

## TUBAME: A Tool for Studying Java Application Migration

*Masato Urano, Kyohei Nakamura, and Masahiro Nagafusa*

### Abstract

The NTT Open Source Software Center is developing a tool that automatically identifies sections of applications needing revision and computes the cost of making the revisions when migrating a Java application, and has made it available as open source software (OSS). This tool reduces the amount of manual work required to perform the migration and supports a quantitative study of the effects of migration on Java applications. This article gives an overview of the tool and its strengths and illustrates its effectiveness through an application example.

*Keywords: OSS, migration, TUBAME*

### 1. Introduction

The NTT Open Source Software (OSS) Center is actively promoting migration from commercial application servers (AP servers) such as WebLogic to open source software (OSS) AP servers such as JBoss<sup>\*1</sup> [1] in order to reduce total cost of ownership (TCO). Various processes are needed to perform a migration, but the first process, the preliminary investigation, is essential for the overall project to run smoothly (**Fig. 1**). This is because the existing Java application operating on the commercial AP server uses an application programming interface (API) specific to that AP server, and configuration files are defined in a way specific to that AP server, so these sections of applications must be revised during the migration. After investigating the number and location of such sections, the results are used in performing the coding work. This investigation must be done methodically and involves quite a lot of work when done manually, so the probability of human error (omissions and errors) increases as the number of these sections increases. This has resulted in increasing demand for a tool that automatically investigates dependencies, such as selecting sections for revision and computing the revision costs.

The NTT OSS Center provides support services for

this sort of migration. The service includes summarizing the sections in the source code and configuration files of an existing Java application that needs revision when performing a migration and estimating the cost of making the revisions. We also systematically summarize the knowledge accumulated from supporting projects in documents and share them internally as guidelines, which have increased the efficiency of this work. Using these guidelines to study a migration has enabled us to conduct more accurate preliminary investigations. Because of this, we began developing a tool that incorporates functions for both dependency analysis and degree of difficulty analysis that uses our knowledge base.

### 2. The TUBAME migration support tool

The TUBAME Java application migration support tool was developed based on the background described above. This tool reduces the amount of work involved in migrating an existing Java application to another AP server by automatically extracting and presenting information regarding the cost of revisions needed and how the revisions need to be made. It also confirms

<sup>\*1</sup> JBoss: An OSS Java Platform Enterprise Edition (JavaEE) application server.

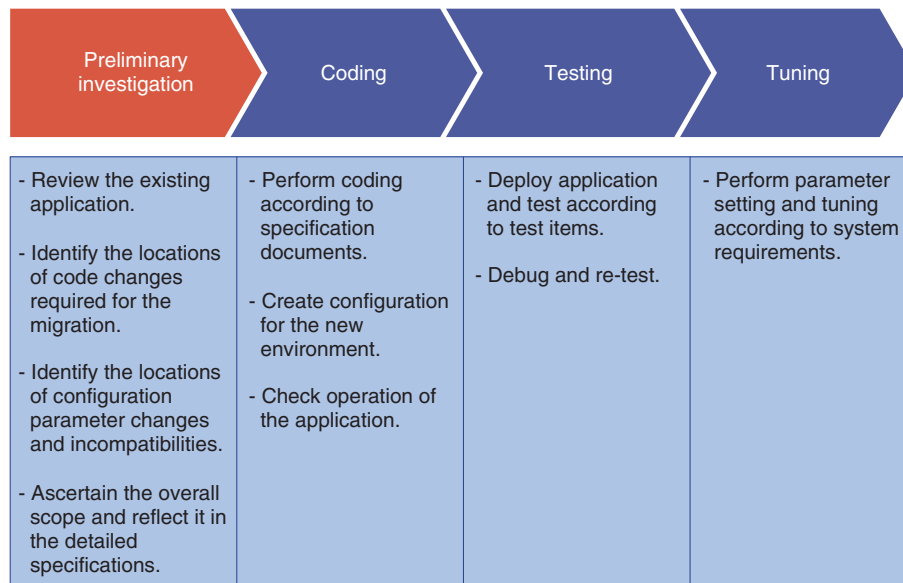


Fig. 1. Migration process.

how revisions should be made for each section to be revised, so the actual work of porting the Java application is more efficient. The tool has been made available as OSS on a community site [2] on GitHub<sup>\*2</sup> since October 2013.

### 3. Operation overview

Here, we describe the operation of TUBAME.

