Report on the 3rd MoDeVa Workshop – Model Design and Validation

Benoît Baudry[1], David Hearnden[2], Nicolas Rapin[3], and Jörn Guy Süß[2]

INRIA, France
University of Queensland, Australia
CEA/LIST, France

Abstract. Software systems are becoming increasingly large and complex, and run the risk of serious failures from unforeseen behaviour. Model driven development (MDD) is emerging as a solution with strong potential for dealing with these difficulties using models and model transformations. However, effective validiation and verification techniques are required to take full advantage of the expected benefits of MDD.

The MoDeVa (Model Design and Validation) series of workshop aims at bringing together researchers and practitioners to discuss links between MDD and model-based validation. This document summarizes the results of MoDeVa'06 that was the third edition of the workshop. Reviewing the workshop, the organisers feel that a community is forming which aims for practical integration of model-driven development and V&V and that specific research topics are being identified and addressed. As an illustration of this, it is important to notice that three papers this year were dedicated to V&V for model transformations. This trend may be due to increasing maturity and scale of use of MDD and thus to a better understanding of hard points and research challenges in this new approach for software development. There is recognition that specific solutions in this area are needed.

Workshop Formation

This year's edition of the MoDeVa series of workshops resulted from a merge of three separate workshop proposals. In addition to the continuation of the MoDeVa workshop, there was a proposal more focused on the Model-Driven Architecture and one other dedicated to model-based testing. 32 participants attended the workshop, despite competition posed by several other scientific gatherings with a similar topic (e.g. the MARTES workshop). The workshop included eight papers organized into three sessions dealing with V&V of models, V&V of transformations and advances in model-based testing. The session topics were used to structure the discussions. The workshop included an organised lunch with ensuing coffee in a bar that gave further opportunity for interaction.

V & V for Models

The session on V&V for models underscored the need to assign a mathematical semantics to UML, in order to apply traditional V&V methods. Works presented

T. Kühne (Ed.): MoDELS 2006 Workshops, LNCS 4364, pp. 179–181, 2007.

in this session underlined the need to identify a portion of the UML amenable to the application of formal semantics. The approaches presented carefully examine possibilities to manage state space growth, recognising it as the major property that can limit the adoption of those methods in practice. The ease of use of these approaches in an industrial context was also discussed. Indeed, adding formal contracts to models or transforming models to feed them into a model-checker are techniques that have been available for some years now but are still not widely adopted. Beyond the challenge of state space explosion, approach usability, i.e. the skill set required of developers in order to manipulate and build formal models, was perceived as a major issue.

V & V for Transformations

The session on V&V of transformations showed that the different approaches to the definition of transformations have significant impact on the preferred style of V&V. Declarative transformation languages, represented in the session by an approach based on triple graph grammars, enable the use of theorem provers. This provides great confidence in the correctness of the solution as crucial properties of the transformation are implicitly proven correct for all cases. Hence, this may be a good V&V approach for code generators that build critical code for embedded systems. However, two issues arise when using this approach. First, if the prover can not prove the property correct, it does not provide support to locate the fault in the transformation. Second, the transformation's specification has to be transformed into the target formalism of the prover, which may be non-trivial. Imperative transformations, exemplified by a business language transformation framework by IBM research, fare better with an intent-guided method of test-case generation, that provides clear counter-examples in the case of errors, but whose coverage is necessarily partial.

Testing

The session on model-based testing showed maturing approaches in this area that took into account the usability of these approaches by proposing lightweight solutions (based on contracts) for model-based test generation and the integration in the development process. The exact nature of a test model in the different approaches arose as a major point of discussion. Even if there seemed to be agreement on the need for a test model that is different from the design or implementation model, the content of this model or moment of its creation in the development cycle differed: a test model might be a refinement of the analysis model, a variant of the design model, or a completely independent model.

Debate

Following the three topic-based sessions, the final part of the workshop was dedicated to a structured debate on the thesis that: "MDA does not need V&V

or Testing". Participants were divided into three groups: one to attack the thesis, one to defend it and a third to judge the quality of the arguments.

An interesting observation in this session was that while participants generally recognised models as the basis of development, the term MDA appeared to be weakly delineated. In preparation for the debate, some immediately generalized MDA to the notion of model-driven development. Those that discussed MDA found it hard to agree on a common definition.

Evaluation

The workshop's format called for the presenters of each session to conclude by forming a common panel for general discussion. Questions on general aspects of the papers were to be delayed for this panel discussion. The panel discussions viability depended on the closeness of the subjects treated in the presentations, however the discussion managed to address general concerns in each domain. The panel discussions may have been even more constructive with more panel members. Both the panel discussions that took place (the Testing panel discussion was cancelled on the day due to time constraints) only had two panel members, however we were expecting the Transformation V&V and Testing V&V panels to have three members. Participants found the final debating exercise interesting and enjoyable.

The two best papers were awarded based on preferential votes cast by the participants, and these votes correlated very well with the scores from the paper reviews. The participant vote had the benefit of being a cheap and transparent form of measurement from a medium-sized population, however it meant that the votes were cast on the presentation rather than the paper. An alternative voting system incorporating the program committee may result in a more objective selection, however it would necessarily introduce a much greater burden on the program committee.

Summarily, the MoDeVa workshop has become a point of reference for the community around MDD and V&V. It is seeing steady growth in the community and helps to promote verification and testing techniques in the MDD domain and to ready them for use in large software development. The organisers feel that the investments into structure and the provision of a high quality environment has paid of through recognition by the participants.

To carry the workshop forward, a dedicated website at the University of Queensland has been established as an information point of reference for past and future MoDeVa editions. A poll of this years participants will be carried out to seek their opinion on this years edition and on recommendations for improvement. Their input will help to improve proposals for the next issue of MoDeVa, at MoDELS 2007 in Nashville, Tennessee.