(ITEA 3 – 17003)

PANORAMA
Boosting Design Efficiency for Heterogeneous^3 Systems

Deliverable: D 6.4
Design Space Exploration Approaches

Work Package: 6
Design flow and traceability

Task: 6.4
Search-based Software Engineering and design space exploration

Document Type:	Deliverable (Software)	**Classification:**	Public
Document Version:	1.0	**Contract Start Date:**	2019-04-01
Document Preparation Date:	30.04.2021	**Duration:**	2022-03-31

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

EUREKA

History

Rev.	Content	Resp. Partner	Date
1.0	D6.4 MEMO v0.8	David Schmelter, Fraunhofer IEM	30.04.2021

Final Approval	Name	Partner
30.04.2021	Jan-Philipp Steghöfer	Chalmers / University of Gothenburg

Contents

List of Figures

List of Tables

1 MEMO

In this deliverable, we present the first release of Multi-objective Evolution of Models (MEMO), a framework for semi-automatic evolution of EMF models.

Model-driven engineering (MDE) endeavors to cope with the complexity of today's software-intensive systems. An example of an MDE approach bases on the Scenario Modeling Language (SML)[1], which is a formal language for scenario-based requirements engineering enabling the verification technique Realizability Check. SML specifications are assume guarantee specifications. SML's Guarantee Scenarios describe how system objects may, must, or must not react to environment events; Assumption Scenarios describe what may, will, or will not happen in the environment, or how the environment may, will, or will not, react to system events. An SML specification is deemed *unrealizable* if there exists a sequence of environment events that forces the system under development into a safety or liveness violation.

A downside of such MDE approaches is that the formal verification results often are hard to understand, and the corresponding models typically have a multitude of adjustment options to fix defects identified by the formal verification results (like unrealizability). Search-based software engineering for MDE is a promising option to automate such tasks for the evolution of models driven by formal verification results, where a large solution space exists and several constraints as well as objectives have to be considered. Following this paradigm, we develop MEMO, a model-driven approach for the evolution of EMF models. In the current release MEMO is especially capable of evolving unrealizable SML models into realizable models.

In the following, we give a brief overview of MEMO's key building blocks using SML as an application example (cf. Figure 1.1).

MEMO's Input As input MEMO expects an unrealizable SML model and a *Solution Candidate Model*.

Solution Candidate Models A Solution Candidate Model specifies production rules that model how MEMO can modify the input model to be evolved. The current release of MEMO provides a domain-specific language for creating Solution Candidate Models. This language is designed to be modeling language independent, such that production rules for any EMF-based modeling language can be specified (i.e., beyond SML). Usually it is sufficient to implement a model transformation once for an EMF-based modeling language that MEMO should support, which automatically transforms an input model into a solution candidate model. This way, for example, different SML input models can be used without the MEMO user having to create a solution candidate model for each input model manually.

Search Algorithms MEMO uses internally among others the MOEAFramework[2]. This provides the MEMO user with a wide range of evolutionary algorithms for model evolution. In

[1] http://scenariotools.org
[2] http://moeaframework.org/

addition, MEMO implements a variant of Tabu search that supports multi-objective optimization.

Fitness Function For each EMF-based modeling language supported by MEMO, the MEMO user must implement a fitness function. The fitness function is used during the search to evaluate each evolved candidate solution. MEMO supports constrained multi-objective optimization. The current release of MEMO includes a fitness function for SML that optimizes for the objectives model size, environment information ratio, and environment restrictiveness. Furthermore, the fitness function evaluates the realizability of a candidate solution by means of a corresponding constraint.

MEMO's Output The output of MEMO is a Pareto-optimal set of candidate solutions. In the case of SML, this is a Pareto-optimal set of realizable SML specifications.

Provided Examples The current MEMO release includes two examples for the evolution of SML models. The EBEAS SML specification models requirements for a fully automated ADAS that cooperatively and autonomously handles hazardous situations. The Production Cell specification models requirements for an industrial control system that controls the production of pressed parts.

In the further course of the PANORAMA project, we want to extend MEMO toward evolution of Eclipse Capra-based traceability models. MEMO is available as open source software under the Eclipse Public License. The GIT respository of MEMO is currently hosted in the Fraunhofer Gitlab infrastructure[3]. In the further course of the PANORAMA project, we also plan to host MEMO under the PANORAMA organization umbrella on GitHub[4].

