# Service Delivery Platform Architecture for the Next-Generation Network

Hiroshi Sunaga, Yoji Yamato, Hiroyuki Ohnishi, Masashi Kaneko,
Masami Iio, and Miki Hirano

*NTT Network Service Systems Laboratories, NTT Corporation*
*3-9-11 Midori-cho, Musashino-shi, Tokyo 180-8585, Japan*
*sunaga.h@lab.ntt.co.jp*

## Abstract

This paper describes an advanced service delivery platform (SDP) that is expected to be used in the Next Generation Network (NGN). To meet diverse demands from NGN customers, it is very effective to integrate state of the art technologies beyond the boundaries of technical domains, such as the telecommunication, enterprise, and Internet fields, to enrich services and mash-up simple service elements into unlooked-for values. We use the service oriented architecture (SOA) concept as the basis and extend it to ease the creation of customised or personalised services. Context-aware dynamic binding of service components, finer-grained SIP call handling, and execution control for guaranteeing the service level agreement are main features of our platform. This is expected to play an important role in the NGN or ubiquitous communication environments.

## 1. Introduction

In the years to come, integrating state of the art technologies beyond the boundaries of technical domains, such as the telecommunication, enterprise, and Internet fields, will become more important. From telecom carrier viewpoints, coordination with Web and Internet technologies will become very effective for enriching services in the next generation network (NGN). To integrate these domains, the service oriented architecture (SOA), originally developed for enterprise networks, can be a basis.

We use this concept as the basis and extend it to ease the creation of customised or personalised services that will be required in the NGN and ubiquitous communication era. Context-aware dynamic binding of service components, finer-grained SIP call handling, and execution control for guaranteeing the service level agreement are specific features of our service delivery platform (SDP). This platform is expected to play an important role in the NGN and so we have developed a prototype.

This paper is organised as follows. In the next section, the basic approach to establish our SDP architecture is presented. Then, Sec. 3 deals with major capabilities required for the SDP on the NGN and Sec. 4 covers ubiquitous context-aware communications [1]-[4]. Next, in Secs. 5 and 6, we show a service creation and propagation mechanism and example applications. Finally, we conclude the paper.

## 2. Basic approach

Considering that Web 2.0 is gaining a foothold in advancing the Internet, telecom operators should provide Web2.0-related services over their NGNs. The major concepts of Web 2.0 are the provision of attractive services that meet various and specific (long-tail) customer demands, a framework where users act as co-developers or contributors, and a fully component-oriented nature.

Our SDP is aimed at easy and quick creation of Web-telecom coordinated services by mashing-up components accessible through networks, e.g., the NGN, the Internet, or sensor networks, and propagating them smoothly [2]. We take an abstract-scenario-based service composition approach for long-tail or personalised service mash-up at a low cost. Although the basic mash-up method is provided by vendors, we add context-awareness to it as described in Sec. 4.

Because our abstract-scenario-based mash-up is easily learned, third parties can participate in the service creation and/or provision as contributors. Even ordinary people can contribute to this service creation-provision cycle. In conjunction with the NGN's QoS guarantee and solid authentication capabilities, services can be provided more comfortably and securely than on the Internet. Also, the service level agreement (SLA) can be guaranteed.

# 3. Configuration of proposed SDP

## 3.1 Overview

The service control positioning in the NGN is described in the IP Multimedia Subsystem (IMS) standard as shown in Fig. 1. Although requirements are discussed, the precise SDP configuration is not defined. Therefore, major vendors are currently developing their own SDP products based on their concepts.

