A TINA-Based Component Modeling for Static Service Composition

Young-Seok Shin, *Member, KIMICS* and **Sun-Hwan Lim,** *Nonmember*

Abstract—This paper describes a modeling of service composition manager based on TINA (Telecommunication Information Networking Architecture). The Service composition function is mainly motivated by the desire to easily generate new service using existing services from retailers or 3rd-party service providers. The TINA-C specification for the service composition does not include the detailed composition procedure and its object models. In this paper, we propose a model of components for the service composition, which adapts a static composition feature in a single provider domain.

To validate the proposed modeling, we implemented prototype service composition function, which combines two multimedia services; a VOD service and a VCS service. As a result, we obtain the specification of the detailed composition architecture between a retailer domain and a 3rd-party service provider domain.

Index Terms—TINA, Service Composition, Open Networking, Service Session Control.

I. INTRODUCTION

In recent years, users of Internet have explosively increased to obtain useful information from a great number of web services. As the usage of modern networks based on computer communication has been greatly activated, user' demands for easy access on functions of various services have also been increasing. For that reason, service providers, who provide distributive multimedia services through networks, continue to develop new services or extend the capability of existing services to meet the various demands of users. But these efforts and resulting products from service providers have not been reaching the users' expectation. Service providers should support expensive payment to develop new services and these services have a short period on the service life cycles. Practically, one of the main reasons is that services developed by different vendors have their own features. A service from a vendor is developed without the consideration of composing with services from other vendors. In other words, these specialized services cannot be composed with others to create new enhanced services with extended features. To realize the composition of services, a generic and common architecture is required for all services.

In TINA-C (Telecommunications Information Networking Architecture-Consortium), actions to define a generic and common service architecture model, which can be applied to various services, have been proceeded. The service architecture currently suggested by TINA-C is to have features of reusability, portability and scalability of service components [1, 2, 4]. Another important feature of this model is that services can be designed and implemented independently from network infrastructures, operating systems or developing languages. In addition to the basic features, this service architecture defines requirements, concepts and functions of service composition for more complicated services adapting object-oriented concept and technology of distributed networking system. But concepts and functions in the architecture are not sufficient to be directly applied in developing new services since the current draft has not been described the specifications on the interface and interactions between service components in detail.

In this paper, we propose a static service composition model, which enables the functional extension of existing services or the creation of new services by combining more than one service. The static proposed service composition model is based on the service architecture of the TINA-C. We prototype a service composition function to validate a proposed model using two multimedia services, such as Video Conference Service (VCS) and VOD (Video On Demand). It is designed by extending TINA information model and computational model. Composition related objects, relations and interfaces between objects are defined in the information model and the computational model.

This paper is organized as follows: Section 2 describes the concept and the scope of service composition. Section 3 and 4 shows the modeling of service composition and the designed service composition architecture, Section 5 shows the environments for prototyping and the validation of proposed composition model. Finally, section 6 describers the conclusion.

II. SCOPE OF SERVICE COMPOSITION

The service composition is a creation of a new service or a service instance by composing services or service components from different services. Many types of composition are possible. Several reference points (RP) for the Service composition can be established between

Manuscript received January 6, 2004.

Y.S. Shin is with Honam University, 59-1 Sebong-Dong, Kwangsan-Gu, Gwangju, 506-714, Korea. He is now with the Dept. of Inform. & Comm. Eng.,(Tel: +82-62-940-5514, E-mail: ysshin@honam,ac,kr)

S.H. Lim is currently a senior member of engineering staff at the Electronics and Telecommunications Research Institute (ETRI) in Korea.(Tel: +82-42-860-1226, E-mail: shlim@etri.re.kr)

two different parties defined in TINA specification. The two parties are Consumers (Users) and Retailers/service providers [1, 3, 4]. Figure 1 shows the scope of the retailer-RP (Ret-RP) and third-party-RP (3Pty-RP) depending on a set of business roles and relationship, defined in [3]. The retailer-RP (Ret-RP) is used as an interface of the consumer business role and that of the retailer business role. It is used to support the consumer's needs for access to services offered by the retailer.

Either a consumer or a retailer can request a service composition in a TINA system. In the service composition, a retailer plays a key role, since it arranges the seamless integration of services and includes various management features in a consistent manner.