#### 3.1 Overall organization

TUBAME has two main components: the dependency analysis tool and the knowledge search tool (Fig. 2). Each tool can be used independently, and both can be run to check the effects of migrating a Java application. The knowledge search tool also incorporates a knowledge management tool, which provides an intuitive GUI (graphical user interface) enabling users to compile the knowledge necessary for a search.

#### 3.2 Dependency analysis tool organization

This tool takes the application source code and configuration files as input and automatically extracts the Java packages and classes that cannot be used on the AP server after migration and displays the results (Fig. 3). By ascertaining the propensity to use code that is strongly dependent on the current AP server in this way, the user can examine the degree of difficulty

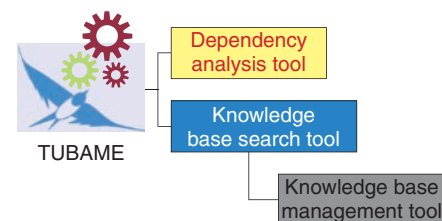


Fig. 2. Overall architecture of TUBAME.

and scope for revising the Java application in concrete terms. The tool can also analyze the effects of multiple migrations such as to different AP servers and to a different framework at the same time. For example, when the AP server is being migrated to JBoss, and the application framework is being upgraded to Struts<sup>\*3</sup> [3], decisions can be made taking the overall effects into consideration. If the scope of revisions for the framework is greater than expected, priority could be given to the JBoss migration first, and the framework upgrade could be deferred to the next development cycle. The dependency results mentioned earlier can also be saved as comma

\*2 GitHub: A web sharing service for software development projects.

\*3 Struts: A Web application development framework using the Java programming language.

