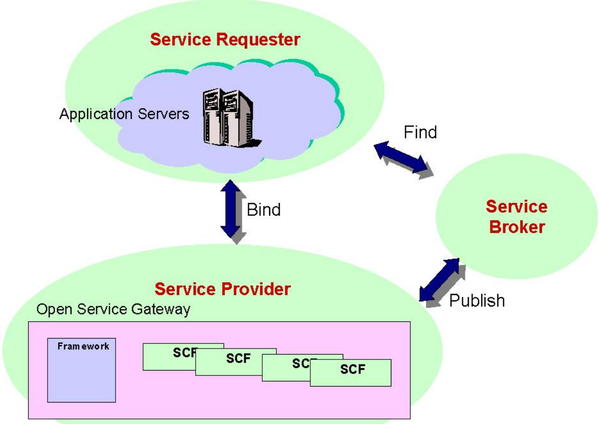
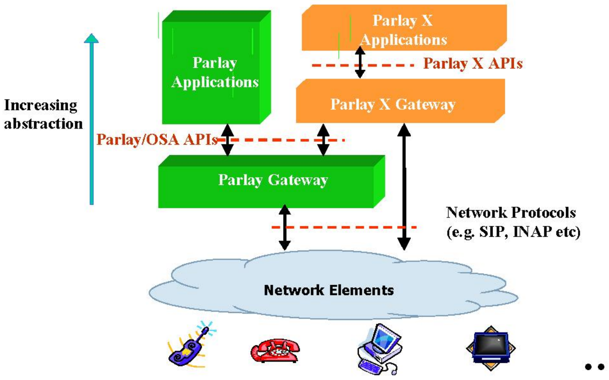


Figure 3. Web-service architecture for Open API

4. Standardization

Parlay/OSA and Parlay X API are typical Open APIs in telecom standardized by Parlay group, 3GPP, and ETSI. Parlay/OSA APIs are set of highly functional APIs defined in UML for professional software developers, who are programming in C++ or Java. While Parlay X APIs are high level, simple, and abstract APIs for web developers and users of scripting languages and vendors who want to use Web-services technologies [2-4].



• •

Figure 4. Relationship between Parlay X and Parlay/OSA APIs

• •

Figure 4 shows the relationship between Parlay X Web-services and Parlay/OSA APIs. Application can interact with the Parlay gateway over a network; using CORBA or Web-services transports. The application itself can, in principle, be written in any language as corresponding responses for the API version that the Parlay/OSA or Parlay X gateway offer. Recently, Parlay/OSA API v5.0 and Parlay X

2.0 (3GPP Rel. 6, ETSI Rel. 2) was published as international standard specifications [4]. Table 1 shows the summary of Parlay X 2.0 APIs and some of them were used for implementation of presence-based call reservation service.

5. Presence-based call reservation service

The presence-based call reservation service, which the calling number is added to a call waiting list during a called party is busy. This service could provide the enterprise with combining capabilities of accessing network resources and manipulating enterprise data resources.

Figure 4 shows an example scenario of presence-based call reservation service. Suppose that the service application would be deployed in the application server on the enterprise network and Open service gateway would be deployed in the telecom operator's network. And it is necessary that the presence of the called party's terminal will be notified to the service application which makes a call automatically when the called party's terminal is available (if not busy).

The presence-based call reservation service can be practically applied for such as order-taking, complain handling, and counseling by telemarketers. If all telemarketers are busy, a call from the new customer can't be connected to any telemarketer. The service application makes a reservation for customer's call during all lines are busy. The application can get a customer's phone number to add it to waiting call lists and request to the Open API service gateway for playing announcement to the customer. After the customer can listen to announcement to inform that his call is made a reservation, the customer will wait for telemarketer's call. Then the application makes a call automatically when the telemarketer's line is available. If the customer couldn't get the announcement, he or she would have tried to call again until call connection to the telemarketer even though his or her call is made a reservation. This service with announcement makes sure that unnecessary calls would be decreased on the telecommunication network.

