

Standardization Trends of OSGi Technology

*Ryutaro Kawamura[†], Hiroyuki Maeomichi,
Ikuo Yamasaki, and Koya Mori*

Abstract

This article describes OSGi technology that has become an industry standard for middleware used to implement modular software. Software modules, called bundles in the OSGi Alliance, can be delivered to communication equipment over a network for purposes such as adding new services, patching bugs, or tailoring services to individual users. This article provides an overview of OSGi technology and the OSGi Alliance and introduces the latest trends in OSGi specifications as an update to previous NTT Technical Review articles in 2004 and 2005. (OSGi originally stood for open services gateway initiative but now has a wider, more abstract meaning).

1. OSGi technology

OSGi technology is *universal middleware* promoted by the OSGi Alliance* [1]–[3]. It provides a service-oriented, component-based environment for developers and offers standardized ways to manage the software lifecycle. These capabilities greatly increase the value of a wide range of computers and devices that use the Java™ platform. In OSGi technology, the functions of communication equipment are configured from software modules (components) called *bundles* (Fig. 1). In this context, communication equipment includes service gateways, routers, AV (audio/visual) hardware, automobiles, mobile phones, PDAs (personal digital assistants), PCs (personal computers), and network equipment. Bundles perform various functions in this equipment—for example, at a service gateway, software providing home automation may be implemented as a bundle. Bundles are downloaded over a network from a server and are run in suitable communication equipment, thereby facilitating the addition of new services, the tailoring of services to individual users, and the patching of bugs.

An example of this technology being used in an ICT (information and communications technology) household using an OSGi service gateway is shown in Fig. 2. In this example, the OSGi technology is implemented inside the service gateway. The OSGi framework provides an execution environment for bundles, and downloaded bundles are started up and run on this OSGi framework. Typical examples of bundles include one that sounds an alarm when a sensor has detected something and one that sends a message across the network to a mobile phone when the recipient is away from home. Besides existing as individual units, the functions of the service gateway can also be implemented inside equipment such as broadband routers, TVs, and set-top boxes. Multiple

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

* The OSGi Alliance is an open forum, whose mission is to specify, create, advance, and promote an open service platform for the delivery and management of multiple applications and services to all types of networked devices in home, vehicle, mobile, and other environments. The letters OSGi originally meant open services gateway initiative, but today, they represent an abstract attribute label (modifier) for objects associated with the organization. In repositioning its public image to better align with its true mission—the advancement of an open, portable, standardized service platform for cross-industry use—the OSGi Alliance wanted to retain the recognition achieved by the organization since its formation in 1999, while clarifying that its specifications and other work products were applicable for uses, venues, and devices far beyond the domains usually associated with typical gateways.

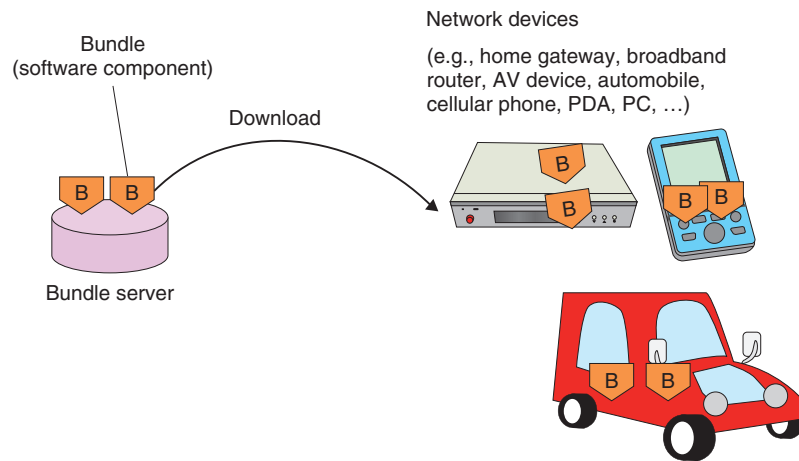


Fig. 1. What is OSGi technology?

