Union and Difference of Models, 10 years later

Ivan Porres

Turku Centre for Computer Science (TUCS)
Department of Information Technologies,Åbo Akademi University
Turku, Finland
ivan.porres@abo.fi

Abstract. This paper contains a summary of the talk given by the author on the occasion of the MODELS 2013 most influential paper award. The talk discussed the original paper as published in 2003, the research work done by others afterwards and the author's personal reflection on the award.

1 Version Control of Software and System Models

There are two main usage scenarios for design models in software and system development: models as sketches, that serve as a communication aid in informal discussions, and models as formal artifacts, to be analyzed, transformed into other artifacts, maintained and evolved during the whole software and system development process.

In this second scenario, models are valuable assets that should be kept in a trusted repository. In a complex development project, these models will be updated often and concurrently by different developers. Therefore, there is a need for a version control system for models with optimistic locking. This is a system to compare, merge and store all versions of all models created within a development project.

We can illustrate the use of a version control system for models as follows. Let us assume that the original model shown at the top of Figure 1 is edited simultaneously by two developers. One developer has decided that the subclass B is no longer necessary in the model. Simultaneously, the other developer has decided that class C should have a subclass D. The problem is to combine the contributions of both developers into a single model. This is the model shown at the bottom of Fig. 1.

We presented the basic algorithms to solve this problem in the original paper published in the proceedings of the UML 2003 conference [1]. The proposed solution is based on calculating the final model as the merge of the differences between the original and the edited models. Figure 2 shows an example of the difference of two models, in this case the difference between the models edited by the developers and the original model. The result of the difference is not always a model, in a similar way that the difference between two natural numbers is not a natural number but a negative one. An example of this is shown in the bottom

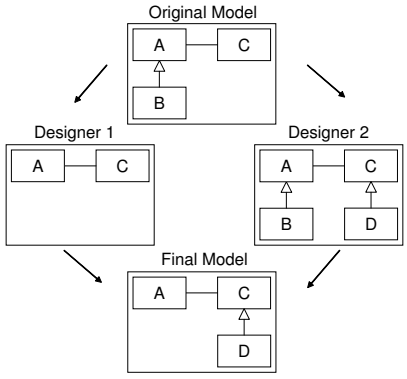

Fig. 1. Example of the Union of Two Versions of a Model

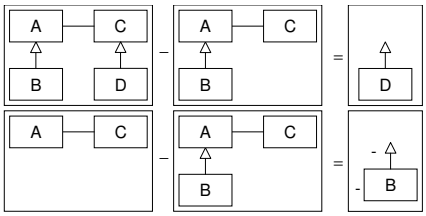

Fig. 2. Example of the Difference of Models

part of Fig. 2. In this case, the difference of the models contains *negative* model elements, i.e., elements that should be removed from a model.

The original UML 2013 paper contains two algorithms to calculate the difference between two models and to merge a model with a difference. Given two models M_1 and M_2 defined in UML or another modeling language based on the Meta Object Facility (MOF) [9], we define the following operations:

Difference of two models	$M_2 - M_1 = \Delta$
Merge of a model and a difference	$M_1 + \Delta = M_2$

Once we know how to operate with differences between two models, we can solve our original problem by computing the union of two versions of a model as follows:

$$M_{\text{final}} = M_{\text{original}} + (M_1 - M_{\text{original}}) + (M_2 - M_{\text{original}})$$

Figure 3 shows an example of the application of this operation.

The definition of these operations is complicated by the fact that two developers may have changed the same subset of a model. This is called a conflict and

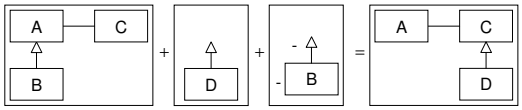

Fig. 3. Example of the Union Based on Differences

it may prevent us to merge all changes into the final model. The original paper also presents an algorithm to resolve some of these conflicts automatically. Automatic conflict resolution is an important component of a version control system with optimistic locking since it can reduce the need for a cumbersome manual intervention when merging models.

2 Assumptions and Limitations

The original paper published in the UML 2003 conference contained many assumptions and limitations that were tackled by other authors after its publication.

An important goal of the original paper was to provide algorithms that are independent of the actual modeling language. This allows us to manage models in many different languages, as long as these languages are defined using a MOF-base metamodel. On the other hand, the original algorithms cannot use information that is specific to a particular modeling language to improve the difference, merge and conflict resolution algorithms. In contrast, other authors have presented algorithms that are specific to one particular modeling language. Examples of this is the work of Nejatil et al., who presented algorithms that are specific to match and merge statecharts [8], and Küster et al., who studied a similar approach for process models [6].

Also, an significant design decision of the algorithms published originally is that model comparison is based on matching model elements by identity. The algorithms assume that each model element has a universally unique identifier (UUID) that is unique and constant. This is an adequate choice for a version control system, where the models to compare are two versions of the same original model. However, it is not a suitable approach to compare two arbitrary models. Other authors have presented model different algorithms that mach elements by similarity [5], and thus lifted the main assumption present in the original paper.

