Comprehensive Commercialization of Open Source Software

Naoki Uchida, Nobuyuki Kobayashi†, Kenji Fujimoto, Takuo Nishihara, and Hitoshi Shibagaki

Abstract

This article introduces commercialization activities at NTT aimed at promoting greater use of open source software, which is expected to bring great changes to the traditional system development by exploiting the worldwide community of software developers and users.

1. Background

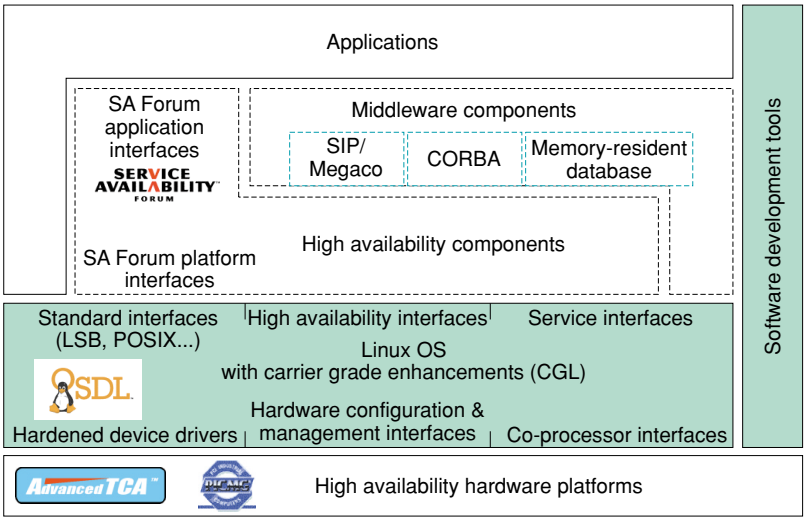
NTT has extensive experience and know-how in software development. In April 2004, it established a comprehensive commercialization system to promote the use of open source software (OSS) in the business operations of the NTT Group. The basic strategy is to eliminate the disadvantages of OSS while exploiting its advantages to the fullest. The comprehensive commercialization system consists of OSS Producers, an OSS Support Center, and an OSS Laboratory [1]. This article discusses OSS use in communication and enterprises services and describes OSS support activities provided by the OSS Support Center.

2. Communication services

The present configuration of open software architecture as commonly recognized by carriers is shown in **Fig. 1**. The hardware block at the bottom of the figure consists of standards-compliant servers (boards) like Compact PCI (cPCI) and Advanced TCA (telecom computing architecture) that are used to provide general-purpose features and reduce costs. The next level up is the operating system, which is Linux with carrier-grade enhancements (Carrier Grade Linux: CGL) [2]. Right above this level is middleware for achieving high-availability functions. This middleware plays an important role in providing communication services—it can quickly detect system faults and switch modules on the fly to provide uninterrupted services. The specifications of high-availability middleware are being studied at the Service Availability Forum (SAF). This scheme therefore uses general-purpose hardware and high-availability functions achieved by software instead of the special (expensive) hardware like switches in redundant configurations used in the past. The end result is a high-reliability system that can be constructed at low cost. Also installed are middleware components like various communication protocols (such as session initiation protocol: SIP) and memory-resident databases that are needed for communication applications. These components can be used to develop service applications.