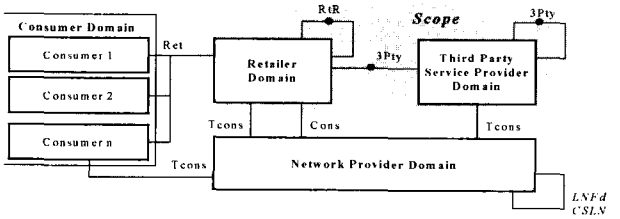


Fig. 1 Scope of the RtR-RP for Service Composition.

A dynamic composition is defined as the on-line construction of a service composition. In other words, a new combined service can be created through the dynamic service composition while the users are on the network waiting for the new service. In a dynamic service composition, the services to be composed are not determined in advance. Some components for the dynamic composition can be implemented to reduce the composition time before a composition process occurs. In a dynamic mechanism, a consumer can request a service composition and the system with the composition request should dynamically create a new service by generating a set of components without consuming too much time for the composition.

While the dynamic mechanism is free to combine any TINA services to create a new service at the run time, a static mechanism is confined to user services which are predetermined in advance. Therefore, the scope of the service composition in a static mechanism should be determined considering a set of requirements from general customers before the service composition function is actually used. As an example of the static service composition, Figure 2 shows a Joint Document Editing (JDE) service. This example is to compose an Shade Document Editor (SDE) service and a Video Conference (VCF) service [1, 4].

In the example, the predetermined components are generated during the composition (e.g. SSM, UAP, etc.) and there are interactions between those components of the JDE, SDE, and VCF services. Several components can be newly generated (e.g. SSM_J, of composed service) or modified (e.g. SSM_V and SSM_S of existing services)

The final goal of this resource is to have a capability of combining any TINA services in various levels, we have a step by step strategy to achieve this goal. As the first level of the composition, in this paper, we propose a component model for the static service composition in a singe provider domain.

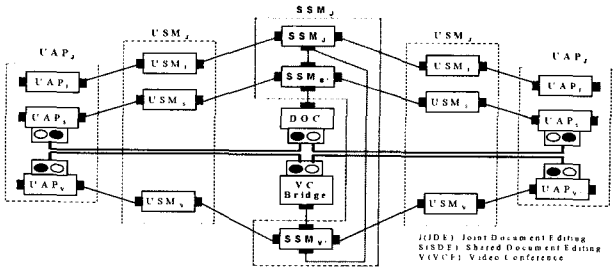


Fig. 2 Example of Static Service Composition.

III. COMPOSITION ARCHITECTURE

In TINA specification[1,3,4], the service composition is to create a new service or a service instance by combining two or more services or service components. A service composition may take place in a single provider domain, between a retailer and a third party service provider, and between a third party service provider and another third party service provider [3]. New services are being created through composing features of many existing services.

To realize the service composition technique, firstly a session model defining abstract service architecture of composition should be designed. And a detailed specification for the composition technique, composed of information model and composition model, should also be defined. An information model defines the functions of information objects and relationships between them, based on the session model. And the computational model defines computational components and interface between them. As a result of these reasons, we should define new objects and components for service composition using information and computational model.

The specification of the TINA service composition framework is general to make it adaptable for various circumstances and environments. It is based on a set of generic paradigms that can be specialized or extended for specific services.

The designed composition architecture in this paper is mainly focused on retailer and third-party service provider domains since a user domain can be very application specific. Figure 3-1, and 3-2 illustrate the designed static service composition architecture which is applied to the video presentation service (VPS) using the direct interface mechanism and the indirect interface mechanism.

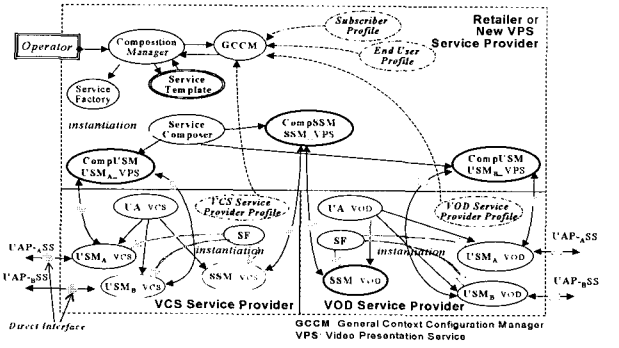


Fig. 3-1 Relationship of Service Composition Objects using Direct Interface Mechanism.