We should note that an important limitation in the original approach is the assumption that the metamodel describing the modeling language used in the artifacts to process never changes. This is not a realistic assumption, specially when using internally developed domain specific modeling languages. This problem was tackled by Gruschko et al. who have studied the problem of metamodel evolution and how to update a model when its underlying metamodel changes [4].

The original paper did not discuss the representation of model differences in any structured format. This problem was studied by other authors that pro-

posed for well defined metamodels to represent model differences. Cicchetti et al presented a model difference approach [3] that allows composition of differences. Another important topic not discussed in the original paper is how to present model differences in a way that is easy to understand by developers. This problem has been studied by Störrle [10], that proposed to present model differences using natural language.

Finally, it is worth to mention that while the original paper contains the key algorithms for a version control system for software models, it does not actually present an implementation of the algorithms into a working system. Other authors have implemented model management tools such as DSMDiff [7] and EMF Compare [1] that can be used by practitioners in actual development projects.

3 Most Influential Paper Award

The original paper published in the UML 2003 conference was selected for the 10 years most influential paper award presented during the MODELS 2013 conference. As a researcher, I consider that achieving long lasting impact among academics and practitioners is the ultimate goal of our work. Therefore, I believe that the reader may be interested in my own personal reflections about the award and the context that made the paper become influential to many other works.

I consider that one of the main reasons for the high impact of the original paper is that is was published timely. The modeling community was then focusing in the MDE research agenda and there was a clear need for this kind of work. The original paper presented the basic algorithms for a version control system, but it also described, sometimes implicitly, many other related problems to be solve. Another important factor that helped the paper to become highly influential is that it was published at the right venue. In 2003, the participants of the UML conference were the right audience for this work, who quickly extended and improved it.

I think it is interesting to discuss what would be the outcome of the original paper if it had been submitted to a high quality conference such as MODELS in the year 2013. This is just my personal speculation, but I consider that the original paper would not have performed that well in the selection process of a contemporary conference. Currently, the program committee of conferences expect an extensive evaluation and validation of the research contributions presented in research papers, something that is actually missing from the original paper. I have the opinion that the requirement for the inclusion of an extensive validation helps to improve the quality of the published papers, but may be detrimental for the quick dissemination of novel ideals, as the ones contained in the original UML 2003 paper.

To conclude, I consider that ten years later the research on model comparison and versioning is not completed yet. For those interested in this problem,

[1] http://www.eclipse.org/emf/compare/

Altmanninger et al. have published a plea for more research work in this area and a concrete research agenda [2].

Acknowledgments

I would like to thank and acknowledge the work of Dr. Marcus Alanen, co-author and main contributor of the original paper "Difference and Unions of Models", published in the UML 2003 Conference.

References

1. Marcus Alanen and Ivan Porres. Difference and union of models. In Perdita Stevens, Jon Whittle, and Grady Booch, editors, *UML*, volume 2863 of *Lecture Notes in Computer Science*, pages 2–17. Springer, 2003.
2. Kerstin Altmanninger, Petra Kaufmann, Gerti Kappel, Philip Langer, Martina Seidl, Konrad Wieland, and Manuel Wimmer. Why model versioning research is needed!? an experience report. In *Proceedings of the Joint MoDSE-MC-CM 2009 Workshop*, 2009.
3. Antonio Cicchetti, Davide Di Ruscio, and Alfonso Pierantonio. A metamodel independent approach to difference representation. *Journal of Object Technology*, 6(9):165–185, October 2007. TOOLS EUROPE 2007 — Objects, Models, Components, Patterns.
4. Boris Gruschko, Dimitrios Kolovos, and Richard Paige. Towards synchronizing models with evolving metamodels. In *In Proc. Int. Workshop on Model-Driven Software Evolution held with the ECSMR*, 2007.
5. Udo Kelter, Jürgen Wehren, and Jörg Niere. A generic difference algorithm for uml models. In Peter Liggesmeyer, Klaus Pohl, and Michael Goedicke, editors, *Software Engineering*, volume 64 of *LNI*, pages 105–116. GI, 2005.
6. Jochen Malte Küster, Christian Gerth, Alexander Förster, and Gregor Engels. Detecting and resolving process model differences in the absence of a change log. In Marlon Dumas, Manfred Reichert, and Ming-Chien Shan, editors, *BPM*, volume 5240 of *Lecture Notes in Computer Science*, pages 244–260. Springer, 2008.
7. Yuehua Lin, Jeff Gray, and Frédéric Jouault. Dsmdiff: a differentiation tool for domain-specific models. *European Journal of Information Systems*, 16(4):349–361, 2007.
8. Shiva Nejati, Mehrdad Sabetzadeh, Marsha Chechik, Steve Easterbrook, and Pamela Zave. Matching and merging of statecharts specifications. In *Proceedings of the 29th international conference on Software Engineering*, ICSE '07, pages 54–64, Washington, DC, USA, 2007. IEEE Computer Society.
9. OMG. Meta Object Facility (MOF) Core Specification. Document formal/2013-06-01. Available at: http://www.omg.org/. Last visited on cCtober 2013, 2011.
10. Harald Störrle. Making sense to modelers: Presenting uml class model differences in prose. In *MODELSWARD 2013 - Proceedings of the 1st International Conference on Model-Driven Engineering and Software Development*, pages 39–48, 2013.