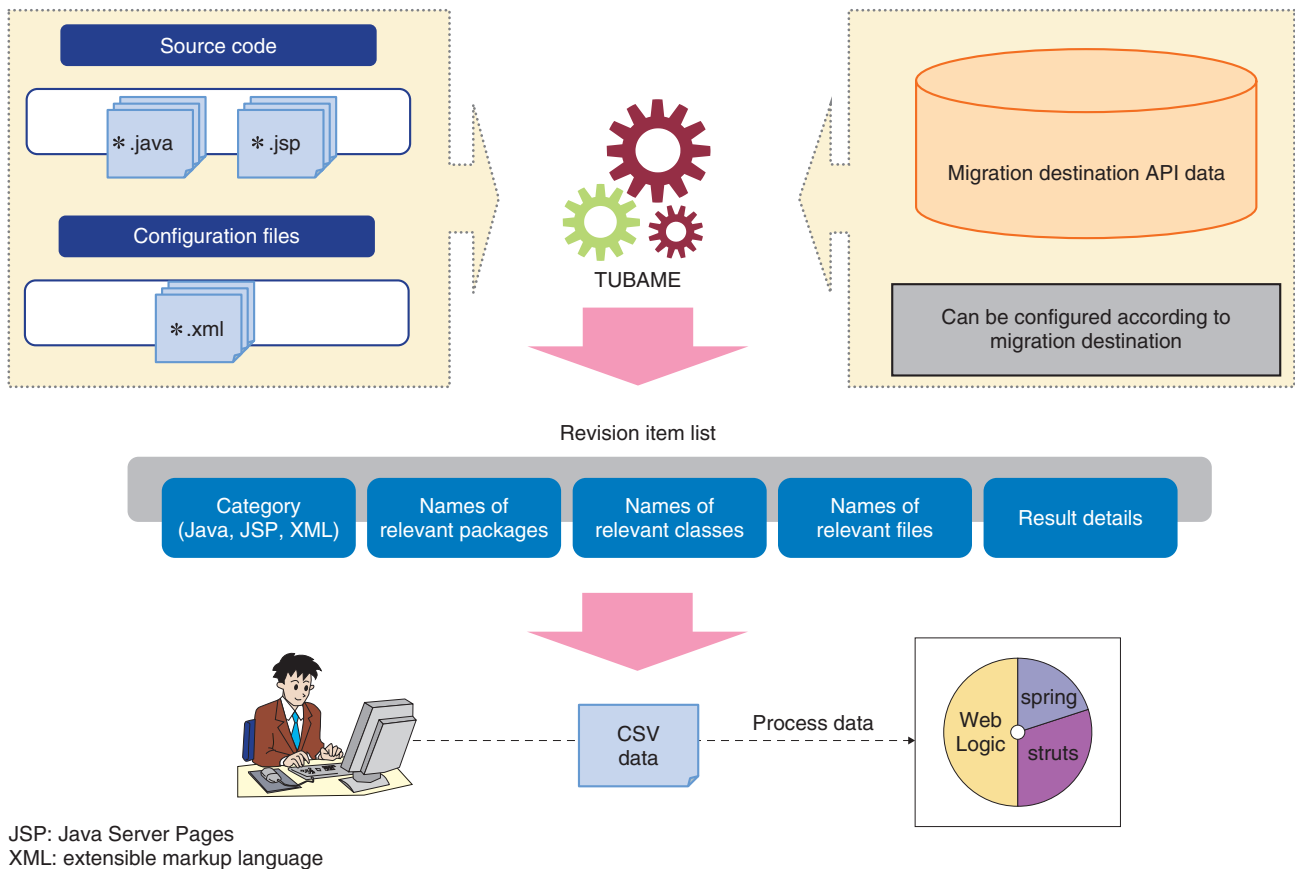


Fig. 3. Organization of dependency analysis tool.

separated value (CSV) data, making it easier to create graphs and other reports showing the proportion of Java packages or classes needing revision.

### 3.3 Knowledge-base search tool organization

This tool extracts sections for revision based on migration knowledge, which cannot be extracted using the dependency analysis tool described above, and presents revision methods (Fig. 4). For example, the formats of configuration files for different AP servers can be very different, so it is important to know beforehand whether a similar configuration method will be available after the migration. If it will, the porting will be easier, but if not, possible alternative approaches must be studied, which may make the porting more difficult. Thus, it is necessary to know the actual method to be used and to compute the cost. The knowledge base is composed of such search methods and keywords for finding the sections needing revision. The tool uses this knowledge to extract the sections requiring revision from source code and

configuration files automatically and presents the level of difficulty and the cost of revisions for each section. As with the dependency analysis tool discussed earlier, this information can be saved as CSV data, so data on sections needing revision can be aggregated, and the total cost of revisions can be estimated.

## 4. TUBAME features

A major strength of TUBAME is its generality; the tool and the knowledge are independent, so users can create their own knowledge and customize it appropriately. Moreover, it can support users who would like to study a variety of migrations. For example, the configuration and other aspects of different versions of JBoss can vary quite a bit. In one case, when an application that worked with an earlier JBoss version was migrated to the latest version, the application worked without major revisions but did not deliver the performance of the earlier version under heavy

