

From Labor Intensive to Knowledge Intensive—Realizing True Shift to Upper Phases of Development with TERASOLUNA Simulator

Takehiko Kaneko

Abstract

As described in the previous article, NTT DATA is promoting TERASOLUNA Suite as an automation tool product suite for system development and is achieving a significant reduction in workload in in-house projects. However, testing still requires a lot of work and man-hours. Thus, a lot of challenges remain to improve productivity. One solution being developed to meet these challenges is TERASOLUNA Simulator, which can reduce the time needed for the entire integration testing phase.

Keywords: software development automation, review assistance, testing workload reduction

1. Introduction

A software development process that uses a coding automation tool always generates design-based source codes automatically. We do not need to verify whether the generated source codes are actually functioning according to the design because the functioning of the automatically generated source codes is guaranteed by the source code generator. However, there is a possibility that the operator may enter the wrong design information if the design document itself was created based on incorrect business specifications. Thus, verifying the accuracy of the design through actual operation of the program becomes the main objective of integration testing when an automatic source code generation tool is applied. In other words, if the design itself is confirmed to be accurate, the integration testing can be omitted.

2. Difficulty in verifying design accuracy

A reviewer analyzing the design review process carries out the following tasks.

(1) Visualizes the necessary data variations in his/

her mind

- (2) Simulates how the program would work if the data of (1) were actually entered
- (3) Checks whether the simulation result is the same as the one expected

If the complexity of the process is within the reviewer's capability, the reviewer can foresee enough variations in mind and can generally secure the quality of the design information. However, if the process is too complex for the reviewer to handle, the reviewer may not be able to anticipate all of the possible variations (**Fig. 1**). As a result, the reviewer cannot secure sufficient design quality. This often results in design mistakes (slippage) in the design phase. Thus, integration testing is necessary to detect the design mistakes.

3. TERASOLUNA Simulator: Review support tool

If design information is complicated, it is generally difficult to predict sufficient data variations ((1) in the review steps above) and to simulate the application of the predicted data variations the reviewer visualized

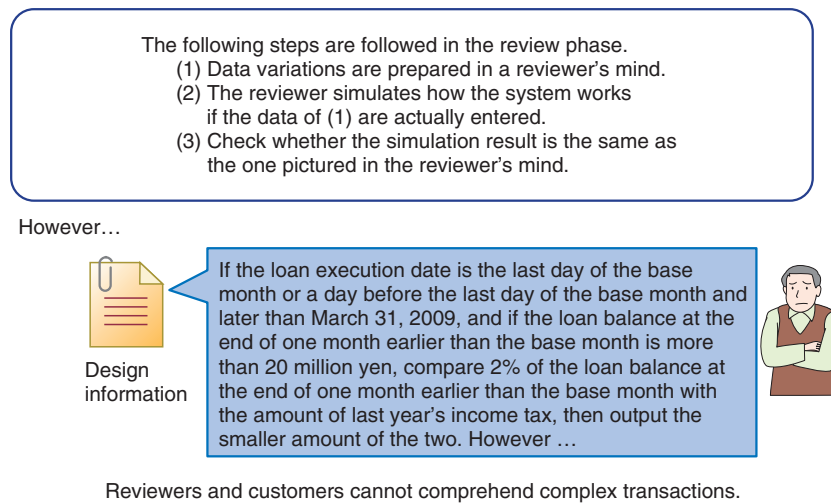


Fig. 1. Why can't errors in design information be fully detected?

((2) in the review steps above). TERASOLUNA Simulator can help to appropriately check the design quality of complex design information in the design phase because it can perform a thorough check by automatically executing processes (1) and (2). More specifically, TERASOLUNA Simulator automatically generates the necessary data variations and simulates the applications of the data variations. The reviewer can check the accuracy of the design by simply verifying the simulation result (the following report) with the expected result ((3) in the review steps above) (See **Fig. 2**). Note that some non-function aspects such as usability and performance cannot be checked.

3.1 Data variations

Data variations generated by TERASOLUNA Simulator are input values for business logic. Data variations need to be information that enables verification of the design of business logic. For example, to verify a data validation check process, we need variations of data that cause errors and that end the process without errors. We also need data variations that can check the thresholds of the design conditions. Data variations are used to verify the validity of the design by checking the process results.

TERASOLUNA Suite can automatically generate data variations from design information by extracting the checking, branching, computing, and editing processes from design information that is provided in logical form. However, there are many challenges in automatic generation of data variations; these are

described later.

3.2 Simulation

Simulation is a process that enables us to visualize the expected execution results after feeding the above data variations into business logic. The word *simulate*, which forms part of the tool's name, usually means pseudo execution. However, this tool applies data variations to actual program source codes (business logic) instead of performing pseudo execution. The reason for this is that there is always some doubt that the results of pseudo execution may be different from the execution results with actual program source codes. Note that it is generally difficult to run an actual program because program source codes are not ready in the design phase. However, it is possible to run a process using actual program source codes with TERASOLUNA Suite because TERASOLUNA Suite generates program source codes that can create expected results based on the design. In the future, TERASOLUNA Simulator will be able to simulate screen transitions once the research and development (R&D) is completed, which will require a few more years.