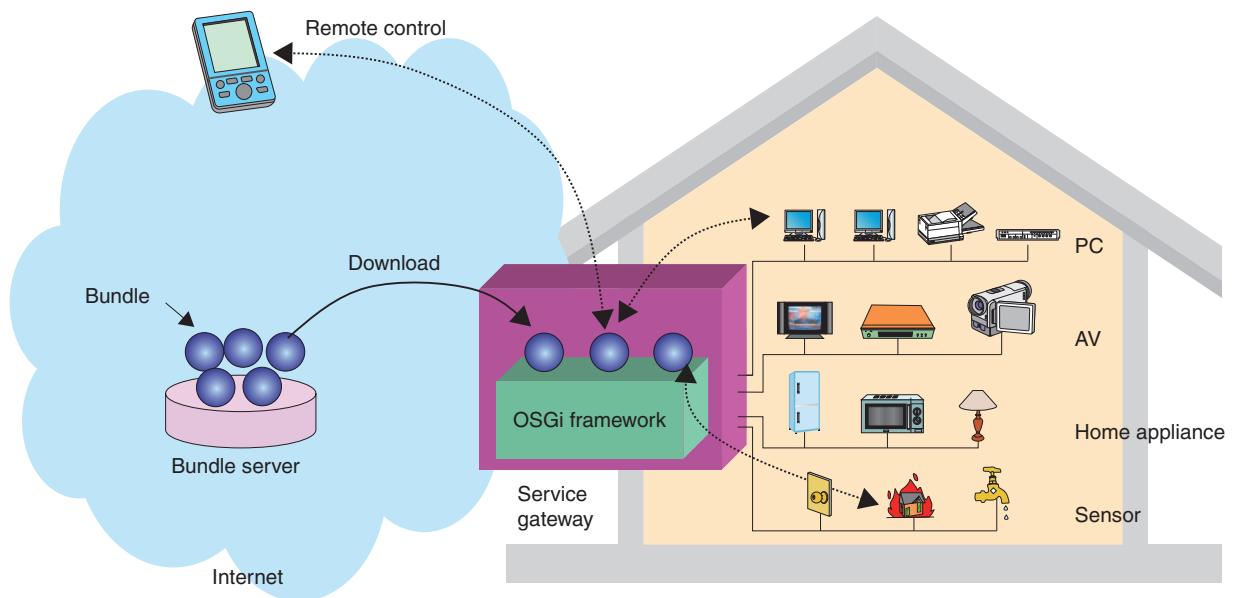


Fig. 2. ICT household using an OSGi service gateway.

OSGi devices can coexist inside the same household. Although an ICT household is given as an example in this article, OSGi technology can be incorporated into other equipment in the same way, and bundles can be used to augment and modify their functions.

2. OSGi configuration and specifications

The structure of OSGi technology is shown in Fig. 3. OSGi technology presumes the availability of the Java language. Java was selected for various reasons, but chiefly because it can be made platform-indepen-

dent. In other words, once a bundle has been developed, it can be run as-is on any target system (including different hardware architectures and operating systems). This reusability of bundles is the most fundamental advantage of software component technologies. It means that bundles can be actively circulated and that new products and services can be developed quickly, either by reusing existing bundles or by procuring bundles from commercial sources.

Bundles can be broadly classified into basic service bundles and application bundles. A basic service bundle is a bundle that conforms to specifications

