# Development of Formal Model Based Test
# – Toward Automatic Testing Framework for Embedded Systems –

Takahiro ANDO*, Shinji KAWASAKI*, Eun Hye CHOI*, Hideaki NISHIHARA*,
Masahiro AOKI†, Keiichi YOSHISAKA†, Keisuke SHIMATANI†, Munekazu FURUICHI†,
and Osamu MIZUNO§
**National Institute of Advanced Industrial Science and Technology*
†*Daikin Industries, Ltd.*
§*Kyoto Institute of Technology*

This presentation proposes a testing framework called "*Formal Model Based Test (FMBT)*". It is a framework to improve embedded system developments with formal specifications, exhaustive test generations, and automatic test executions. Formal descriptions make specifications consistent, and exhaustive test generations avoid an oversight of tests. Thus, we expect a reliable test process can be obtained by FMBT.

**PROBLEM DEFINITION:** We recognize a problem in current functional tests for an entire system, which rely on individual efforts. When the tests are generated by hand from specifications written in a natural language, their quality depends on test engineers' experiences and intuitions. That might cause a lack of necessary tests.Moreover, manpowered testing might cause oversights or mistakes of tests.

**SOLUTIONS:** We attempt to minimize manpowered tasks in testing processes and to improve them by using the FMBT. Test engineers will execute testing processes in the FMBT as follows: (1) Write specifications of a developing system in a formal language. (2) Generate tests exhaustively from the formal specifications. (3) Execute tests automatically and exhaustively.

This way has the following effects: First, describing the specifications in formal removes ambiguities and makes the gist of the designers clear. Next, we can obtain the tests independent of test engineers' skills. Moreover, we can avoid oversights and mistakes by executing automatically tests exhaustively.

We show our activities for building the FMBT framework. First, we have designed and developed a formal specification language[1] for test generation, called "*Sens*". The specifications in the Sens description are constraints for state transitions of the systems. We can verify the consistency of the specifications and detect defects of them, and thus we can improve the quality of the specifications.

The important feature of Sens language is that the domain of variables can be limited for the test generation, independent of specifications. It makes clear designer's focus for test. This supports a test-oriented design and an efficient test generation from Sens specifications.

Second, we are developing a test generator which generates tests exhaustively from Sens specifications. The generated tests satisfy the specifications, and do not include lack of necessary tests.

Our test generator uses a SAT solver as its core engine, which solves satisfiability problems at high speed. With a SAT solver, our generator can find distinct and enormous test quickly. In our feasibility study for a real air-conditioning system, it succeeded in generating about 900,000 tests in 30 minutes.

Testing embedded systems is generally performed on actual hardware. In addition, the various environments of the system executions are considered in the testing. In many industrial companies, a test engineer executes tests and checks the results by hand. This traditional way takes heavy cost and time, and thus cannot give sufficient test executions, as well as exhaustive tests.

**TOWARD PRACTICAL APPLICATIONS:** To overcome the deficiency, we propose to adopt automatic test executions by "*Cluster-In-the-Loop Simulation (CILS)*" in our framework. CILS is a technique that simulates a whole embedded system including hardware components, network devices, and environment, in a cluster computing system. In CILS, software applications can be executed and tested on hardware simulation environment even before hardware is completed. In the future, we will integrate our test generator and the CILS system, to develop the automatic test environment for embedded systems.

---

[1] http://cfv.jp/cvs/introduction/pdf/PS2011-002.pdf