10421 Abstracts Collection
Model-Based Testing in Practice
— Dagstuhl Seminar —

Wolfgang Grieskamp[1], Robert Hierons[2] and Alexander Pretschner[3]

[1] Microsoft Corp. - Redmond, US
wrwg@microsoft.com
[2] Brunel University, GB
rob.hierons@brunel.ac.uk
[3] TU Kaiserslautern, DE
alexander.pretschner@kit.edu

Abstract. From 17.10. to 22.10.2010, the Dagstuhl Seminar 10421 "Model-Based Testing in Practice" was held in Schloss Dagstuhl – Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Testing, Modeling, Model-Driven Development

10421 Summary – Model-Based Testing in Practice

Software testing is one of the most cost-intensive tasks in the modern software production process. Model-based testing is a light-weight formal method which enables the automatic derivation of tests from software models and their environment. Model-based testing (MBT) has matured as a rich research area in the last decade, with a significant body of research and applications. The academic community is well established with many conferences, workshops, and research projects. Tools for model-based testing have been developed both as research prototypes and as commercial or semi-commercial applications brought to users by midsize and enterprise-level companies, and applied in large scale projects. In the family of model-driven approaches, model-based testing can be seen as a success story in particular with respect to the degree of mechanical processing and automation that has been achieved, and the adoption in practice. The successful deployment of model-based testing in industrial settings can be seen in the telecommunication domain, chip cards, specific Windows components, and embedded systems in general. An interesting issue is under which circumstances we can expect these successes to carry over to other domains and families of systems as well (e.g., distributed systems; testing the cloud).

Dagstuhl Seminar Proceedings 10421
Model-Based Testing in Practice
http://drops.dagstuhl.de/opus/volltexte/2011/2924

Keywords: Testing, Modeling, Model-Driven Development

Joint work of: Grieskamp, Wolfgang; Hierons, RObert H.; Pretschner, Alexander

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2011/2925

Efficient Mutation Killers in Action

Bernhard K. Aichernig (TU Graz, AT)

In this talk we report on our results and experiences of applying model-based mutation testing in the FP7 STREP project MOGENTES. We have developed a tool chain that translates UML state transition diagrams to Back's action systems. The state transition diagrams may be nested and may involve parallel regions. Given an original model and a number of mutants, an ioco-conformance checker implemented in Prolog searches for killing test cases. In addition, the action system models can be translated to the CADP tool set, giving access to its model checkers, simplifiers and the TGV test case generator. We will discuss our different strategies for generating test cases and will present the empirical results of an automotive demonstrator.

Keywords: Model-Based Mutation Testing, UML, Action Systems, Mutation Analysis, Embedded Systems

Slicing State-based Models

Kelly Androustopoulos (University College London, GB)

Slicing is a technique, based on dependence analysis, for simplifying a program and focusing on selected parts of interest. Although it has been widely studied for nearly three decades, there has been comparatively little work on slicing for state machines. One of the primary challenges that currently presents a barrier to wider application of state machine slicing is the problem of determining control dependence. In this talk, we introduce a new definition of control dependence and illustrate our slicing approach that is based on this definition. Our slices respect Weiser slicing's termination behaviour.

Keywords: Slicing, state machines, EFSM

Abstraction for Model Validation and Test Selection

Victor Braberman (University of Buenos Aires, AR)

Models are rich and complex artifacts that require validation activities. I will present some advances in statically creating conservative behavioral abstractions in the form of FSM from infinite state models.

We believe such models are potentially useful to conduct manual review by complementing detailed exploration. I will also mention some preliminary research on understanding if coverage of such abstraction may be (or not) a useful goal to pursue when generating test suites.

Joint work of: Braberman, Victor; Guido de Caso; Diego Garbervetsky; Hernán Czemerinsky; Sebastián Uchitel

Model-based Testing and Verification @Simula: Industrial Experiences

Lionel C. Briand (Simula Reseach Laboratory - Lysaker, NO)

This presentation aims a summarizing my group's experiences with doing applied research on model-based testing and verification in collaboration with industry partners

Full Paper:
http://simula.no/research/approve

Microsoft's Protocol Documentation Program: A Success Story for Model-Based Testing

Yiming Cao (Microsoft China - Beijing, CN)

The talk introduces the success story of applying model based testing inside of Microsoft for the quality assurance of over 200 technical documents of network protocols related to Microsoft Windows. It gives background information of the "BlueLine" document testing project, and then introduces the way how MBT tool "Spec Explorer 2010" is used in the project. It finally evaluates the effectiveness of MBT comparing to traditional testing, lists several pain points observed, and draws the conclusion that the application of MBT brings big effeciency gain.