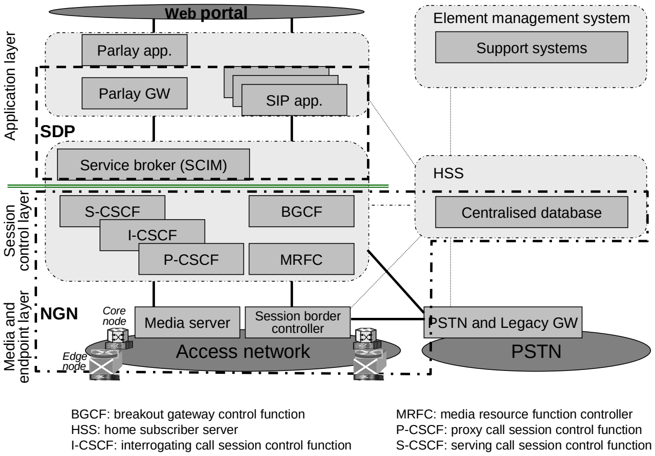

Web portal
Element management system
Support systems
Application layer
Parlay app.
Parlay GW
SIP app.
SDP
Service broker (SCIM)
HSS
Session control layer
S-CSCF
I-CSCF
P-CSCF
BGCF
MRFC
Centralised database
Media and endpoint layer
NGN
Core node
Edge node
Media server
Session border controller
PSTN and Legacy GW
Access network
PSTN
BGCF: breakout gateway control function
HSS: home subscriber server
I-CSCF: interrogating call session control function
MRFC: media resource function controller
P-CSCF: proxy call session control function
S-CSCF: serving call session control function

**Fig. 1 NGN functional blocks defined in IMS**

However, they have basically the same architecture. An orchestration server controls the coordination of components and enablers abstract the underlying telecommunication network capabilities as shown in Fig. 2. Our SDP uses this basic architecture but some specific values are added. First, it places the SCIM block between the NGN and SDP as defined in the IMS standard (Sec. 3.3). Then, it also provides capabilities for facilitating context-aware communication in the forthcoming ubiquitous era or for activating the service creation cycle (Sec. 4).

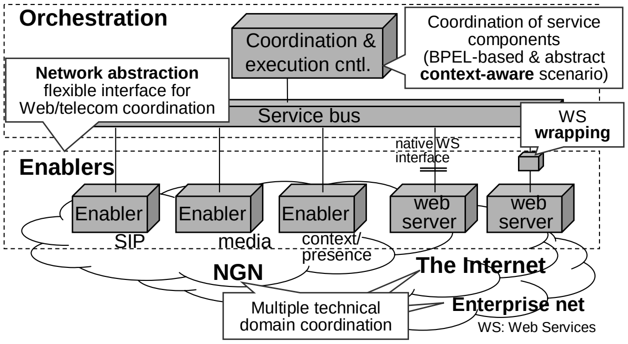

Orchestration
Coordination & execution cntl.
Coordination of service components (BPEL-based & abstract context-aware scenario)
Network abstraction flexible interface for Web/telecom coordination
Service bus
WS wrapping
native WS interface
Enablers
Enabler
Enabler
Enabler
web server
web server
SIP
media
context/ presence
NGN
The Internet
Enterprise net
Multiple technical domain coordination
WS: Web Services

**Fig. 2 Common configuration of SDPs**

## 3.2 Enablers

We introduce a SIP session control enabler, a media handling enabler, and a context handling enabler over the NGN to abstract and offer network-related capabilities in the form of well-defined and standardised interfaces. Most SIP enablers provide Parlay-X session-handling interfaces, but we define more elemental application programming interfaces (APIs) as shown in Fig. 3.

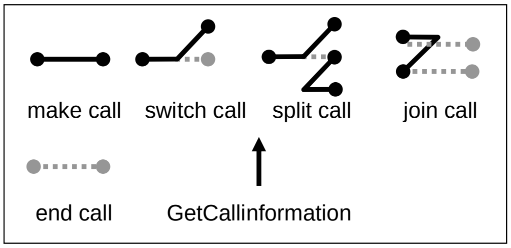


**Fig. 3 Our proposed APIs**

Because the granularity of Parlay-X is rather large (e.g., the unit of three-way calling or a conference call), we introduce a set of basic operations that can be used in any combination [4]. Our APIs can not only be used independently, but also as attachable components to enhance Parlay-X based enablers. Also, the media handling enabler provides finer-grained media control capabilities for service coordination to service providers that cannot provide an expensive media server by themselves [5].