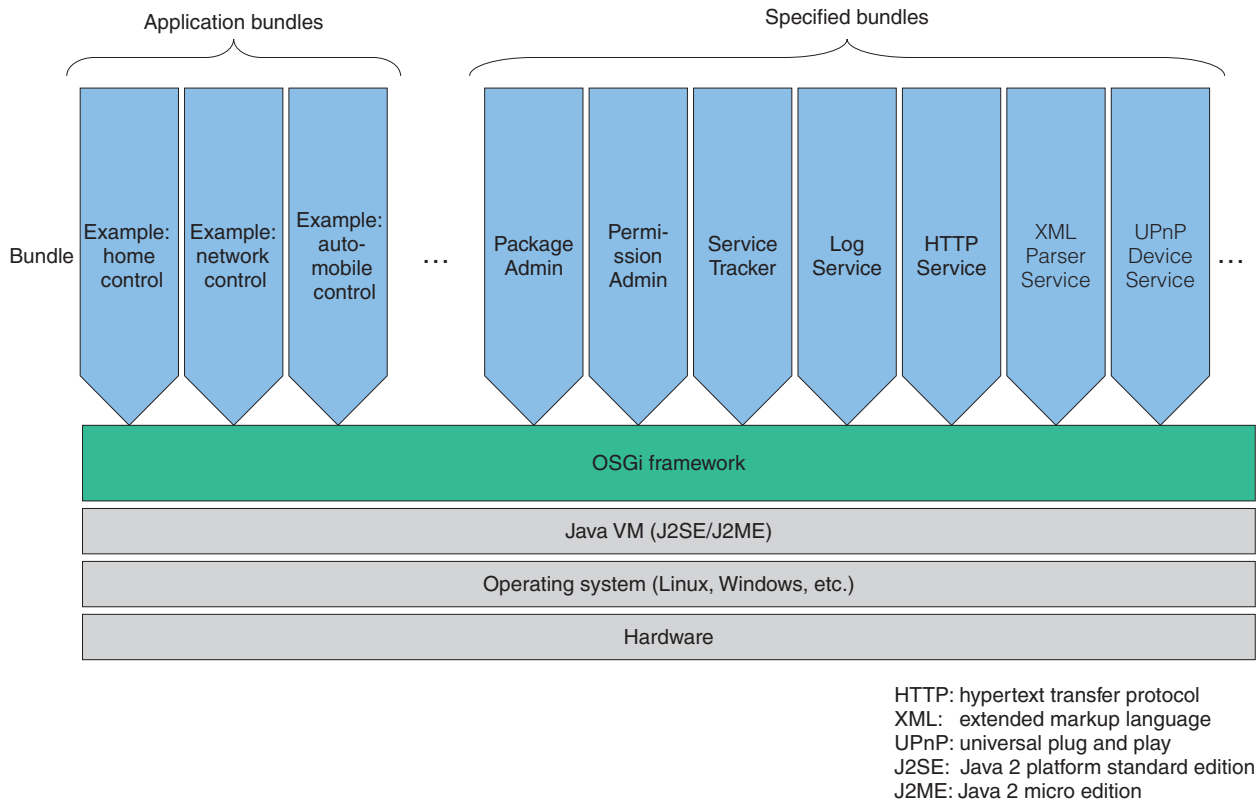


Fig. 3. OSGi structure.

issued by the OSGi Alliance. Although OSGi technology is used in a diverse range of applications, functions that have common user values across these applications are standardized. On the other hand, application bundles are tied to particular businesses and equipment and are not subject to common specifications. The OSGi framework specifies that only the bare minimum of bundles including the basic service bundle needs to be operated. Because the framework parts are minimized and bundles are made optional, the technology can be scaled from the humblest embedded equipment all the way up to large-scale servers. The OSGi framework, which is a functional unit that is preloaded into the OSGi equipment, provides basic functions for the execution of bundles, including the downloading, starting up, and stopping of bundles; bundle interaction services; security mechanisms; and event processing.

The evolution of OSGi technology from release 1 through release 4 is shown in Fig. 4. The process of extending the specifications is performed centered on the work of adding basic service bundles that are useful in new fields to which OSGi technology is applied, while the OSGi framework part has remained largely

unchanged to maintain upward compatibility. To raise awareness of the OSGi specifications, efforts are also being made to have OSGi technology recognized as a standard by the Java Community Process (JCP). To date, release 4 has obtained approval for two specifications: JSR-232 and JSR-291.

3. OSGi Alliance

The OSGi Alliance [1] was established in 1999. It has two classes of members, and there are currently 41 different bodies participating in its activities. It makes its specifications freely available to non-members. Thus, there are many businesses that participate only when the specifications being drawn up are relevant to their own fields of operation. There are currently a relatively large number of members from businesses related to mobile equipment and application servers because these businesses are closely related to topics treated in releases 4 and 5.

