

## Improving the Efficiency of Application Development Based on the Macchinetta Framework

*Gengo Suzuki, Akira Kanamaru, Takuya Iwatsuka, Junya Katada, Satoshi Okada, Seiichiro Mochida, Katsuyuki Natsukawa, Kenji Motohashi, Takanori Hishiki, Takehiko Kaneko, Kenji Tanabe, Hiroshi Izumoto, Miyotaka Sakai, Katsuyuki Yamashita, and Yosuke Iwaki*

### Abstract

At the NTT Software Innovation Center, we are working to improve the stability and productivity of software development throughout the entire NTT Group by conducting research and development on new software development techniques and making these techniques accessible to developers. This article introduces our work on the Macchinetta framework that we developed in order to produce web-based enterprise applications more efficiently on the Macchinetta tool suite that commonizes development tools, and on the Macchinetta talent pool that provides talented developers to support efficient development.

*Keywords: software development, application framework, OSS*

### 1. Introduction

The NTT Group develops software on a very large scale in order to create a diverse range of enterprise applications (enterprise APs) and systems for the management of information about our networks and customers. Increasing the efficiency of this development work is therefore a very important issue. Very little of our software has been developed entirely from scratch or entirely by manual effort in recent years. Development companies and individual developers strive to improve their productivity by making use of frameworks geared towards efficient software development and development tools that allow software to be designed, produced, and tested more efficiently through the use of automation.

However, if individual developers use different frameworks and development tools, they will develop enterprise APs in different ways with different operating methods. This is problematic because developers have to spend a lot of time familiarizing themselves with frameworks, porting systems to new frameworks, and getting disparate frameworks to cooperate with one another. Even if systems are developed using the same framework, operational problems can still arise if the systems use different versions of the framework. Because the NTT Group develops software on a large scale, there are also cases where each company has developed its own set of tools. The inability to share these tools and know-how is a problem with regard to improving productivity.

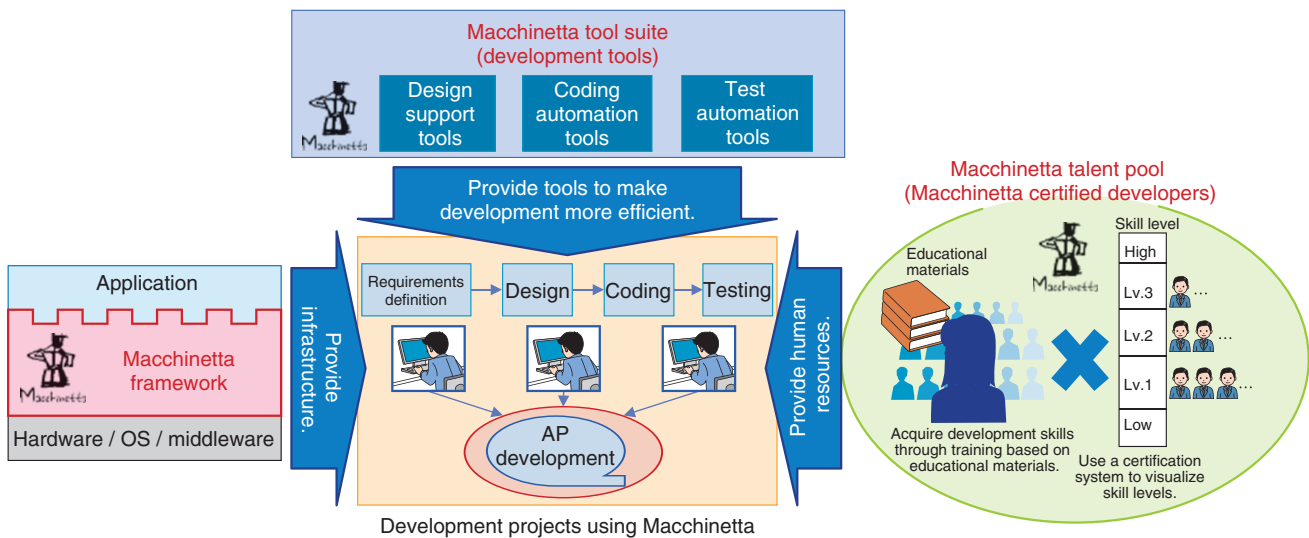


Fig. 1. Efforts to improve the efficiency of application.

To solve these issues, we have developed an application framework to support the development of enterprise APs at NTT Software Innovation Center. Furthermore, to maximize the efficiency of application development based on this framework, we aim to promote the use of a common set of development tools and to cultivate developers who are all skilled in the use of the framework and the tools.