Based on our APIs, we can create and provide a differentiated service as shown in Fig. 4. This is a call-centre application. First, a customer asks operator A something. The original connection is split into two by the *SplitCall* interface, one for between the customer and the guidance server, and the other for between operator A and the expert (operator B). After a conversation between the two operators, a connection between the customer and operator B is established by using the *JoinCall* interface. If a current Parlay-X interface is used, the call is disconnected once during the reestablishment procedure.

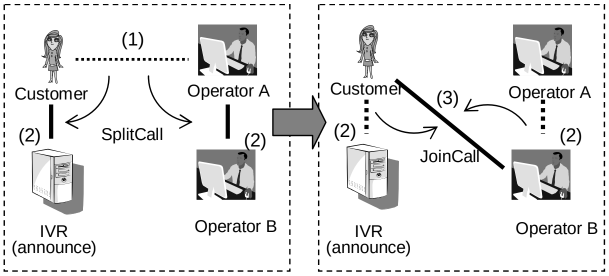


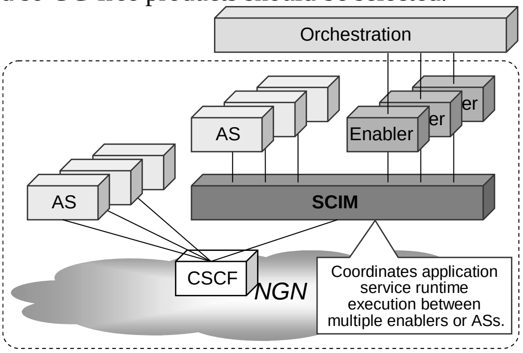
**Fig. 4 Differentiated service based on our enablers**

The context/presence enabler provides Web Services-based user context APIs by converting presence information managed in the underlying network, location servers, or sensor/intelligent transport system (ITS) networks.

### 3.3 SCIM

As defined in the IMS standard, the Service Capability Interaction Manager (SCIM) function block is necessary to control SIP messages between the network and enablers. In our architecture, the SCIM can be included if desired [6]. The SCIM distributes messages to appropriate destinations based on customer profiles or SIP parameters, blocks inconsistent messages, relays messages between Application Servers (ASs) and enablers, and controls service contention.

Because most service control messages pass the SCIM, the performance is one of the major items to be solved. One approach is the use of carrier-grade middleware and the C language, but it is not easy to add or change new service logic. Therefore, we currently think that the use of an off-the-shelf SIP Servlet product as the core protocol handling part for this block as well as for the SIP session enabler is effective. Its use gives programmers a Java-based easy software development environment, but several countermeasures are required against, for example, the garbage collection (GC) procedure. Ordinary Java virtual machines allow the service stoppage by GC, and so GC-free products should be selected.


Orchestration
er
er
AS
Enabler
AS
SCIM
CSCF
NGN
Coordinates application service runtime execution between multiple enablers or ASs.

AS: application server, CSCF: call session control function

**Fig. 5 SCIM positioning**

### 3.4 Orchestration server

The orchestration server consists of a service bus that efficiently connects all the elements and a coordination server that includes the following capabilities [3].

- BPEL engine: executes scenarios for service coordination.
- Execution control: guarantees secure service coordination by providing SLA, authentication, and supervision capabilities.
- Ubiquitous scenario translation: see Sec. 4.

We basically use off-the-shelf products for the service bus and BPEL engine parts, but we have devised and prototyped special elements for the execution control and ubiquitous handling. In particular, the priority control of the packet flow in the bus is a key to guaranteeing the SLA, and ubiquitous context-awareness will enhance personalised-service creation.

For the execution control, we introduce the concept of Policy Evaluation, Enforcement and Management (PEEM), one of the policy enforcers standardised in the Open Mobile Alliance (OMA). Considering the performance, we prototyped a proxy type system (Fig. 6), where our orchestration server performs single-sign-on authentication by the user identifier (ID) and authentication ticket contained in the request, converts the ID in accordance with the designated service, controls traffic, and transfers the request. In addition, it guarantees the SLA by observing the response time.