Keywords: MBT, Protocol Documentation

Specification coverage for testing in *Circus*

Ana Cavalcanti (University of York, GB)

Circus is a state-rich process algebra for refinement. We have previously presented a theory of testing for *Circus*; it gives a symbolic characterisation of tests.

Each symbolic test specifies constraints that capture the effect of the possibly nondeterministic state operations, and their interaction. This is a sound basis for testing techniques based on constraint solving for generation of concrete tests, but does not support well selection criteria based on the structure of the specification. We present here a labelled transition system that captures information about a *Circus* model and its structure. It is useful for testing techniques based on specification coverage. The soundness argument for the new transition system follows Hoare and He's Unifying Theories of Programming style, but relates the new transition relation to the *Circus* relational model and its operational semantics.

Keywords: Process algebra, symbolic testing

Joint work of: Cavalcanti, Ana; Gaudel, Marie-Claude

Component-based framework for model based testing

Victor Kuliamin (Academy of Sciences - Moscow, RU)

The talk outlines possible architecture of component-based model based testing framework using non-invasive composition techniques for flexible test construction and test configuration.

Keywords: Model based testing, component-based development, non-invasive composition, dependency injection

Experiences from MBT industrial usage

Andres Kull (Elvior - Tallin, EE)

The presentation gives an overview of the experiences gained from MBT industrial usage. Elviori model-based testing technology has been applied to four industrial case. The presentation gives an overview of the Elvior testing technology and test generation tool TestCast Generator. Subsequently, the use cases are briefly introduced. Then, benefits from MBT usage in the industrial use cases are outlined. Finally, the challenges of MBT wider industrial adoption are discussed.

Keywords: Model-based testing

Challenges on deploying MBT for functional testing of large-scale information systems

Bruno Legeard (Smartesting - Besancon, FR)

In this talk, we address the Challenges linked to Model-based testing in the IT domain. The test of large Information Systems is facing specific issues such as the importance of End-to-End testing, manual testing and the large number of applications that are part of systems.

Composing Business Process models and behavioral models for automated test generation, is a way to address the challenges of MBT for IT and support the cooperation between Business Analysts and Quality Assurance people.

Keywords: Model-based Testing, Functional Testing, Testing Information Systems

XACML Model Based Testing

Eda Marchetti (CNR - Pisa, IT)

A widely adopted security mechanism is the specification of access control policies by means of the XACML language. We propose a framework, called XCREATE, for the systematic generation of test inputs (XACML requests). Differently from existing tools, X-CREATE exploits the XACML Context Schema. In particular, the tool applies a XML-based methodology (XPT) to systematically produce a set of intermediate instances, covering the XACML Context Schema. Moreover, for request generation, X-CREATE applies a procedure for parsing the policy under test and assigning values to the generated intermediate instances. The aim of the proposed framework is twofold: testing of policy evaluation engines and testing of access control policies. The experimental results show that the fault detection effectiveness of X-CREATE is similar or higher than that of existing approaches.

Keywords: Policy Testing, XACML, Test Cases Generation

Joint work of: Marchetti, Eda; Francesca Lonetti; Antonia Bertolino

Software Architecture-based Testing: how MBT may help?

Henry Muccini (Univ. degli Studi di L'Aquila, IT)

A software architecture description captures the overall structure of a software system, the rationale applied to make decisions, and structures the architecture specification into different viewpoints, each one aimed at solving certain stakeholdersŠ concerns. Architecture-based testing (ABT) consists in testing the implementation compliance to architectural design decisions and to the selected solution (i.e., the architecture itself). Being architecture-based testing a form of model-based testing, goal of this talk is to raise discussion about how MBT can be leveraged for ABT. An ongoing project, and its related activities, is also presented.