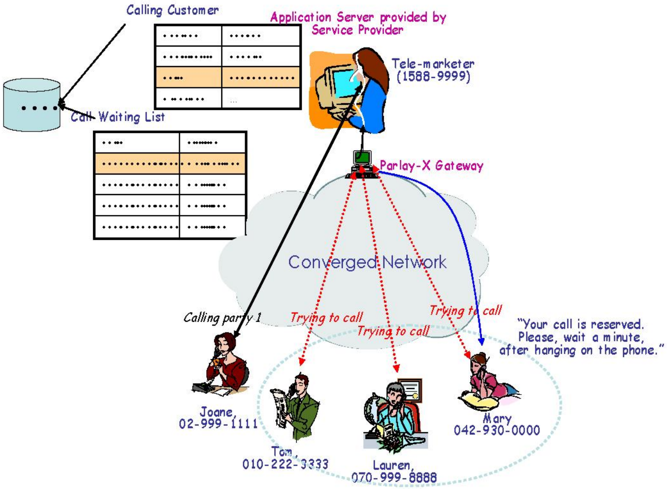


Figure 5. Service Architecture of Presence-based call reservation service

Table 1. Summary of Parlay X APIs [4]

SCF (Service Capability Feature)	APIs
Third Party Call	MakeCall(), GetCallInformation(), EndCall(), CancelCall()
Call Notification	HandleBusy(), HandleNotReachable(), HandleNoAnswer(), HandleCalledNumber(), NotifyBusy(), NotifyNotReachable(), NotifyNoAnswer(), NotifyCalledNumber()
Short Messaging	SendSms(), SendSmsLogo(), SendSmsRingtone(), GetDeliveryStatus(), NotifySmsReception(), GetReceivedSms()
Multimedia Messaging	SendMessage(), GetMessageDeliveryStatus(), GetReceivedMessages(), GetMessageURIs(), GetMessage(), NotifyMessageReception()
Payment	ChargeAmount(), RefundAmount(), ChargeVolume(), GetAmount(), RefundVolume(), ReserveAmount(), ReserveAdditionalAmount(), ChargeReservation(), ReleaseReservation(), GetAmount(), ReserveVolume(), ReserveAdditionalVolume(), ChargeReservation(), ReleaseReservation()
Account management	GetBalance(), GetCreditExpiryData(), BalanceUpdate(), VoucherUpdate(), GetHistory()
Terminal Status	getStatus(), getStatusForGroup(), startNotification(), endNotification(), statusNotification(), statusError(), statusEnd()
Terminal Location	getLocation(), getTerminalDistance(), getLocationForGroup(), startGeographicalNotification(), startperiodicNotification(), endNotification(), locationNotification(), locationError(), locationEnd()
Call Handling	SetRules(), SerRulesForGroup(), GetRules(), ClearRules()
Audio Call	playTextMessage(), playAudioMessage(), playVoiceXmlMessage(), getMessageStatus(), endMessage()
Multimedia Conference	createConference(), getConferenceInfo(), endConference(), inviteParticipant(), disconnectParticipant(), getParticipantInfo(), getParticipant(), addMediaForParticipant(), deleteMediaForParticipant()
Address List Management	createGroup(), deleteGroup(), queryGroups(), setAccess(), queryAccess(), addMember(), addMembers(), deleteMember(), deleteMembers(), queryMembers(), addGroupAttribute(), deleteGroupAttribute(), queryGroupAttributes(), addGroupMemberAttribute(), deleteGroupMemberAttribute(), queryGroupMemberAttributes(), addMemberAttribute(), queryMemberAttributes(), deleteMemberAttribute(),
Presence	subscribePresence(), getUserPresence(), startPresenceNotification(), endPresenceNotification(), statusChanged(), statusEnd(), notifySubscription(), subscriptionEnded(), publish(), getOpenSubscritptions(), updateSubscriptionAuthorization(), getMyWatchers(), getSubscribedAttributes(), blockSubscription

If the customer couldn't get the announcement, he or she would have tried to call again until call connection to the telemarketer even though his or her call is made a reservation. This service with announcement makes sure that unnecessary calls would be decreased on the telecommunication network.

Figure 6 shows a call flow example of this scenario in the IMS network. Some network servers are omitted to simplify a diagram. We assume that there is only one telemarketer for the service and he or she has a presence-enabled phone to notify its presence to the application. If the customer2 tries to call to a telemarketer during the telemarketer is talking to customer1, the customer2 can listen to the announcement instead of conventional busy tone. The announcement can inform the customer2 that his or her call would be made a reservation. Otherwise, customer2 would try to call repeatedly because he or she doesn't know that his or her call is made a reservation in waiting call lists. After the application is notified that telemarketer's phone is available from the underlying network, it makes a call automatically between telemarketer and the first waiting customer, customer2 in this case, at that time. The application is a web application that has been used for call management including call reservation and third party call attempt. It requests to access network resource using Open API through Open service gateway. Open service gateway analyzes the requests received from application, converts the request to proper network protocols dependent on underlying networks, and forwards it to appropriate network element with their network protocol [6].