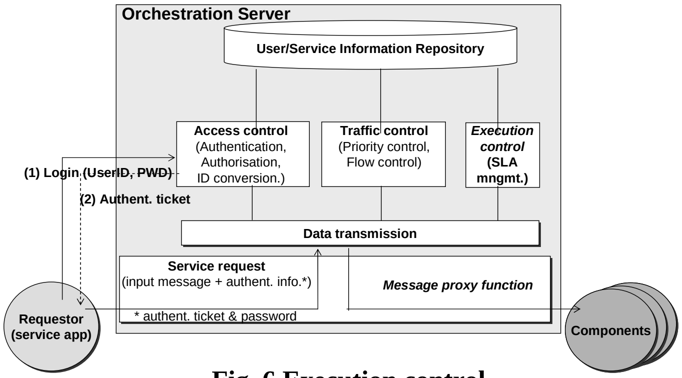

Orchestration Server
User/Service Information Repository
Access control (Authentication, Authorisation, ID conversion.)
Traffic control (Priority control, Flow control)
Execution control (SLA mngmt.)
(1) Login (UserID, PWD)
(2) Authent. ticket
Data transmission
Service request (input message + authent. info.*)
Message proxy function
* authent. ticket & password
Requestor (service app)
Components

**Fig. 6 Execution control**

To support service creation, development, and service provision, we have also devised a development environment that supports service creation for developers, i.e., a graphical service creation support tool [7] and a wrapper tool that converts interfaces between ordinary Web pages (HTML) and Web Services [8]. Basically, the SOA-based framework requires a standardised interface for invoking components, i.e., the Web Services interface, but this tool allows ordinary Web applications written in HTML to be used as components.

## 4. Towards ubiquitous SOA

We regard context-awareness as the key software technology in the ubiquitous era. We have established a context-aware service composition framework [9], and we provide it as an advanced service. In this framework, the service scenario is described in a more abstract format than BPEL. BPEL requires specific or exact component names and parameters for invocation as a C-language function-call, but in our case, abstractly described process names in a scenario are translated at runtime and the most appropriate components are selected and invoked in accordance with user context. For this purpose, every component is given a semantic description by the OWL-S standard [10], and the engine translates the process name into an exact component that meets the user context.

For example (Fig. 7), if we describe a scenario in which a user's schedule is retrieved from the scheduler and the user is notified of route search results, the sequence of "schedule retrieval", "route search", and "user notification" category names is translated into a set of exact components and each component is invoked. Differences in parameters are adjusted by checking their Web Services Definition Language (WSDL) definitions, and a SOAP message is parsed and exchanged.

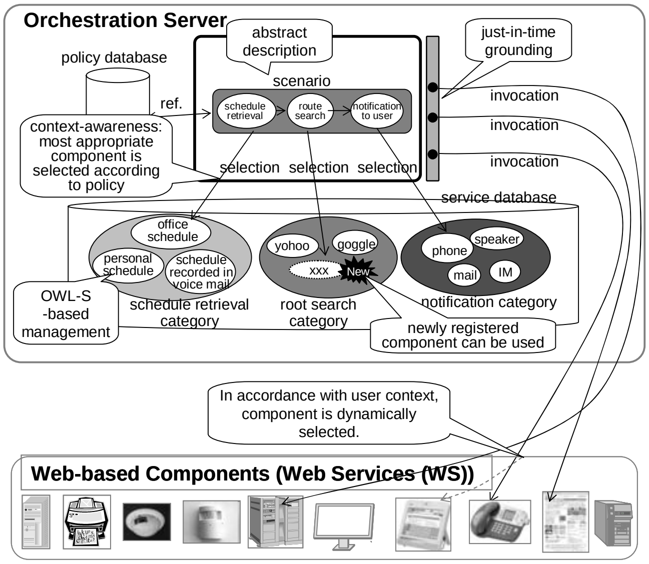


**Fig. 7 Context aware service coordination**

## 5. Activation of Coordinated Service Creation and Propagation

An overview of the future network service distribution environment based on our platform is shown in Fig. 8. Elementary service components of telecommunications, enterprise network, Web, and sensor network domains can be composed by a service scenario that describes the sequence of components (or categories) to be invoked, usage conditions, and exchanged parameters. Once a service has been created, it can be used as a basis for other enhanced services by adding new components or scenarios. This framework will be used by third-party venture companies as well as the development section of an operator. In addition, billing and settlement capabilities are offered to support the activities of third parties.