Keywords: Software Architecture, Software Architecture-based Testing, MBT

Full Paper:
http://www.henrymuccini.com/publications.htm

Multi-model-based Analysis

Henry Muccini (Univ. degli Studi di L'Aquila, IT)

A Critical System can be defined as a system that requires to deal with different quality attributes. Such quality attributes cannot be managed separately or sequentially, but needs to be taken into account in a coordinated way, dealing with a multi-optimization problem.

This presentation presents an ongoing work on how to manage analysis of multiple specifications. The idea is to use model transformation techniques, to keep various models synchronized, and model differencing for enabling regression analysis.

Keywords: Multiple models, multiple analysis, synchronization, regression

Full Paper:
http://www.henrymuccini.com/publications.htm

Automatic generation of high quality test sets via CBMC

Gabriele Palma (Genova University, IT)

Software Testing is the most used technique for software verification in industry. In the case of safety critical software, the test set can be required to cover a high percentage (up to 100%) of the software code according to some metrics. Unfortunately, attaining such high percentages is not easy using standard automatic tools for tests generation, and manual generation by domain experts is often necessary, thereby significantly increasing the associated costs. In previous papers, we have shown how it is possible to automatize the test generation process of C programs via the bounded model checker CBMC. In particular, we have shown how it is possible to productively use CBMC for the automatic generation of test sets covering 100% of branches of 5 modules of ERTMS/ETCS, a safety critical industrial software by Ansaldo STS. Unfortunately, the test set we automatically generated, is of lower "quality" if compared to the test set manually generated by domain experts: Both test sets attained the desired 100% branch coverage, but the sizes of the automatically generated test sets are roughly twice the sizes of the corresponding manually generated ones. Indeed, the automatically generated test sets contain redundant tests, i.e. tests that do not contribute to reach the desired 100% branch coverage. These redundant tests are useless from the perspective of the branch coverage, are not easy to detect and then to eliminate a posteriori, and, if maintained, imply additional costs during the verification process.

In this paper we present a new methodology for the automatic generation of "high quality" test sets guaranteeing full branch coverage. Given an initially empty test set T, the basic idea is to extend T with a test covering as many as possible of the branches which are not covered by T. This requires an analysis of

the control flow graph of the program in order to first individuate a path p with the desired property, and then the run of a tool (CBMC [6] in our case) able to return either a test causing the execution of p or that such a test does not exist (under the given assumptions). We have experimented the methodology on 31 modules of the Ansaldo STS ERTMS/ETCS software, thus greatly extending the benchmarking set. For 27 of the 31 modules we succeeded in our goal to automatically generate "high quality" test sets attaining full branch coverage: All the feasible branches are executed by at least one test and the sizes of our test sets are significantly smaller than the sizes of the test sets manually generated by domain experts (and thus are also significantly smaller than the test sets automatically generated with our previous methodology). However, for 4 modules, we have been unable to automatically generate test sets attaining full branch coverage: These modules contain complex functions falling out of CBMC capacity. Our analysis on 31 modules greatly extends our previous analysis based on 5 modules, confirming that automatic test generation tools based on CBMC can be productively used in industry for attaining full branch coverage. Further, the methodology presented in this paper leads to a further increase in the productivity by substantially reducing the number of generated tests and thus the costs of the testing phase.

Joint work of: Di Rosa, Emanuele; Giunchiglia, Enrico; Narizzano, Massimo; Palma, Gabriele; Puddu, Alessandra

From Test Purposes and Specifications to Test Cases

Alexandre Petrenko (CRIM - Montreal, CA)

In this talk, we discuss the problem of constructing a test case for a given test purpose and specification modelled by input/output transition systems (IOTS). The communication between the tester and the implementation under test is assumed to be asynchronous, performed via queues. Differently from synchronous tests, when issuing verdicts, asynchronous tests should take into account the distortion caused by the queues in the observed interactions. We demonstrate that IOCO tests designed for synchronous communication may not be sound for queued asynchronous communication. A class of IOTS specifications, called Mealy IOTS, without input/output conflicts is identified, for which synchronous and asynchronous tests coincide. We also discuss how sound asynchronous tests can be derived for a given test purpose without actually composing the specification with the queue, avoiding state explosion.

