Report on the International Workshop on Comparison and Versioning of Software Models (CVSM 2013) February 27, 2013, Aachen

Udo Kelter, Pit Pietsch
Electrical Engineering and Computer Science
University of Siegen, D-57068 Siegen
{kelter,pietsch}@informatik.uni-siegen.de

Jan Oliver Ringert
Software Engineering
RWTH Aachen, D-52074 Aachen
ringert@se-rwth.de

Abstract

This report summarizes the main results of the International Workshop on Comparison and Versioning of Software Models (CVSM 2013) February 27, 2013, Aachen.

1 Introduction

This edition of the CVSM workshop series was a follow-up action of plans developed at the previous edition, CVSM 2012 in Essen (co-located with ASE 2012). The main goal was to advance the state of the art in the area of model versioning by developing benchmarks and later organising tool contests. As a first step in this direction, CVSM 2013 was planned to discuss proposals of benchmarks for model versioning tools.

A Call for Benchmarks (s. next section) was issued in advance of the workshop and resulted in 10 submissions. The workshop attracted 25 participants from 4 countries and around 10 academic working groups and/or commercial enterprises.

The workshop was co-located with the conference "Software Engineering 2013", organized by the Software Engineering Group at RWTH Aachen University. The CVSM workshop organisers would like to thank the local organisers and their team for the excellent organization and service and for making sure everything went smoothly.

2 Call for Benchmarks

The call for benchmarks [1] invited three types of benchmarks:

Performance Benchmarks: Performance benchmarks are intended to measure the runtime of algorithms and to compare performance-related properties of different approaches. They typically involve large models because performance-related practical problems do not occur with small models.

Challenges: Challenges are usually small, artificially created test cases which can be used to highlight certain quality aspects of algorithms. Challenges usually reveal problems in state-of-the-art tools and algorithms. A set of challenges can be used to identify implicit or explicit constraints that exist for the algorithms. Furthermore they help evaluating whether an algorithm is a good choice in a given application context.

Real Use-Cases: Real use-cases must stem from real-world application scenarios and projects. Real use-cases should help assessing the usefulness of known algorithms in the context of real application scenarios or trigger research into new, better algorithms.

3 Discussion of Submissions

A total of 10 benchmark proposals were submitted, among them 2 performance benchmarks (no. 9 and 10 below) and 8 challenges:

1. Pit Pietsch: Model Matching Challenge: Moving Elements
2. Klaus Müller, Bernhard Rumpe: Model Matching Challenge: Moving Elements
3. Pit Pietsch: Model Matching Challenge: Renaming Elements

4. Petra Brosch, Martina Seidl, Magdalena Widl: Semantics-Aware Versioning Challenge: Merging Sequence Diagrams along with State Machine Diagrams
5. Timo Kehrer: CVSM 2013 Challenge: Recognizing High-level Edit Operations in Evolving Models
6. Timo Kehrer, Udo Kelter, Dennis Koch: CVSM 2013 Challenge: Model Patching
7. Manuel Wimmer, Philip Langer, Vienna University of Technology: Benchmark for Model Matching Systems: The Heterogeneous Metamodel Case
8. Philip Langer, Manuel Wimmer, Vienna University of Technology: A Benchmark for Conflict Detection Components of Model Versioning Systems
9. Pit Pietsch: A Benchmark Set: for the Evaluation of Model Differencing Algorithms
10. Pit Pietsch, Hamed Shariat Yazdi: A Benchmark Set to Assess Scalability and Runtime Aspects of Model Versioning Algorithms

The first three submissions by Pietsch and Müller et al. cover real use-cases from two independent research projects where state-of-the-art matching algorithms produced differences whose quality was perceived as too low. In fact, the quality of some results was so low that the differences were unusable in the context of the given research question. The reasons of the low-quality turned out to be a combined occurrence of certain edit operations, implicit assumptions inherent in the matching algorithms and subjective preferences of the model developers. It also became clear that these examples are general matching problems which occur in most model types. Because of the similar content the authors decided to merge their results in [2].

Submission 4 [3] by Brosch et al. describes a novel approach for merging two sequence diagrams in cases where a corresponding state machine is available. The semantic information contained in the state machine can be used in the merging process to produce a merged version of the sequence diagrams which is consistent with the state machine. The approach is presented in more detail in the current issue of the newsletter.

Kehrer et al. [4] discuss in Submission 5 problems wih low-level differences, i.e. differences which are based on elementary operation on the Abstract Syntax Graph (ASG) representation of the model and which do not use the (high-level) edit operations developers are used to from their editors. Low-level differences are often hard to understand for model developers. The challenge presents test cases for accessing the quality of algorithms which detect high-level edit operations in differences.