Users can use various enriched or personalised services over a safe and high-throughput network, i.e., the NGN. Operators can create various enriched/personalised services quickly and cost-effectively because anyone, even ordinary people, can participate in creating/providing their own services. From the viewpoint of third-party service providers, telecommunication-specific capabilities such as telephone handling, security management, and bandwidth control are effective as components to build one's own Web-telecom coordination business even if the provider does not precisely know complicated specification or procedures about telecommunication capabilities.

## 6. Applications

There are two approaches to create applications over the SDP. One is to enhance the basic voice/video call session by adding Web or enterprise components. This is similar to the Advanced Intelligent Network (AIN) services. The other is to trigger a voice/video call session from a Web or enterprise application. Internet developers, or even ordinary users, can easily participate in this type of approach. The main target of service propagation discussed in Sec. 5 will be this case.

An example of the first approach is shown in Fig. 9. The basic videophone communication service is enhanced by composing Internet components such as commercial (CM) content, merchandise information, word-of-mouth (WOM) information, media mixer (media enabler), and electronic commerce (EC) sites. In addition, the SIP adaptor conceals complicated SIP call control procedures so that existing EC sites can operate on the NGN.

The example shown in Fig. 10 is of the second approach. If there are components such as an OA scheduler, a text-to-speech device, and a SIP session control enabler, we can create a service that informs the user of the schedule by an artificial voice through an automatically set-up call.

Of course, many other service applications can be proposed, and we have prototyped several ones. Because the context aware mechanism allows us to designate an abstract process name instead of a particular action (function call), a current service application can be easily modified by replacing the existing component with a new one belonging to the same category. This helps to effectively develop service applications.

## 7. Conclusion

We showed an advanced service delivery platform for the NGN. This platform is based on state of the art technologies beyond the boundaries of technical domains, i.e., the telecommunication, enterprise, and Internet fields, to enrich services and mash-up simple service elements into unlooked-for values. In addition, we used the SOA concept as the basis and extended it to ease the creation of customised or personalised services. Context-aware dynamic binding of service components, finer-grained SIP call handling, and execution control for guaranteeing the service level agreement are main features of our platform. These features help us to create more attractive enriched service capabilities and activate the NGN business scene more than those of major vendor products. Our platform is expected to play an important role in the NGN or ubiquitous communication environments.

