논문 2011-48TC-9-3

개방형 서비스 게이트웨이를 위한 유비쿼터스 지역서비스 자동구성

(Autonomic Service Composition of Localized Ubiquitous Services for Open Service Gateways)

노 지 훈*, 윤 원 식**

(Jihoon No and Wonsik Yoon)

요 약

무선 네트워크 기술 및 모바일 컴퓨팅 기기의 발전과 함께 웹 서비스 기술이 발전하여 유비쿼터스 환경에서 새로운 서비스가 출현하게 되었다. 이러한 서비스들은 홈, 사무실, 공항, 전시장 등과 같은 위치 환경과 깊은 관계가 있으며, 위치환경에 따라 차별화된다. 사용자의 환경정보에 따라 다양한 지역서비스를 사용하는데 필요한 프로그램들이 사용자의 기기에 동적으로 전달되어지고, 지역에 따라 분산된 다양한 서비스들을 효과적으로 관리해 주는 시스템이 요구된다. 본 논문에서는 편리하게 서비스 선택을 할 수 있으며 유용한 서비스를 사용자에게 제공해 줄 수 있는 자동 서비스 구성 시스템을 제안한다. 제안한 시스템은 사용자가 지역 서비스 존에 들어가게 되면 모바일 기기에 자동으로 지역서비스 프로그램을 설치하고 지역서비스를 제공하여 준다. 또한 본 시스템은 분산된 지역서비스를 원격에서 효율적으로 관리할 수 있으며, 원격 관리자, 서비스 게이트웨이, 모바일 기기로 구성되어 있다. 본 시스템을 802.11b 무선 네트워크와 OSGi 프레임워크 기반으로 구현하였다.

Abstract

Many types of services appear in a ubiquitous environment promulgated by the evolution of web service technology with the advances in wireless network technologies and mobile computing devices. These services differ according to their location environments, such as home, office, airport, and exhibition. It is required that a different set of services dynamically drops into the mobile user's device depending on their context and the distributed localized services are efficiently managed to seamlessly provide these localized services to the user. This paper proposes an Autonomic Service Composition System (ASCS) to provide useful services to the user with minimal or no effort for service selection. ASCS seamlessly installs the programs of localized services in the user's mobile device. It automatically provides localized services to users that are in the local service zone. Also ASCS can manage distributed localized services remotely and efficiently. ASCS is composed of a Remote Manager, Service Gateway, and Mobile Device. The prototype implementation uses 802.11b Wireless Network and Bundles using the OSGi Framework.

Keywords : Autonomic service composition, localized ubiquitous service, open service gateway

Ⅰ. Introduction

Many technologies for ubiquitous computing and network have been developed. Broadband access and nomadic access are the subjects of major research efforts. Practical techniques to utilize the CPU power and storage areas of user machines are now in use [1]. Much progress has been made in implementation technologies for sensors from which dynamic context information about the real world is obtained. In general, the progress of software system, nomadic

* 정회원, 삼성전자
(Samsung Electronics)
** 정회원-교신저자, 아주대학교 전자공학부
(Department of Electrical and Computer Eng., Ajou University)
접수일자: 2011년4월26일, 수정완료일: 2011년9월16일

computing, and sensor network technologies is bringing the ubiquitous computing environment ever closer [2].

A ubiquitous computing environment suggests many kinds of services related to the user's situation or context. These services may be dependent on their location, such as home, school, airport, exhibition, and office. If the services fit the user's situation or context, the user can readily use the services to be personalized to his/her needs. Research on context-aware computing has been done in recent years, though not all specifically dealing with localized services[3~5].

Contextual information and user characteristics have been employed to tailor applications. The Open Services Gateway Initiative (OSGi) attempts to provide a managed and extensible framework to connect various devices in a local network, such as in a home, office, or automobile. By defining a standard execution environment and service interfaces, OSGi promotes the dynamic discovery and collaboration of devices and services from different sources. The framework is designed to ensure smooth space evolution over time and to support connectivity to the outside world, allowing remote control, diagnosis, and management. The OSGi bundle yields unprecedented flexibility in terms of computation location and reduction in set up time and effort for applications[6].

In this paper, we propose an Autonomic Service Composition System (ASCS) to provide useful services to the user with minimal or no effort for service selection. Context is used to suggest services that are useful and relevant to the user. ASCS contexts include location, user, device resources, and time. In our system, we mainly use location, device, and user contexts. The location context is represented by the zone of an indoor logical area. When these contexts are sensed, a service gateway computes available service/device/user list, and sends this list to the user's devices. The user selects the service to use and the service bundle is automatically downloaded and set up, effectively getting a required code.

