MoDeVVa 2010 Workshop Summary

Levi Lúcio[1], Elisangela Vieira[2], Stephan Weißleder[3]

[1] University of Luxembourg, Luxembourg
levi.lucio@uni.lu
[2] Alcatel-Lucent, France
elisangela.vieira@alcatel-lucent.com
[3] Fraunhofer FIRST, Germany
stephan.weissleder@first.fraunhofer.de

1 Modeling, Verification, and Validation

The MoDeVVa workshop series is focused on Model-Driven Engineering, Verification, and Validation.

Models are purposeful abstractions. They are used to support the focus on the important aspects and to make complex systems easy to understand. Beyond their use as documentation, models can also be used for automatic transformation or code generation. For this, a formal foundation of models with fixed semantics is necessary. One typical application is the automatic generation of large parts of system source code. The automation can result in a decrease of system engineering costs. Thus, the usage of models, model transformations, and code generation is becoming more and more important for industrial applications. As one of the most important representatives for the application of models, Model-Driven Engineering (MDE) is a development methodology that is based on models, meta models, and model transformations. There is already a lot of tool support for models, (domain-specific) modeling languages, model transformations, and code generation. The constant pace at which scientific and industrial development of MDE-related techniques moves forward shows that MDE is quickly changing and that new approaches and corresponding issues arise frequently. Most important, there is a crucial need for verification and validation (V&V) techniques in the context of MDE. Likewise, V&V is very important in many domains (e.g., automotive or avionics) and the use of MDE techniques in the context of V&V is an interesting topic. One prominent representative of this approach is model-based testing (MBT).

2 Objectives of the Workshop

The objective of the workshop on model-driven engineering, verification, and validation (MoDeVVa) in 2010 was to offer a forum for researchers and practitioners who are working on V&V and MDE. The main goals of the workshop were to identify the mutual impact of MDE and V&V: How can MDE improve V&V and how can V&V leverage the techniques around MDE? Thus, we asked for submissions that target the following areas:

- V&V techniques for MDE activities, e.g. V&V for model-to-model or model-to-text transformations;
- V&V at the level of the models: techniques for validating a model or generating test cases from models, including simulation, model-checking, model-based testing, etc.;
- V&V at the level of meta models: techniques for validating meta-models (languages) or for using meta-models to validate other artifacts;
- The application of MDE to validation, testing and verification;
- Impact analysis of (meta) model changes on validation, i.e. the result of a change in a (meta-)model on the previous results of validation;
- V&V techniques supporting refinement, abstraction and structuring;
- Difficulties and gains of applying V&V to MDE and vice versa;
- Case studies and experience reports;
- Tools and automation.

This year we especially encouraged the submission of papers on the most discussed topic in MoDeVVa 2009: the combination of model transformations and model-based testing.

3 Submissions and Acceptance in 2010

In 2010, there were 14 submissions to MoDeVVa from 8 countries. In the review process, we chose 9 papers on mainly three topics: *transformation verification*, *modeling* and *model-based testing*. The transformation verification session was the largest one, with topics such as the application of mutation analysis to transformation verification or the formal validation of an implementation of a transformation language. Given that transformations are taking on a fundamental role in MDE, their verification seems to be also assuming an important role in the community's research.

4 Discussions During the Workshop

One of the major topics was the applicability of MBT in safety-relevant domains: "Would you fly an airplane that had been tested ONLY by model-based testing?"

The use of a development based on models is a common practice in several domains of application. The applicability of model-based strategies in the validation phase, such as model-based testing, is less common in daily practice of large companies. The main question is: if MBT offers interesting advantages, why is it not widely applied? What are the thresholds to overcome?

This question has caught some attention of the academic community and also of the industrial one. In MoDeVVA 2010, the participation of Antti Huima as keynote speaker created an interesting opportunity for discussion. Some of the identified challenges that need to be overcome are:

- *The need for mature tools*
  One inconvenience of applying MBT in industry is that mature tools able to automatically generate test code are in general a costly investment. Worst, the return on investment is not immediately observed when the users are not familiar with models. On the other hand, in general academic centers develop their own prototype tools, languages, and small case studies. If this context allows valuable time to investigate new alternatives on modeling, algorithms, and strategies for test case generation and selection, it does necessarily promote of applying such approaches in industrial cases.
  One alternative to tackle this issue is the collaboration of both worlds by introducing MBT in industry using prototype tools. This scenario seems interesting: industrial reduces the risks on investments and academics have real case studies to analyze. In practice, however, this is not easy to implement. It requires both knowledge of tools proposed by research groups and identifying interested groups from the industry.