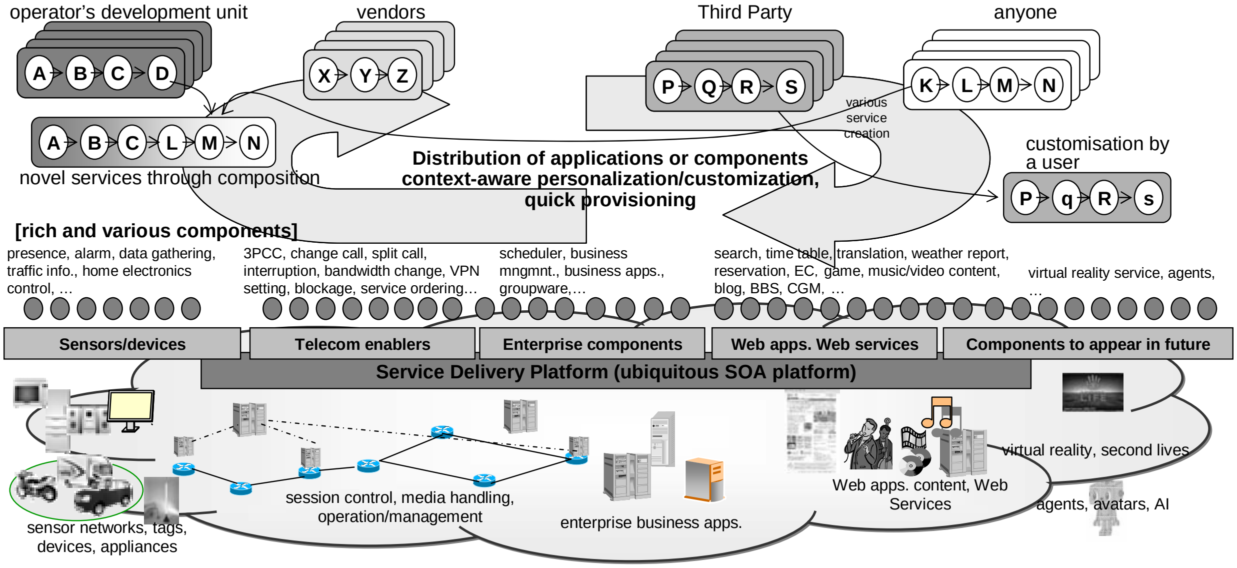

operator's development unit
A B C D
vendors
X Y Z
Third Party
P Q R S
anyone
K L M N
various service creation
A B C L M N
novel services through composition
Distribution of applications or components
context-aware personalization/customization,
quick provisioning
customisation by a user
P q R s
[rich and various components]
presence, alarm, data gathering, traffic info., home electronics control, …
3PCC, change call, split call, interruption, bandwidth change, VPN setting, blockage, service ordering…
scheduler, business mngmnt., business apps., groupware,…
search, time table, translation, weather report, reservation, EC, game, music/video content, blog, BBS, CGM, …
virtual reality service, agents, …
Sensors/devices
Telecom enablers
Enterprise components
Web apps. Web services
Components to appear in future
Service Delivery Platform (ubiquitous SOA platform)
sensor networks, tags, devices, appliances
session control, media handling, operation/management
enterprise business apps.
Web apps. content, Web Services
virtual reality, second lives
agents, avatars, AI

**Fig. 8 Service creation and delivery by using our service delivery platform**

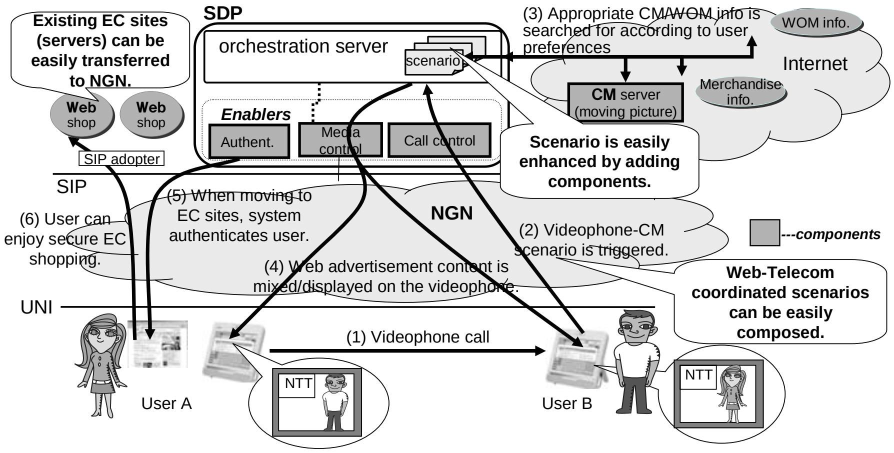
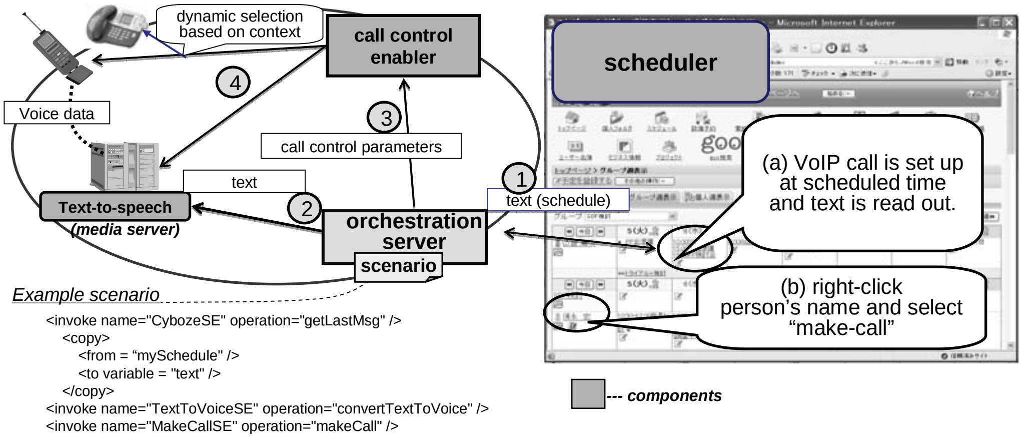

