Operation-based Model Differencing meets State-based Model Comparison

Timo Kehrer, Udo Kelter, Pit Pietsch, Maik Schmidt
Software Engineering Group
University of Siegen, Germany
{kehrer, kelter, pietsch, mschmidt}@informatik.uni-siegen.de

ABSTRACT

The computation of a difference between two models is the most basic function of model versioning tools. Two fundamental approaches to obtain model differences have been proposed: Operation-based differencing and state-based comparison. They are commonly regarded as disjoint and thus evolved mostly independent of each other. Recent advances in model comparison, which lift low-level differences to invocations of complex editing commands, brought both approaches closer together. The overall goal of the CVSM working session proposed in this paper is to analyse common challenges and to derive conceptual and tooling-related issues which offer the potential for future collaborations of researchers originating from working groups of both approaches.

1. INTRODUCTION

A lot of research into methods and algorithms for model version management was stimulated in the last ten years: the bibliography [2] compiles about 300 publications in this field, most of them dating from 2003 or later. The computation of a difference between two models is the most basic required function; two main approaches to obtain such differences have been proposed: operation-based differencing and state-based comparison.

Operation-based approaches, e.g. [3, 6, 8], basically obtain model differences by "recording" the editing steps that are performed by a modeler. The obvious idea is to exploit the logging facilities in syntax-based editors to generate change logs, i.e. asymmetric differences of the models that are being edited.

State-based methods for model differencing are only based on the state of the models which are to be compared, a survey can be found in [5]. Virtually all of these methods consider models as graphs. Their basic processing structure is to compute a set of corresponding elements; elements not involved in a correspondence are considered to be deleted or created. Structural model comparison approaches initially deliver low-level differences which are often hard to understand. Thus, first approaches to lift these low-level differences to invocations of user-level edit operations have been proposed recently [4, 7, 1, 9]. Lifting low-level changes to invocations of complex editing commands inherently leads to asymmetric differences which basically have the same conceptual content as edit logs which are obtained by operation-based differencing approaches. Consequently, a set of common challenges of both approaches can be identified.

2. COMMON CHALLENGES

Model versioning approaches based on user-level edit operations assume an asymmetric notion of a difference, no matter whether the difference was obtained by logging edit commands or by lifting low-level differences. Thus, operation-based differencing and model comparison share a set of common challenges. The following incomplete list of common challenges shall serve as input for in-depth discussions at the workshop.

2.1 Identification of Meaningful Edit Operations

Model editors should offer the same set of edit operations which is used to explain the difference between two models. Edit operations which are offered by standard model editors most often are still elementary and mainly oriented at the ASG representation of the underling models. However, developers prefer changes to be explained in terms of conceptually meaningful edit operations including refactorings and other complex edit operations. In principle, one can construct arbitrarily many complex edit operations from elementary ones. However, most of them are not useful for the aim to make model editing more convenient or to make model differences better understandable.

A first important criterion for selecting a complex operation is that it is easier to understand than the single contained elementary edit operations. This criterion depends obviously a lot on the modeling language and in addition on design and usage rules and patterns, which may be project-specific. A second important restriction is that the set of complex edit operations must be small enough to be understood by developers. Thus, the identification of the relevant edit operations, especially in terms of complex edit operations, is rather a conceptual question than being a technical issue related to the ASG-based representation of models.

2.2 Visualising Asymmetric Differences

Most use cases that are based on the differencing of models require an adequate presentation of the obtained differences. If model differences are inherently asymmetric, we need alternative presentation concepts that differ from the well-known integrated parallel display metaphor. Simple textual edit scripts, such as the result of the Unix diff command, are not applicable to models with graphical representations, as they cannot adequately handle nameless model elements which have no user-readable identifier.

2.3 Editing of Differences

In some cases, differences must not be simply visualized but have rather to be edited. For example, if differences are considered as patches, irrelevant changes might have to be excluded from the patch or a modified patch with a subset of the original changes must be produced. A particular challenge is to keep patches consistent when being edited.

2.4 Handling of Conflicts

Use cases which lead to the synthetisation of new models, namely patching and merging, involve the recognition and resolution of merge conflicts. This is still challenging for models, as consistency requirements are much higher than in case of textual documents. Consequently, handling conflicts should be lifted at the level of complex edit operations.

3. CONCLUSION

Research on model differencing has lead to two fundamental approaches, logging of edit commands and state-based model comparison. They are often regarded as disjoint and thus evolved mostly independent of each other. This workshop session is clearly not intended to recall a discussion about the advantages and disadvantages of both approaches. In fact, both approaches share a set of common challenges, especially if state-based methods are extended by lifting low-level differences to invocations of complex editing commands, which usually are inherently asymmetric.

Consequently, we propose a working group for the CVSM workshop which brings researchers from both fundamental approaches to model differencing together. The agenda for the intended workshop session might look as follows: Firstly, common challenges of the approaches will be analysed in more detail. Secondly, the working group shall elaborate a set of common problems/issues which offer the potential for future collaborations of researchers originating from both approaches. The following incomplete set of questions may guide the discussions at the workshop:

- Can we share and maintain a common repository of edit operations/edit rules for the most common modeling languages (UML, Matlab/Simulink, etc.)?
- How can transformation rules that implement conceptual edit operations be loosely integrated into operation recorders of logging-based difference architectures?
- Are the architectures of both types of difference tools (operation-based and state-based) flexible enough to mutually exchange components of later phases of the difference tool processing pipeline, e.g. difference visualization, patching or merging?
- State-based approaches to model differencing have already proposed several difference models to be used for difference representation. Can we also establish a standard representation for asymmetric differences?

4. REFERENCES

[1] Brosch, P.; Langer, P.; Seidl, M.; et al.: An Example is Worth a Thousand Words: Composite Operation Modeling By-Example; p.271-285 in: Proc. Intl. Conf. Model Driven Engineering Languages and Systems 2009, Denver; LNiCS 5795, Springer; 2009

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

[3] Herrmannsdörfer, M.; Kögel, M.: Towards a Generic Operation Recorder for Model Evolution; p.76-81 in: Proc. 1st Intl. Workshop on Model Comparison in Practice, Malaga; ACM; 2010

[4] Kehrer, T.; Kelter, U.; Taentzer, G.: A Rule-Based Approach to the Semantic Lifting of Model Differences in the Context of Model Versioning; p.163-172 in: Proc. 26th IEEE/ACM Intl. Conf. Automated Software Engineering (ASE 2011); ACM; 2011

[5] 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

[6] Kögel, M.; Helming, J.; Seyboth, S.: Operation-based Conflict Detection And Resolution; p.43-48 in: Proc. 2009 ICSE Workshop on Comparison and Versioning of Software Models, May 17, 2009, Vancouver; IEEE; 2009

[7] Könemann, P.: Capturing the Intention of Model Changes; p.108-122 in: Proc. Intl. Conf. Model Driven Engineering Languages and Systems 2010, Oslo, Part I; LNCS 6394, Springer; 2010

[8] Schneider, C.; Zündorf, A.; Niere, J.: CoObRA - a small step for development tools to collaborative environments; Workshop on Directions in Software Engineering Environments at ICSE 2004, Edinburgh; 2004

[9] UML2 specific extensions to EMF Compare; `http://bit.ly/iNeV3q`