## 2. Setting up an enterprise AP framework: the Macchinetta framework

When developing enterprise APs, it is becoming common practice to use a software infrastructure called an application framework (AP framework). An AP framework consists of a software framework used by software developers to implement the standard structure of an application using a software development environment consisting of elements such as an operating system, programming language, and middleware. The shared functions of an AP framework make it possible to improve the productivity of software development. Also, by developing systems according to established routines that reduce the differences between individual developers, it is possible to achieve uniform quality. Furthermore, the cultivation of developers who are all experienced with this framework facilitates their understanding of these systems so they can maintain them more efficiently.

However, the following issues have arisen when using an AP framework. First, in order to reap the

abovementioned benefits, developers had to learn the established routines for using the AP framework, and had to become familiar with diverse know-how in actual development. In addition, it was necessary to choose from a wide variety of existing frameworks. In particular, even in the NTT Group, there was an issue in that the expansion of similar AP frameworks resulted in duplication of technical development and maintenance costs incurred by each company.

To resolve these issues, reduce the initial training costs, and avoid duplicated maintenance costs, the NTT Software Innovation Center has developed the Macchinetta framework as a new AP framework that can be used in common throughout the NTT Group [1] (Fig. 1). The Macchinetta framework uses globally standard open source software (OSS) as a basic principle, and it can be applied to a wide range of different information systems without being tied to proprietary technologies. The Macchinetta framework consists of an OSS stack recommendation model as a common feature that has been verified to be capable of being used stably in multiple APs, and technical documentation to facilitate the smooth implementation of development using the AP framework (Fig. 2). By expanding the scope to incorporate online processing, batch processing, client development, and the like, we can use the framework for enterprise APs in a wide range of fields.

The Macchinetta framework has been supplied by the NTT Software Innovation Center to NTT Group companies that carry out system development and has

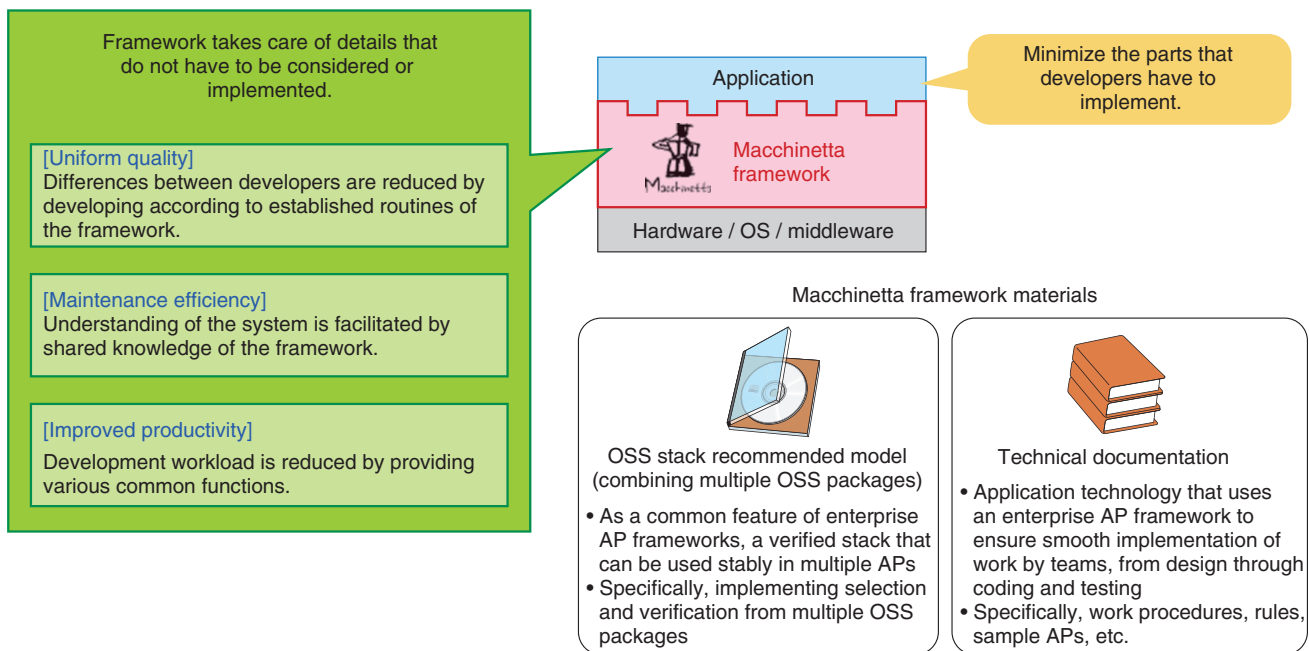


Fig. 2. Overview of Macchinetta framework.

already demonstrated its worth in over 100 projects. In particular, we are not only actively introducing this technology in projects at NTT DATA and NTT COMWARE but are also feeding back knowledge and know-how from these projects in order to promote close cooperative relationships. NTT DATA is deploying the Macchinetta framework with libraries needed for business as the TERASOLUNA Server Framework for Java (5.x) and is applying this framework to large-scale software development in many enterprise systems, which is a major achievement. Also, at NTT COMWARE, we are gaining a lot of experience with applications centered on projects for the NTT Group.