Existing EC sites (servers) can be easily transferred to NGN.
Web shop
Web shop
SIP adopter
SDP
orchestration server
scenario
Enablers
Authent.
Media control
Call control
(3) Appropriate CM/WOM info is searched for according to user preferences
WOM info.
Internet
CM server (moving picture)
Merchandise info.
Scenario is easily enhanced by adding components.
SIP
(6) User can enjoy secure EC shopping.
(5) When moving to EC sites, system authenticates user.
NGN
(2) Videophone-CM scenario is triggered.
---components
(4) Web advertisement content is mixed/displayed on the videophone.
Web-Telecom coordinated scenarios can be easily composed.
UNI
(1) Videophone call
User A
User B
NTT
NTT

**Fig. 9 Application example 1 –SIP video call service with Internet content**



```
<invoke name="CybozeSE" operation="getLastMsg" />
   <copy>
      <from = "mySchedule" />
      <to variable = "text" />
   </copy>
<invoke name="TextToVoiceSE" operation="convertTextToVoice" />
<invoke name="MakeCallSE" operation="makeCall" />
```

**Fig. 10 Application example 2 – Scheduler with SIP voice calling**

## References

[1] H. Sunaga, M. Takemoto, Y. Yamato, Y. Yokohata, Y. Nakano, and M. Hamada, "Ubiquitous Life Creation through Service Composition Technologies", WTC2006, May 2006

[2] H. Ohnishi, Y. Yamato, M. Kaneko, T. Moriya, M. Hirano, and H. Sunaga, "Service Delivery Platform for Telecom-Enterprise-Internet Combined Services", IEEE Globecom2007, pp. 108-112, November 2007

[3] Y. Yamato, and H. Sunaga, "Context-aware Ubiquitous Service Composition Technology", CONFENIS 2006, pp. 51-61, April 2006

[4] M. Kaneko et al., "Novel SIP Session Handling in a Service Delivery Platform for Telecom-Web Integration", IEICE APSITT2008

[5] M. Irie et al., "Highly Flexible Media Handling Enabler for the NGN Service Delivery Platform", IEICE APSITT2008

[6] M. Nakajime et al., "SIP Servlet Dialogue Handling for NGN Service Capability Interaction Manager", IEICE APSITT2008

[7] T. Moriya, H. Ohnishi, M. Yoshida, and M. Hirano, "Dataflow Generation for Service Composition to Incorporate Web and Telecommunication", IEEE Globecom2007, pp. 118-123, November 2007

[8] Y. Nakano, Y. Yamato, M. Takemoto, H. Sunaga, "Method of creating web services from web applications", IEEE International Conference on Service-Oriented Computing and Applications (SOCA)2007, June 2007

[9] Y. Yamato and H. Sunaga, "Context-Aware Service Composition and Component Change-over using Semantic Web Techniques", IEEE ICWS 2007, July 2007

[10] W3C, "OWL Web Ontology Language for Services (OWL-S)", http://www.w3.org/Submission/OWL-S/