# Representation of model differences

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

## ABSTRACT

Model-driven development requires a full set of development tools. While approaches and tools for constructing most aspects of model driven tool chains are readily available, there is a lack of tools and well evaluated approaches which compute and/or processes model differences. Difference tools for models must be adapted to different types of models, use-cases and application contexts. So the tools and approaches must be realized every time completely anew according different algorithms and functions. As a possible solution, we propose a common difference representation and data exchange format as a foundation to integrate and evaluate individual activities. Thus we propose to analyze common and individual requirements and discuss issues of the state-of-the-art difference representation. With this we can support future collaborations of researchers, integrate approaches, and enhance interoperability leading to more efficient research.

## 1. INTRODUCTION

Model-driven development strongly depends on good tool support for the comparison and versioning of software models. Thus, a lot of ongoing research activities have been stimulated in this field in the last decade. Most challenges and problems such as model matching, difference visualization, patching or merging are directly or indirectly related to some notion of a difference. Prototypical implementations which are needed to evaluate the feasibility of research approaches are even based on concrete representations of model differences. However, the CVSM community has not yet established a "standard" representation of model differences. This lack of a common representation of model differences causes a number of serious problems:

- Different approach-specific terminologies or meanings lead to misunderstandings and confusion.
- Model differences cannot be exchanged between different tools.
- Tool components or user interface concepts cannot be flexibly exchanged or reused.
- Functions that provide solutions to particular subproblems cannot be integrated or developed independently. On the one hand providing comprehensive solutions for end users is not yet possible in an ideal way. On the other hand, rapid prototyping of research approaches has serious limitations and shortcomings.
- Difference representations are often not use-case agnostic and are based upon assumptions with respect to the functions which are operating with differences[1]. Thus, results of different approaches are often hardly comparable and objective benchmarks are difficult to implement.

etc. These circumstances lead to inefficient research. Consequently, we propose a CVSM working group that aims at integrating existing models of difference representations and initiating to elaborate a common, extensible, generic difference model. A very brief state-of-the art survey is given in Section 2, the agenda outline for the workshop is sketched in Section 3.

## 2. STATE-OF-THE-ART

Most approaches provide a use-case, as well as an adapted algorithm working on specifically designed data structures. Generally, the proposed representations can be divided into three categories.

*Static Difference Representation Approaches.* This category includes approaches which assume a fixed model type and a use-case specific representation of the data. Some approaches assume non-model-based representations, e.g. implementation of models with usual data structures such as lists, sets, bags, maps and trees. Especially commercial tools in closed environments and tool-chains without need for external data-exchange or adaptations are typical examples of this approach.

Further, even model-based approaches can be tailored to specific tools by integrating additional information into the meta-model or by discarding constraints. For example, Niere [5] has extended the Fujaba meta-model by additional, visualization related elements such as DiffUMLClass, DiffUMLAttr, and DiffUMLAssoc. These kinds of approaches are typically efficient and effective. However, they are hard to follow for external persons and to maintain and extend.

*Dynamically Created Difference Models.* The universal model-based approach proposed by Cicchetti [2] uses a data

[1]For example computation, representation and visualization of model differences are often strongly related and restricted to each other even in the same use-case.

model of differences which is dynamically generated from the meta-model of the compared models. This approach supports four generic edit operations, namely insertions, deletions, and modifications of elements. Edit operations for a concrete element type are modelled as subtypes of this element type; the data model of differences thus contains four subtypes for each concrete element type, which are generated by a transformation of metamodel. The data model of differences will thus increase dramatically and becomes incompatible with the original. Thus, generically operating with model differences is not possible and related components such as graphical user interfaces have to be generated. However the generation of such functions or components is not always possible due to the granularity of modifications, redundancy, and missing movement information.

*Generic Difference Models.* The most universal encoding to be used with different model types and use-cases is a static but generic difference model. This idea has been proposed several times. Rivera und Vallecillo proposed an approach [1] similar to [2]. They do not change the meta-model of the models, but classify the changes in the same way. However, information of unchanged model elements is not available and the need for duplication of modified elements leads to redundancy. Further the assumed traceability of such elements by their id-properties is not always guaranteed and dependencies between modifications are not regarded.

In the Eclipse framework EMF Compare [3] is used for computation and visualization of model differences. In this use-case several components as tree views, matching-algorithms, and difference computation are connected via their difference model. This model limits differences to those representable in tree visualization and the needs of merge purposes. In consequence some modifications, like shiftings or rearrangements cannot be expressed.

Könemann [6] considers model differences as patches. The difference is expressed as a more "fuzzy" specification of the desired changes, i.e. atomic changes are grouped into higher-level changes which capture a user's intention. Model elements serving as parameters of changes are identified through symbolic references. Thus, differences are independent of the original models of which they have been obtained from. The atomic change types are additions, removals, and updates of model elements, their attributes, or references. Other possible change types, for instance order changes of an ordered reference, are currently not supported. Hence this approach is clearly focused on patching models. Consequently this representation is not suitable to visualize differences, since typical visualizations as parallel views require exact references to the changed elements.

Another generic approach to difference representation is presented in [4]. Semantic change sets are used to group low-level changes into representations of user-level operations. The main intention of the approach is to provide a methodology to consistently integrate model editing and model comparison technologies. Thus, the types of low-level changes defined by the difference model are adapted to the capabilities of the technology which is used to implement the edit rules for a dedicated modeling language.

# 3. FURTHER WORK

We can conclude that state-of-the-art approaches to difference representation significantly differ from each other. Most difference representations are focused on a small set of dedicated use cases, sometimes only a single use case is really supported. Thus, a set of implicit assumptions restrict them from being adopted by other use cases or functions. We propose to establish a generic difference model that is capable of supporting all difference-related use cases in the domain of model versioning. Such a model should be the foundation to establish a standard exchange format, this will allow different components to interoperate as well as different approaches to be evaluated and compared . Existing approaches to generic difference representation should be fully included and mappings of their respective difference models into a universal difference model must be provided. This also includes different aspects and abstraction-levels of a model difference. In order to achieve such a generic difference model, different use-cases and research areas should be clarified first. Afterwards the related requirements and the limitations of the already known approaches can be discussed. This hopefully leads to a common agreement of the conceptual model difference as the core and the use-case specific needs and constraints which fulfills all participants needs as the extensions. The following incomplete set of questions may guide the discussions at the workshop:

- The benefits and restrictions of the previously proposed difference representations
- Do they completely solve the stated problem?
- Which kind of changes have been considered? Which changes are possible?
- Are all requirements of all use-cases satisfied? Which ones are not?
- Which levels of abstraction are needed?
- What will be the standard conceptual difference?
- What are the specific requirements of different research groups, and how should these be integrated?
- What are suitable technologies for providing a standard exchange representation?

# 4. REFERENCES

[1] Rivera, J.E.; Vallecillo, A.: Representing and Operating with Model Differences; p.141-160 in: Proc. TOOLS EUROPE 2008; LNBIP 11, Springer-Verlag; 2008

[2] A. Cicchetti: Difference Representation and Conflict Management in Model-Driven Engineering; PhD Thesis; Universita' degli Studi dell'Aquila; April 2008

[3] EMF-Compare; `http://www.eclipse.org/emf/compare/`

[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] Niere, Jörg: Visualizing Differences of UML Diagrams With Fujaba; in: Proceedings of the International Fujaba Days 2004, Paderborn, Germany, September 2004; 2004

[6] 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