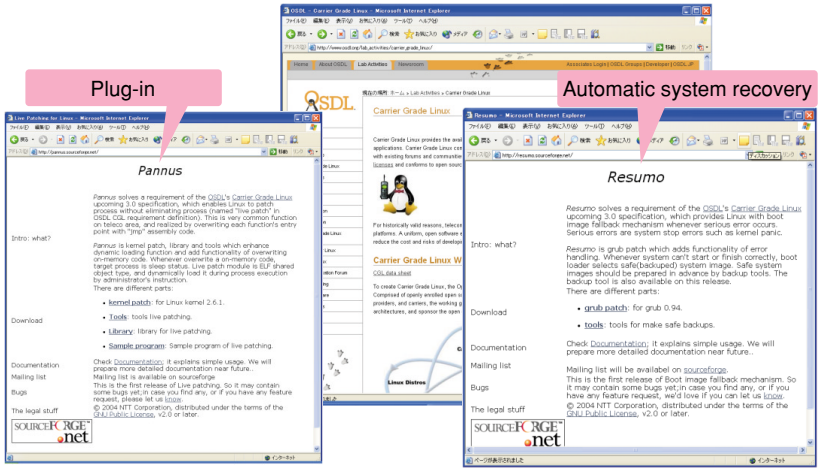
CGL plays an important role in the above framework. Researchers are vigorously studying how to use CGL to construct a high-performance low-cost communication system that is as reliable as traditional switches. There are a number of reasons for using CGL as the operating system. It avoids dependence on specific vendors and gives us more control of source code redistributions by using the GNU General Public License (GPL) rather than FreeBSD and greater openness in adding functions. The most important reason, however, is that CGL makes it easy for carriers to add the functions they need at the operating system level.

† NTT Department III
Chiyoda-ku, Tokyo, 100-8116 Japan
E-mail: oss-pro@hco.ntt.co.jp



Source: http://www.osdl.org with modifications

Fig. 1. Open software architecture.



Source: http://www.osdl.org/lab_activities/carrier_grade_linux/

Fig. 2. Examples of proof of concept (POC).

2.1 CGL-related activities

NTT is actively involved in efforts to incorporate carrier-grade requirements in the Linux kernel [3] as a member in the CGL working group at the Open Source Development Labs (OSDL) [4]. It is proposing specifications for CGL 3.1 to be released in June 2005 and is assembling the requirements for "availability", which is an important feature for carriers, as a section editor for this area. NTT is also setting up within the OSDL some kernel patch projects conforming to these requirements as a proof-of-concept activity and is interfacing with developers throughout the world at the code level. These projects are currently working on a software "live patch" called "Pannus" and an automatic system recovery mechanism called "Resumo" (**Fig. 2**). There are also plans to set up other projects such as one for congestion-control functions.

2.2 Middleware-related activities

In terms of middleware, NTT is participating in the technology working group at SAF and is working to investigate SAF-conforming middleware. In September 2004, NTT held Japan's first joint meeting with SAF and OSDL in Makuhari, Chiba. Through cooperative forums such as these, NTT aims to establish a carrier-grade software platform employing OSS.

3. Enterprise services

Within the NTT Group, various types of systems are being developed in the enterprise-services field. Within individual companies, there are various types of in-house information systems to support communication-related businesses. For corporate businesses as well, various systems ranging from small-scale to large-scale ones are being developed. NTT is involved in the following activities centered on producers and NTT Laboratories with the goal of expanding the use of OSS in these various types of development projects (**Fig. 3**).

3.1 R&D targeting existing OSS

To make use of existing OSS in the most efficient manner possible, its features mostly related to performance and functionality must be exploited and new functions must be developed if necessary. Source code dealing with application servers, database management systems, and operating systems is being analyzed and ways of enhancing functions and raising performance toward expanded use of OSS are being investigated.

3.2 R&D targeting middleware

It would be difficult at present to achieve enterprise services with only existing OSS, so key functions for expanding the use of OSS are being investigated with an emphasis on middleware. Operation support tools and high-availability middleware are two examples of software targeted by this R&D effort.

- **Operation support tools**

One important consideration when constructing a system is that the completed system must be operated and monitored. At present, however, no suitable operation support tools exist for systems configured with OSS products. To therefore promote OSS-based systems that are now attracting attention as a means of reducing system costs, NTT is investigating operation support tools for system operations, monitoring, etc., employing Java Management Extensions (JMX) technology incorporating Java standard specifications. This strategy also considers the spread of Web services and systems using Java 2 Platform, Enterprise Edition (J2EE) in the years to come [5].