### 3. Commonization of development tools: the Macchinetta tool suite

In modern software development, various development tools are generally used to improve work efficiency and automate as much of the work as possible in the design, coding, and testing processes. These development tools are often developed separately at various companies, resulting in problems such as duplicated functions and compatibility issues. Even if commercially available technology is used, different companies may use different technologies, resulting in larger training costs for the NTT Group as a whole.

We have therefore been working to organize and select suitable development tools to eliminate redundant development efforts and prevent the dispersion of know-how. This selected set of development tools is called the Macchinetta tool suite.

In developing the Macchinetta tool suite, we assessed the effectiveness of commercial tools and existing tools and compiled a list of the development tools that should be adopted. We also compared the functions of these development tools in detail and checked them for duplicated functions. On this basis, we set up the Macchinetta tool suite by deciding whether or not to adopt these tools and continue with their development. An overview of the types of development tools included in Macchinetta and each tool's role in the software development process are presented in **Table 1**. In this way, we were able to prepare development tools to support every step from design to coding and testing, and to make these tools available throughout the NTT Group.

These development tools make it possible to achieve benefits such as improved productivity over a broad range of products, although the characteristics and/or conditions of the products to which they are applied mean that it may not be possible to achieve the desired effect. Therefore, while anticipating advances in development techniques, we will also continue to examine, integrate and develop tools,

Table 1. The Macchinetta tool suite (partial list).

Tool	Target process	Overview of tool
UI design support tool	Design	Integrated development environment that automatically produces source code, settings files, and design documents from UI design information
Automatic code generation tool	Design - coding	Automatic generation of source code from business logic design information
Unit test code generation support tool	Unit testing	Automatic generation of unit test code and test data for source code
Integration test case generation support tool	Integration/system testing	Automatic generation of integration test case list
Continuous integration support tool	Unit - integration/system testing	Automation of repeated build/testing/deploy cycles, support for continuous integration
Source code diagnostic tool	Coding - unit testing	Discovery of memory leaks and other inherent issues in source code
Test execution automation tool	Integration/system testing	Automation of test execution involving the use of UIs
DB access code creation support tool	Coding	Input support for generation of DB access codes

DB: database

UI: user interface

collect and share know-how on the use of these tools among NTT Group companies, and provide feedback through the development of tools that make the greatest possible use of this know-how.

#### 4. Development of human resources to make use of common techniques: the Macchinetta talent pool

The use of a common set of techniques has major benefits in terms of ensuring the availability of human resources. Before these techniques can be used in common, however, it is necessary to have specific technical personnel allocated to each technique. This can make it difficult to ensure that technical personnel are available during busy seasons, resulting in higher development costs, and can also result in development companies becoming entrenched in the use of particular techniques. However, if common development techniques such as frameworks and tools are provided as an infrastructure as described above, then it should be possible to reduce the cost of technical education and make it even easier to secure the necessary human resources. The environment for the education of talented individuals with this common set of skills is called the Macchinetta talent pool. We are working to support the cultivation of this sort of talent pool.

As a part of this initiative, we are working to prepare educational materials relating to these common techniques. An effective way of cultivating high quality developers is to provide them with hands-on

access to the frameworks and other technologies in order to learn by building things. However, developers currently have to learn by extracting the required information from large amounts of information such as ordinary commercial primers and web pages, which is a time-consuming and inefficient process. Also, the number of developers assigned to a single development project may run into the hundreds, and training them is no easy task. Therefore, to solve this problem and improve the efficiency of our human resources, we prepared educational materials for the Macchinetta framework.

The Macchinetta educational materials are targeted at beginners with just a basic understanding of Java and are constructed in order to provide a combination of basic instruction and hands-on experience with programming exercises. These materials pick out commonly used functions and arrange them for maximum educational benefit in order to provide a plain description of the recommended implementation methods and rules, thereby facilitating rapid learning. By preparing a hand-made selection of exercises and solutions, we made it possible to gain practical knowledge through simulated development experience. This made it easy to train the human resources needed for Macchinetta (**Fig. 3**). These educational materials are made available to every company in the NTT Group and are used as educational programs for companies involved with development projects using the Macchinetta framework, thereby contributing to a large increase in the efficiency of education.