Kehrer et al. describe in Submission 6 a challenge for model patching tools. A patch is essentially the difference of two revisions of a model which is prepared to be applicable to a third model. Patching occurs, e.g., in the evolution of product families where changes in one branch have to be added to one or more concurrent branches. However, the third model is independently developed and elements which are necessary for the execution of a patch can be altered or missing. The challenge in this scenario is to resolve such cases. In principle, the developer has to interfere and adapt the patch appropriately.

Submission 7 [5] by Wimmer et al. discusses another challenge for model matching algorithms. Matching algorithms usually work on different versions of one model, i.e. models of the same type. However, there are also cases where heterogeneous models have to be compared. [5] proposes a benchmark consisting of 10 scenarios based on independently designed versions of meta models where the expected result, i.e. the correct matching, was manually controlled.

Langer et al. present in Submission 8 [6] a synthethic benchmark which allows conflict detection functions in model merging tools to be automatically evaluated regarding their accuracy and execution time. This benchmark can be used by state-based as well as operation-based approaches. Several state-of-the-art conflict detection tools were evaluated using this benchmark.

Submission 9 by Pietsch is a proposal for a performance benchmark. The benchmark set contains 135 pairs of synthetically created models. Since these pairs were generated by a model generator the exact evolution as well as the correct matching are known. Hence, this benchmark can be used to assess the quality of model differencing algorithms.

The Submission 10 by Pietsch et al. contains 6599 models which are reverse engineered from large open source projects. Because these models are rather large, some contain even more than 25k elements, they can be used to assess the scalability and runtime of model versioning tools.

All presentations were intensively discussed at the workshop. All authors were invited to submit revised, possibly united versions of their proposals for publication in "Softwaretechnik-Trends", the newsletter of the SIG on Software Engineering within the German Informatics Society.

4 Conclusions and Further Plans

Based on the submissions and discussions at the workshop it was decided to join the efforts of the community and create a *Model Matching Challenge* (MMC). The MMC is intended to evaluate and compare the capabilities of current state-of-the-art matching algorithms in a way that the strengths and weaknesses of the different approaches become visible. These results will give model developers guidance when they have to decide on a model-matching tool for their specific use-case. The first version of the MMC will contain four different kinds of benchmarks:

- Challenging edit operations, e.g. as presented in [2].
- Model evolution benchmarks from real projects, i.e. if available.
- Synthetically created model evolution benchmarks, e.g. as presented in Submission 9.
- Other types challenges, e.g. the heterogeneous metamodel case [5].

All benchmarks test data will be represented as Ecore models, which is the de facto standard technology in model driven engineering. Each benchmark will define the expected outcome as well as alternative and wrong solutions. The first version of the MMC is planned to be available in late September 2013.

Developers and users of model comparison tools and algorithms will be invited to evaluate available state-of-the-art model comparison tools and algorithms using these benchmarks. Results of these evaluations should be submitted to, and will be discussed at, the next edition of the CVSM workshop series in Kiel in late February 2014.

All participants of the workshop will also have the chance to discuss and revise the first version of the MMC based on the experience gained during this first iteration. Furthermore, every member of the community will be invited to submit new benchmark proposals for inclusion in the MMC.

In case of other challenges discussed at the workshop, e.g. semantic aware merging [3] and detection of complex edit operations [4], chances for a successful tool contest were considered too low at this time because the number of tools which support the specific functionality appears to be too small or because the benchmark needs to be further refined and elaborated. These decisions may be revised in the future when new, refined proposals are made by the community or when the state of the art is more advanced.

References

[1] Call For Benchmarks, http://pi.informatik.uni-siegen.de/CVSM2013/cvsm13_cfp.pdf

[2] Pietsch, P.; Müller, K., Rumpe, B: Model Matching Challenge: Benchmarks for Ecore and BPMN Diagrams, Aachen; 2013

[3] Brosch, P.; Seidl, M., Widl, M: Semantics-Aware Versioning Challenge: Merging Sequence Dia- grams along with State Machine Diagrams, Aachen; 2013

[4] Kehrer, T.; Gerth, C.: CVSM 2013 Challenge: Recognizing High-level Edit Operations in Evolving Models, Aachen; 2013

[5] Wimmer, M; Langer, P.: Benchmark for Model Matching Systems: The Heterogeneous Metamodel Case, Aachen; 2013

[6] Langer, P.; Wimmer, M: A Benchmark for Conflict Detection Components of Model Versioning Systems, Aachen; 2013