Model-Driven Engineering (MDE) is a paradigm that promotes the use of models as the central artifacts for solving problems. In MDE, problem domains are specified using domain-specific modeling languages and models are concrete problem instances that abstract from reality to reduce complexity. At the heart of MDE, model transformations are used to systematically manipulate these problem models to find good solutions to the problem at hand. However, reasoning about how the transformation needs to be orchestrated to find good solutions is a non-trivial task due to the large or even infinite transformation space. As a result, this task is either performed automatically, e.g., by following an apply-as-long-as-possible approach, which does not necessarily produce satisfactory results, or it is carried out manually by the respective engineer. This, in turn, hampers the application of MDE techniques on complex problems which usually cannot be solved manually or by enumerating all possible solutions.

Therefore, we present in this thesis an approach that facilitates to solve these problems by stating clear objectives operationalized through model-based analysis techniques and elevating search-based optimization methods to the model level to find optimal transformation orchestrations. As first contribution, we introduce a model-based analysis approach that measures dynamic, timed properties that consider the contention of resources directly on the model level using the fUML standard. As second contribution, we provide a generic encoding of the transformation orchestration problem on which many different optimization methods can be applied. Using this encoding, we propose an approach that enables to solve problems by providing a model, a set of transformation rules, a set of objectives that are optimized during the process and a set of constraints that mark invalid solutions. The optimization process is configured through a dedicated language which provides information on the optimization concepts and immediate feedback for the concrete configuration. The results consist of the respective orchestrated transformations, the solution models, the objective and constraint values as well as analysis details about the optimization process. Our approach is based on graph transformations and has been implemented as an open-source framework called MOMoT. Based on this implementation, we provide an extensive evaluation of our approach using several case studies from the area of model-driven software engineering as well as two novel problem formulations that tackle the modularization of model transformations and the generic modularization of modeling languages. The obtained evaluation results validate the effectiveness of our approach and give rise to interesting lines of research.