A Graph-Based Framework for Model-Driven Optimization Facilitating Impact Analysis of Mutation Operator Properties – Summary

Stefan John[1], Jens Kosiol[1], Leen Lambers[2], Gabriele Taentzer[1]

Abstract: We summarize our paper *A Graph-Based Framework for Model-Driven Optimization Facilitating Impact Analysis of Mutation Operator Properties* that has been published in the International Journal on Software and Systems Modeling (SoSyM) in 2023.

Keywords: Search-Based Software Engineering; Model-Driven Engineering; Evolutionary Computation

1 Summary

Search-based Software Engineering (SBSE) [HJ01] explores the application of metaheuristic techniques to software engineering problems such as software modularization, software testing and release planning. One of the widely used approaches to efficiently explore a search space is the application of evolutionary algorithms [HMZ12]. In this approach, elements of the search space are generated from existing elements using evolutionary operators such as mutation operators. However, the proper application of SBSE techniques is often not an easy task. As pointed out in [ZM16], "the problem domains in software engineering are too complex to be effectively captured with traditional representations as they are typically used in search-based systems".

In Model-Driven Optimization (MDO), SBSE is combined with Model-Driven Engineering (MDE) [Sc06], so that domain-specific models are used to represent the structural information while evolutionary algorithms are often used to solve optimization problems. However, designing appropriate evolutionary algorithms to evolve structures is not always straightforward. Domain experts still need a deep knowledge of how to configure an evolutionary algorithm. According to Harman et al. [HMZ12], the initial excitement about SBSE is over; it is now time for consolidation, i.e., "to develop a deeper understanding and scientific basis for the results obtained so far." This statement has motivated us to develop a formal framework for MDO that will hopefully lead to a deeper understanding of MDO.

In [Jo23], we present a graph-based framework for MDO that identifies and clarifies the core concepts of MDO and relies on mutation operators to specify evolutionary changes. This

[1] Philipps-Universität Marburg, Marburg, Germany, {johns,kosiolje,taentzer}@mathematik.uni-marburg.de
[2] Brandenburgische Technische Universität Cottbus-Senftenberg, Germany, leen.lambers@b-tu.de

doi:10.18420/sw2024_06

framework is intended to help domain experts develop and study evolutionary algorithms based on domain-specific models and operators. In addition, it can help to clarify the critical factors for conducting reproducible experiments in MDO. Based on the framework, we are able to take a first step towards identifying and studying important properties of evolutionary operators in the context of MDO. As a showcase, we investigate the impact of *soundness* and *completeness* at the level of mutation operator sets on the *effectiveness* and *efficiency* of evolutionary algorithms. In the experiment conducted, we select three state-of-the-art evolutionary algorithms (NSGA-II, PESA-II, and SPEA2) and compare different sets of mutation operators for three optimization problems: the class responsibility assignment problem, the scrum planning problem, and the next release problem. The experiment is based on the tools MDEOptimiser and Henshin. The results show that unsound operators can lead to low-quality solutions as they increase the probability of getting stuck in local optima. A set of incomplete operators discards part of the search space, which can also affect the solution quality.

2 Data Availability

All evaluation data can be found at `https://github.com/Leative/SoSyM22-MDO-framework-evaluation`. They include the results of our experiments and all artifacts needed to reproduce them.

Acknowledgements. This work was partly supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) project "Model-Driven Optimization in Software Engineering" (TA 294/19-1).

Bibliography

[HJ01] Harman, Mark; Jones, Bryan F.: Search-based Software Engineering. Inf. Softw. Technol., 43(14):833–839, 2001.

[HMZ12] Harman, Mark; Mansouri, S. Afshin; Zhang, Yuanyuan: Search-based Software Engineering: Trends, Techniques and Applications. ACM Comput. Surv., 45(1):11:1–11:61, 2012.

[Jo23] John, Stefan; Kosiol, Jens; Lambers, Leen; Taentzer, Gabriele: A graph-based framework for model-driven optimization facilitating impact analysis of mutation operator properties. Softw. Syst. Model., 22(4):1281–1318, 2023.

[Sc06] Schmidt, Douglas C.: Guest Editor's Introduction: Model-Driven Engineering. Computer, 39(2):25–31, 2006.

[ZM16] Zschaler, Steffen; Mandow, Lawrence: Towards Model-Based Optimisation: Using Domain Knowledge Explicitly. In: Software Technologies: Applications and Foundations – STAF 2016 Collocated Workshops: Revised Selected Papers. volume 9946 of LNCS. Springer, pp. 317–329, 2016.