Standardization Activity of OSGi (Open Services Gateway Initiative)

Ryutaro Kawamura[†] *and Hiroyuki Maeomichi*

Abstract

This article provides an overview of OSGi (open services gateway initiative) standardization activities for software technology that is applicable to gateways that control and manage, in a coordinated fashion, various services and devices connected to networks in homes, offices, vehicles, and factories. The OSGi architecture allows for the timely provision of various new network-based services, such as home security and home automation, energy conservation, telematics (in which automobiles and the network interwork with each other), and factory control.

1. What is OSGi?

OSGi (open services gateway initiative) is a standards body that was founded in March 1999 to define open specifications for software technology providing a gateway function. A gateway function is needed for the connection of internal devices to an external network or for interconnections between the different protocols that are used in devices in homes, offices, vehicles, and factories [1]. With the advent of ubiquitous communication, PCs as well as diverse devices, such as sensors and information home appliances, are now being linked together, often using heterogeneous communication protocols. In addition, broadband access has become much more widespread and it is common to see homes and offices with always-on connections to the Internet. Against this backdrop, increased attention is being paid to gateways that provide key capabilities for functional interworking between devices and portal capability for using services offered by external networks, including the Internet.

The OSGi specification is a basic software specification that can be used to implement such gateway capabilities, and is built on software component technology using the Java language. Specifically, the specification assumes that an appropriate set of gateway functions will be implemented in home gateways, set-top boxes, PCs, computers in vehicles, broadband routers, and television sets.

2. Architecture of the OSGi specification

The architecture of the OSGi specification is explained below using a home application as a typical example (**Fig. 1**). A range of diverse devices that are found in a home environment, such as PCs and sensors, are connected to the home network using several different protocols including non-IP-based protocols (IP: Internet protocol). A home gateway connects all these devices to the network and controls them. In this case, the OSGi specification is implemented as a home gateway.

The OSGi architecture consists of the OSGi Framework and a set of bundles. The OSGi Framework provides the basic functionality for executing OSGi functions. Namely, it provides the environment for downloading and executing bundles. A bundle is a software component that contains algorithms and protocols for controlling a device. When a bundle is required, it can be downloaded from a server on the network and then executed. This feature makes it possible to download and use the latest and most optimal bundles and allows customization of gateway functions for each user. Since only the bundles that are needed are downloaded and stored, little memory space is required.

The OSGi architecture implemented within a gate-

† NTT Cyber Solutions Laboratories
Yokosuka-shi, 239-0847 Japan
E-mail: kawamura.ryutaro@lab.ntt.co.jp

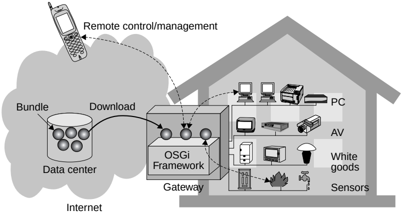

Fig. 1. OSGi control model (in a home).

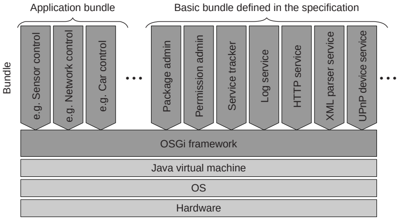

Fig. 2. Structure of OSGi gateway.

way is shown in **Fig. 2**. Both the OSGi Framework and bundles use the Java language, and can run on Java-compatible devices. (Considering the application to set-top boxes and other small devices, the operational conditions of the Java virtual machine (VM) for OSGi are specified in the latest version of the OSGi specification and these allow J2SE as well as J2ME and other VMs to be used). The actual implementation of the framework is a type of Java application. The framework contains the basic functionality for executing bundles, such as downloading, starting, and stopping bundles, and providing inter-bundle services, a security mechanism, and event processing. It is pre-installed on the gateway.

Bundles contain the algorithms and protocols to control individual devices. They can be either downloaded from a server or read from a local disk and then executed. Bundles are classified as either basic service bundles or application bundles. Basic service bundles provide basic functions for OSGi gateways, and are defined as part of the OSGi specification. Application bundles perform the individual application functions that are executed in the gateway. In other words, if a unique service based on the OSGi specification is implemented, it is usually necessary to develop and provide an application bundle. Links between bundles are provided by the framework. For example, the network control bundle in Fig. 2 can easily perform UPnP-based control by using the service provided by the UPnP Device Service Bundle (UPnP: universal plug and play). A bundle is actually a JAR file, which contains Java class files with specified interfaces and a MANIFEST file. By specifying a relevant URL, a bundle can be downloaded from a server on the Internet or from a local disk using Java's class loading function.