As contexts are changed, the list of services/devices/users is automatically updated on the user's device and the downloaded bundles are managed according to device resources. Context-awareness of the service gateway is important and the bundle takes advantage that the service can be used immediately without the effort to download and install the service software to offer the user proactive behavior. As ASCS service is based on the OSGi bundle, it is ideal to deliver mobile applications that make use of context information. We also suggest ubiquitous exhibition guide services, as a mobile application and explain ASCS using this scenario.

The remainder of the paper is structured as follows. In Section Ⅱ, we present the components of ASCS. In Section Ⅲ, we describe the system architecture and system flow of ASCS. In Section Ⅳ, we suggest a mobile application of ASCS and evaluate ASCS. The paper ends with our conclusions in Section Ⅴ.

Ⅱ. ASC System Components

1. Context-Awareness

The context is valuable information for a device or service operation. Based on [4], we categorize four kinds of context. (a) Device context: This can be any information about devices that provides a service (e.g. printer, scanner, home automation server, streaming server, and so on). This involves device identification, device type, device name, device IP address, user of this device, the location of the device, and so on. (b) Service context: This refers to any knowledge that relates to service information, such as service identification, service name, device identification of this service, URL of service server, and the URL of the client mobile code. (c) User context: User context refers to any information about the users, such as user identification, and user name.

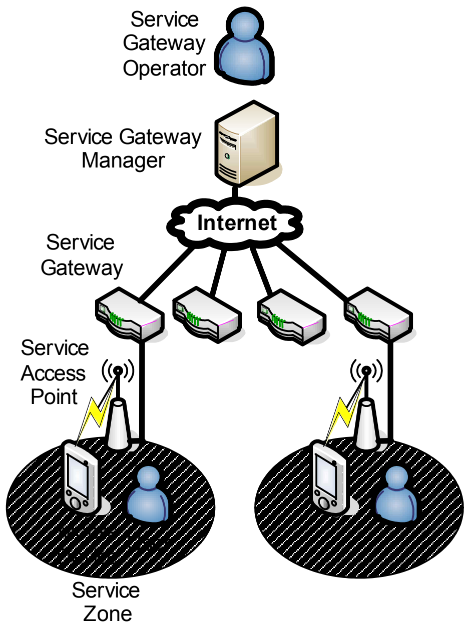

그림 1. 자동 서비스 구성 (ACS) 시스템 구성도
Fig. 1. High level architecture of ASC system.

(d) History context: This context includes any event history that occurs in the mobile device, such as location changes, service start and stop, and user changes. The history context is high level context created from device context, service context, and user context. Each context has information about the time it was created. The contexts are used to reason the user's situation, mobility management, and proactive services.

2. Autonomic Service Composition System

ASCS composes service environments automatically when a user comes to a new service zone. The system manages distributed localized services and provides localized services to the mobile device in the service area. When a user comes to the area of the localized service, the user's mobile device downloads and installs the localized service program seamlessly. The high level architecture of ASCS is shown in Fig. 1. When the service gateway operator manages localized services, it is too difficult to match and manage localized services according to each service gateway. The ASC system also provides an infrastructure that enables the service gateway operator easily manage localized applications. The components of the system are described.

A. Remote manager

Remote manager is a server to manage a set of localized services based on the service gateway. A service gateway operator wishing to register a new localized service of a service gateway, does so in three steps. First, a service gateway operator should register the new localized service and insert the information of the new localize service. Second, the service gateway operator should register the new service gateway. The service gateway operator inputs its information and registers the services to each service gateway.

When a new service gateway is installed and connected to the network, the remote manager identifies the service gateway using the basic service gateway information input by the service gateway operator. Then, the remote manager installs localized services using the service gateway information and the service information, and checks the status of the registered service gateways.

B. Service Gateway

Service gateway is the gateway between different services. It is similar in that a network gateway is the gateway for different networks to have compatibility. The service gateway provides the inter-operation between different services by defined interfaces. The service gateway also manages the local contexts, such as service context, user context, and device context.

When the service gateway is connected to the IP network, it registers its information to the remote manager and gets the information of the localized service from the remote manager. The service context is already registered by the service gateway operator. The service gateway also manages contexts of the user and devices in the local service zone. When a new mobile device comes to the local service zone, it requests the context information of the local service zone and the service gateway provides service context, device context, and user context to the new mobile device.