- **High-availability middleware**

The provision of Internet-based services 24 hours a day 365 days a year has become standard practice. High-availability middleware is attracting attention as a means of ensuring such continuous operation. This software constructs a redundant configuration made up of active and standby servers. When an active server crashes and can no longer provide services, the service is switched over to a standby server to provide automatic continuation of services. But to raise availability in large-scale services, we need functions that can provide redundancy using fewer standby servers than active ones. Such functions do not exist in currently available OSS. NTT is working

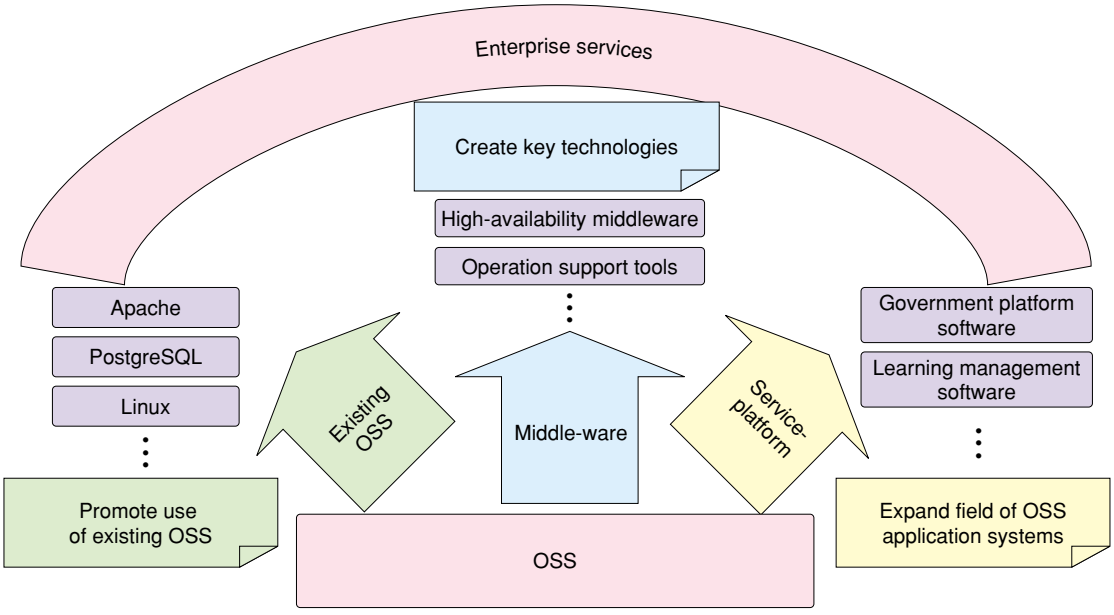


Fig. 3. NTT activities in the enterprise field.

on creating functions of this type in software currently under development and on applying them to OSS database management systems.

3.3 R&D targeting service-platform software using OSS

With the aim of increasing OSS applications in enterprise services, NTT is investigating service-platform software in fields where the use of OSS will be most effective. One example is service-platform software for an OSS version of e-government services. NTT is leading the development of various kinds of government-service systems. E-government service-platform software must be able to interface with a security platform (e.g., one for authentication purposes), provide basic functions such as electronic applications and electronic bidding, and provide diverse government services in a secure and reliable manner with a high level of convenience for both residents and businesses. In promoting the introduction of this software in small- and medium-size local governments, a key requirement is low cost. In recent years, improvements in functionality and stability have raised OSS to the level of commercial products. Against this background, NTT has developed e-government service-platform software using OSS such as Linux, Tomcat, and PostgreSQL. Furthermore, in the field of education, NTT is working to develop learning management systems whose use can be expected to grow through the application of OSS.

