MODEL BASED TESTING: EXPERIENCES FROM TTCN-3 POINT OF VIEW

TIBOR CSÖNDES, ERICSSON
FERENC BOZÓKI, ERICSSON
GYÖRGY RÉTHY, ERICSSON
STEPHAN SCHULZ, CONFORMIQ
ATHANASIOS KARAPANTHELAKIS, CONFORMIQ
JANI KOIVULAINEN, CONFORMIQ
OUTLINE

› Motivation
› Why Model Based Testing?
› MBT Impact on Test Suite Design
› Approaches for Test Harness Implementation
› Workflow
› Catches and traps
MOTIVATION

› Introduction of Model Based Testing in context of TTCN-3

› Give a summary about the differences of manually designed and model based test suites

› Investigate the different approaches of test harness implementation

› Share our experiences with model generated TTCN-3 test suites
TEST AUTOMATION

› “Classical” test automation: automation of test execution
› MBT: automation of test design (automatic test generation from a model)

MBT is a higher level problem solution

Trend

MBT
Automated test
Manual test

Happy advanced tester
Typical tester
“Classical” automated testing

- each test case checks one or a few transitions
- each test case is developed separately
- each test case is maintained separately
- each test engineer is exposed to details of SUT interfaces

testcase tc_TP#1 ()
runs on MTCType_CT system MTCType_CT
{
    map(mtc:My_PCO, system:My_PCO);
    My_PCO.send (t_SYN(A));
    alt {
        My_PCO.receive(t_SYN_ACK(A+1,B))
        My_PCO.send(t_ACK(A+1,B+1))
        setverdict(pass);
    }
}
Model Based Testing

- tests are generated from an SUT model
- at SUT change the model is updated and test cases are re-generated
- models only include interface aspects & data related to the functionality to be tested
- tests are generated based on coverage criteria
MODEL BASED TESTING ON FIELD

› Pros and Cons of Model Based Testing
  – Reduces fault slip through

**Design phase**

Model development of the Design and model development of the Testing could take place parallel

→ model development for testing verifies the model of the design

→ some faults could be found in the “development phase”

→ Reduces development time

→ Model Driven Engineering

**Testing phase**

Model development of the Design and model development of the Testing could take place parallel

→ model development for testing verifies the model of the design

→ some faults could be found in the “development phase”

→ Reduces development time

→ Model Driven Engineering
CASE STUDIES: TEST ARRANGEMENT

- Generated tests: abstract TTCN-3 test cases (not directly executable)

- Test harness: all the extras that makes the abstract test cases executable (TTCN-3 code, adapters, TTCN-3 tool environment etc.)
APPROACHES FOR TEST HARNESS

› Hand written glue code
  - Demands advanced knowledge of TTCN-3 and the TTCN-3 tool
  - Demands advanced knowledge of the underlying test harness
  - Repeated development if the tested scenario changes
  - Test harness is project-specific

› Using generic glue code
  - Built on top of already existing generic SW libraries (TitanSim)
  - Requires only minor project-specific adaptation
  - Generic part: write once, use several times: additional gain to test case generation
WORKFLOW

- Test Goals
- SUT Specification
- Modelling
  - Defining the Model interfaces
  - Defining the Model behavior
  - Model
  - Test Generation (automatic)
  - Test Cases
  - Test Script Generation (automatic)
  - Test harness Implementation (manual)
  - Test Framework
- Test Scripts
- Test Executor Tool
- Test Results
- Test Result Analysis (manual)
- SUT
EXPERIENCES, RECOMMENDATIONS

› MBT is a paradigm shift
› “Right” competence is required, training is needed
› New roles should be established within the test organisation, especially the model designer/”test architect”
› When designing the **good** model, the tester shall not think in terms of test cases – the tester should, ultimately, only think of the system behaviour
› The generated test cases cover several events (Model/Test requirements), while the traditional test cases normally only cover one event/situation
› Start with a smaller, well defined, well encapsulated, area/functionality
› Save time and money! On average: ~20-30%