Internally, the OSGi Alliance consists of a Board of Directors that manages the organization, Expert Groups that draw up specifications, and Committees that are responsible for handling marketing and com-

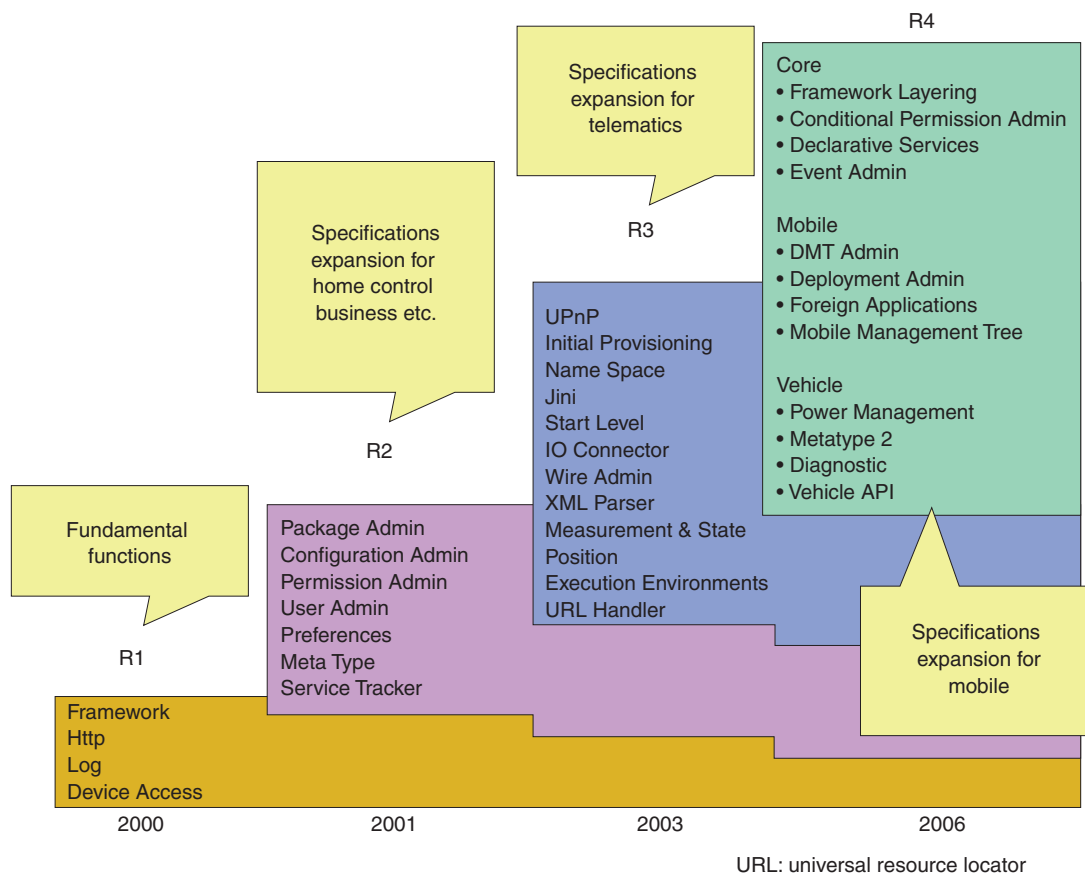


Fig. 4. OSGi specification releases.

mercial requirements. The Expert Groups currently involved in drafting specifications are as follows:

- Core Platform Expert Group
- Vehicle Expert Group
- Mobile Expert Group
- Enterprise Expert Group (newly established)
- Residential Expert Group (newly established)

4. Latest OSGi specifications and open source activities

Most of the work on developing specifications for the forthcoming release 5 is currently being undertaken by the Enterprise Expert Group and the Residential Expert Group.