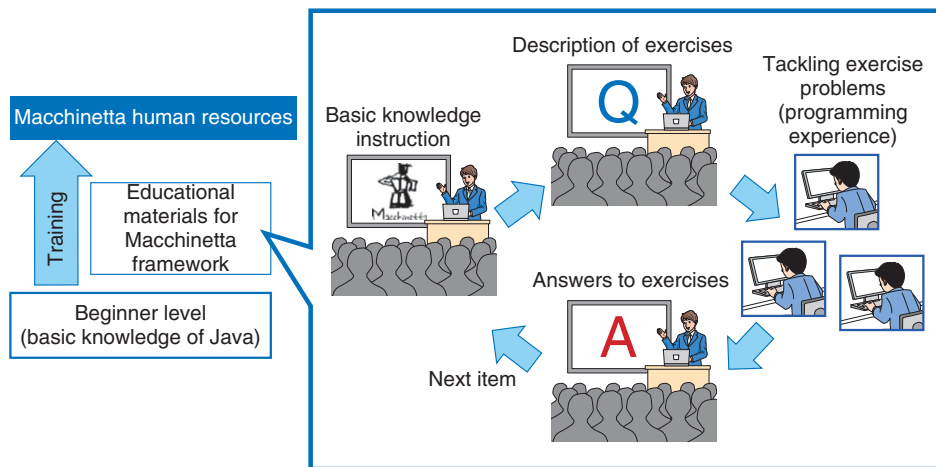


Fig. 3. Educational materials for Macchinetta framework.

### 5. Future prospects

The environment surrounding software development is evolving every day. By taking advantage of the system infrastructure offered by cloud services, it is becoming unnecessary these days for developers to build their own system infrastructure. However, it is necessary to adapt to a wide variety of client environments with diverse user environments, terminals, operating systems, and browsers. Furthermore, it is expected that application technology will be developed for technical areas with a higher level of difficulty than system development. We therefore aim to continue with research and development (R&D) aimed at expanding the applicable range and operating environment of frameworks and tool suites in order to achieve higher productivity. Also, by modifying these environments, we will further elevate their importance and the opportunities for porting systems to new frameworks and environments. We believe that it will become important to prepare procedures and tools for improving the productivity of migration.

The Macchinetta infrastructure consists of various

OSS packages, but in order to provide higher-quality support and improve the reliability of its frameworks and tools, we are actively contributing to the improvement of these OSS packages. In addition to improving quality by providing patches to fix bugs, we are also conducting R&D together with the OSS community to provide new functionality. In particular, we plan to work together with the other NTT Group companies to actively contribute to Spring [2], which is an essential part of the OSS framework.

To cultivate human resources, we are investigating the preparation and operation of a Macchinetta developer certification system. This certification system will make it possible to clearly visualize developers and their skill levels, thereby facilitating the marshaling of human resources.

### References

- [1] "Development of an OSS Enterprise AP Framework Aimed at Improving Software Development Efficiency and Productivity," Business Communication, Vol. 51, No. 9, pp. 4-7, 2014 (in Japanese).
- [2] NTT DATA Corporation, "Spring Primer: Developing Java Applications in the Spring Framework," Shoeisha, 2016 (in Japanese).



#### Gengo Suzuki

Senior Research Engineer, Software Engineering Project, NTT Software Innovation Center.

He received a B.S. and M.S. in mathematics from Tohoku University, Miyagi, in 1988 and 1990, and a Ph.D. in computer science from the University of Electro-Communications, Tokyo, in 2013. He joined NTT in 1990. His research interests include database design, multi-database systems, graph data management, and software engineering. He is a member of the Information Processing Society of Japan and the Institute of Electronics, Information and Communication Engineers.



#### Seiichiro Mochida

Senior Research Engineer, Software Engineering Project, NTT Software Innovation Center.

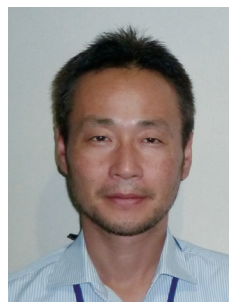
He received a B.E. in science and engineering from Waseda University, Tokyo, in 2002 and an M.E. in engineering science from the University of Tokyo in 2004. He joined NTT in 2004. His interests include software engineering.



