Testing Techniques 2016 – 2017
Assignment 3

Model-Based Testing II

This is the second Model-Based Testing (MBT) assignment of the course Testing Techniques, which continues the previous assignment. The purpose of this assignment is to apply Model-Based Testing (MBT) to your System Under Test (SUT) using a second MBT tool, and to compare. You can freely select any MBT tool; below some ideas are given.

Try to reuse as much as possible from what you did in the second assignment, i.e., the test architecture and (the structure of) the model, though you will have to express your model in another syntax, of course.

1. Model-Based Testing

1. **MBT Tool Selection**

   Search for model-based testing tools, select an MBT tool, and give some arguments why you selected that tool. (See the list of MBT tools below.)

2. **MBT Modeling**

   Make a model for your SUT, or part of your SUT, in the input language of the selected MBT tool. In order to allow comparison, it is easiest if you model the same part as you used in the second assignment (first MBT assignment). Explain your model.

3. **MBT Test Environment**

   Adapt the test environment of the previous assignment, if necessary, for model-based testing with the selected MBT tool.

4. **MBT Testing**

   Use the selected MBT tool to generate tests, and execute them on your SUT. Explain your observations and analyse the test results.

5. **Deliverable**

   Give the models, code, adapters, etc. in such a way that we can run it; provide a 'README'. Be prepared to give a demo.

2. Comparison

Compare the selected MBT tool of this assignment with TorXakis; consider the following aspects:

- implementation relation;
- support for test input generation as well as output checking;
- support for non-determinism;
- method of test selection;
- modeling notation: its expressiveness and ease of use;
- on-line vs. off-line testing, i.e., on-the-fly vs. batch.

Hints - Model-Based Testing Tools

There are many MBT tools: academic and commercial (often with 30 days free trial period), using various modeling languages, with abstract models and with models as programs, on-line and off-line, supported and unsupported, serious and less serious, ...

Have a look at websites like this one:

`http://robertvbinder.com/open-source-tools-for-model-based-testing`

or this one:

`http://mit.bme.hu/~micskeiz/pages/modelbased_testing.html`

or search on the web, or have a look at this list (underlining is used to indicate a slight preference):

AETG, Agatha, Agedis, All4Tec MaTeLo, Autolink, Axini Test Manager, Conformiq Qtronic, Cooper, fMBT, Gast, Gotcha, JTorX, NModel, ParTeG, Phact/The Kit, QuickCheck, Reactis, RT-Tester, SaMsTaG, SeppMed MBTsuite, Smartesting CertifyIt, Spec Explorer, Statemate, STG, TestGen (Stirling), TestGen (INT), TestComposer, TGV, TorX, T-Vec, Uppaal Cover, Uppaal Tron, Tveda, TestOptimal, GraphWalker, Tigris, OSMO, ...