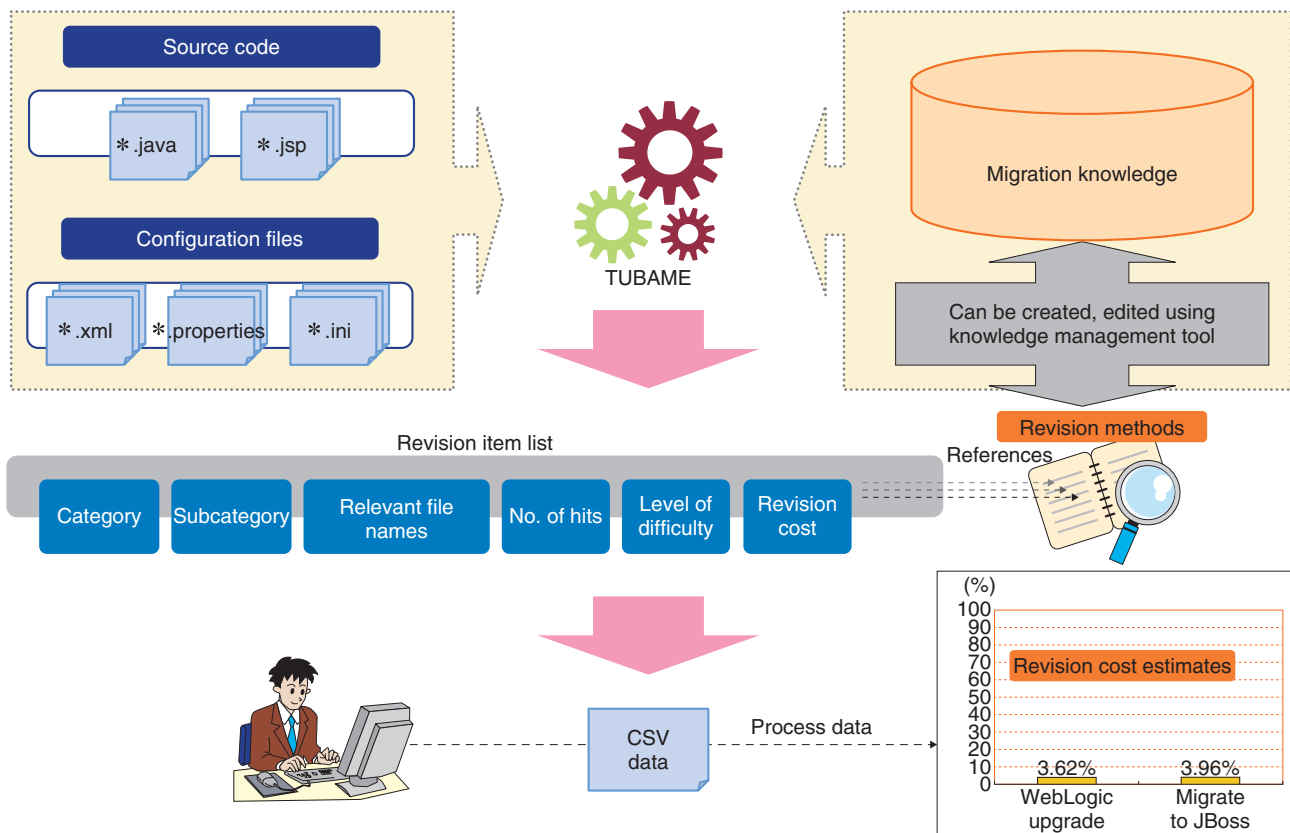


Fig. 4. Organization of knowledge base search tool.

load. This was because the default size of the instance pool\*<sup>4</sup> had changed. This provided a lesson that testing can go more quickly if default values are set to those of the previous version. We are adding such lessons to our knowledge regarding upgrades (migration) to new JBoss versions.

We have also added mechanisms that store API data used with the dependency analysis tool in the past, which can be reused when investigating similar AP server migrations in the future, so that the knowledge base search tool can check the sections needing revision and the methods for making the revisions.

### 5. Target users and usage

We expect that TUBAME will be used for projects involving the actual migration or possible migration of Java applications. It operates as a plug-in for Eclipse [4], an integrated development environment widely used around the world for development of Java applications, so it can be installed easily, even on compact, lightweight work machines.

### 6. Example application and effects

We present here the results of comparing the amount of work that is necessary to investigate a migration using TUBAME with that using conventional manual techniques (Fig. 5). A comparison of the work done by the NTT OSS Center for migration support services using the tool and conventional techniques is shown in Fig. 5 and shows an average reduction of approximately 40% by introducing the tool. Currently, the tool has been used to investigate more than ten migrations. These projects vary in the amount of source code by a factor of approximately 20 from the smallest to the largest. When working manually, the amount of time required is proportional to the scale of the source code, but with the tool, revision sections of even a relatively large Java application of over two million lines of code can be extracted in less than three hours, and this effect increases as the scale increases (Fig. 6). The tool can also be used

\*<sup>4</sup> Instance pool: An entity that pools a given data set in memory.

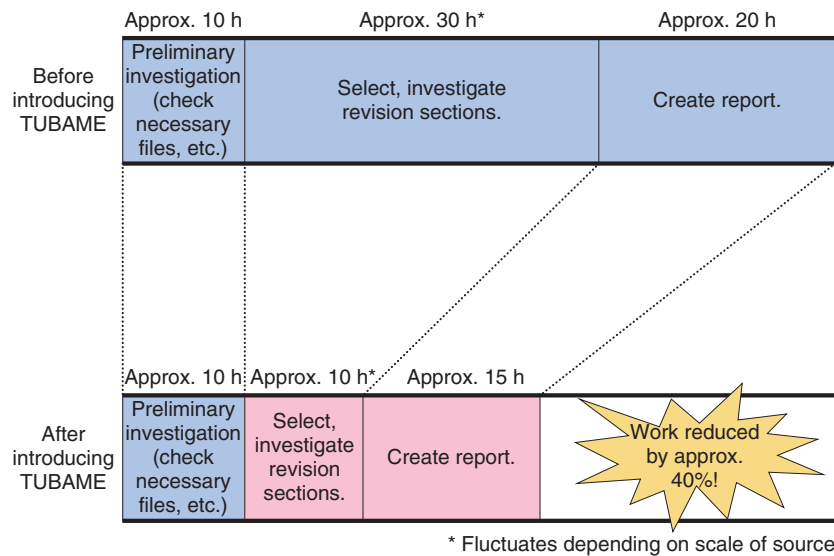


Fig. 5. Amount of work required using TUBAME and conventional techniques.

