Models and Evolution - ME2010

Dirk Deridder[1], Alfonso Pierantonio[2], Bernhard Schätz[3], and Dalila Tamzalit[4]

[1] Vrije Universiteit Brussel, Belgium
dirk.deridder@vub.ac.be
[2] University of L'Aquila, Italy
alfonso@di.univaq.it
[3] fortiss GmbH, Germany
schaetz@fortiss.org
[4] University of Nantes, France
dalila.tamzalit@univ-nantes.fr

1 Introduction

In general, software artifacts and applications are subject to many kinds of changes, which range from technical changes due to rapidly evolving technology platforms, to modifications in the applications themselves due to the natural evolution of the businesses supported by those software applications. These modifications include changes at all levels, from requirements through architecture and design, to source code, documentation and test suites. They typically affect various kinds of models including data models, behavioral models, domain models, source code models or goal models. Coping with and managing the changes that accompany the evolution of software assets is therefore an essential aspect of Software Engineering as a discipline.

In this context, models can play an important role. They can help and guide software evolution and can enforce and reduce critical risks and important resources (e.g., costs, personnel, time) involved in software evolution, by employing high-level abstractions. Models can thus help to direct evolution. Model-Driven Engineering (MDE) is an approach to software design and development in which models are the primary artifacts, and play a key role. The major objective of MDE is to increase productivity and reduce time-to-market by raising the level of abstraction and using concepts closer to the problem domain at hand, rather than those offered by programming languages. Models represent domain-specific concepts and conform to metamodels. A core task of MDE is the manipulation and transformation of models. Manipulating and transforming models can be very useful to manage software evolution. The objective is to enforce and reduce critical risks and important costs involved in software evolution, by employing high-level abstractions and by considering several facets.

Similar to traditional software engineering approaches, MDE is also susceptible to evolution. The MDE context poses unique challenges which require the conception and development of novel techniques, dedicated approaches, and advanced tool support. In fact, there is an increasing need for more research investigating disciplined techniques and engineering tools to support a wide range

J. Dingel and A. Solberg (Eds.): MODELS 2010 Workshops, LNCS 6627, pp. 180–183, 2011.

of model evolution activities, including model-driven software evolution, model differencing, model comparison, model refactoring, model inconsistency management, model versioning and merging, and (co-)evolution of models.

The different dimensions of evolution in the MDE context make the problem intrinsically difficult. This is because modifications can reflect coherent adaptations of correlated artifacts at several layers of the metamodeling architecture and at several levels of abstractions. For example, some well-formed rules can be invalidated when a metamodel evolves. The same happens with the associated model transformations. Furthermore, model adaptations should be propagated to artifacts interconnected by means of model transformations. Finally, evolution of model transformations should be reflected in both source and target models. In addition, the exploitation of differences is an appropriate solution for version management, because in general the complete system model is far larger than the modifications that occur from one version to another.

Furthermore, there is a substantial difference between the modeling of evolution and the evolution of models. There are plenty of works on the former topic, while the focus of this workshop is on the evolution of models, hence its name: “Models and Evolution” (ME). ME 2010 was the result of merging two successful series of international yearly workshops that were in existence since 2007 (for the MoDSE workshop), and 2008 (for the MCCM workshop).

2 Workshop Contributions

In addition to the content-wise objectives described above, one of the main goals of the workshop series is to provide an open discussion space where the MDE and software evolution communities can meet on a yearly basis. As usual we also encouraged young researchers to participate and submit their work to get into contact with this growing community.

The full-day workshop included four thematic sessions which we used to group the presentations and focus the discussions:

Semantics

- **A Manifesto for Semantic Model Differencing**,
  by Shahar Maoz, Jan Ringert and Bernhard Rumpe
- **Towards Semantics-Preserving Model Migration**,
  by Markus Herrmannsdoerfer and Maximilian Koegel
- **Documenting Stepwise Model Refinement using Executable Design Decisions**,
  by Matthias Biehl

Model merging

- **Representation and Visualization of Merge Conflicts with UML Profiles**,
  by Petra Brosch, Philip Langer, Martina Seidl, Konrad Wieland, Manuel Wimmer, Horst Kargl and Gerti Kappel