C. Mobile Device

A user can access localized services in the service zone, using a mobile device. It sends the device context and the user context to a service gateway and gets contexts of the local service zone from the service gateway. One should input the user context (user ID and username) in the mobile device before a user uses a mobile device. When a mobile device comes to a new service zone, the mobile device requests its registration of a service gateway at the local service zone. After the registration of the mobile device, the mobile device obtains the contexts of the local service zone, such as users, devices, and services. A mobile device downloads and installs client programs of localized services based on the service contexts automatically and a user can use localized services in the service zone.

3. Mobile Code

Mobile code is transmitted to a node across the network and executed on the node. It is no longer constrained to execute on the nodes where they reside. Unlike mobile computing, in which hardware moves, mobile code changes the machines where the program executes. Mobility allows vendors of localized services, reconfigure software without shipping a physical medium. Mobile code can help distributed systems adapt autonomously. It also includes downloading and installing software for new features.

In our system, we use client mobile code to support code mobility. Client mobile code is the client-side mobile code of services. When a new mobile device enters a new service zone and is registered to a service gateway of the service zone, it obtains service context from the service gateway. A service context includes the URL of the mobile client code. Based on this, the new device can download and install the client mobile code of localized services. Mobile code increases system flexibility, scalability, and reliability.

4. Localized Services

Localized services are characterized by the location area and have been developed for a number of everyday scenarios:

- Office applications, such as nearest printer services.
- Tour and museum guides can help people navigate an unfamiliar space.
- Conference aids can track presentation attendance and facilitate note taking and discussion.
- Home applications can help with household management and home entertainment, as well as aid the aged and disabled in performing everyday tasks.

If these localized services are used, a client service program is installed or a special device, such as a remote controller, is required for them. In ubiquitous environments, a user wants to add a new service to the mobile node. This is done by the mobile code and the service gateway that manages localized service contexts. The service expiry time is used to manage the situation in which a mobile device moves out of the service range. The service gateway broadcasts an advertisement message periodically and the mobile node checks the message. If the message is changed or the mobile node cannot find it, the mobile node thinks that it moved out of the previous service zone and tries to find the same type of service that the user used in the previous service zone. If there is a service of the same type as the previous service, the mobile node downloads the client program. If none is the same as the previous service, the mobile node stops the service.

Ⅲ. ASCS Architecture and Flow

1. Architectural Design

The system architecture is shown in Fig. 2. In this section, we describe the system architecture and explain components of our system and system flow. ASCS is composed of three main services: Service Gateway Management Service, Autonomic Service Composition Service, and Location Aware Service.

A. Service Gateway Management Service

Service gateway management service manages the distributed service gateway and localized service efficiently. The main components are service context manager, service gateway management server, service gateway management client, and service repository of service gateway.

Service context manager handles service contexts and service gateway contexts. They are registered by the service gateway operator and are stored in databases of the service context manager. The service gateway management server manages service gateways by interacting with the service gateway management client. The service gateway operator can control services of service gateways and check the status of them remotely. The service gateway management client controls localized service based on the command of the service gateway management server and reports the service gateway status. When the service gateway management client requests the service context of localized services, the service gateway management server requests them of the service context manager and send them to the service gateway client. The service gateway management client also downloads and stores mobile code of localized services in the service repository of the service gateway.

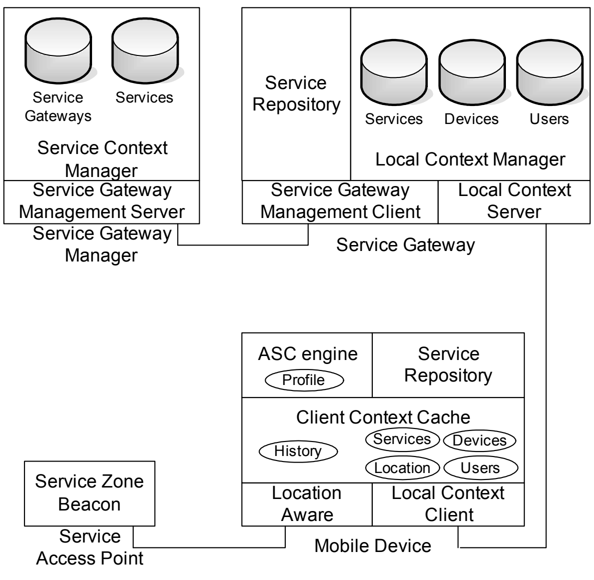

그림 2. ASCS 구조도
Fig. 2. Autonomic service composition system architecture.

