AN EFFICIENT APPROACH FOR MODEL-BASED TESTING: SIEMENS USE CASE IN THE MBAT EUROPEAN PROJECT

HELENE.LEGUEN@ALL4TEC.NET,
DEEPA.VIJAYARAGHAVAN@MBTECH-GROUP.COM,
RALF.PINGER@SIEMENS.COM
Test duration is still too important and testing can take 40% of the total project costs. For this reason, Siemens seeks new approaches to reduce these costs. The challenge is first to propose adequate methodologies and then to make them applicable in practice with powerful tool support.

The methodology is under study in the ARTEMIS research project MBAT (Combined Model-based Analysis and Testing of Embedded Systems). MBAT aims at efficient validation and verification of embedded systems by using model-based technologies and effective combinations of analysis and testing activities. The improvements are evaluated in case studies provided by industrial partners.

SCADE Suite doesn’t provide capabilities in terms of model based testing, test case generation, test execution, test management.
INTEGRATED TOOLCHAIN

- PROVEtech:TA
- SCADE
- KCG
- C code
- Test execution
- Signals
- Test functions
- Test cases
- MaTeLo

Requirements

Integration

control
The shutter has an up and a low position.

User can activate a movement up (or down) when the shutter in the lower (or upper) position or in a intermediate position.

When the user presses a long time on the button, the shutter moves all the way up (or down).
TOOLS PRESENTATION: SCADE
TOOLS PRESENTATION: PROVETECH:TA

**PROVEtech:TA – Test Automation**

- Visualization
- Test Management & Automation
- Integrated Diagnostics
- Fault Simulation
- Real-time Automation
- Integrated XCP & CCP
- Project & User management
- Statistics module
- Compatible with multiple standard HIL technologies

**Key Features**

- **VisionCore**
- **MATLAB Simulink**
- **dSPACE**
- **ETAS**
- **LabVIEW**
- **National Instruments**
- **SCADE System**
- **VeriStand**
MaTeLo is a model-based testing tool which contains its own modeller. MaTeLo is used to generate test cases from the usage model.

With MaTeLo, Test engineers do not need to consider the implementation details (Black-box testing), but just focus on the way the system is used.

MaTeLo model is based on Markov chains where stimulations and expected results are basically associated with transitions. Facilities are provided to define the expected results, to link the test model with system requirements and also to automate test execution.
WHAT IS NEW? WHAT IS DIFFERENT?

- A specific tool for each specific need: the language used in each tool is appropriate for each test level.
  - No need to know the script language when you design test cases (only basic test operations)
  - No need to know the specification when you are at the automation level
- Tools are interconnected: there is no need to enter the same information multiple times in multiple tools
  - Scade -> PROVEtech:TA: signals list
  - PROVEtech:TA -> MaTeLo: inputs/outputs/test operations
  - MaTeLo -> PROVEtech:TA: test cases
  - PROVEtech:TA -> Scade: test cases execution
1. Connection between the System Under Test (SUT; here the Scade model) with a test execution tool (here PROVEtech:TA)

2. At PROVEtech level, user can create some specific “Test operations” in order to stimulate input or make a check on output in a specific manner.

3. Test model creation using MaTeLo

4. Test cases generation with MaTeLo

5. Test cases execution with PROVEtech:TA

6. Coverage of Scade model (implementation level) and coverage of MaTeLo model (requirements level) are calculated
INTEREST OF USING MODEL BASED TESTING FOR SIEMENS

- Using MBT means build a test model to automatically generate test cases instead of manual test cases identification.
- Design of test model is based on the system usage.
- Automatic test generation of relevant test cases according to specific criteria -> quality improvement.
  - Focused on the software parts where risk is higher.
  - Coverage-based criteria or Reliability-based criteria.
- Multiple generation options make it possible to generate many test cases from the same usage model i.e. for the same initial cost.
- Models reuse is possible.
BENEFITS FOR USING AN INTEGRATED TOOLCHAIN

- Only relevant information is entered in each tool, bridges are available to import/export information in the other tools.
- Manual adaptations are no longer needed to translate test cases to test scripts.
CONCLUSION

- Automatic test generation and automatic test execution interconnected in a toolchain
- Duration and effort saving
- Quality improvement