Keywords: MBT, conformance relations, IOCO, asynchronous testing, queued testing

Integration and testing of driver-assistance systems

Christopher Robinson-Mallett (Berner & Mattner Systemtechnik - Berlin, DE)

The trend towards constantly growing numbers of product variants and features in automotive industry makes the improvement of specification, analysis and testing techniques a key efficiency enabler during development and validation of driver assistance systems. The development of a single generic functional specification applicable to a whole product family can help saving costs and time to market significantly. Similarly, the specification of generic test-cases executable with minor changes to a whole family of target platforms can help to save further costs and time. However, the introduction of product-line approaches into a system manufacturer's electronics development processes is a challenging task, prone to human error, with the risk of spreading single faults across a whole platform of product variants.

Additionally, the introduction of safety-standards, such as the ISO 26262, into automotive systems development raises the demand for efficient and traceable validation and verification approaches specific to product properties and Automotive Safety Integrity Level (ASIL). Although, such standards and validation and verification approaches have been experienced in industries with a long tradition in the development of safety-critical systems, such as in avionics or railway domains, we have seen that their application is challenging in automotive systems development; to a large degree caused by significantly shorter times to market and cost pressure of a mass product on a highly competitive market.

In order to comply with emerging safety-standards, while meeting tight cost and time goals, we have identified three key efficiency enablers for system integration and validation that are currently being introduced or have been exercised in car-manufacturers driver-assistance development projects:

a) improving the testability and traceability of specifications,

b) introducing variant-management into the test-case specification process,

c) introducing new systematic and model-based analysis and testing techniques.

This contribution presents experiences with model-based and formal approaches on system integration and testing. Practical issues with an emphasis on product-line development and academical approaches on model-based testing and analysis are discussed and a conclusive summary is being presented.

Keywords: Model-based testing, automotive, system integration

Driving Technology Through An Industrial Challenge: The Microsoft Hypervisor Verification

Thomas Santen (European Microsoft Innovation Center - Aachen, DE)

Is it feasible to verify a substantial piece of product software, with all the intricacies it comes with such as low-level operations, high degree of course-grained and fine-grained concurrency, and written not with formal verification but with optimization for performance in mind? This question motivated the project of formally verifying the kernel of the Microsoft Hyper- V hypervisor. The talk reports on the results of the project, which produced the verification system VCC for concurrent C. The talk concludes with lessons learned and future challenges in driving research results toward industrial applicability.

Model-Based Testing of a Wireless Sensor Network

Holger Schlingloff (HU Berlin, DE)

Within the BMBF Project SPES2020, we develop a body area network (BAN) to be worn by elderly people and patients in need for supervision. The BAN consists of different sensor nodes which are able to jointly detect a fall of the bearer and, should the situation arise, autonomously call for emergency help. In this talk, we describe the challenges and projected solutions for model-based testing of this system.

Keywords: Model-based testing, sensor networks, distributed consensus, HiL-Testing

Joint work of: Schlingloff, Holger; Lackner, Hartmut

MBT and WSN: a practical issue

Julien Schmaltz (Radboud University Nijmegen, NL)

In this talk, we present a modeling issue that we encountered while testing a clock synchronization algorithm for nodes of a Wireless Sensor Network.

Keywords: Model-based testing, wireless sensor networks, real-time systems, simulated time

Applying Microsoft Spec Explorer to Engine Control Unit Safety Tests

Arne Schmenkel (Adam Opel GmbH - GMAPC Europe - Rüsselsheim, DE)

Our Engine Control Units contain safety critical functions. Examples are torque, high voltage and fuel control.

The software functions are time dependent, distributed over several control units and software is also reused from existing product lines.

We were able to take over the methodology based on Spec Explorer. It shows many conceptual analogies to modern model based development of controls software as well as power-train simulations.

We developed a small timer-library and a small component framework. Test case scenarios can easily be extended to models now and allow systematic and precise tests.

Keywords: Functional safety, Microsoft Spec Explorer, model based testing, power-train control, real-time system, timed system, time planning

Alternating Simulation and IOCO

Margus Veanes (Microsoft Research - Redmond, US)

In this talk I'll provide a formal proof of the relation between ioco and alternating simulation for symbolic labeled transition systems and talk about practical implications of this result.

