| | |
|---|---|
| Title | モデルベースによる体系的テスト駆動開発環境の研究 |
| Author(s) | 北，篤 |
| Citation | |
| Issue Date | 2010-03 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/8947 |
| Rights | |
| Description | Supervisor:DEFAGO Xavier，情報科学研究科，修士 |

# A Systematic Model-based Environment for Test-driven development

Atsushi Kita (0810017)

School of Information Science,
Japan Advanced Institute of Science and Technology

2010 年 2 月 9 日

**Keywords:** Test-driven development, Testing view, Configuration management,Change impact analysis.

Among various software development approaches, test-driven development is one of those that attract attention. It is a code-based software development approach, and its benefits are development of the software that surely executed, early detection of degraded software, and refining development process. However, test-driven development has the following problems;

**problem1** Developers tend to develop test cases ad-hoc

**problem2** When requirements change, it is difficult to understand which test case is changed

Problem 1 is caused because test-driven approach is biased by developers' stand points. Developers want to test against their intentions but often lack quality assurance views; as a result test cases tend to be ad-hoc.

Problem 2 is caused because test-driven approach is code-centric. In this development style, source code and test cases are primary artifact and it is difficult to understand the relationship among requirements and test cases.

To solve the above-mentioned problems, we propose a model-driven environment that supports systematic development of test cases from multiple test views and also supports management of artifacts and related information for test-driven development. The environment has the following three functions:

**function1** it supports test case design from various view points

**function2** it systematically manages artifacts and information for test-driven development

**function3** it supports tracing the effect of requirement changes

Here, function1 is for problem1. Function2 and function3 are for problem2.

To realize function1, we propose a testing views template based on Nishi's testing view. This template hierarchically shows the testing views which are known to be important when we test software. This template supports software developers to design test cases systematically without forgetting important testing views.

To realize function2 we extend existing UML models to be able to manage related artifacts and information; we define the extended model in terms of meta model.

To realize function3, we propose a mechanism to generate trace information from models and also show how to analyze the impact of requirement changes. Based on our proposal, software developers identify test cases that are affected by requirement changes.

Moreover, we develop the proposed environment based on meta model we define, and apply it to simple calculator example.

Concretely speaking, we design test cases based on various testing view by using our template, manage an information in the test-driven testing systematically in our defined figures, and, generate a trace information from managed information.

We also analyze impacts caused by requirement changes utilizing these trace information. In this sample example, we identify test cases that potentially needs modification automatically.

We demonstrate that our proposed environment has capability to manage artifacts and information used for test-driven development along with trace information used for impact analysis.