Searching for Optimal Models: Comparing Two Encoding Approaches (Summary)

Stefan John [1], Alexandru Burdusel [2], Robert Bill [3], Daniel Strüber [4], Gabriele Taentzer [5], Steffen Zschaler [6], Manuel Wimmer [7]

Abstract: This work summarizes our paper originally published in The Journal of Object Technology in the course of the International Conference on Model Transformations 2019 [Jo19].

Keywords: Model-driven Engineering; Search-based Software Engineering; Optimization; Encoding; Comparative evaluation

Many software engineering problems give rise to a tremendous space of possible solutions that differ in various qualities, such as their performance, resource efficiency, and understandability. To find optimal solutions, *search-based software engineering (SBSE)* seeks to formulate these problems as optimization problems and applies metaheuristic search techniques, to efficiently explore the solution space. Model-driven engineering (MDE) is a paradigm that aims to raise the level of abstraction in a broad range of application domains by the use of models, which are continuously refined and transformed.

Research combining SBSE and MDE under the umbrella term *search-based model-driven engineering* (SBMDE) has become increasingly popular. One particular line of research in SBMDE, which we call *model-driven optimization* (MDO), aims to reduce the level of expertise required by users of SBSE techniques[8]. In MDO, models are used to specify optimization problems and transformation rules are used to explore the search space. Thus, rather than becoming involved in the intricacies of the used optimization technology, users interact with a domain-specific formulation of their problem. They can rely on the familiar modeling and model transformation tools to inspect solutions and specify change operations.

Recently, a variety of MDO frameworks has emerged with their key distinction being the way in which solutions are encoded [ZM16]: The *model-based encoding* approach represents solutions as models. In the *rule-based encoding* approach, a solution is a sequence of rule

[1] Philipps-Universität Marburg, Marburg, Germany, stefan.john@uni-marburg.de
[2] King's College London, London, United Kingdom, alexandru.burdusel@kcl.ac.uk
[3] TU Wien | Austrian Center for Digital Production, Wien, Austria, bill@big.tuwien.ac.at
[4] Chalmers University | University of Gothenburg, Gothenburg, Sweden, danstru@chalmers.se
[5] Philipps-Universität Marburg, Marburg, Germany, taentzer@informatik.uni-marburg.de
[6] King's College London, London, United Kingdom, szschaler@acm.org
[7] CDL-MINT, Johannes Kepler University, Linz, Austria, manuel.wimmer@jku.at
[8] As opposed to applying SBSE techniques to solve MDE problems.

doi:10.18420/SE2020_30

calls in the context of a given input model. While both encodings have been applied to different use cases, no study has yet compared them systematically. Hence, we evaluate both approaches on a common set of optimization problems, investigating their impact on optimization performance. To that end, we rely on two state-of-the-art MDO frameworks (MOMoT [Bi17] and MDEOptimiser [BZS18]) that differ in the encoding approach used, but otherwise share the same technological basis: EMF[9]. (as the modeling platform), Henshin [Ar10; St17] (as model transformation language), and the MOEA evolutionary search framework[10]. Additionally, we discuss the differences, strengths, and weaknesses of both encoding approaches laying the foundation for a knowledgeable choice of when to use which encoding. Consequently, the main contributions of the paper are as follows:

1. A *qualitative comparison* between the model-based and the rule-based encoding in MDO frameworks, based on a systematic study of their features.
2. A *quantitative comparison* of both encodings with their implementations in MOMoT and MDEOptimiser, based on their performance (regarding solution quality and execution time) in a set of three diverse use cases.
3. *Insights into the applicability* of both encoding approaches; their strengths and weaknesses. We study whether the differences can be attributed to the different encoding approaches.

References

[Ar10] Arendt, T.; Biermann, E.; Jurack, S.; Krause, C.; Taentzer, G.: Henshin: Advanced concepts and tools for in-place EMF model transformations. In: Int. Conference on Model Driven Engineering Languages and Systems. Pp. 121–135, 2010.

[Bi17] Bill, R.; Fleck, M.; Troya, J.; Mayerhofer, T.; Wimmer, M.: A local and global tour on MOMoT. Software & Systems Modeling/, pp. 1–30, 2017.

[BZS18] Burdusel, A.; Zschaler, S.; Strüber, D.: MDEOptimiser: A search based model engineering tool. In: Int. Conference on Model Driven Engineering Languages and Systems. Pp. 12–16, 2018.

[Jo19] John, S.; Burdusel, A.; Bill, R.; Strüber, D.; Taentzer, G.; Zschaler, S.; Wimmer, M.: Searching for optimal models: Comparing two encoding approaches. Journal of Object Technology 18/3, 2019.

[St17] Strüber, D.; Born, K.; Gill, K. D.; Groner, R.; Kehrer, T.; Ohrndorf, M.; Tichy, M.: Henshin: A usability-focused framework for EMF model transformation development. In: Int. Conference on Graph Transformation. Pp. 196–208, 2017.

[9] https://www.eclipse.org/modeling/emf/ (last visited: December 2019)
[10] http://moeaframework.org (last visited: December 2019)

[ZM16] Zschaler, S.; Mandow, L.: Towards model-based optimisation: Using domain knowledge explicitly. In: Workshop on Model-Driven Engineering, Logic and Optimization. Pp. 317–329, 2016.