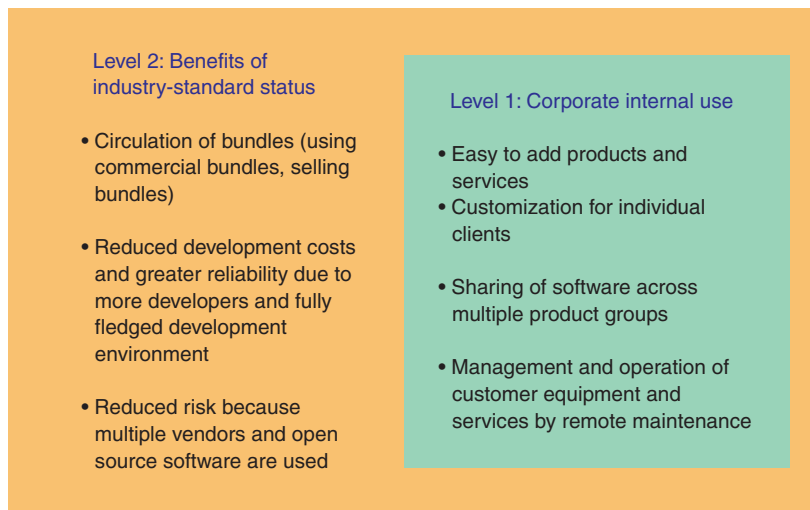
(1) Enterprise Expert Group

This group aims to establish specifications for corporate systems, and its main area of concern is application servers. At present, JavaEE (Java Enterprise Edition) has the largest share of the application server field. The OSGi Alliance's development efforts were originally geared towards end devices, but recently it

has also been recognized that the OSGi component structure mechanism is useful for developing and operating programs on application servers, which has led to it being adopted enthusiastically by the open source community and some businesses. These circumstances started off the process of drawing up specifications at the OSGi. This technology is attracting attention not only for use in servers but also as a form of *rich client technology*, which has been a recent subject of interest. It is of course likely that business-oriented terminals will subsequently be used at the heart of rich client technology in consumer-oriented terminals.

(2) Residential Expert Group

The scope of specifications developed so far has been limited to standard bundles and OSGi frameworks in OSGi-based equipment. However, as OSGi equipment becomes more widespread, there has been an increase in the number of modes of business that require functions for remote management of equipment containing OSGi technology. Examples include businesses that provide services to clients using



OSS: open source software

Fig. 5. Benefits of OSGi technology from a business perspective.

mobile phones or service gateways that incorporate OSGi technology. This situation led to the commencement of specifications related to remote management. The scope of the specification process includes basic management architecture models, protocols for interfacing between management systems and OSGi equipment, and methods for separating service providers when a service is offered by more than one provider. NTT is playing a leading role in this technical field with the OSGi Service Aggregation Platform (OSAP), which is described below, and is playing a central role in standardization.

Like other major software technologies, OSGi technology has been the focus of intense activity in the open source software community. The largest open source project is currently Eclipse [4], which is an integrated development environment (IDE) for Java software and perhaps the most popular Java IDE available today. To meet the diverse needs of software developers, Eclipse has a mechanism that allows various functions to be added as plug-ins. Since release 3.0, this facility has been implemented using OSGi technology. In other words, Eclipse itself consists of an OSGi runtime architecture running various bundles. This has had a major impact above and beyond simply using OSGi technology to implement the IDE. The use of OSGi technology for Eclipse has resulted not only in a massive increase in the number of OSGi developers, but also in the appearance of a full-scale OSGi development environment. Furthermore, by extension, there has been a great deal of

interest in the platform (Equinox) obtained by extracting the OSGi runtime architecture from Eclipse, and in Eclipse RCP (rich client platform), in which Eclipse is used not just as an IDE but also as an RCP runtime environment, which has already been used in many fields.

5. Benefits of OSGi technology and issues to be addressed

The benefits of OSGi technology are shown in **Fig. 5**. Most of these benefits are common to all software component technologies. The starting level is the level of corporate internal use. In cases where a company produces an appreciable number of product groups, this makes it possible to share bundles between different products by using OSGi technology as a unified software platform for these products, with the result that the software productivity and maintainability is greatly increased. This is why it quickly became universally adopted by companies with product lineups that have a high proportion of software [5]. The next level corresponds to the benefits realized from an industry standard. The attainment of industry standard status leads to active circulation of bundles, thereby maximizing the value of software components. For example, the bundle repository of the OSGi Alliance has already registered nearly 2000 bundles. Furthermore, it is thought that OSGi has already entered the level 2 phase.

