ICSE 2005 Workshop
Advances in Model-Based Software Testing (A-MOST)
St. Louis, MO, USA
May 15-16, 2005
http://a-most.argreenhouse.com

Siddhartha R. Dalal
Xerox Corporation
Webster, NY, USA
+1-585-422-7485
sdalal@crt.xerox.com

Ashish Jain
Telcordia
Piscataway, NJ, USA
+1-732-699-2625
jain@research.telcordia.com

Jesse Poore
U of Tennessee, Knoxville
Knoxville, TN, USA
+1-865-974-5784
poore@cs.utk.edu

Introduction
This is a summary of the Advances in Model-Based Software Testing (A-MOST) workshop held on May 15-16, 2005, in St. Louis, Missouri. The workshop had approximately 40 participants. The goals of this workshop were to offer a comprehensive overview of model-based testing to the ICSE community, and bring the researchers and practitioners together to discuss advances, applications, and the complex problems yet to be solved in model-based testing. The workshop was lively and maintained participant interest for the entire two-day program. A brief summary based upon agenda items is provided. Details on the workshops are archived at http://aetgweb.argreenhouse.com.

Keynote I: *Failure Mode Avoidance by Tim Davis, Henry Ford Technical Fellow*
The workshop began with an excellent review of an approach used in industry to integrate testing into all phases of design and production. Heavily emphasizing the fundamentals of science and engineering, Davis explained the underlying causes of error introduction in engineered products and the vigilance required to remove them. Although the presentation was essentially hardware oriented, the audience appreciated descriptions from a product line, where failures in the field are generally under control. The presentation ended with some speculation about the impact on reliability of the growing software content in automobiles.

Session 1: Test Selection Criteria
Renee Bryce presented analysis and algorithms to prioritize tests generated by the combinatorial design methods, where user-specified importance is treated as bias in the algorithms. Empirical evidence indicates that the bias results in potentially significant improvements in cumulative weight of test suites. This material was based on foundational material presented by Chris Lott in Session 3.

Marcus Clermont reported on selection of test cases based on specifications. More specifically, test-case selection is a form of stress testing, which focuses on test cases drawn from interesting points in the input space, e.g., points where properties of a function change. This work is in the stream of research from the Software Quality Research Lab of the University of Limerick (and McMaster University) and is based on the tabular representation of specifications.

Session 2: Usage and Simulation Models
Kirk Sayre reported on a large undertaking to automate testing of a library of C++ class templates, the scientific code for computational materials research being developed and maintained at the Oak Ridge National Laboratory. The approach is based on Markov chain usage models and uses methods and tools developed in the Software Quality Research Lab at the University of Tennessee. While early in the overall program, the practicality and value of the approach have been demonstrated through numerous test cases and error discovery.

David McGuiness described a performance simulation model for servers built on enterprise Java beans. The model has a set of user-modifiable parameters (memory size, CPU speed, for example) and a set of model parameters based on the server and system (memory in use, for example). The model computes statistics on performance and allows what-if analysis and prediction, based on parameter changes (increase in number of CPUs, for example.)

Erika Olimpiew described a method for producing test cases for an application derived from a software product line. The software product line spawns applications that have many features in common. Relationships between model elements of the product line and the application facilitate automatic selection and generation of tests.

Albert Schilling reported on test designs based on human-computer interaction. The work focuses on usability tests, such as the appropriateness of interface design to user preferences. Testing produces better user interfaces, and better user interfaces improve testability.

Session 3: Requirements and Test Generation
Chris Lott presented a tutorial through examples of the combinatorial approach to testing. Example system requirements and corresponding models for achieving pair wise coverage were presented. The terminology and notation were from the AETG. This material was the foundation for Bryce's presentation in session one.

Bruno Legeard reported on a requirements-traceability process used successfully in Smart Card applications. The process based on annotating a formal model and then generating a traceability matrix calls for management of all the links among the requirements, the model, and the test cases. It is embedded in the LEIROS test generator. Effectiveness data on field application was presented.

Christopher Robinson-Mallett discussed "shortest distinguishing sequences" generated from a class of extended finite state machines. This algorithm is based on computational tree logic and can be integrated into any model checker that uses tree logic to generate test cases. Execution time complexity is not increased, but space requirements are.

Mikhail Auguston described the use of "attributed event grammars" to model environments surrounding real-time systems. Environment models represented in this way can automatically generate a large number of pseudo random tests. A prototype of a test generator that takes a grammar and generates a test driver in the C language was developed.

Panel Discussion I: *Black-Box Model-Based Testing – Current State and Future Vision moderated by Peter Lakey***.**

Peter Lakey introduced the panel discussion as a forum to (1) build awareness of current approaches to black-box, model-based testing (MBT), (2) evaluate the current state of practice, and (3) explore future directions.

Peter Becker described functional MBT, a technique that uses a cause-effect graphical model in automated test-case generation. This technique achieves complete functional coverage, with roughly an order of magnitude fewer test cases than typically required in manual testing. The modeling process itself often reveals over 50 percent of all detected defects.