B. Autonomic Service Composition Service

Autonomic Service Composition Service is a core service that composes localized services in a service zone. It is composed of two main components. One is local context management for service composition; the other is autonomic service composition to download and install the client software of localized service based on local context.

The local context management component consists of local context manager, local context server, local context client, and client context cache. The local context manager handles all local context information provided by the mobile node. The local context server provides local context stored in the local context manager of the service gateway to the local context client of the mobile node. The local context client provides its context to the local context server of the service gateway and requests registration and local context that is required for service composition. The client context cache stores local context for service composition and a location context is provided by the location-aware service. It also has a profile that includes the mobile node's information, ht euser's information using the mobile device, and the service history. The service history is log information of services the user used.

The autonomic service composition part is done by the ASC engine. The ASC engine gets local context from the client context cache and displays them to the user. When a user selects a localized service, the ASC engine downloads the client mobile code of the selected service from the service gateway and installs it to perform the service to the user. After the installation procedure, it is initiated and the user can access the service. If a user selects a device, the ASC engine shows the control operation of the selected device and the user can control the device. If a user context is selected, the ASC engine shows the user information. The ASC engine also writes the events to the profile of the client context cache. When a mobile device moves to a new service zone, the ASC engine tries to find similar services that are used in previous service zones. Thus the ASC engine enables environments to access localized service automatically.

C. Location aware service

This service is for the mobile node to be aware of the service zone. It is composed of the service zone beacon of the service access point and location aware program of the mobile node. The service zone beacon broadcasts the location context message periodically and the location aware program of the mobile node gets the location context of the message. This location context is stored in the client context cache.

2. System Flow

The three main services of the ASC system are the service gateway management service, location aware service, and autonomic service composition service. Fig. 3 shows the ASC system procedure. First, the service gateway management service operates and localized services are deployed to service gateways. The localized services are ready to provide their service to the user in the service zone. The next procedure is initiated by the mobile node. The mobile node gets the location context from the location aware service. It registers itself to the service gateway and gets local context from the service gateway from the autonomic service composition service. We now describe each step in Fig. 3.

1a. Enter information of Service Gateway and Services : The service gateway operator registers service gateway information and services in the remote manager. The service context manager of the remote manager writes them to their repository.

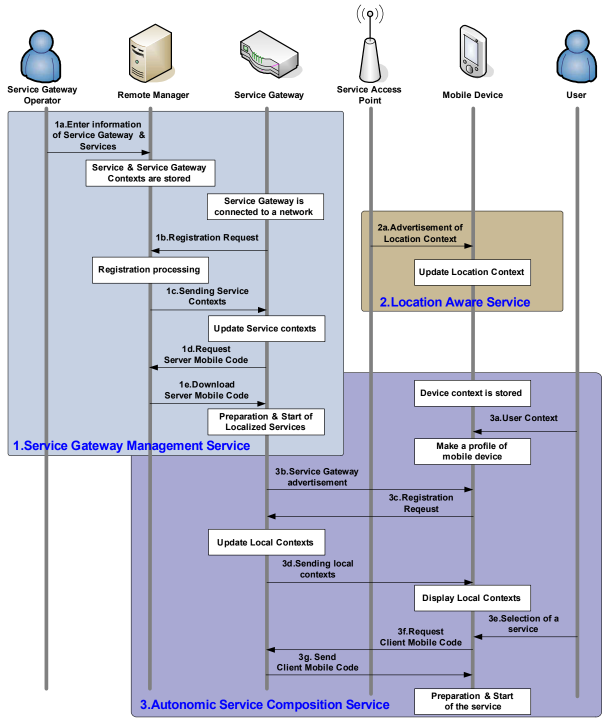

그림 3. ASCS 흐름도

Fig. 3. Autonomic service composition system flow

1b. Registration request : When a service gateway is connected to an IP network, a service gateway sends the registration request message to the remote manager. If the service gateway is valid, the remote manager registers it.

1c. Sending Service contexts : After the registration procedure, the remote manager sends context to the service gateway.

1d. Request server mobile code : The service gateway requests the server mobile code from the remote manager based on the service context of the service gateway.