4. NTT's OSS Support Center

OSS products can be used in many ways. To clarify their applications, we categorized various OSS products using the Diffusion of Innovations Theory [6] developed by Rogers (**Fig. 4**). According to this theory, there are five steps in the diffusion of innovative products based on the attitude of consumers to those products. Applying these steps to the adoption of OSS, we can categorize adopters as innovators who accept innovative technology if only out of great interest for the possibilities it holds, visionaries who recognize the value of the technology and are eager to see it spread, and early and late majority adopters who contribute to the explosive spread of that technology. Many OSS products lie in the innovators step. And, of course, there are some that belong to the late majority step as is natural with products that have been used successfully for many years. Since a wealth of data on effective use of OSS must have been accumulated about products of this type, we

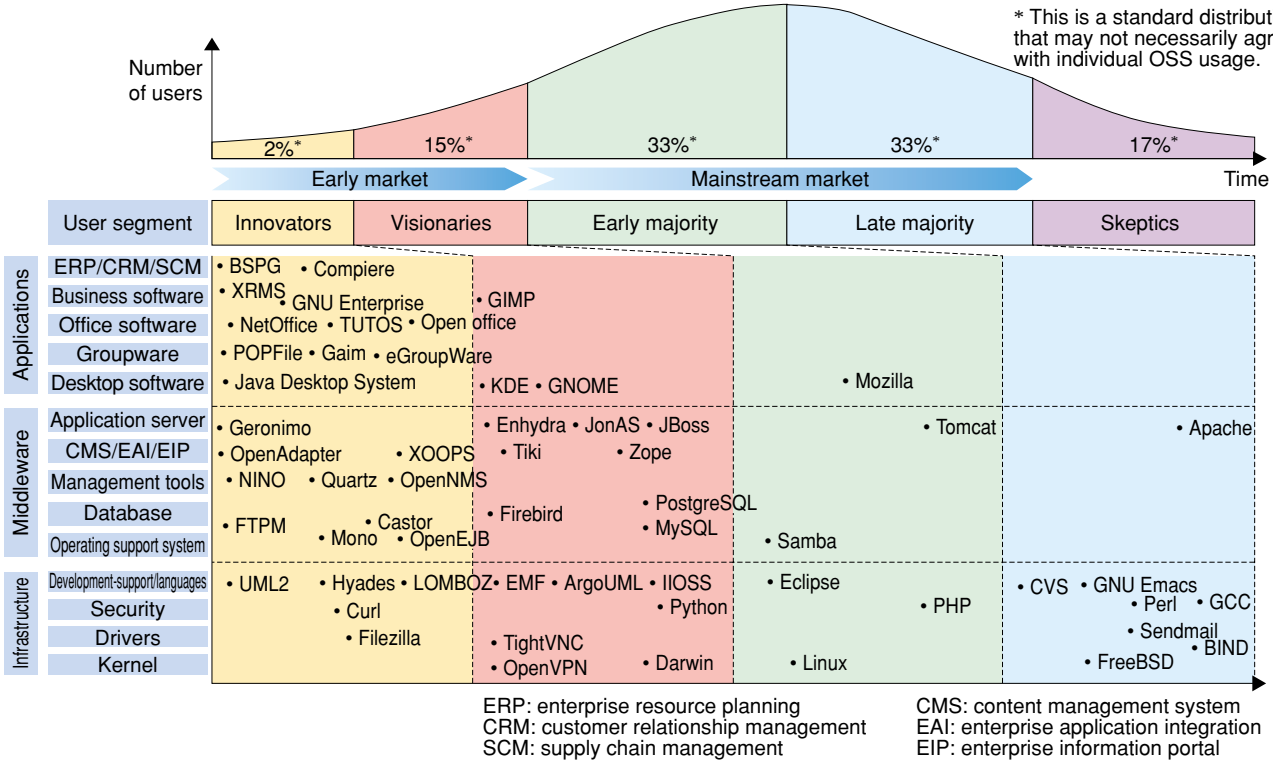


Fig. 4. Life cycle of OSS adoption.