In this way, the OSGi specification provides a means to build gateway functions. It exhibits its greatest benefit when a connection to an external network, such as the Internet, is available allowing bun-

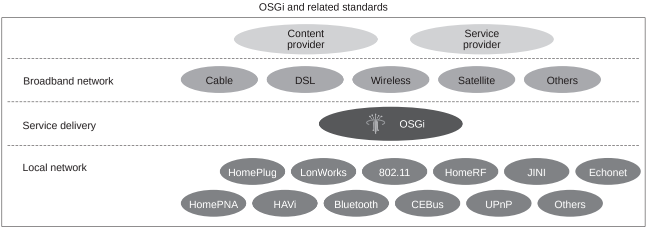

Fig. 3. Relationship to conventional specifications and protocols.

dles to be dynamically changed. The design of the OSGi specification assumes that multiple protocols can be used within the target device. The OSGi specification is not intended as a new protocol technology that will replace existing technologies. **Figure 3** shows the relationship between the OSGi specification and other standardized technologies. Multiple protocols can be accommodated by installing appropriate bundles for individual protocols, such as the UPnP Bundle. If required, a bundle provides interworking between different protocols to allow device control.

The OSGi specification consists of the OSGi Framework specification and the specifications for a set of basic service bundles. The latest specification is Release 3 [1]. The requirements of the hardware and OS (JavaVM and underlying environment) of a gateway are not included in the OSGi specification. Likewise, specific control algorithms for individual devices and the methods of interworking between protocols are to be individually implemented by bundle developers.

3. Examples of services using the OSGi specification

OSGi initially focused on home automation using a DSL modem or a set-top box as a gateway. Recently, the application areas have expanded to factories, automobiles, and ubiquitous communication. Typical application areas of the OSGi specification are listed below:

- home security
- home automation
- remote maintenance of home appliances and office devices
- health care
- services for the elderly and handicapped
- energy conservation
- factory automation
- in-vehicle applications (telematics)

Because of its features, the OSGi is particularly attractive for applications with the following properties.

- Addition or modification of functions/services/devices is required.
- Remote control is important, such as control from a data center, (provision of services to consumers).
- Different protocols are to be used.
- The gateway function needs to be implemented in a device with a small memory space.

At present, a number of organizations are developing and providing OSGi Frameworks and a variety of bundles. They are used mainly by OSGi members. Some are also used for commercial services [2].

4. Standardization activities and NTT's participation

Table 1 shows the main member organizations. IBM, Motorola, and other U.S. companies and telecommunication carriers in Europe are particularly active. The OSGi specification is open and may be used by non-member organizations. In recent years, automobile manufacturers, such as BMW, Daimler Chrysler, Audi, and Volvo, have been particularly prominent in making public their use of the OSGi specification.

NTT Laboratories is undertaking R&D on a proprietary communication middleware technology called "Communication Service Concierge (CSC)" [3], [4]

Table 1. Major members.

Acunia	Motorola
Alpine Electronics Europe Gmbh	NTT
AMI-C	Oracle
BMW	Panasonic Technologies
Deutsche Telekom	Philips
Echelon	ProSyst Software AG
Electricite de France (EDF)	Robert Bosch Gmbh
Ericsson	Samsung Electronics
ETRI	Sharp
France Telecom	Siemens VDO Automotive
Fraunhofer Gesellschaft	Sun Microsystems
Gatespace AB	Telcordia Technologies
IBM	Telefonica I+D
KDDI R&D Laboratories	Texas Instruments
Mitsubishi Electric	Toshiba

and joined OSGi in 2001 to improve CSC development by incorporating the OSGi specification. In addition to incorporating the OSGi specification in our CSC, we also pursue R&D and standardization so that OSGi's gateway design technology will be applicable to NTT's communication services.

References

[1] http://www.osgi.org/

[2] http://www.e2-home.org/

[3] H. Maeomichi, R. Kawamura, A. Tsutsui and K. Yata, "A QoS Management Framework based on COP oriented Communication Resource Coordination," ACM/IFIP/USENIX Middleware 2003, Rio de Janeiro, Brazil, June 2003. (http://www.cs.wustl.edu/~corsaro/papers/RM2003/p23-hiroyuki.pdf)

[4] A. Tsutsui, H. Maeomichi, R. Kawamura and K. Yata, "An Adaptive Communication Middleware for Network Service Coordination," IEEE CCNC 2004, Las Vegas, U.S.A., Jan. 2004.

Ryutaro Kawamura

Senior Manager, Home Network Service Innovation Project, NTT Cyber Solutions Laboratories.

He received the B.S.and M.S. degrees in precision engineering and a Ph.D. in electronics and information engineering from Hokkaido University, Hokkaido, in 1987, 1989, and 1996, respectively. In 1989 he joined NTT Optical Network Systems Laboratories. From 1998 to 1999, he was a visiting researcher at CTR in Columbia University. He is engaged in research on network reliability techniques, network control and management, high-speed computer networks, active networks, network middleware, and the next-generation Internet architecture. For 2003-2004, he is a member of the Board of Directors of the OSGi Alliance. He is a member of the IEEE Communications Society and the Institute of Electronics, Information and Communication Engineers (IEICE).

Hiroyuki Maeomichi

Home Network Service Innovation Project, NTT Cyber Solutions Laboratories.

He received the B.S. and M.S. degrees in physics from Okayama University, Okayama in 1993 and 1995, respectively. He joined NTT in 1995 and has researched middleware and development methods for network operation support systems. Since 2001 he has focused on middleware for QoS management. He is a member of IEICE.