1e. Download server mobile code : When the remote manager receives the request message from the server mobile code, it sends it to the service gateway. The service gateway installs and runs the server mobile code. The localized service is ready to await the mobile devices that wish to use it.

2a. Advertisement of location context : The service access point broadcasts the advertisement message periodically. The mobile device gets this message and creates the location context from it.

3a. User context : A user writes user context in the mobile device and the mobile device forms the profile of the mobile device using the device context and user context. This profile is used in the registration procedure of the mobile device.

3b. Service gateway advertisement and 3c. Registration request : The service gateway advertises itself periodically using the advertisement message. When the mobile device receives this message, the mobile device requests registration to the service gateway with the mobile device profile. The service gateway receives this registration request message and updates local context with the profile of the registration request message.

3d. Sending local context : After the registration of the mobile device and updating local context, the service gateway sends updated local context to the mobile device in the service zone. The mobile device's ASC engine displays them to the user.

3e. Selection of a service : A user selects a service from them.

3f. Request client mobile code : The ASC engine of the mobile node requests the client mobile code of the selected service from the service gateway.

3g. Send client mobile code : The ASC engine of the service gateway sends the client mobile code to the mobile device. The ASC engine installs and runs the received mobile code. Finally, a user can use the selected service.

3. Implementation Aspects

The service gateway and mobile device of autonomic service composition system is based on the OSGi framework, since the OSGi framework is good for dynamic service deployment, a service oriented approach, mobile code, and remote management. We also use the OSGi bundle for the mobile code

The OSGi Alliance is an independent non-profit corporation to define and promote open specifications for the delivery of managed services to networked

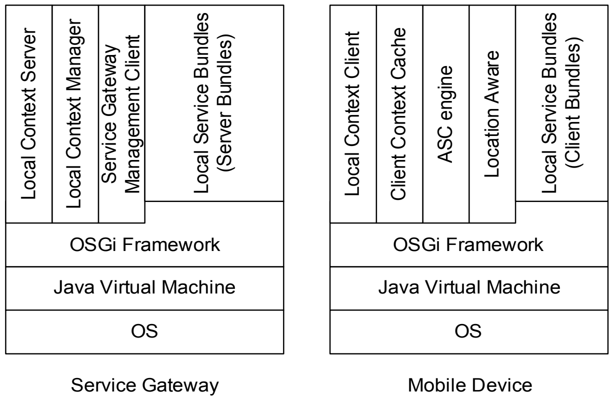

그림 4. 서비스 게이트웨이 및 모바일 기기 구조
Fig. 4. Architecture of service gateway and mobile device.

environments. The OSGi Alliance defined a Service Platform specification to achieve this goal. The OSGi Service Platform specification consists of two phases: the OSGi framework and a set of standard service definitions. The OSGi framework is an execution environment for dynamically loadable services. The standard service definitions specify the interfaces and semantics for several useful reusable services.

Recent trends marry the notion of service orientation to Web services. Web services are one realization of service-oriented principles based on Web-related standards and protocols. Service orientation is more general. It is founded on the concepts of late non-explicit bindings between clients and servers using mechanisms of functionality description and discovery. Some key traits of the service oriented approach are dynamism and substitutability. Dynamism occurs in two phase : service providers may offer or retract service at any time, and service requesters may bind available services. These traits of service orientation are very relevant to the OSGi vision of a dynamic networked environment, in which many service providers participate to offer functionality to the end user. The OSGi specification anticipates a highly dynamic networked environment[7].

The framework portion of the OSGi specification defines the execution environment for services and includes a minimal component model, management services for the components, and a service registry. Since the service interface is separate from any implementation, it is possible for numerous implementations of any service to exist. Service implementations are delivered and deployed to the framework in a physical and logical unit called a bundle. Physically, a bundle corresponds to a Java archive file that contains code, resources, and deployment manifest. Bundles are installed into the framework via networks.

The OSGi Service Platform is specifically designed for devices that can operate unattended or under the control of a platform operator. These are the devices that need remote management. The OSGi Alliance decided that no management protocol can be preferred over others, as no protocol is suitable for all cases. They chose an architecture that provides a management API to be used by an authorized bundle[8]. This authorized bundle can then act as a Management Bundle, where this bundle maps a protocol to API calls. The flexible OSGi remote management model allows the OSGi Service Platform to be used for the ASC system.