- *Does validation require formal verification?*
  Some points that seem natural steps for academics may seem expensive for industrials. For instance, the idea that models used for MBT purposes should be verified. This brings questions such as: what is the definition of verification? How far should we go in this phase? Why are MBT tools and model verification tools are seldom integrated?
  Those familiar with model verification know by experience that it is not possible to do test generation using a model that has not been verified. A minimum of verification is required, e.g., to detect deadlocks, live locks, or dead transitions.
  The practice in the industry shows that, except for organizations dealing with critical and safety properties, industrials do not use or even mention verification in formal terms (for instance, model checking). In general, it is expected that the test generation tool will verify some basic properties before launching any test generation. However, in general, only in tools that come from the academic world these checks seem to be explicit.

- *Modeling for test generation*
  MBT requires time for modeling. This phase can be long, according to the modeling level of the testers and the knowledge about the feature/system/use case to be modeled. One reason for this difficulty comes from the university years: engineers are prepared for modeling for the development phase but not for validation purposes.
  Another problem is the number of modeling languages proposed to describe a system. How to distinguish a model that describes the system under test from the model that is going be used for test generation? A possible response to this question is that we can restrict ourselves to particular modeling languages for which MBT can be done automatically and for which coverage metrics have been studied and are available.

- *Meta models, models, or code?*
  Many approaches, languages, tools, and methodologies for V&V and MDE have been proposed, but the trend seems to be on "keeping on proposing" rather than evaluating. A possible answer to this question would be to apply in larger scale at the industrial level or let the questions from industry permeate into research in a more "open" fashion, i.e. to rethink the research questions in terms of what is demanded from industry. This may also mean that a step up from current research is needed in order to find "meta" tools that will allow this research to take place.

- *Model transformations*
  Model transformations are becoming part of MBT. Transformations can be used in order to add information relevant for testing to models used by the development team. Development and validation phases may start from a common model, however, according to the features under test, it may be required to abstract some information or go further in detailing other information.
  And the problem may go even further. When we want the generation of test cases able to be automatically executed, it requires including information for code generation in the modeling phase. The model may become very complex and discourage beginners in this practice.
  An alternative, especially for those starting on MBT, is going step by step. First, modeling without taking into consideration the test execution phase. Second, when automation of test case execution is required, the model is transformed to include additional information for code generation.
  There again, tools developed in academia can be helpful. These tools are developed having other primary goals than the test execution phase. For instance, they are developed for the analysis of new algorithms for generating scenarios or for test case reduction. In addition, academics are interested in having case studies coming from different industrial domains. Depending on the industrial partner a distinct programming language may be used during test execution. The result is thus not automatically executed, but it can be if a test harness is associated to it.
  Another usage of model transformation is for coping automatically with model changes during the whole test phase.

- *Test case reduction*
  Model transformation has also been used to deal with test case reduction, test case selection, and even to avoid state explosion problem.
  The slicing technique, for instance, is a way to avoid the state explosion problem. From an initial model, several sub-models are sliced that will contain only the essential information for the test generation. Each slice or sub-model is generated according to the test objectives. Ideally, this slice is automatically done such that all sub-models put together recreate the global model. In addition we guarantee that all parts of the initial global model are in at least one sub-model.

Concerning the selection or reduction of automatically generated test cases, this is a problem that is very present in the industry even for manual test generation. Which test cases should be prioritized? How to select the most important ones? Another way to see this problem in MBT is: how to automatically identify a given requirement in a model? A possibility is that in the modeling phase the tester would be able to identify important transitions or states by tagging or labeling. The criteria for test generation from such labeling could be: cover all scenarios where a certain requirement is involved. The results may still require a phase of test case reduction, but using such a technique could provide an idea about how many tests are required to cover a specific requirement from a given model. Nowadays, few tools are able to provide these features.

5 Acknowledgments

We would like to thank all the authors for their contributions and high quality presentations. A big thank you also to the Program Committee for their help and advice while selecting the best contributions: Vasco Amaral, Paul Ammann, Benoit Baudry, Fabrice Bouquet, Ruth Breu, Gregor Engels, John Derrick, Alain Faivre, Mark Harman, Antti Huima, Eliane Martins, Mercedes Merayo, Roman Nagy, Alexander Pretschner, Holger Schlingloff, Dehla Sokenou, and Yves Le Traon.

Finally, we would like to especially thank Antti Huima for having helped us kicking off MoDeVVa with his keynote speech. His enthusiasm for MBT and his many application and research questions on the topic have added value to this year's edition of MoDeVVa.