On the other hand, OSGi is not without issues that

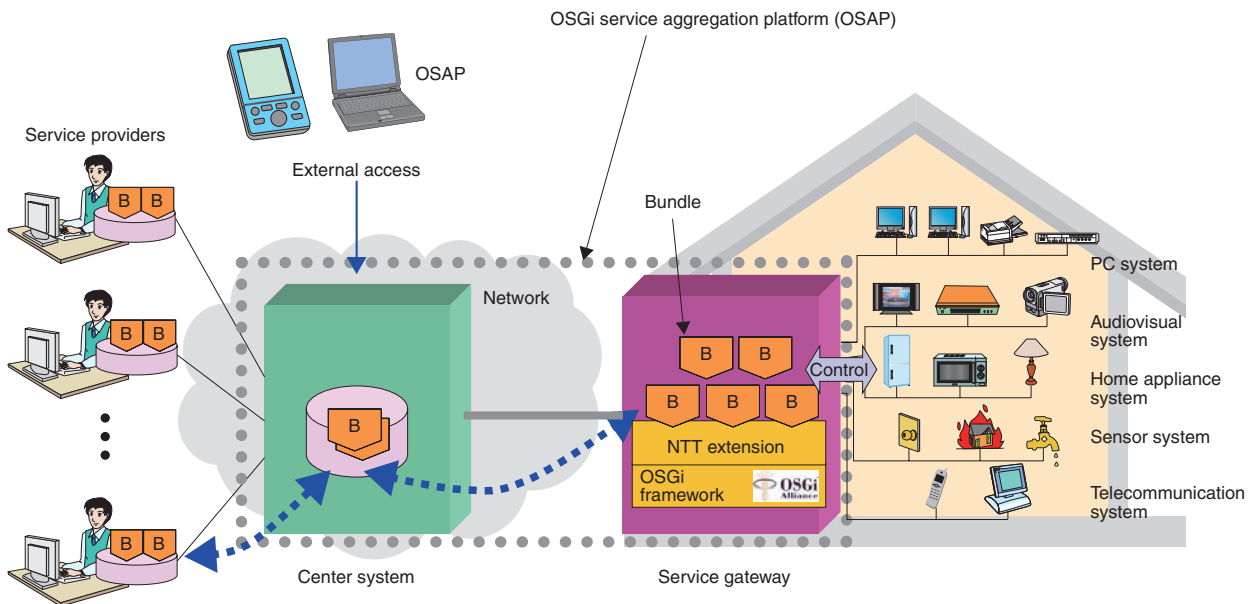


Fig. 6. NTT's OSAP (OSGi service aggregation platform).

need to be addressed. Some problems arise from the fact that—as mentioned above—Java is used to achieve platform independence. One of these is the cost involved in running JavaVM (Java virtual machine). When any software (not just OSGi software) is run on JavaVM, it generally requires more CPU (central processing unit) and memory resources than software written in a language such as C. This is what you might call the downside of achieving platform independence with a virtual machine architecture. But with Java recently becoming the main programming language used in mobile phones, this problem will gradually diminish over the years as intensive efforts result in improved performance, so it will no longer be the main problem.

The other problem is the limitations of the Java language. Since Java is designed to operate in a virtual machine, its specifications are not suited to applications that have strict realtime operation requirements or ones that need to process broadband communication signals. A typical example would be an application that performs video encoding and communication at the same time. Applications of this type would have to be written using native code in a language like C. Although such programs are unsuitable for implementation in Java, the OSGi Alliance distributes this sort of native code in bundles, thereby providing developers with functions that can be used from within the bundles. In such cases, the native code is platform-dependent, but considering the

growth in the OSGi and Java user bases, there are unlikely to be many such applications for which Java is unsuitable [3].

6. Status of OSGi in Japan

Many Japanese businesses, especially those in the consumer electronics field, occupy an important position in that they have a very large effect on the software industry. For this reason, the OSGi Users' Forum Japan [6] was established in September 2004 to promote OSGi as a de facto standard. This organization was the world's first OSGi user group to be authorized by the OSGi Alliance. Subsequently, chapters of the OSGi Users' Forum have also been established for South Korea, France, Spain, Germany, and the USA. OSGi Users' Forum Japan holds regular workshops where members can share information and where the latest technological and business trends are introduced. There are now 74 organizations participating in this forum.