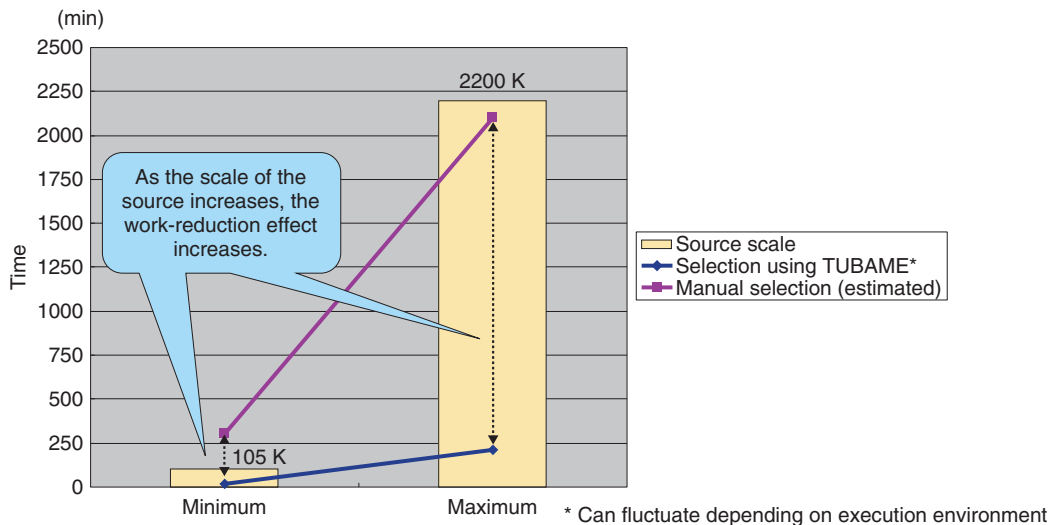


Fig. 6. Relation between revision ratio and selection time.

by people who are not migration experts and enables them to partition the work; thus, they can conduct a rough analysis first and have a specialist perform a more detailed analysis later.

### 7. Community activity

Currently in the TUBAME community, two knowledge bases that can be downloaded and used with the tool have been published: one for migrating from

WebLogic to JBoss and another for upgrading JBoss. We plan to continue to update the migration knowledge mainly for migration to JBoss OSS. We can also receive requests to add or revise knowledge discovered by users in this community. We will investigate such user requests and periodically update the knowledge to increase its accuracy. We are also considering expanding the knowledge to add information regarding migration to JBoss from commercial AP servers besides WebLogic (WebSphere etc.) to our current

knowledge, in cooperation with engineers in the JBoss community.

## 8. Future development

The NTT OSS Center intends to continue to provide support in the migration of applications to OSS AP servers by promoting this tool to those in development projects who are migrating, or considering migrating, Java applications, and by actively expanding our migration support service. As of December 2013, we had had more than 1000 accesses from more than ten countries, so we are attracting substantial interest. With such knowledge provision and feedback from around the world, we hope to enrich

and improve a wide body of knowledge and develop it as a tool that can dramatically reduce the costs of migration from various commercial AP servers to OSS AP servers and migration between different versions of AP servers.

## References

- [1] NTT OSS Center (in Japanese).  
<http://www.oss.ecl.ntt.co.jp/oss/techno/jboss.html>
- [2] TUBAME, GitHub.  
<https://github.com/TUBAME>
- [3] Struts, The Apache Software Foundation.  
<http://struts.apache.org/>
- [4] Eclipse, the Eclipse Foundation.  
<http://www.eclipse.org/>



**Masato Urano**

Senior Expert, Application Software Technology Unit, NTT Open Source Software Center.

He received the M.E. in industrial engineering from Yokohama National University in Yokohama, in 2000. He joined NTT in 2011. He provides JBoss support to NTT Group companies and develops tools related to JBoss such as TUBAME.



**Masahiro Nagafusa**

Leader, Application Software Technology Unit, NTT Open Source Software Center.

He received the B.L.S. from the University of Library and Information Science (currently, the School of Informatics at the University of Tsukuba, Ibaraki), in 1989. He joined the NTT Group in 1989. He manages the support operations of more than 15 web/OSS products including JBoss.



**Kyohei Nakamura**

Expert, Application Software Technology Unit, NTT Open Source Software Center.

He received the BSIT in computer science from Chiba Institute of Technology in Chiba, in 2010. He joined NTT in 2013. He is the developer of TUBAME and provides Tomcat and JBossEAP support for NTT Group companies.