Keywords: LTS SMT

Satisfying yet unsupported Coverage Criteria or Simulated Satisfaction for MBT

Stephan Weißleder (Fraunhofer Institut - Berlin, DE)

UML state machines are widely used as test models in model-based testing. Coverage criteria are applied to them, e.g. to measure a test suite's coverage of the state machine or to steer automatic test suite generation based on the state machine. The model elements to cover as described by the applied coverage criterion depend on the structure of the state machine. Model transformations can be used to change this structure. In this talk, we present semantic-preserving state machine transformations that are used to influence the result of the applied coverage criteria. The contribution is that almost every feasible coverage criterion that is applied to the transformed state machine can have at least the same effect as any other feasible, possibly stronger coverage criterion that is applied to the original state machine. We introduce simulated satisfaction as a corresponding

relation between coverage criteria. We provide formal definitions for coverage criteria and use them to prove the correctness of the model transformations that substantiate the simulated satisfaction relations. The results of this approach are especially important for model-based test generation tools, which are often limited to satisfy a restricted set of coverage criteria.

Keywords: Model-based testing, coverage criteria, model transformation, simulated satisfaction

Full Paper:
http://dx.doi.org/10.1109/ICST.2010.28

See also: Stephan Weißleder, Simulated Satisfaction of Coverage Criteria on UML State Machines. In Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation (ICST'10), 2010, pp. 117–126, IEEE Computer Society, USA.

Model-based testing in the enterprise software domain - past and future initiatives

Sebastian Wieczorek (SAP Research - Darmstadt, DE)

The testing of enterprise software poses various challenges that require a lot of effort using classical testing approaches. Model-based testing (MBT) showed its potential to improve this on several dimensions including better coverage and increased productivity both on system integration and UI-based scenario testing. In this talk we describe SAP's recent history of MBT initiatives. A special focus is put on the identified industrial challenges, future directions inside SAP, and proposals for a global alignment of industry and academia.

Keywords: Enterprise software, mbt, industry

Joint work of: Wieczorek, Sebastian; Stefanescu, Alin (University of Pitesti)

The Isabelle Platform: A Perspective for Collaborative and Fine-grained Parallel TestCase-Generation

Burkhart Wolff (Université Paris Sud, FR)

Isabelle is meanwhile a formal methods implementation platform — so to speak: the Eclipse of FM Tools — offering fine-grain parallelism, a powerful symbolic computation environment, a theorem prover kernel allowing for logically safe extensions, substantial logical libraries, a component plug-in mechanism, code- and documentation generators and a GUI-Framework both making the internal parallelism accessible and bridging the gap to the JAVA world. Our TestCase-Generation System HOL-TestGen is a PlugIn in this framework. We will outline recent advances for the Isabelle Framework and the consequences for HOL-TestGen and its design.

Keywords: Formal Methods, Model-based Testing, Theorem Proving, Isabelle, LCF Architecture, Parallel Symbolic Computation

Model-based Security Testing of a Health-Care System Architecture: A Case Study

Burkhart Wolff (Université Paris Sud, FR)

We present a generic modular policy modelling framework and instantiate it with a substantial case study for model-based testing of some key security mechanisms of the NPfIT. NPfIT, "the National Program for IT" is a very large- scale development project aiming to modernise the IT infrastructure in the English health care system (NHS). Consisting of heterogeneous and distributed code, it is an ideal target for model-based testing techniques of a very large system exhibiting critical security features. We will model the four information governance principles, comprising a role-based access control model, as well as policy rules governing the concepts of patient consent, sealed envelopes and legitimate relationship. The model is given in higher-order logic (HOL) and processed together with suitable test-specifications in the HOL-TestGen system, that generates semi-automatically test sequences according to them.

Particular emphasis is put on the modular description of security policies and their generic combination and its consequences for model-based testing.

Keywords: Model-based Testing, Security Testing, Access-Control, NPfIT, Isabelle/HOL, HOL-TestGen

Joint work of: Brugger, Lukas; Brucker, Achim; Kearney, Paul; Wolff, Burkhart