ought to utilize that data to determine how best to use OSS. However, the OSS products that we must concentrate on are those in the visionaries step because the quality of products in this step is considered to be equal to that of existing commercial products in terms of performance and reliability. At the same time, we must check what quality means for the fields that we are interested in. Furthermore, the lack of long-term data about these products means that we need support services to eliminate any trouble that might occur with that software. NTT plans to define and expand support contents in the form of an OSS Support Center under the assumption that OSS products will come to be widely used in business systems. The focus here will be on middleware OSS products, which will be applied in a relatively common-use format and on OSS belonging to the visionaries step.

We examined what kind of services the OSS Support Center should provide in terms of the system development phases. As shown in **Fig. 5**, the required support differs among the phases from system design to construction and operation. In the initial phase, it is important to consider what kind of system needs to be constructed. Answering this question requires quality-related information such as the performance and reliability of individual products and cost-related information with respect to implementation and operation. Another important factor is facilities and an environment for measuring quality. Next, once the system configuration has been decided and we enter the construction phase, an important point is how to assemble people proficient in the technology in question. To this end, it is important to increase the population of developers through training and certification programs. Finally, in the testing and operation phases, fault analysis and recovery measures are needed. Unlike commercial products, OSS products do not have a sufficient lineup of tools, so the lack of resources for performing such work could be a deciding factor for rejecting OSS. In Moore's Chasm Theory [7], there is a huge chasm between the visionaries and early majority steps that must be crossed to make that transition. For OSS, a major factor in crossing that chasm is determining how to provide the services described above in both a high-quality and low-cost manner. Our response to this problem is to select OSS products having the potential to provide those services and build up knowledge in various areas to provide support in the following steps.

(1) Information gathering step: Collect and analyze case studies and operation results and obtain users' needs.
(2) Measurement step for unit products: Measure functionality, performance, and reliability.
(3) Measurement step for pseudo-model applications: Test combinations of products, measure portability and migratability of applications, data, etc., and check operation and maintenance procedures.

The data and know-how obtained through such studies and tests will differ as the hardware environment evolves and products mature. We plan to investigate effective methods of maintaining such data as a future research topic.

