# RuleMerger: Automatic Construction of Variability-Based Model Transformation Rules

Daniel Strüber[1], Julia Rubin[2], Thorsten Arendt[3], Marsha Chechik[4], Gabriele Taentzer[5] and Jennifer Plöger[6]

**Abstract:** We present a summary of our paper of the same title, published in the proceedings of the International Conference on Fundamental Approaches to Software Engineering (FASE) 2016. Unifying similar model transformation rules into variability-based ones can improve both the maintainability and the performance of a model transformation system. Yet, manual identification and unification of such similar rules is a tedious and error-prone task. In this work, we propose a novel merging approach for automating this task. The approach employs clone detection for identifying overlapping rule portions and clustering for selecting groups of rules to be unified. Our instantiation of the approach harnesses state-of-the-art clone detection and clustering techniques and includes a specialized merge construction algorithm. We formally prove correctness of the approach and demonstrate its ability to produce high-quality outcomes in two real-life case-studies.

## 1 Summary

Model transformation is a key enabling technology for Model-Driven Engineering, pervasive in all of its activities, including the translation, optimization, and synchronization of models. Model transformation systems often contain rules that are substantially similar to each other. Yet, until recently, various model transformation languages lacked constructs to capture such similar rules [St16b]. Therefore, developers often resorted to cloning, i.e., producing rules by copying and modifying existing ones. The drawbacks of cloning are well-known. For instance, when a bug is found in one of the rules, all rules need to be updated correspondingly. Furthermore, creating many mutually similar rules also impairs the performance of transformation systems, possibly rendering the transformation infeasible.

*Variability-based* (VB) rules are an approach to address these issues [St15, St16a]. A VB rule encodes a set of similar rules in a single-copy representation, explicating common and variable portions. In [St15], we provide an algorithm for applying VB rules and show that it outperforms the application of classical rules in terms of execution time. Yet, the VB rules in [St15] were created manually, a tedious and error-prone task relying on the precise identification of (i) sets of similar rules, each to be unified into a single VB rule; (ii) rule portions that should be merged versus portions that should remain separate. Choices made during these steps have a substantial impact on the quality of the produced VB rules.

[1] Universität Koblenz-Landau; Philipps-Universität Marburg, strueber@uni-koblenz.de
[2] Massachusetts Institute of Technology; University of British Columbia, mjulia@ece.ubc.ca
[3] GFFT Innovationsförderung GmbH; Philipps-Universität Marburg, arendt@mathematik.uni-marburg.de
[4] University of Toronto, chechik@cs.toronto.edu
[5] Philipps-Universität Marburg, taentzer@mathematik.uni-marburg.de
[6] Philipps-Universität Marburg, ploeger@mathematik.uni-marburg.de

To address this shortcoming, we present *RuleMerger* [St16c], an approach for automating the merging of model transformation rules. The approach includes a three-component framework (see Fig. 1). It applies *clone detection* [SPA16] to identify cloned portions between rules and *clustering* to identify disjoint groups of similar rules. During *merge construction*, common portions are unified and variable ones are annotated to create VB rules. Each component can be instantiated and customized with respect to specific quality goals, e.g., to produce rules optimized for the background execution of a large rule set or for convenient editing using a customized editor [SS16]. Since the framework guarantees that all created rule sets are semantically equivalent, we envision a system that enables users to edit rules in a convenient representation and to automatically derive a highly efficient one.

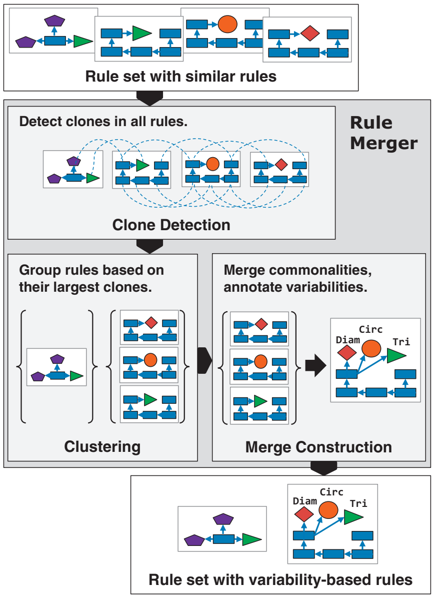
Rule set with similar rules
Rule Merger
Detect clones in all rules.
Clone Detection
Group rules based on their largest clones.
Clustering
Merge commonalities, annotate variabilities.
Circ
Diam
Tri
Merge Construction
Circ
Diam
Tri
Rule set with variability-based rules

Fig. 1: Overview of *RuleMerger*

**Contributions.** We make the following contributions: (1) a novel merging approach for model transformation rules, (2) a correctness proof for the approach, showing the equivalence of the produced VB rules to their classical counterparts, (3) an instantiation of the approach via state-of-the-art clone detection and clustering techniques and a novel *merge construction* algorithm, and (4) an experimental validation, indicating that the approach allows improving the performance and compactness of rule sets considerably.

## References

[SPA16] Strüber, Daniel; Plöger, Jennifer; Acretoaie, Vlad: Clone Detection for Graph-Based Model Transformation Languages. In: International Conference on the Theory and Practice of Model Transformations (ICMT). Springer, pp. 191–206, 2016.

[SS16] Strüber, Daniel; Schulz, Stefan: A Tool Environment for Managing Families of Model Transformation Rules. In: International Conference on Graph Transformation (ICGT), pp. 89–101. Springer, 2016.

[St15] Strüber, Daniel; Rubin, Julia; Chechik, Marsha; Taentzer, Gabriele: A Variability-Based Approach to Reusable and Efficient Model Transformations. In: International Conf. on Fundamental Approaches to Software Engineering (FASE). Springer, pp. 283–298, 2015.

[St16a] Strüber, Daniel: Model-Driven Engineering in the Large: Refactoring Techniques for Models and Model Transformation Systems. PhD thesis, Philipps-Universität Marburg, 2016.

[St16b] Strüber, Daniel; Kehrer, Timo; Arendt, Thorsten; Pietsch, Christopher; Reuling, Dennis: Scalability of Model Transformations: Position Paper and Benchmark Set. In: Workshop on Scalable Model Driven Engineering (BigMDE). pp. 21–30, 2016.

[St16c] Strüber, Daniel; Rubin, Julia; Arendt, Thorsten; Chechik, Marsha; Taentzer, Gabriele; Plöger, Jennifer: RuleMerger: Automatic Construction of Variability-Based Model Transformation Rules. In: International Conference on Fundamental Approaches to Software Engineering (FASE). Springer, pp. 122–140, 2016.