Abstract

Blom, J. 2016. Model-Based Protocol Testing in an Erlang Environment. Built: February 26, 2016. Acta Universitatis Upsaliensis. 1214. 250 pp. Uppsala. ISBN

Testing is dominant technique for quality assurance of software systems. In general, it consumes significant amounts of resources in development projects, but are many times ad hoc and error prone. This thesis is concerned with model-based testing, which is an approach to make testing more systematic. Despite the significant improvements on model-based testing in the last 20 years, acceptance by industry is limited. This thesis address this problem by suggesting an approach based on symbolic model checking techniques and a novel test suite generation technique in which different coverage criteria can be easily expressed. All wrapped up and presented in an Erlang environment.

The general idea in model-based testing is to start from a formal model, which captures the intended behavior of the software system to be tested. On the basis of this model, test cases can be generated in a systematic way. Since the model is formal, the generation of test suites can be automated and with adequate tool support one can automatically quantify to which degree they exercise the tested software. An important application area in model-based testing is that of testing communication protocols and other similar classes of reactive systems. For such systems, detailed behavioral specifications can be used as models, and serve both as a basis for test suite generation, and as an oracle for assigning verdicts to test cases.

This thesis describes a methodology and associated tool support, which is intended to be used for model-based testing of communication protocol implementations in industry. It presents several technical contributions to the area of model-based testing, of which the main ones are

- a new specification language based on the functional programming language Erlang,
- a novel technique for specifying coverage criteria for test suite generation, and
- a technique for automatically generating test suites.

Based on these developments, we have implemented a complete tool chain that generates and executes complete test suites, given a model in our specification language. The thesis also presents a substantial industrial case study, where our technical contributions and the implemented tool chain are evaluated.

Keywords:

Johan Blom, Department of Information Technology, Uppsala University, Lägerhyddsvägen 2, SE-752 37 Uppsala, Sweden

ISSN
ISBN
urn:nbn:se:uu:diva-3344 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-3344)