Assessing the Quality of Model Differencing Engines

Pit Pietsch, Hamed Shariat Yazdi, Udo Kelter, Timo Kehrer
Software Engineering Group
University of Siegen, Germany
{pietsch, shariatyazdi, kelter, kehrer}@informatik.uni-siegen.de

ABSTRACT

In recent years many tools and algorithms for model comparison and differencing were proposed. Typically, the main focus of the research laid on being able to compute the difference in the first place. Only very few papers addressed the quality of the delivered differences sufficiently. Hence, this is a general shortcoming in the state-of-the-art. Currently, there are no established community standards how to assess the quality of differences and it is neither possible to compare the quality of different algorithms, nor can developers decide whether or not an algorithm is able to produce adequate results in a given application scenario. We propose a parallel working session to be held to discuss this general problem and its implications. The goal of the working session is to achieve a common understanding of what the crucial factors in assessing the quality of differences are. Furthermore, it is planed to discuss possible solutions that help the research community as whole, e.g. by drafting the design of an initial benchmark corpus which later could be turned into a standardized, openly available benchmark set.

1. MOTIVATION

Many different algorithms for model differencing were proposed in recent years; surveys can be found in [5, 6, 7]. Typically, the main emphasis of these approaches is focused on the algorithms used to compute the difference between two revisions of a model. The evaluation of the algorithms, if conducted at all, is usually based only on sets consisting of few, small and sometimes even specially created, test models. Obviously, such an evaluation is not enough to assess the quality of the algorithms objectively, nor is it possible to compare the quality of different algorithms this way. Hence, developers which must choose a model differencing engine in their day-to-day work can not make informed decisions which tool fits their needs best.

The reasons why very few papers proposing model differencing algorithms address the quality of the delivered differences sufficiently are manifold:

- Ultimately, the quality of a difference can only be assessed in the context of the use case in which the difference is used. If the difference between two models is to be displayed to developers then understandability and compactness [2] are highly relevant; in the context of merging, the avoidance of merge conflicts is important. If a difference is to be used for internal delta storage or for batch patching of models, none of the above properties are relevant and it is sufficient to simply save the smallest representation of the difference.

- Properties such as *quality* and *understandability* are not generally defined and must be refined further for each specific model type and paradigm.

- Test models where the evolution is sufficiently known and documented are not available for many domains. While model generators [8, 9] can be used to create realistic test models synthetically, configuring these tool is time intensive and requires in-depth knowledge of edit processes for a given domain.

- Some generic algorithms make inherent assumptions on how models are composed, edited and compared. These assumptions are hard-wired into the code. Obviously, this leads to situations where such optimized algorithms perform well for certain model types, but fail for others [10].

- Adaptable differencing algorithms and engines do not determine properties of differences by themselves; in fact, the quality of the delivered differences depends strongly on the choice of the configuration options. Thus, the responsibility for the quality is passed on to the configurator of the engine, i.e. the person in charge of configuring the differencing engine and integrating it in the environment. Often configurators have a different perception of the quality of differences than developers, which can lead to unsatisfactory situations.

2. CONCLUSION

These research questions are addressed recently in the QuDiMo project[1] [4]. The main goal of QuDiMo is to asses and compare the quality of differences computed by the state-of-the-art differencing algorithms. One step to achieve this is to

[1]Which is supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under grant KE499/5-1.

create standardized benchmark sets which will be available to the research community.

Because there is not one differencing algorithm that works well in all use cases [11] the benchmark should be as diverse as necessary to reflect the different application scenarios. Furthermore, different types of models show vastly different requirements for model comparison, e.g. structural, hierarchically ordered model types like class diagrams in contrast to behavioural, flat model types like activity diagrams. Hence, the benchmark sets should cover model types with different characteristics.

We propose a parallel working group for the CVSM workshop in which quality aspects of model differencing are discussed. In particularly, questions like

- Is the quality of differences delivered by the state-of-the-art model comparison algorithms sufficient? If not, which are the determining factors for their failure?
- Which quality measures can be used to assess the quality of differences?
- How can the quality of model differencing algorithms and their computed differences be better assessed?
- Can the use of technologies like test model generators help to improve the quality of delivered differences?
- Is it possible to establish standardized benchmark sets in the model differencing community? If so, what is a desirable design for these benchmark sets. What are the obstacles that must be overcome to establish a benchmark set that is supported by the majority of the community?
- Can a model differencing challenge, similar to the MSR mining challenge, be established in the community? If so, which rules, date formats, etc. can be commonly agreed on?

To address these open questions it is of particular interest to not only hear opinions from researchers, but also from practitioners.

3. REFERENCES

[1] Bibliography on Comparison and Versioning of Software Models; http://pi.informatik.uni-siegen.de/CVSM

[2] T. Kehrer, U. Kelter, and G. Taentzer, "A rule-based approach to the semantic lifting of model differences in the context of model versioning," in *ASE*, 2011, pp. 163–172.

[3] Kolovos, D.S.; Ruscio, D.D.; Pierantonio, A.; Paige, R.F.: Different Models for Model Matching: An Analysis Of Approaches To Support Model Differencing; p.1-6 in: Proc. 2009 ICSE Workshop on Comparison and Versioning of Software Models; IEEE; 2009

[4] P. Pietsch and H. S. Yazdi, "The QuDiMo Project," `http://pi.informatik.uni-siegen.de/qudimo/`, 2011, [Accessed 26-June-2012].

[5] S. Förtsch and B. Westfechtel, "Differencing and merging of software diagrams - state of the art and challenges," in *ICSOFT (SE)*, 2007, pp. 90–99.

[6] D. S. Kolovos, D. D. Ruscio, A. Pierantonio, and R. F. Paige, "Different models for model matching: An analysis of approaches to support model differencing," in *CVSM 09*, 2009.

[7] P. Selonen, "A review of UML model comparison techniques," in *Nordic Workshop on Model Driven Engineering*, 2007.

[8] P. Pietsch, H. Shariat Yazdi, and U. Kelter, "Generating realistic test models for model processing tools," in *ASE 11*, 2011, pp. 660-623.

[9] P. Pietsch, H. Shariat Yazdi, and U. Kelter, "Controlled Generation of Models with Defined Properties," in *SE 12*, 2012.

[10] P. Pietsch, S. Wenzel, "Comparison of BPMN2 Diagrams," in *4th International Workshop on the Business Process Model and Notation* , 2012.

[11] T. Kehrer, U. Kelter, P. Pietsch, and M. Schmidt, "Adaptability of Model Comparison Algorithms," in *ASE 12*, 2012.