The service was designed and implemented by using Parlay X APIs 2.0 in Table 1 on Web-service architecture. It has exploited Parlay X Presence, Call Notification, TPC, SMS, and Audio call web services [4] which were invoked in the application or Open service gateway to handle network initiated call, presence of terminal, and to make a call, and playing announcement as shown in Figure 6.

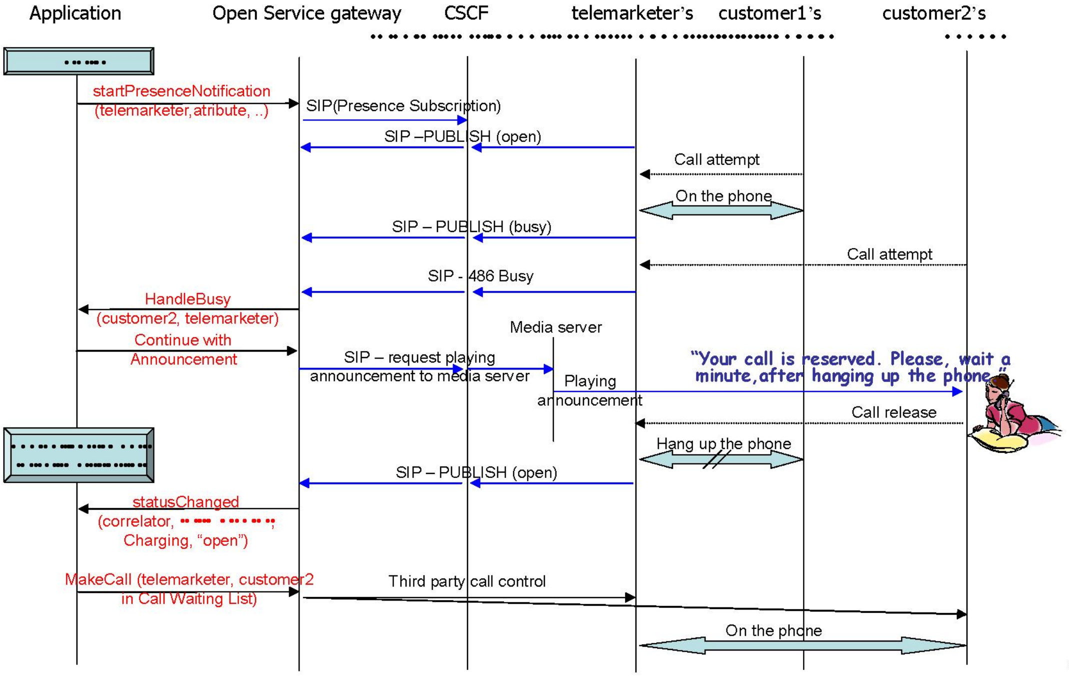


Figure 6. Call flow example of presence-based call reservation service

4. Conclusions

Open API is an emerging technology standards that facilitate the convergence of the IT and Telecom communities. Telecom operators can disclose their network resources such as call handling and terminal status to 3rd parties using Open API, and IT application developer can develop new value-added service to be combined with internet resources.

This paper described an introduction of Open API and service example using it on the converged network. It could be possible for IT application programmers to implement easily new creative services on the fixed-mobile and IT converged network. Open API could be expected to be a new paradigm of the value-added service development in the next generation network. Implementing Open API could provide network operators a migration path to future networks independent of network technology and allows operators to deploy new and innovative services quickly and cheaply.

References

[1] Ard-Jan Moerdijk and Lucas Klostermann, "OPEN SERVICE ARCHITECTURE : CONCEPTS AND STANDARDS," Computer Communications Journal for the special issue on IN and 3G Mobile. 2001

[2] Zygmunt Lozinski, • Parlay/OSA - a New Way to Create Wireless Services," IEC Mobile Wireless Data, pp.61-70. Jun. 2003.

[3] Zygmunt Lozinski, • Parlay Web Services,• •12th Parlay Meeting, May 2004.

[4] 3GPP TS 29.199-mn V6.2.0, "3GPP: Technical Specification Group Core Network and Terminals; Open Service Access (OSA); Parlay X Web Services; (Release 6)," June 2005.

[5] Roch H. Glitho, • Developing Applications for Internet Telephony: A Case Study on the Use of Parlay Call Control APIs in SIP Networks,• • IEEE Network, pp. 58 ~64, May/June 2004.