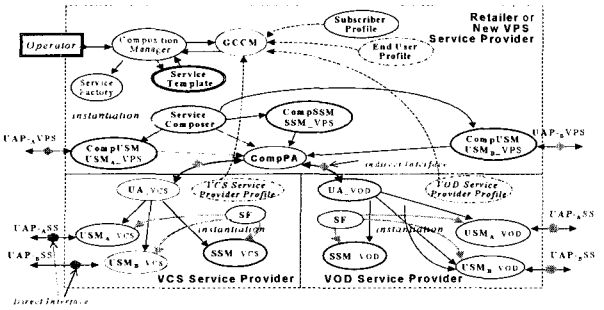


Fig. 3-2 Relationship of Service Composition Objects using Indirect Interface Mechanism.

It is worth to note that the indirect interface mechanism has the advantage of separating different providers, and a consistent interface with in the TINA specification. We model the composition components using the indirect interface mechanism, since it has the extendible provider domains and effective interfaces, and it is easy to implement the static composition.

IV. COMPOSION MODELING

A. Composition Objects

Fig. 3-2 shows the composition objects on the service composition architecture. Through this architecture model based on the TINA specification [1], the domain access session can act as a user, as a provider or as a peer function (User + Provider) depending on the access role in TINA specification. If the role is a user, it requests for service functions to a provider. Each service session may perform a service composition process with the request issued by access related sessions. On the composition step, a service type manager object determines whether or not the composition is possible. And it also provides the own attributes of services to the access session. The composition description object manages the information of the newly composed session. Table I and Table II describers information objects that are related to the service composition. A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

That construct a service are classified into access session objects and service session objects. The functions of the objects are as follows. For the detailed functions of the computational objects, refer to the reference [1], [3], and [4].

Table 1. Service Type Manager

Information	Content	Descriptions
Svc_Name	String	Identification of Service
Svc_Provider	String	Identification of Service Provider
IR_of_SF	String	Interface Reference of Service Factory Component
Max_Usr_Num	Integer	Maximum Number of User
Comp_Svc_Lst	List	List of Svc_Name
Stream_Svc	Bool	Requirement of Stream Service

Table 2. Composition Description.

Information	Content	Descriptions
Relation	Enum	Relation between Initiator and Initiated
Initiator	String	Composition Requester
Initiated	String	Composition Requestee
Started_Time	Time	Started Time of Composition

A.1 Access Session Objects

- Object: UAP-AS (User APplication-Access Session), PA, UA (including CompUA), IA, STM, CompA
- Establish access relation between domains
- Deliver users' service requests to the provider domain

A.2 Service Session Objects

- Object: UAP-SS (User APplication-Service Session, SF, USM (including CompUSM), SSM
- Establish service session as response of users' services requests
- Provide services to users

Among these computational objects, composition related objects are STM (Service Type Manager), SF (Service Factory), CompA, IA (Initial Agent), CompUA (Composition User Agent), CompUSM (Composition User service Session Manager) and SSM (Service Session Manager, CompSSM). These objects have to provide additional interfaces for service composition as follows.

A.3 STM

The STM contains and manages information about specific services, and the interface that should be added to provide composition related information is as follows.

```
Interface STM {
        . . .       //Basic Operations
        Request_Composible_Svcs (svcs);
};
```

A.4 SF

The SF, which creates and manages service session objects (USM and SSM) has to provide following interface to initiate or terminate CompUSM for the service composition.

```
Interface SF {
        . . .       //Basic Operations
        Create_Composition_USM
(IR_of_Other_Domain);
        Delete_Composition_USM ( );
};
```

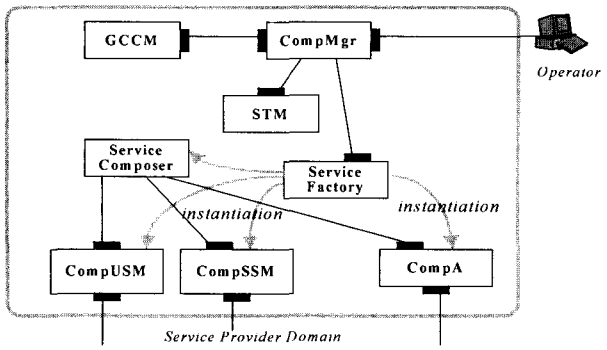


Fig. 4 Service Composition Objects.

