Abstract

Blom, J. 2016. Model-Based Protocol Testing in an Erlang Environment. Built: March 8, 2016. Acta Universitatis Upsaliensis. 1214. 250 pp. Uppsala. ISBN

Testing is the dominant technique for quality assurance of software systems. It typically consumes considerable resources in development projects, and is often performed in an ad hoc manner. This thesis is concerned with model-based testing, which is an approach to make testing more systematic and more automated. 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.

Despite the significant improvements on model-based testing in the last 20 years, acceptance by industry has so far been limited. A number of commercially available tools exist, but still most testing in industry relies on manually constructed test cases.

This thesis address this problem by presenting a methodology and associated tool support, which is intended to be used for model-based testing of communication protocol implementations in industry. A major goal was to make the developed tool suitable for industrial usage, implying that we had to consider several problems that typically are not addressed by the literature on model-based testing. The thesis presents several technical contributions to the area of model-based testing, including

- 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. Findings from the case study include that test suites generated using (model) coverage criteria have at least as good fault-detection capability as equally large randomly generated test suites, and that model-based testing could discover faults in previously well-tested software where previous testing had employed a relaxed validation of requirements.

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)