MoMuT::UML
UNIFIED MODELING LANGUAGE

THE CHALLENGE

Software has become an integral part of most technical systems and often is the major contributor of value. Put differently, software never has been more important than today. Take a commodity like a car as an example. In 1981 car manufacturers were using microprocessor-based engine controls running about 50,000 lines of code (Charette, 2009). Today, a premium-class vehicle will contain upwards of 100 million lines of software code. The big challenge manufacturers are facing is to make this code as safe and fault free as possible. Accepting this challenge turns out to be difficult as it is known that average code will contain 15 to 50 faults per 1,000 lines delivered (McConnell, 2004).

A SOLUTION

To mitigate the threat posed by low software quality to business, software testing is generally applied and will improve the error rate considerably – provided it is done in a correct way. Keeping the increasing product complexity in mind, automated and model-based software testing is of inevitable value. Not only will automation help counter balancing the ever increasing size of software systems but it will also generate a “proper set” of test cases that guarantees a certain (selectable) test coverage over the given specification that would be difficult and very expensive to attain with manual testing.

MoMuT::UML was designed with this in mind. At heart it is an efficient, automated test case generation tool. It will take a UML model of your system and support you in unit, integration, system and acceptance testing. Apart from automated test design it can also provide feedback about the quality of your existing test suite, extend a given test suite, and help with fault localization. MoMuT::UML can easily be integrated into existing workflows.

MoMuT::UML has demonstrated its value in several international research projects in close cooperation with leading partners from industry in the areas of automotive and rail.

HOW DO I PROFIT FROM MoMuT::UML?

► Reduction in the costs of testing (safety) critical systems.
► Improved software quality through the use of an automated and model-based test case generation technique.
► Customizable and very fine grained control over the test coverage.
► Assessment and extension of an existing test suite.
► Different types of tests for regression testing (long) and debugging (short).
► Integration into existing workflows via OSLC.

MoMuT::UML

UNIFIED MODELING LANGUAGE

TECHNOLOGY

MoMuT::UML uniquely combines a powerful new fault-based test case generation strategy with standard techniques to deliver high quality test suites with an excellent cost/benefit ratio. Heart of this new technology is the concept of fault seeding or mutation. As can be seen in Figure 1, MoMuT::UML uses customizable mutation operators to derive mutated models from the original test model. A mutated model is an exact copy of the original minus one change introduced by the mutation operator. Given a mutant and the original specification, MoMuT::UML searches for a sequence of inputs and outputs that uncovers any design refining (“implementing”) the mutant instead of the original. It is in the nature of mutation-based test case generation that one such sequence, i.e. test, finds (“kills”) multiple mutated models and, hence, has the ability to find faults that are not directly modeled by a mutation operator.

Mutation-based test case generation is the most fine-grained and versatile test generation technique available today. In principle, mutation-based test case generation can not only be used to test functional properties of designs but also to generate tests that detect certain non-functional defects. It allows MoMuT::UML to know exactly which faults are caught by a particular test case, analyze or extend existing test sets, and help localizing faults by (a) automatically selecting a set of mutants that can explain faulty behavior and (b) create a short test-case to help with debugging.

MoMuT::UML is grounded on solid theory: it uses the IOCO conformance relation and translates UML to an internal representation with clearly defined semantics (action systems). The backend can connect to state-of-the-art model checkers for further design verification and uses both, concrete and symbolic test case generation engines. It also supports a model animation feature and OSLC integration.

MoMuT::UML is actively maintained by a group of dedicated scientists and engineers at the AIT Austrian Institute of Technology in close cooperation with Graz University of Technology. Please contact us for further information on MoMuT::UML.

CONTACT

AIT Austrian Institute of Technology
Safety & Security Department
Donau-City-Straße 1, 1220 Vienna

RUPERT SCHLICK

Safe and Autonomous Systems
Phone: +43(0) 50550 - 4124
Fax: +43(0) 50550 - 4150
E-mail: rupert.schlick@ait.ac.at
Web: www.ait.ac.at/safety_security