A.5 CompA

The CompA is an additional object solely for service composition capabilities. It has the same capability with that of PA except that it has following additional interface for requesting composition and decomposition.

```
Interface CompA {
        . . .       //Basic Operations of PA
        Compose_Svc ( );
        Decompose_Svc ( );
```

A.6 CompUA

The CompUA is a component that inherits from UA. And it issues a request for the establishment of a service session and controls that session. So it should provide following interface to deliver an initialization request for composition session.

```
Interface UA {
        . . .       //Basic Operations of UA
        Create_Comp_Ses ( );
};
```

A.7 CompUSM

The CompUSM inherits from USM that manages the service session on user domain. And it should provide following additional interfaces to deliver composition-related controls to the service sessions on the provider domains.

```
Interface USM {
        ....    //Basic Operations of USM
        Send_Comp_Controls (control);
        Compose (IR_of_Other_USM);
        Decompose ( );
};
```

B. Composition Scenario

This section describes the steps of the service composition using a possible scenario. We show the information flow for the designed components of the static service composition using the event trace diagram as shown in Figure 5. The interactions between components are described in detail as below.

B.1 Precondition

The operator of a retailer domain requests service composition operation using CompMgr. As a result of composition, a new service, e.g., video presentation service (VPS), is composed two services: with a VCS and a VOD services. All service composition objects reside in the retailer domain.

B.2 Scenario

1. *An operator requests a new service creation to Service Composition Manager (SCM).*
2. *The GCCM (General Context Configuration Manager) gets the service profiles of VCS and VOD.*
3. *The SCM requests for the temptation of the new requested service (VPS) to STM using service profiles of VCS and VOD and STM returns object references.*
4. *The SCM requests for invocation of the Service Composer (SC) to SF.*

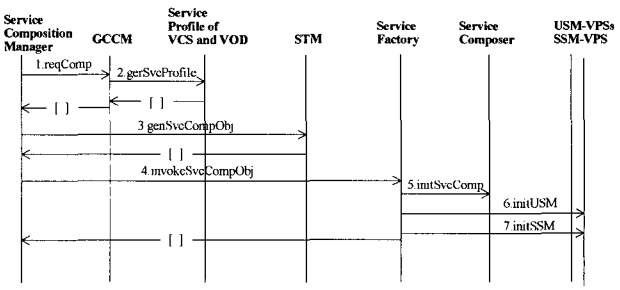
5-7. *The SF creates service composer, CompUSM (USM-VPS) and CompSSM (SSM-VPS) and SF returns interface references of SC, CompUSM and CompSSM.*

8-9. *The SC tests status of CompUSM and CompSSM, and manages session status.*

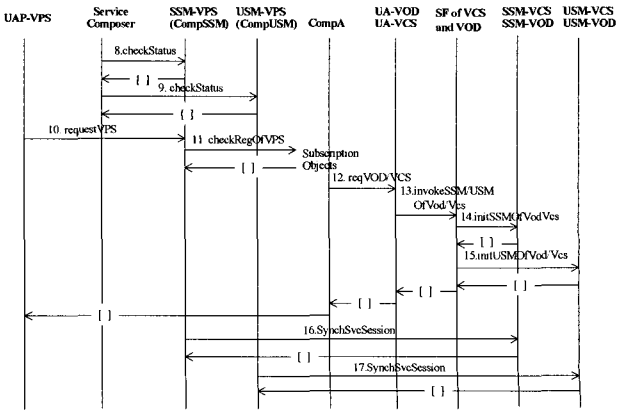
10. *A user requests a new service session of VPS through UAP-VPS.*
11. *The SC checks registration of VPS to subscriber objects.*
12. *The CompA requests a new service session of VPS to VOD provider and VCS provider.*

13-15. *In the provider domain, the SF creates USMs and a SSM for a new service session and returns interface references.*

16-17. *The CompSSM synchronizes the CompUSM to support interactions between the service session managers in service provider domain.*



(a) Access Scenario of Service Composition Manager Objects



(b) Access Scenario between Retailer and Service Provider

Fig. 5 Typical Access Scenario at RtR-RP.

V. IMPLEMENTATION OF THE PROTOTYPE FUNCTION

A. Environments

The designed service composition model was implemented and tested on an ATM network. The ATM network platform consists of three switching nodes such as the ASX-200 Series switches (product by FORE Ltd.). For the VCS and the VOD services, there are two types of the connections: DPE connections for operation interface between stakeholders and stream interface connections between end users. As the user terminals, we used IBM Pentium PC on windows NT with a video codec card (Osprey-1000) and a Fore ATM adapter card. At the provider site, we developed the service components and the connection management components on a SUN and a DEC workstation using IONA orbix2.3 MT which is an OMG CORBA compliance.