- **Decoupling Operation-Based Merging from Model Change Recording**,
  by Stephen Barrett, Patrice Chalin and Greg Butler
- **Merging Model Refactorings - An Empirical Study**,
  by Maximilian Koegel, Markus Herrmannsdoerfer, Otto von Wesendonk and Jonas Helming
- **The Case for Batch Merge of Models Issues and Challenges**,
  by Lars Bendix, Maximilian Koegel and Antonio Martini

Model Inconsistencies, Model Differences

- **Automated Planning for Resolving Model Inconsistencies: A Scalability Study**,
  by Jorge Pinna Puissant, Ragnhild Van Der Straeten and Tom Mens
- **RCVDiff - a stand-alone tool for representation, calculation and visualization of model differences** ,
  by Zvezdan Protic, Mark van den Brand and Tom Verhoeff

Meta models and transformations

- **Semi-Automated Correction of Model-to-Text Transformations**,
  by Gábor Guta, András Pataricza, Wolfgang Schreiner and Dániel Varró
- **Transformation Migration After Metamodel Evolution**,
  by David Méndez, Anne Etien, Alexis Muller and Rubby Casallas
- **Towards Metamodel Evolution of EMF Models with Henshin**,
  by Stefan Jurack and Florian Mantz
- **Comparing Model-Metamodel and Transformation-Metamodel Co-evolution**,
  by Louis Rose, Anne Etien, David Méndez, Dimitrios Kolovos, Richard Paige and Fiona Polack

As a tradition in the workshop series we always organise a plenary debate. The basis for these debates comes from the questions and challenges provided by the authors, program chairs, and participants. Below we briefly summarize the ones that were used to guide and fuel this year's discussion(s):

- Why don't we focus more on proactive support for model evolution (possibly facilitated and backed up by a MDE process), instead of fixing problems after they occur?
- How to deal with the semantic aspect of model evolution?
- How to preserve model semantics when evolving models, meta-models, transformations, ...?
- When to use semantic model differencing instead of syntactic model differencing?
- How to manage model-metamodel co-evolution and transformation-metamodel co-evolution? And what about the link between both?
- Isn't it time to move towards a unification of model evolution approaches (terminology, frameworks, ...)? And how would this impact/advance our research (and collaborations)?

- There is still a lot of work to be done in resolving the problems detected when evolving models. Knowing the problem is one thing, resolving it another?
- Wouldn't a benchmarking methodology/tool be benificial for comparing our different approaches?
- Which types of models do we want to consider in ME research (e.g. physical models, process models, domain specific models, design models, hardware models, ...)?
- What are the limitations of evolution approaches with respect to the kinds of models they can address and is it possible to make these more generic?
- How to make models and transformations more resilient to changes in the meta-models? Is there a way to encapsulate changes (e.g. in a first class artifact) and apply them to all impacted artifacts?
- How to deal with ME when confronted with a heterogeneous set of meta-models?
- What about large scale models and evolution?

3 Summary and Outlook

The Models and Evolution workshop series is maturing, and every year we witness an increase of interest by the MoDELS community. This is clearly evidenced by the steady growing number of participants that we are happy to welcome each year (over 50). The quality of the submissions we received was exceptionally high, which is why we have set up a theme issue on Models and Evolution in the Springer *Software and Systems Modelling* journal (`www.sosym.org`). Given the fact that the Models and Evolution research community is a very lively one, we are looking forward to learn about the results in future editions of the ME workshops. Full versions of the presented contributions are available online at `www.modse.fr`.

Acknowledgements

We thank our programme committee members for their help in the paper selection process: Hubert Dubois, Jeff Gray, Tom Mens, Pierre-Yves Schobbens, Stefan Wagner, Arnaud Albinet, Benoit Baudry, Mireille Blay-Fornarino, Jean-Michel Bruel, Jordi Cabot, Rubby Casallas, Antonio Cicchetti, Davide Di Ruscio, Anne Etien, Jean-Marie Favre, Mehdi Jazayri, Gerti Kappel, Udo Kelter, Olivier Le Goaer, Richard Paige, Antonio Vallecillo, Ragnhild Van Der Straeten, Hans Vangheluwe, Dennis Wagelaar.