Model-Based Testing: Challenges Ahead

Mats P. E. Heimdahl
Department of Computer Science and Engineering, University of Minnesota
heimdahl@cs.umn.edu

In model-based testing, models derived from the informal requirements (or models developed as part of the requirements process) are used to drive the testing; these models are used to generate the tests as well as serve as oracles, and the testing process can be largely automated. This move towards models, tools, and automation holds enormous promise, but it also raises new challenges that, in our experience, must be addressed before we can reap the full benefits.

Model Validation: As we are moving towards a testing framework where models are the central artifacts, the verification and validation (V&V) effort will be largely focused on assuring that the models are correct—the models must be properly validated. Unfortunately, current certification standards, for example, DO-178B, as well as research efforts to date focus almost exclusively on various verification activities. Our current inability to adequately validate our requirements and models raises a serious concern regarding the adoption of model-based testing; if the models are wrong, the testing effort will be inefficient or possibly outright deceiving (if we blindly accept the results of testing using poorly validated models). Robust techniques addressing the requirements validation and model validation problems are of critical importance. There are several questions that must be answered: What techniques can we use for model validation? Can these techniques be automated? How do we know when we have validated the model sufficiently? How can we determine whether there are missing or unstated high-level requirements?

Model-Coverage Criteria: When using tools to automatically generate tests-suites from models, the selection of coverage criterion that guides the generation process is of imperative importance. In our work, we have observed that although a coverage criterion may seem reasonable when instrumenting a model or code to measure the adequacy of a test suite, it may be unsuitable when formalized and used to guide the model checker to generate a test suite [1]; the generated tests technically provide adequate coverage according to the formalization, but do so in a way that exercises only small portions of the system under study and finds few faults. Based on our experience, we conclude that fully automated test-suite generation techniques must be pursued with caution and that coverage criteria specifically addressing test-suite generation from models are needed. We must develop and empirically evaluate a new-class of “model coverage criteria” addressing the needs specific to model-based testing.

Loss of Collateral V&V: Cost savings in model-based testing will be achieved by replacing time consuming and costly manual processes with tools. Manual processes, however, draw on the collective experience and vigilance of many dedicated software professionals; professionals that provide “collateral validation and verification” as they are working on the software testing tasks. Experienced testers developing test-cases provide additional validation of the software system; if there is a problem with the specified functionality of the system, they have a chance of noticing and taking corrective action. When replacing these manual efforts with automation there may be no safeguards in the development activities to catch critical flaws in the model or the tools used in the test automation.

Nevertheless, as we ponder the challenges with model-based testing we cannot lose track of one important fact—although perfection is the goal, perfection is not necessary for deployment and highly effective use. After all, our aim is to replace costly, time consuming, and error prone manual tasks, and all that is really necessary is that our automation is better than the manual tasks we replace—much important analytical and empirical research is needed to help determine if that is the case.

References