The stream interface between end-to-end users was realized using the multicasting protocol on the ATM service platform as in Figure 6 and the prototyping of stream interface connectivity was designed following the connection management architecture on TINA-C.

The connection management architecture for stream interface consists of CSM (Communication Session Manager), CC (Connection Coordinator), LNC (Layer Network Coordinator), NMP-CP (Network Management Layer Connection Performer), EML-CPs (Element Management Layer Connection Performers) and SA (Switch Agent) objects on a single network provider domain. We used the SNMP protocol between EML-CP and ATM switches using an switch agent object.

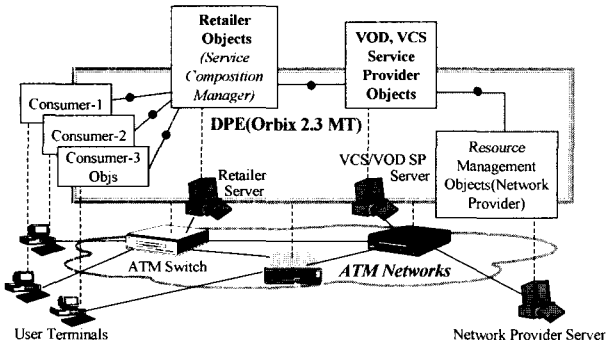


Fig. 6 Prototype Configuration for Service Composition Test.

B. Testing

In this test we use the VCS as the first (initializing) service, and the VOD as the second service. For the service composition – new service creation, an operator on the retailer domain runs the service composition manager as shown in Figure 7. During the composition, components of the new service should be generated in the retailer domain. The newly generated components include CompUSM, SSM, etc. If the VCS and the VOD were composed into a single composite service successfully, customers on the network can be serviced with a VPS as a new service. As a result of the service composition, the VCS and the VOD are composed into a single composite service successfully as shown in Figure 8.

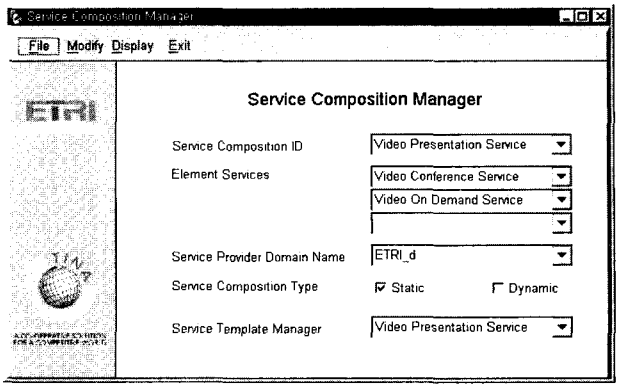


Fig. 7 Operation Interface of Service Composition Manager.

VI. CONCLUSION

Modern multimedia services to adapt client-server architecture become more sophisticated and complicated to provide newly introduced features for users. To efficiently cope with such situations, standard-making efforts for multimedia services are currently under way. Service architectures from the efforts should include the basic features of reusability, portability as well as the service composition capability or the object-oriented design on distributed networks. In order to be capable of combining several services from different vendors into a new service, service composition techniques should soon be realized.

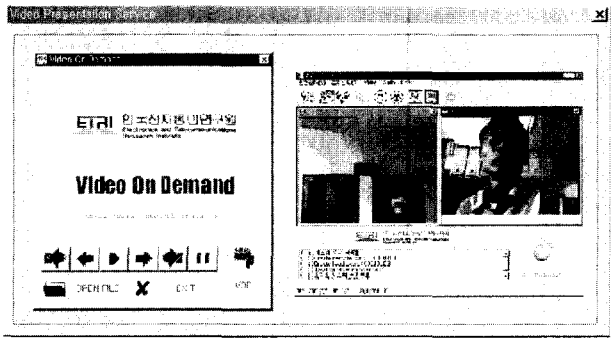


Fig. 8 The Example of User Interface for VOD and VCS Service.

In the paper, we proposed a service composition model and designed a composition architecture based on the service architecture of the TINA-C. The designed model is able to extend functions of a service by combining functions from different services. It can also provide a feature of creating new services with two or more existing services. For this composition capability, information and computational models from the TINA architecture are defined in terms of service components or objects, and relations or interfaces between them.

The validation of the designed service composition model using two multimedia services is proceeded. And we proved the designed model by combining the two multimedia services into a composite service. We obtain the specification of the detailed composition architecture between a retailer domain and a third-party service provider domain.