Jim Hamadani presented a case study where functional MBT was applied to a web-based product configurator application. The testing efficiency of automated functional MBT enabled Sun to not only shorten release cycles by a factor of two but also reduce the defect escape rate.

Harry Robinson presented a case study of MBT applied to a Trouble Ticket Management System. Using a variety of test-case generation algorithms, tests were generated and run continually throughout the project. Only one defect was reported in the first eight months of deployment, where approximately 100 defects had been predicted, based on code volume.

Tom Swain described Model-Based Statistical Testing (MBST), emphasizing the usefulness of quantitative techniques available when testing is based on Markov chain usage models. Markov analysis of usage models facilitates model validation, enables quantitative test planning, and provides software reliability estimates.

Sid Dalal discussed model-based testing challenges, including modeling complexity versus design complexity, portability across operating systems and hardware, and the ability to adapt to various development methods.

Peter Lakey described application of MBT to a complex weapon system. Decomposing the usage model into 19 component models and developing a specialized tool set mitigated the model's complexity. The resulting modeling process was more efficient and less error-prone than methods that attempt to treat a complete system directly.

The panel session concluded with a lively discussion of the current state and future directions of MBT. Much of the discussion focused on barriers to the adoption of MBT, widely seen as (1) the failure of management to understand the value of testing in the first place; (2) the lack of appropriate metrics to measure testing value; (3) the lack of industrial-strength tools; and (4) inadequate training and education of current practitioners. The consensus seemed to be that, with automation support, the effectiveness of MBT for mission critical software is undeniable, and that its adoption would increase slowly as the aforementioned barriers are removed.

Keynote II. *Trustworthy System through Quantitative Software Engineering by Larry Bernstein, Senior Industry Research Professor, Stevens Institute of Technology.*

Larry Bernstein discussed the role of modeling in the overall software development process, with examples and anecdotes from the early Anti-Ballistic Missile program. He pointed out that a system model is the key component of requirements analysis. Once created and refined the model drives architecture, system design, and test-case design.

Session 4: Measurements

Mark Sherriff discussed a research program centered on constructing and validating a set of process metrics useful for predicting external system defect density. Metric suites for the Haskell were developed and field-tested. Field data collection and study methods were described for five independent variables, used in multiple regression analysis with random snapshots taken during the development cycle. Analysis showed that future defect densities could be predicted from historical data in the experimental situation.

Amit Paradkar presented a two-part paper: a review of different classes of model-based testing techniques, followed by results from two case studies comparing techniques in terms of fault detection effectiveness. Case studies with methods that use mutation and explicitly generate state verification sequences had better fault detection than did those based on boundary values and predicate coverage criteria. Discussion of the role of test objectives and the problem of defining objectives followed.

Alan Karr recasts performance analysis as a model-based experimental design and execution problem. Commercial systems have hundreds of configuration parameters, combinations of which affect performance. The approach moves some parts of performance analysis from the laboratory to end-user systems. The experience suggests that the approach has merit, but work remains to be done. Future research will focus on performance oriented test selection, optimization and variability reduction, and requirements satisfaction.

Xia Cai reported on hypotheses that predict the effect of code coverage testing on fault detection under different testing profiles. Using data from previous experiments, the work demonstrates that the effect of code coverage varies under different profiles. These findings need support from further experimental work.

Session 5: New Applications

Bob Binder presented a system for performing end-to-end testing of distributed mobile applications. The system uses model-based strategies to generate realistic test suites that represent external actors, system requirements, and the radio-frequency propagation environment of the system under test. Test input rates can be varied to represent aggregate workload. Post conditions can be specified for model parameters that can, in turn, be checked automatically with exception reporting.

Xing Li reported on up-front modeling of image processing systems in an environment of short development cycles and high failure costs. By separating out concerns related to hardware-application interactions, the architecture of the model enables software to be tested even before the hardware becomes available.

Peter Lakey reported on feasibility issues related to the use of model-based testing for a defense system. The large number of states in the model and the large number of test cases run in the demonstrations characterize the work. While demonstrating technical and economic feasibility, with significant benefits, the support tools need to be improved and extended before the application of model-based testing can achieve widespread among defense contractors.

Panel Discussion II: *Model-Based Testing – Making it Stick moderated b Bob Binder, mVerify Corporation.*

Panelists summarized his or her background and point of view on model-based testing. This was followed by a moderated discussion based on the following questions.

1. What can be done to reducing the degree of difficulty for MBT tool use?
2. How can MBT be integrated with other development environment capabilities?
3. How can we best support and achieve sufficient modeling skills to effectively use an MBT system?
4. How should we decide which SUT capabilities are/aren't a good fit with MBT?
5. How can we best maintain a model repository once it has been established?
6. How can we verify models and coordinate model development with other developers?
7. How can we generate cost-effective oracles from models?
8. How can we reduce rework costs resulting from changes in requirements and interfaces?
9. How can we produce meaningful metrics for model coverage and reliability growth?
10. How can we manage model configurations to support regression testing?

This panel was an excellent close to the workshop. Attendance remained at nearly full registration for the entire workshop