3.3 Reports

A report represents a comprehensible form of information of system input and its corresponding output. It is essential to have a report form that can be easily understood by human operators. This is because the purpose of a design review is to verify the accuracy of the design, and the verification decision depends

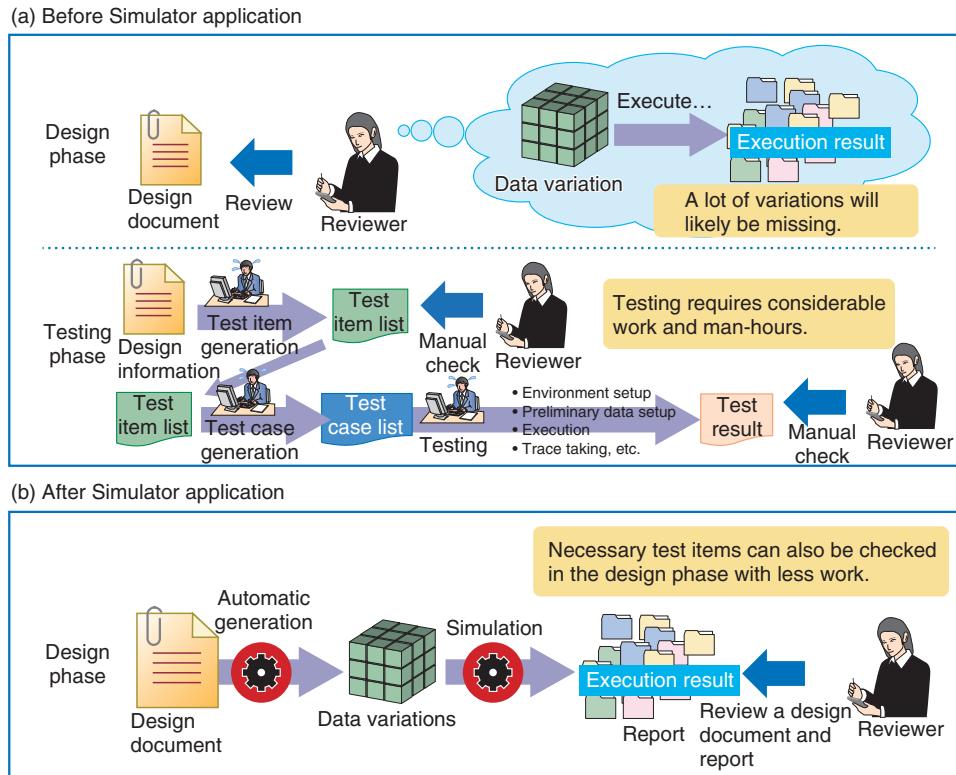


Fig. 2. TERASOLUNA Simulator overview.

solely on the reviewer. Therefore, a report needs to be in a form that is comprehensible to the reviewer, who will make a decision based on the report.

4. Expected effects of TERASOLUNA Simulator

The effects of using TERASOLUNA Simulator in different processes are outlined in this section.

4.1 Effects in design process

- The time to prepare data variations and simulate process executions is reduced.
- The quality of reviews is improved because reviews can be performed by checking execution results.
- Man-hours for rework are reduced because possible errors expected during execution can be identified in the design phase.
- Extra work for reviewing reports is needed.

4.2 Effects in testing process

- Man-hours required in the testing phase are

expected to be reduced because data variations are checked on reports and do not need to be re-checked in the testing phase.

4.3 Summary

A summary of the overall effects indicates that although the man-hours needed for reviewing reports in the design phase increase, the time required to prepare data variations and to simulate processes decreases. Thus, the total man-hours will not change. However, the testing work in the testing phase is expected to be drastically reduced, and therefore, the time needed for process integration support and functions integration testing is also expected to be drastically reduced.

5. Challenges

The following challenges remain in the ongoing R&D of TERASOLUNA Simulator.

5.1 Challenge 1: Generation of data variations

Data variations are generated based on design

Table 1. Challenges in generating data variations.

Item attributes	(int/String/double..., null, <u>cast</u> , ...)
Data structures	(<u>array</u> , <u>list</u> , <u>structure</u> , ...)
Variables	(variable, constant, environment variable, <u>session information</u> , ...)
Control structures	(branch, <u>loop</u> , ...)
Various processes	(checking, editing, calculation, call, form editing, ...)
String operations	(<u>partial reference</u> , concatenation, number of characters, ...)
External information	(<u>DB</u> , <u>file</u> , ...)
Java dependent	(Exception, annotation, inheritance, class, ...)

information; consequently, if the design information is not correctly analyzed, the expected data variations cannot be generated. Typical challenges in this area are listed in **Table 1**. In this case, program components are used for convenience because it is difficult to define necessary patterns if we use design information for a natural language. The underlined components represent difficult components. Loop, DB (database), and, unexpectedly, cast descriptions are difficult. In addition, although it is not included in the table, it is also difficult to output appropriate boundary value data from design information.