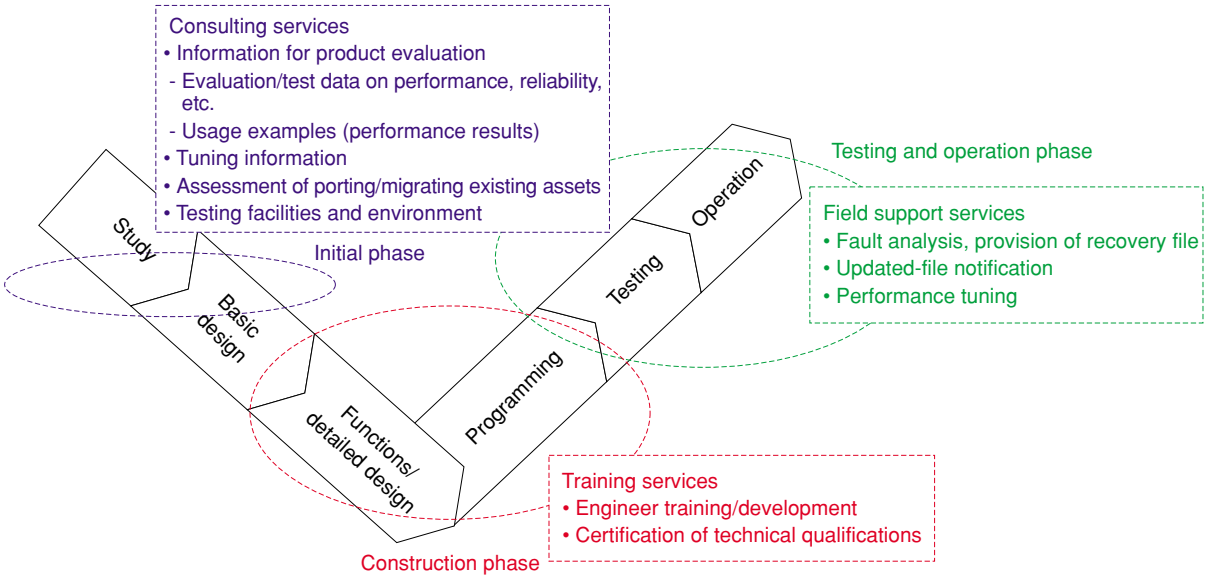


Fig. 5. Relationships among system development phases and support.

References

[1] N. Kobayashi, H. Shibagaki, N. Uchida, and T. Nishihara, “Open Source Software Research Activities in NTT Laboratories,” NTT Technical Review, Vol. 3, No. 2, pp. 12-14, 2005.

[2] Y. Manabe, Y. Kawarasaki, and T. Ikebe, “Linux for Telecommunication Carrier Systems,” NTT Technical Review, Vol. 2, No. 11, pp. 72-74, 2004.

[3] http://www.kernel.org/

[4] http://www.osdl.org/

[5] N. Hayashi, S. Yura, K. Ito, T. Shimma, Y. Nishizawa, and S. Nagai, “Trend of J2EE Servers in Open Source Software,” NTT Technical Review, Vol. 3, No. 2, pp. 20-24, 2005.

[6] “Diffusion of Innovations 5th edition,” E. M. Rogers, The Free Press, New York, 2003, ISBN: 0743222091.

[7] “Crossing the Chasm Revised,” G. A. Moore, HarperBusiness, New York, 2002, ISBN: 0060517123.

Naoki Uchida

Senior Manager, Open Source Software Producer, NTT Department III (R&D Strategy Department).

He received the B.E. and M.E. degrees in electro-communications from the University of Electro-Communications, Chofu, Tokyo in 1983 and 1985, respectively. In 1985, he joined NTT, Tokyo, Japan. Since then, he has been engaged in the development of network services and IP services and research on the next-generation network. He is currently engaged in R&D strategy planning for open source software. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan and a delegate of OSDL and SAF.

Nobuyuki Kobayashi

Senior Research Engineer, Supervisor, Open Source Software Computing Project, NTT Cyber Space Laboratories.

He received the B.E. and M.E. degrees in applied mathematics and physics from Kyoto University, Kyoto in 1987 and 1989, respectively. In 1989, he joined NTT Electrical Communication Laboratories, Kanagawa, Japan. He was engaged in R&D strategy planning for open source software in Department III. In 2005, he moved to his present post. He is a member of the Information Processing Society of Japan (IPSJ).

Kenji Fujimoto

Senior Research Engineer, Open Source Software Computing Project, NTT Cyber Space Laboratories.

He received the B.S. and M.S. degrees in physical science from Chiba University, Chiba in 1986 and 1988, respectively. In 1988, he joined NTT Electrical Communication Laboratories, Kanagawa, Japan. He is a member of IPSJ.

Takuo Nishihara

Executive Manager, Software Architecture Project, NTT Information Sharing Platform Laboratories.

He received the B.S. degree in information science from Kyoto University, Kyoto in 1980. In 1980, he joined the Electrical Communication Laboratories, Nippon Telegraph and Telephone Public Corporation (now NTT), Tokyo, Japan. He is a member of IPSJ.

Hitoshi Shibagaki

Executive Director, Open Source Software Chief Producer, NTT Department III (R&D Strategy Department).

He received the B.E. and M.E. degrees in electrical engineering from the University of Tokyo, Tokyo in 1978 and 1980, respectively. In 1980, he joined the Electrical Communication Laboratories, Nippon Telegraph and Telephone Public Corporation (now NTT), Kanagawa, Japan. He is currently engaged in R&D strategy planning for open source software. He is a member of IEICE and IPSJ.