Improving Collaborative Modeling by an Operation-Based Versioning Approach

Joeri Exelmans [1], Jakob Pietron [2], Alexander Raschke [2], Hans Vangheluwe [1], and Matthias Tichy [2]

Abstract: The presented work concerns our recent research on advanced collaboration and versioning techniques supporting blended modeling, originally published in the Journal of Computer Languages in August 2023 [Ex23]. Collaboration is a rising topic not only but also in the domain of model-driven engineering. The more collaborators working together on a joint model or artifact, the more adequate tooling becomes important, especially if the collaborators use different, but for their specific task most appropriate notations/concrete syntaxes (CS) simultaneously. The presented approach supports collaboration in this context by applying and extending operation-based versioning to support bi-directional change propagation between multiple CSs and a common single abstract syntax (AS). We further broaden the topic of our talk by presenting benefits for the user of modeling tools introduced by operation-based versioning and the information included in the recorded operations. We especially focus on how we facilitate the comprehension of a model's evolution over time and the impact of a distinct edit operation on the whole model. This allows modelers to understand how, why, when, and by whom the model or specific elements were modified [PFT22].

Keywords: Modeling, Collaboration, Versioning, Blended Modeling, Evolution, Understandability

1 Extended Abstract

Model-driven engineering is an answer to the problems arising from increasingly larger and more complex systems. However, the original problems have shifted as the models are getting larger and larger and can only be created in a reasonable amount of time if several (domain) experts work collaboratively together. If each expert can edit the shared model (abstract syntax (AS)) in the most appropriate (graphical) language (concrete syntax (CS)) depending on the concern, the efficiency increases again. However, the more collaborators work on a joint artifact, the more it becomes essential for each modeler to trace and understand changes performed by others.

In general, versioning can be implemented in two different ways: Either persisting snapshots of *states* or persisting the *operations* that transformed one state into another [Br12].

1 University of Antwerp, Flanders Make, Antwerp, Belgium,
joeri.exelmans@uantwerpen.be, https://orcid.org/0000-0002-6916-5140;
hans.vangheluwe@uantwerpen.be, https://orcid.org/0000-0003-2079-6643
2 Ulm University, Institute of Software Engineering and Programming Languages, Ulm, Germany,
jakob.pietron@uni-ulm.de, https://orcid.org/0000-0001-8308-6636;
alexander.raschke@uni-ulm.de, https://orcid.org/0000-0002-6088-8393;
matthias.tichy@uni-ulm.de, https://orcid.org/0000-0002-9067-3748

doi:10.18420/sw2024_07

Various approaches exist to add collaboration support to modeling tools, often using a state-based versioning, such as EMF Compare[3] or AMOR[4]. However, these tools support only a single CS to AS mapping, which in turn prevents blended modeling: the modification of a single AS through multiple different CSs. Conversely, tools that support blended modeling do not provide versioning or only in a restricted way.

In contrast, our novel operation-based versioning approach enables blended modeling in a collaborative environment by establishing and maintaining relations between the elements of CSs and the AS in a separate correspondence model. It supports branching, merging, conflict detection, and -resolution in an orthogonal way across different CSs. We provide insights into this approach and emphasize why operation-based recording of edit operations brings many benefits compared to state-based versioning.

One of these benefits is the support of modelers in understanding a model's evolution over time. Due to its detailed resolution of changes, i.e., type of change, affected elements, changed values, and further meta-information, such as author or edit time, operation-based versioning is an ideal basis for model evolution analyses. Nevertheless, recorded information requires a suitable visualization to be usable by domain experts.

We demonstrate a set of visual and interactive tools integrated into a graphical modeling tool, supporting domain experts in figuring out how, why, when, and by whom the model or specific elements were modified and performing change impact analyses.

Acknowledgment

Author J. Exelmans is an SB Ph.D. fellow at FWO (1S70622N). Author J. Pietron is partly funded by the project GENIAL!, which is partly funded by the German Federal Ministry of Education and Research (BMBF) within the research programme ICT 2020 (reference number: 16ES0875).

Bibliography

[Br12] Brosch, Petra; Kappel, Gerti; Langer, Philip; Seidl, Martina; Wieland, Konrad; Wimmer, Manuel: An Introduction to Model Versioning. In: Formal Methods for Model-Driven Engineering - 12th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2012, Bertinoro, Italy. 2012.

[Ex23] Exelmans, Joeri; Pietron, Jakob; Raschke, Alexander; Vangheluwe, Hans; Tichy, Matthias: A new versioning approach for collaboration in blended modeling. J. Comput. Lang., 76:101221, 2023.

[PFT22] Pietron, Jakob; Funk, Lenard; Tichy, Matthias: Improving the Comprehension of Evolving Graphical Models. In: Working Conference on Software Visualization, VISSOFT 2022, Limassol, Cyprus, October 3-4, 2022. IEEE, pp. 96–107, 2022.

3 https://eclipse.dev/emf/compare/

4 http://modelversioning.org/