One solution is to combine independent methods. Current approaches to dealing with this challenge mostly target program source codes. Therefore, one of the important factors that makes our approach more feasible is that the target areas can be limited because our approach is based on TERASOLUNA Suite.

5.2 Challenge 2: Report forms

Reports need to be in a form that can be easily understood by reviewers and customers. Although the levels of understanding vary from person to person, the general goals are to create reports in which (1) the volume of information is not unnecessarily large and is limited to a size that contains necessary information only, and (2) the relationship between data variations is described in a meaningful context in order for people to identify design errors and shortcomings. Specific examples of (2) include clear descriptions for checking the purpose of each input data variation and correct sorting of a report sequence and categories for checking a report sequence.

In addition, TERASOLUNA Simulator is expected to be used in two scenarios: the reviewers' review process and customers' verification process. Therefore, the actual reports that are required will differ depending on each scenario. Reviewers require a

report form that enables them to thoroughly check all data variations. In contrast, customers need an easily comprehensible report form that enables them to quickly review important points.

5.3 Challenge 3: Volume and quality of data variations

Another challenge is that there is no clear definition of data variations required to verify system quality (**Fig. 3**). TERASOLUNA Simulator is a tool for verifying a system concept visualized in a reviewer's or customer's mind with the design prepared for that concept. However, there is no clear answer that explains how many data variations are required for verification. In our case, it is difficult to review many data variations. Thus, balancing volume and quality requirements is a difficult issue.

Moreover, one type of variation that cannot be generated by TERASOLUNA Simulator for quality verification is a variation for a process that is not defined in the design information due to initial design shortcomings. Variations that are not defined in the design information need to be identified during the review process.

5.4 Challenge 4: Very long processing time

Our approach is expected to take a very long time to completely execute all processes. Thus, reducing the processing time is an important issue. There are three possibilities to achieve this: (1) speeding up the process by improving the logic; (2) speeding up the process by using high-performance computing; and (3) making a tool that requires intervention of human knowledge instead of pursuing a fully automated tool. Approach (3) is expected to be the key for solving realistic problems.

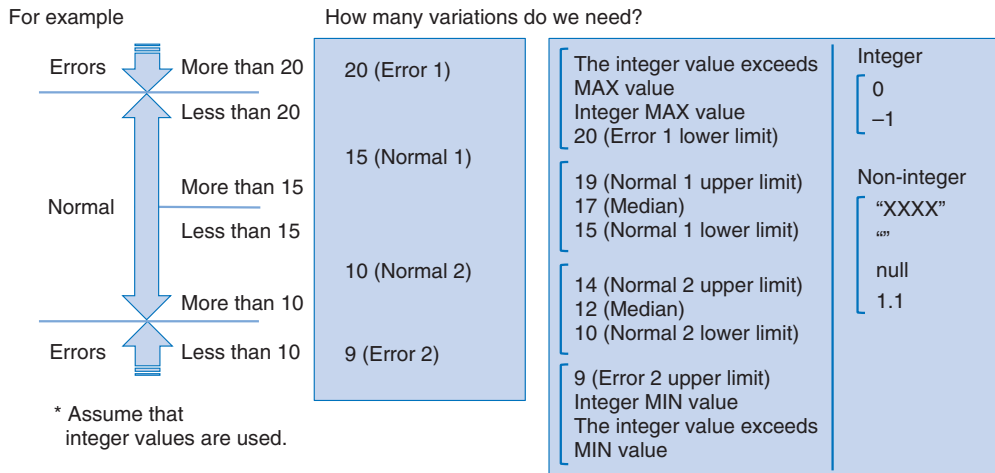


Fig. 3. Data variations required to establish quality.

6. Future outlook

Full-scale development of TERASOLUNA Simulator began in fiscal year 2014. The basic components

will be completed by the end of the fiscal year, and the tool will be elevated to a practical level within fiscal year 2015.



Takehiko Kaneko

Research Manager, Center for Applied Software Engineering, Research and Development Headquarters, NTT DATA Corporation.

He received the Master of Industrial Engineering, Aoyama Gakuin University, Tokyo, in 1997. He joined NTT DATA in 1997 and was assigned to the Public System Division, where he developed a large-scale trade-related system. He also worked on the development of automated tools. He was transferred to his current position in 2010, where he is in charge of R&D and deployment of automated tools, including TERASOLUNA Suite.