REFERENCES

[1] C. Abarce, P. Farley, et al, "Service Architecture Version 5.0", *TINA-C Documentation*, June 1997.

[2] C. Abarce, P. Farley, et al, "Service Component Specification Computational Model and Dynamics", *TINA-C Documentation*, Jan 1998.

[3] Martin Yates, Wataru Takita, et al, "TINA Business Model and Reference Points (Version 4.0)", *TINA-C Documentation*, May 1997.

[4] Mark Bagley, Raul Gutierrez and Hidetsugu Kobayashi, "Service Composition", *TINA-C Documentation*, July 1995.

[5] C. Abarce, P. Farley, et al, "Service Architecture Version 5.0", *TINA-C Documentation*, June 1997.

[6] C. Abarce, P. Farley, et al, "Service Component Specification Computational Model and Dynamics", *TINA-C Documentation*, Jan 1998.

[7] Martin Yates, Wataru Takita, et al, "TINA Business Model and Reference Points (Version 4.0)", *TINA-C Documentation*, May 1997.

[8] Mark Bagley, Raul Gutierrez and Hidetsugu Kobayashi, "Service Composition", *TINA-C Documentation*, July 1995.

[9] Chelo Abarca and Marcel Mampaey, "Specification of a Videoconference Service for Validation", *TINA-C Documentation*, April 1996.

[10] Jogensen Mikael, et al, "Multimedia Support in the TINA Architecture", *Proc. of TINA'96*, Sep 1996.

[11] A Limongiello, "ORCHSTRA: An Experimental Software Architecture to Support Multimedia Services", *Proc. of TINA'96*, Sep 1996.

[12] Patrick Hellemans, Juan C. Yelmo, et el, "TINA Service Architecture: From Specification to Implementation", *Proc. of TINA'97*, Nov 1997.

[13] Patrick Farly and Richard Westerga, "The Ret Reference Point Definition and a Prototype Implementation", *Proc. of TINA'97*, Nov 1997.

[14] S.H. Lee, K.H. Lee, et al, "Design of a Service Composition Based on TINA Service Platform", *Proc. of CSTN'98 (Korea)*, Nov 1998.

[15] Y.S. Shin, H.J. Oh and B.D. Ko, "The Design of Video Conference Service Based on Open Networking Architecture", *Proc. of ICOIN-12*, July 1997.

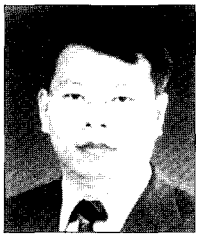
[16] S.B. Lee, D.S. Park, Y.S. Shin , et al, "Design of a Composition Architecture based on TINA", *Proc. of ICOIN-13*, Jan 1999.

[17] Tuncay Saydam, Xavier Logean and Simon Znaty, "A Service Management Architecture", *Proc. of ICT'97*, April 1997.

[18] A. Campos Flores, Ch. Lecluse and T. Landegem, "Prototyping TINA based services – the ALCIN project", *Alcatel Telecom. Review*, 1st Quarter, 1996.

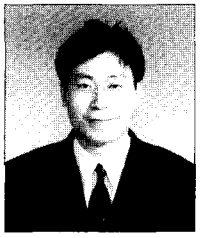
[19] Stephan Paschke, "Management of the BERKOM Multimedia Collaboration Service", *Proc. of DSOM'94*, 1994.

[20] David G. Boyer and Michael E. Lukacs, "The Personal Presence System: a Wide-area network resource for the real-time composition of multipoint multimedia communications", *Springer-Verlag, Multimedia Systems*, pp 122-130, 1996.



Young-Seok Shin

Received his B.E., M.S. and Ph. D. degrees in Electronic Engineering from Chon Buk National University in 1982, 1984 and 1993, respectively. From 1984 to 1998, he joined at ETRI, where he worked as Senior Member of Technical Staff of Broadband ISDN session. In 1998, he joined the department of Information and Communication Engineering, Honam University, Korea, where he is presently a associate professor. His research interest is in the area of Policy-Based Network Management, High-Speed Protocol, Open Network Signaling and Wireless Sensor Network.



Sun-Hwan Lim

Received his B.E. and M.S. degrees in Electronic Engineering from Chon Buk National University in 1997 and 1999, respectively. He is currently a senior member of engineering staff at the Electronics and Telecommunications Research Institute (ETRI) in Korea. His research interests include parlay/osa api, parlay x api, intelligent networks, CAMEL, and OSS/BSS.