A Bluetooth Beacon System is developed for location-aware service. It is composed of Bluetooth Beacons and the Bluetooth Module of the mobile node. Bluetooth beacons have a unique ID for each area. When a location-aware service of a mobile device requests IDs of the Bluetooth Beacons, the Bluetooth beacons send their ID to the mobile device.

Ⅳ. Evaluation and Scenarios

1. Evaluation

The ASC system sends local context to a user and installs the client mobile code. We measure the time from when a user enters the service zone to when the client code of the service the user selects is installed and started to evaluate the ASC system. It is assumed that the service gateway already gets the information of localized services from a remote manager using service gateway management services. We measure the start time and the end time of the location aware service and autonomic service composition service. We evaluate how long it takes for the selected service to be provided to a user.

The Service Composition Time (Tc) is defined as the total time spent to provide a localized service to a user. It is composed of Location Aware Time (T_L), Mobile Device Registration Time (T_R), Local Context Update Time (T_U), and Mobile Code Download and Run Time(T_{DR}) :

$$T_C = T_L + T_R + T_U + T_{DR} \quad (1)$$

The Location Aware Time is the time spent for a mobile device to enter the service zone and to be aware of the location context by the advertisement of the service gateway. Mobile Device Registration Time is the time spent by a mobile device to request the registration to the service gateway and to obtain the local context list from the service gateway. Local Context Update Time is the time spent for a mobile to get local context and display them to a user. Mobile Code Download and Run Time is the time spent for a mobile device to download and run the client mobile code of the selected service.

We used Pentium 4 2.66GHz Desktop PCs for the remote manager and service gateway and used a notebook with a Centrio mobile 1.5GHz, 802.11b WLAN Card for the mobile device. It is assumed that

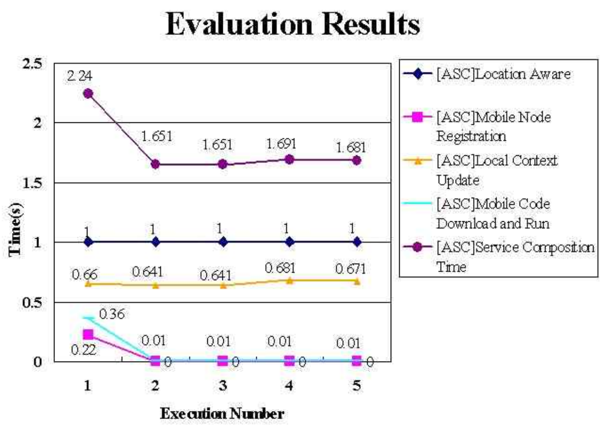

그림 5. ASC 시스템 평가
Fig. 5. ASC system evaluation.

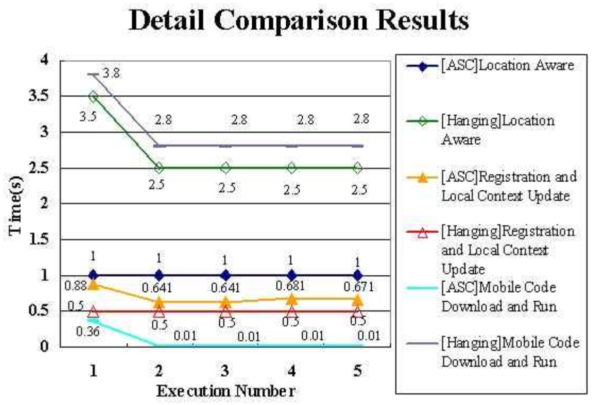

그림 6. Hanging 서비스 시스템과 ASC 시스템의 비교

Fig. 6. Comparison of the Hanging Service System and ASC system.

five mobile devices register at a service gateway and the service gateway sends an advertisement message per second. When the sixth mobile device enters the service zone, TL , TR , TU , and TDR are measured to calculate TC. Fig. 5 shows the results. The ASC system Service Composition Time is from 1.5s to 2.5s. The first try is shorter than the others by about one second, because TR and TDR are necessary in the first try. After the first try, the mobile device does not need the registration procedure and the client mobile code of the selected service is not downloaded and is run directly.

We also compare the result with the Hanging Service System[5] in Fig. 6 and Fig. 7. The service composition time of theASC system is shorter than that of the Hanging Service System, because a mobile device gets the client mobile code from the service gateway and the mobile code does not need to compile the code and the location aware time of ASC system is shorter that that of the Hanging Service System.

