
I. OVERVIEW

Independently, both modelling and search are efficient, practical, and innovative approaches to software engineering.

Models enable the software engineer to capture and manipulate relevant information about an engineering task, while avoiding irrelevant details. By focussing only on the most important characteristics of the problem, model-based solution techniques are often very efficient. Practicality derives from the use of mature, domain-independent tools to create, transform, analyse, and interact with models. Techniques such as Model-Driven Engineering enable the software engineer to work at an appropriate level of abstraction and thereby create innovative solutions to problems.

Search-Based Software Engineering (SBSE) techniques use metaheuristic search algorithms to efficiently traverse the space of potential solutions to an engineering problem. Practicality derives from the high degree of automation possible using such algorithms. In particular, affordable high performance computing resources—such as cloud computing and commodity GPU cards—may be used by SBSE techniques to solve engineering tasks that would be impractical using manual engineering techniques. This is also the source of the innovation: the low cost and high performance of SBSE techniques enable the engineer to explore the space of potential solutions in a manner that would be impossible if each solution had to be derived manually. For example, the engineer can use SBSE to discover optimal trade-offs between competing objectives, and so work at a higher level of abstraction than that of deriving individual solutions.

In this tutorial, I will expand on the how these properties of efficiency, practicality, and innovation are demonstrated by the two approaches independently. The purpose is provide a foundation for the presentations at this workshop that propose and demonstrate how these desirable properties of modelling and search may be combined in order to realise even greater benefits in the domain of software engineering.