Speaker	Title	Abstract
Ciaran McCreesh	Modelling and Optimisation with Graphs	In constraint programming, we describe problems in terms of a set of variables, each of which has a (typically finite) domain of values, together with a set of constraints that forbid certain combinations of variable-value assignments. The objective is to find a value for each variable, such that all the constraints are satisfied (or to find the best such assignment, subject to some scoring function). I'll give a very quick overview of constraint solving technology, followed by an introduction to high level modelling languages that are used to make the technology accessible to non-experts. Finally, I'll discuss my current research, which is on extending both the high- and low-level technologies to make it easier to model and solve constraint problems that involve graphs.
Yannic Noller	Differential Program Analysis with Fuzzing and Symbolic Execution	Differential program analysis means to identify the behavioral divergences between two program versions for the same input (i.e. regression testing), but also to identify divergent behavior for different inputs for the same program (e.g., worst-case complexity and side-channel analysis). Most of the existent approaches for both subproblems try to solve it with single techniques which suffer from its weaknesses like scalability issues or imprecision. In order to tackle these problems and to provide scalable solutions for real-world applications, we are performing research on how to combine search-based techniques with semantics-based techniques, namely fuzzing and symbolic execution.
Stefan John	Rule-based Crossover Operators for Model-driven Optimization	In search-based model-driven engineering (SBMDE), search-based optimization techniques are applied in a model-driven environment. One of many emerging application areas in that field is model-driven optimization (MDO), which focuses on models to express and solve optimization problems. Algorithm-wise, meta-heuristics relying on the evolutionary concepts of mutation and crossover, implemented as model transformations, have caught the most attention. While meta-models are commonly used to define the problem at hand, two different approaches have been proposed to encode solutions. In the direct encoding approach, models are used as solutions and changed directly by applying evolutionary operators. In the transformation sequence approach, in contrast, sequences of model transformations are optimized and evaluated by applying them to a common initial model. For the latter approach, known crossover techniques

		for sequential encodings can be easily adapted and applied. Directly working on models, however, new rule-based approaches for crossover operators have to be investigated. To that end, “What are the building blocks of a model?”, “How can structural features of a model be exchanged in a performant and consistency preserving way?”, and “Can we generalize model crossover?” are among the questions of interest.
Daniel Strüber	Generating Efficient Mutation Operators for Search-Based Model Engineering	Search-based model engineering combines the abstraction power of models with the versatility of meta-heuristic search algorithms. While current approaches in this area use genetic algorithms with fixed mutation operators to explore the solution space, the efficiency of these operators may heavily depend on the problem at hand. In this work, we propose FitnessStudio, a technique for generating efficient problem-tailored mutation operators automatically based on a two-tier framework. The lower tier is a regular meta-heuristic search whose mutation operator is constrained by an upper-tier search using a higher-order model transformation. We implemented this framework using the Henshin transformation language and evaluated it in a benchmark case, where the generated mutation operators enabled an improvement to the state of the art in terms of result quality, without sacrificing performance.
Steffen Zschaler	Search-Based Model Engineering with MDEOptimiser	Finding models that optimise a set of objective functions is a non-trivial task. It often requires encoding the search space in a mathematical format and using complicated tools to implement a suitable search strategy. In this talk, I will present MDEOptimiser, a model-driven tool that implements evolutionary search on top of domain-specific modelling languages. As a result, a natural encoding of an optimisation problem can be used and a standard search algorithm can be easily deployed on top of this.
Daniel Varro	Design space exploration for graph model generation	In this talk, I will provide an overview on how design space exploration (DSE) techniques and tools can be used as a core component of a model generator framework which derives graph models that are simultaneously consistent and diverse graph models without mapping them to back-end logic solvers. I will discuss key challenges as (1) how to encode partial graphs as abstract states, (2) how to continuously evaluate consistency constraints over them and (3) how different traditional SAT-solving techniques can be implemented as DSE exploration strategies.

Timo Kehrer	Where Search-Based Model Engineering Meets Model Management and Evolution	In this talk, I will give an overview of promising research synergies arising from integrating techniques from the field of Model Management and Evolution with Search-Based Model Engineering. Examples of this include but are not limited to the adoption of model merging for the sake of breeding new candidate solutions, the incorporation of model repair for automatically resolving inconsistencies in candidate solutions, and, on the meta-tool level, the usage of high-order transformation rule generation and inference for the sake of (semi-)automatically synthesizing mutation operators. The goal is to provide the necessary background information for fostering discussions on how to exploit these techniques while meeting the specific requirements in the field of Search-Based Model Engineering.