2. Implemented Scenarios

We suggest two applications for the application using the ASC system: Home Cooking Guide service and Context Aware Exhibition Guide service. The Home Cooking Guide service helps a user cook when s/he enters the kitchen. The Context Aware

그림 7. Hanging 서비스 시스템과 ASC 시스템의 서비스 구성 시간

Fig. 7. Service composition time of the Hanging Service System and ASC system.

Exhibition Guide service guides a user to explain the exhibit near the user. If a user enters a home with a mobile device, the mobile device registers itself to a home service gateway and obtains services from the home service gateway. When the user enters the kitchen, the Home Cooking Guide service is performed. If the user enters a exhibition with the same mobile device, the mobile device registers itself to a exhibition service gateway and obtains services from the exhibition and runs the Context Aware Exhibition Guide service. In this scenario, the ASC system displays the information of the local service area to a user and helps a user access the selected service. A Context Aware Streaming Service using the ASC system is suggested. When a user listens to music in his/her office using a Network Audio and moves from his/her office to his/her home, the Network Audio of his/her office is stopped and the Network Audio of his/her home plays the music that was played in his/her home.

V. Conclusion

In this paper, we proposed an ASC system to manage distributed services efficiently and remotely and to provide localized services to users automatically based on local context. In the ASC system, these localized services are managed by a remote manager efficiently and are seemlessly provided to a user. The OSGi Framework is used for a service-oriented approach and for remote management. The ASC system can update localized service dynamically and quickly by utilizing the OSGi Bundle for the mobile code of services. We composed a testbed and developed applications for the ASC system. The measurement and applications showed that the service composition is very short. We can see the future possibilities of using the ASC system. The ASC system is used in a local service system, such as the home network, telematics, intelligent building, and smart mobile phone.

References

[1] A. Grimshaw, M. Morgan, D. Merrill, H. Kishimoto, A. Savva, D. Snelling, C. Smith, and D. Berry, "An Open Grid Services Architecture Primer," *IEEE Computer*, vol. 42, pp. 27-34, Feb. 2009.
[2] J.S. Erickson, S. Spence, M. Rhodes, D. Banks, J. Rutherford, E. Simpson, G. Belrose, and R. Perry, "Content-Centered Collaboration Spaces in the Cloud," *IEEE Internet Computing*, vol. 13, pp. 34-42, Sep. 2009.
[3] G. D. Abowd, M. Ebling, H. W. Gellersen, G. Hunt & H. Lei (eds), "Special Issue on Context-Aware Computing," *IEEE Pervasive Computing,* vol. 1, 2002.
[4] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," Dartmouth Computer Science Technical Report TR2000-381. Available at ftp://ftp.cs.dartmouth.edu/TR/TR2000-381.pdf.
[5] E. Syukur, D. Cooney, L. S. Wai, and P. Stanski, "Hanging Services: an investigation of context-sensitivity and mobile code for localised services," in Proc. Mobile Data Management, pp. 62 - 73, 2004.
[6] C. Lee, D. Nordstedt, and S. Helal, "Enabling smart spaces with OSGi," *IEEE Pervasive Computing,* vol. 2, pp. 89 - 94, July 2003.
[7] R. S. Hall and H. Cervantes, "Challenges in building service-oriented applications for OSGi," *IEEE Communications Magazine,* vol. 42, pp. 144-149, May 2004.
[8] The Technical Whitepaper of Prosyst, "About the OSGi Service Platform," OSGi Alliance, 2004.
[9] Open Service Gateway Initiative, "OSGi Service Platform (Version 3)," Mar. 2003, http://www.osgi.org/

저 자 소 개

노 지 훈(정회원)
2003년 아주대학교 전자공학부 학사 졸업
2005년 아주대학교 전자공학부 석사 졸업.
2005년 - 현재 삼성전자 책임연구원
<주관심분야 : 무선통신, 무선단말>

윤 원 식(정회원)-교신저자
1984년 서울대학교 제어계측 공학과 학사 졸업.
1986년 한국과학기술원 전기 및 전자공학과 석사 졸업.
1991년 한국과학기술원 전기 및 전자공학과 박사 졸업.
1991년 금성전기 선임연구원
1994년 LG정밀 책임연구원
1995년 University of Victoria 방문교수
2001년 콘텔라 CTO
2007년 University of Florida 방문교수
1994년～현재 아주대학교 전자공학부 교수
<주관심분야 : 이동통신, 무선네트워크>