Development of Software Testing Techniques Based on Extended Formal Models

Tomohiko Takagi (Associate Professor)
Faculty of Engineering and Design, Kagawa University

Software testing is an important process to evaluate and improve the quality of software. MBT (Model-Based Testing), that is, the technique to generate test cases based on formal models that represent the abstracted expected behavior of software, is effective for testing large and complex software systematically. In this laboratory, we develop new techniques based on formal models that focus on the change of the state and data of software, in order to improve the effectiveness of MBT.

(1) Development of Extended Formal Models

In MBT, test cases are generated from formal models. Therefore, it is important that (*a*) formal models can exactly represent the aspects of software that should be tested and that (*b*) they can be interpreted and executed by computers, in order to generate high-quality test cases. For example, in order to enable generation of test cases that can improve software reliability effectively, we develop the construction techniques of extended formal models in which software metrics that is related to use and failure probability on each functional element is introduced. Also, in order to enable verification of essential internal processing and constraints of software, we develop the construction techniques of extended formal models in which other formal specification description languages are introduced, and the techniques to convert the extended formal models to other formal specifications.

(2) Development of Test Coverage Criteria and Test Case Priority Evaluation Techniques

In MBT, test cases are usually generated so as to satisfy a test coverage criterion that is specified by test engineers. For example, we develop new criteria that are aimed at covering the executable sequences of state transitions or combinations of data that meet the specified conditions on an extended formal model or an intermediate model (a model that is generated from the extended formal model in order to generate test cases effectively). Also, in order to detect serious failures (for example, failures that have high probability of occurring in actual operational environments, or that make big impacts on stakeholders when they have occurred) in an earlier stage of testing, we develop techniques to evaluate the priority of test cases based on software metrics.

(3) Development of Test Case Generation Techniques

Test cases that satisfy a specified test coverage criterion in a smaller size and also are ordered by priority are greatly to be desired, since the effort that test engineers can spend on test processes is finite. However, it is not easy to generate such test cases from extended formal models that have large search spaces and complex constraints. Therefore, we develop techniques to generate intermediate models suitable for test case generation, and techniques to generate optimized test cases from extended formal models or their intermediate models at reasonable cost by applying metaheuristics. Also, we develop techniques to evaluate and improve the effectiveness of test cases based on simulated failure detection ratio and degree of impact on software quality before test case execution.

(4) Development of Test Engineer Support Techniques

Test engineers need to construct extended formal models based on various software development documents, and will require training in the constructing skills. For example, we develop techniques to support the training by visualizing the execution results of extended formal models under construction and giving hints to complete the construction. Also, test engineers usually need to deal with great amount of data in MBT, and therefore we develop management and visualization techniques based on extended formal models to easily grasp the tendencies of software quality and the state of work progress. We develop prototype tools to support (1)-(4) in order to not only evaluate their effectiveness but also discuss the effective support manners by tools.