7. NTT's involvement in the OSGi Alliance

NTT has not only been using OSGi technology in communication equipment right from the start, but has also researched and developed a platform that builds upon OSGi technology to allow multiple service providers to provide customers with application services in the form of bundles. This platform is

called the OSGi Service Aggregation Platform (OSAP) (Fig. 6) [7]. If each service provider were to use its own gateway equipment for domestic applications, then people's homes would end up becoming cluttered with different gateways from different providers. It is therefore very important to enable devices to link together. With OSAP, this can be achieved without difficulty, with the added benefit that the service providers can implement their services with little investment. The techniques used in OSAP are currently being standardized by the Residential Expert Group mentioned above. NTT is represented on the board of directors of the OSGi Alliance and provides one of its vice presidents. It is also represented at the managerial and secretarial levels of the OSGi Users' Forum Japan. We are thereby contributing to the adoption of OSGi as a de facto standard. We will

continue to create new value in optical services and mobile telephone services, and we will continue our efforts in R&D and standardization with regard to OSGi technology and OSAP for enterprise business applications.

References

- [1] <http://www.osgi.org>
- [2] R. Kawamura and H. Maeomichi, "Standardization Activity of OSGi (Open Service Gateway Initiative)," NTT Technical Review, Vol. 2, No. 1, pp. 94–97, 2004.
- [3] N. Fujii and R. Kawamura, "Activities of HGI and OSGi Alliance," NTT Technical Review, Vol. 3, No. 9, pp. 72–75, 2005.
- [4] <http://www.eclipse.org>
- [5] "OSGi makes quiet progress," Nikkei Electronics, 26th July 2007.
- [6] <http://www.osgi-ufj.org>
- [7] H. Maeomichi, M. Tamaki, and R. Kawamura, "Service Aggregation Platform for Aggregation of OSGi based Service," Proc. of the 2006 IEICE General Conference, B-7-56, 2006.



Ryutaro Kawamura

Senior Manager, Supervisor, First Promotion Project, NTT Cyber Solutions Laboratories.

He received the B.S. and M.S. degrees in precision engineering and the Ph.D. degree in electronics and information engineering from Hokkaido University, Hokkaido, in 1987, 1989, and 1996, respectively. He joined NTT Transmission Systems Laboratories in 1989. From 1998 to 1999, he was a visiting researcher at CTR in Columbia University, USA. He has been engaged in research on network reliability techniques, network control and management, high-speed computer networks, active networks, network middleware, and next-generation Internet architecture. Since 2003, he has been a member of the Board of Directors of the OSGi Alliance, and since 2005, he has been the Vice President for the Asia-Pacific region. He is a member of the IEEE Communications Society and the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan.



Ikuo Yamasaki

Research Engineer, First Promotion Project, NTT Cyber Solutions Laboratories.

He received the B.S. and M.S. degrees in electronic engineering from the University of Tokyo, Tokyo, in 1996 and 1998, respectively. He joined NTT in 1998. From 2006 to 2007, he was a visiting researcher at IBM Ottawa Software Laboratories, IBM Canada. After initially researching traffic control and management technologies in IP networks, he is now engaged in researching and developing a software deployment and service management system on OSGi service platforms. He is a member of IEICE and a committer of the Equinox Incubator Project of the Eclipse Foundation.



Hiroyuki Maeomichi

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. His research interests include network management system, reusable component programming, aspect-oriented programming, and distributed computing. He is a member of IEICE.



Koya Mori

First Promotion Project, NTT Cyber Solutions Laboratories.

He received the B.S. and M.S. degrees in bioinformatics from Keio University, Kanagawa, in 2002 and 2004, respectively. He joined NTT in 2004 and has researched access control methods for privacy protection. Since 2006, he has focused on Java middleware; in particular, the OSGi service platform. He is a member of IEICE.