[3] https://gitlab.cc-asp.fraunhofer.de/david-schmelter/memo

[4] https://github.com/panorama-research

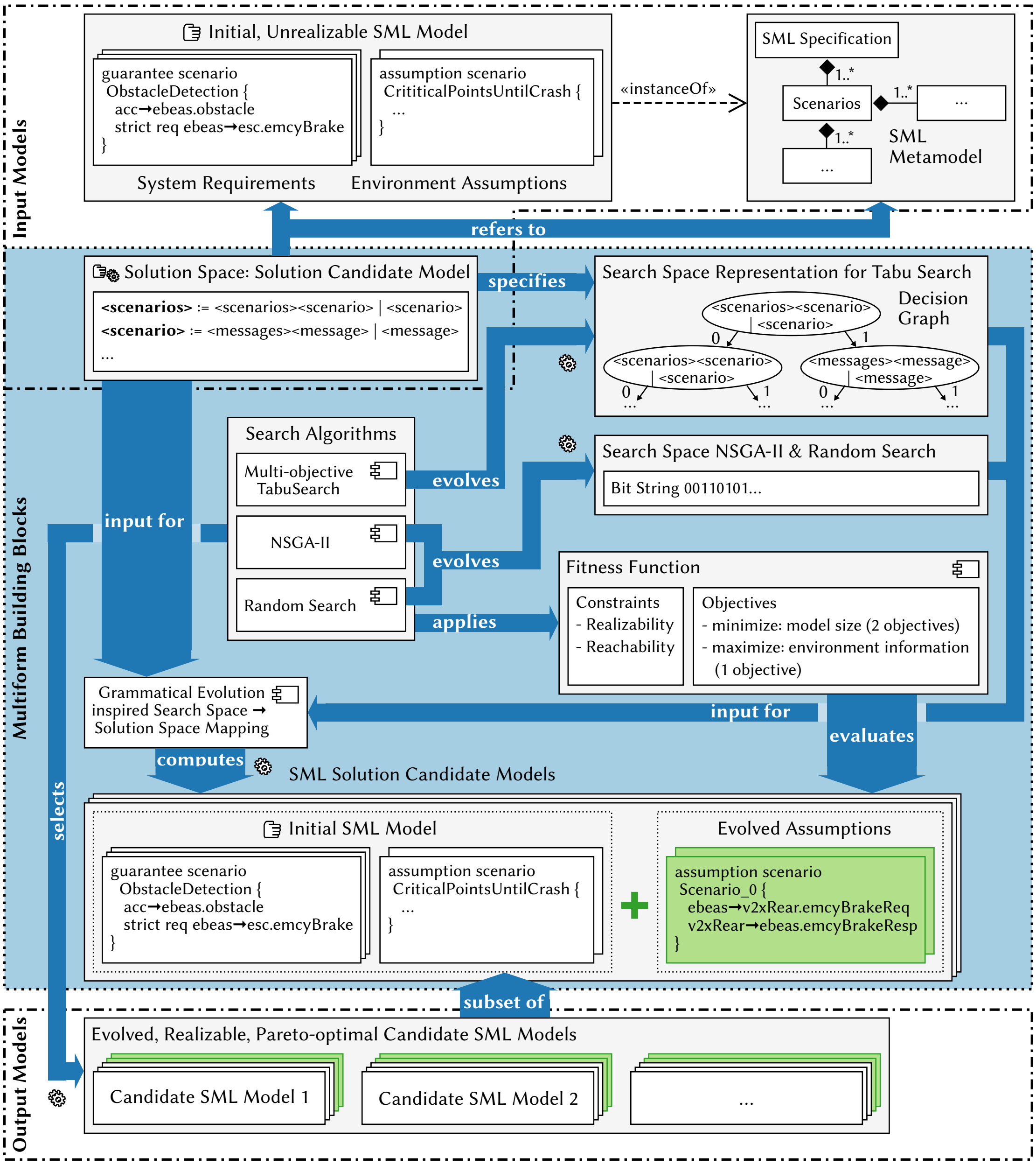

Figure 1.1: MEMO Overview