#### Akira Kanamaru

Senior Research Engineer, Supervisor, Software Engineering Project, NTT Software Innovation Center.

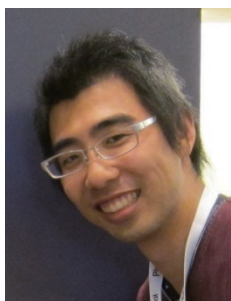
He received his B.E., M.E., and Ph.D. from Tohoku University, Miyagi, in 1996, 1998, and 2001. He joined NTT in 2001. His research interests include computer networks and software engineering.



#### Katsuyuki Natsukawa

Senior Research Engineer, Supervisor, Software Engineering Project, NTT Software Innovation Center.

He received an M.E. from Nara Institute of Science and Technology in 1996. He joined NTT in 1996. His current research interests include software engineering.



#### Takuya Iwatsuka

Researcher, Software Engineering Project, NTT Software Innovation Center.

He received a B.E. and M.E. in computer science from Nagoya University, Aichi, in 2011 and 2013. He joined NTT in 2013. His research interests include programming language theory, program verification, and software engineering.



#### Kenji Motohashi

Manager, System Engineering Headquarters, NTT DATA Corporation.

He received a B.E. in computer science and an M.E. in communications and integrated systems from Tokyo Institute of Technology in 2003 and 2005. He joined NTT DATA in 2005. He is engaged in developing thin client systems, cloud computing systems, and application frameworks.



#### Junya Katada

Researcher, Software Engineering Project, NTT Software Innovation Center.

He received a B.E. and M.E. in computer science from Tokyo Institute of Technology in 2003 and 2005. He joined NTT COMWARE in 2005. He moved to NTT in 2016. His interests include software engineering.



#### Takanori Hishiki

Manager, System Engineering Headquarters, NTT DATA Corporation.

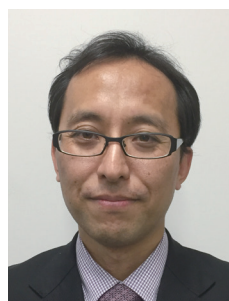
He received a B.E. in computer science and an M.E. in communications and integrated systems from Tokyo Institute of Technology in 2003 and 2005. He joined NTT DATA in 2005. He is involved in developing thin client systems, cloud computing systems, and application frameworks.



#### Satoshi Okada

Senior Research Engineer, Software Engineering Project, NTT Software Innovation Center.

He received a B.E. and M.E. from Toyohashi University of Technology, Aichi, in 1993 and 1995. His current research interests include software engineering.



#### Takehiko Kaneko

Research Manager, Center for Applied Software Engineering, Production Engineering Department, System Engineering Headquarters, NTT DATA Corporation.

He received an M.E. in industrial engineering from Aoyama Gakuin University, Tokyo, in 1997. He joined the Public System Division of NTT DATA in 1997 and developed a large-scale trading system. He was also involved in developing automated tools. He is currently in charge of R&D and deployment of automated tools including the TERASOLUNA Suite.



**Kenji Tanabe**

Manager, System Engineering Headquarters, NTT DATA Corporation.

He received a B.E. in electrical engineering from Gunma University in 1988. He joined NTT DATA in 1989. His current work involves technical strategies and public relations in the field of software engineering.



**Katsuyuki Yamashita**

Manager, SE Department in Core Technology Quality Management and Engineering Division, NTT COMWARE CORPORATION.

He received a B.E. and M.E. in communication engineering from Osaka University in 1997 and 1999. He joined NTT COMWARE in 1999. His main interest is improving the efficiency of software development.



**Hiroshi Izumoto**

Manager, System Engineering Headquarters, NTT DATA Corporation.

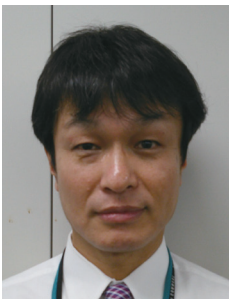
He received a B.E. and M.E. in information engineering from Hiroshima University in 1986 and 1988. He joined NTT DATA in 1989. He is currently engaged in developing technical strategies and working in public relations in the field of software engineering.



**Yosuke Iwaki**

Manager, SE Department in Core Technology Quality Management and Engineering Division, NTT COMWARE CORPORATION.

He joined NTT in 1990. He has contributed to the development of the software development environment. His main interest lies in the engineering of software development tools.



**Miyotaka Sakai**

Manager, SE Department in Core Technology Quality Management and Engineering Division, NTT COMWARE CORPORATION.

He joined NTT in 1985. His main interest is studying best practices for software development projects in order to improve software development methods.