

# Comparing Multiple MATLAB/Simulink Models Using Static Connectivity Matrix Analysis

Alexander Schlie,<sup>1</sup> Sandro Schulze,<sup>2</sup> Ina Schaefer<sup>3</sup>

**Abstract:** Model-based languages such as MATLAB/Simulink are crucial for the development of embedded software systems. To adapt to changing requirements, engineers commonly copy and modify existing systems to create new variants. Denoted clone-and-own, this straightforward reuse strategy entails severe maintenance and consistency issues as redundant and similar assets proliferate. Software product lines can be a remedy but require all existing variants to be compared prior to their actual migration. However, current work mostly revolves around comparing only two systems and those approaches coping with more are not applicable to embedded software systems such as MATLAB/Simulink. We bridge this gap and propose Static Connectivity Matrix Analysis (SCMA), a novel comparison procedure that evaluates multiple MATLAB/Simulink model variants at once. We transfer models into matrix form and identify all similar structures between them, even with parts being completely relocated during clone-and-own. Moreover, we allow engineers to tailor results and to focus on any arbitrary variant subset, enabling individual reasoning prior to migration. We provide a feasibility study from the automotive domain, showing our matrix representation to be suitable and SCMA to be fast and precise.

**Keywords:** MATLAB/Simulink; clone-and-own; software product lines; variability; descriptors

## Overview

Industrial domains such as automotive, rail and avionic facilitate on embedded systems to develop safety-critical functionality. With reliability and maintainability being pivotal factors, the paradigm of *model-driven engineering (MDE)* prevails in such fields, with languages such as MATLAB/Simulink utilizing function-block based designs. Overall system development remains a challenging and time-intensive task, and to address changing requirements, engineers commonly resort to ad-hoc reuse of existing systems by copying and subsequently modifying them. Denoted clone-and-own, maintenance and evolution of the emerging system portfolio are impaired, as redundancies proliferate and knowledge about commonalities and differences between system variants is rendered void. *Software product lines (SPLs)* can be a remedy, facilitating strategic reuse and promoting maintainability. Unfortunately, the necessity to transitioning towards an SPL practice only becomes evident in the aftermath, then posing an enormous challenge to practitioners. In the context of MDE however, current approaches addressing SPL migration strategies mostly revolve around an incremental comparison of only two related systems. We argue that such restricted evaluation is insufficient for strategically migrating a portfolio towards an SPL, but rather consider a comprehensive assessment of all variants to be indispensable.

---

<sup>1</sup> TU Braunschweig, Braunschweig, Germany a.schlie@tu-braunschweig.de

<sup>2</sup> Otto-von-Guericke-University in Magdeburg, Magdeburg, Germany sandro.schulze@iti.cs.uni-magdeburg.de

<sup>3</sup> TU Braunschweig, Braunschweig, Germany i.schaefer@tu-braunschweig.de

To compare multiple MATLAB/Simulink model at once, hence addressing aforementioned problems, we propose *Static Connectivity Matrix Analysis (SCMA)*, which we presented at the 34th IEEE International Conference on Software Maintenance and Evolution [SSS18]. With our work, allow for a comprehensive overview of all system variants, and furthermore, for practitioners to focus on arbitrary variant combinations to facilitate strategic decisions prior to portfolio migration. We introduce the *Connectivity Matrix (CM)*, a descriptor, which abstracts salient system information from MATLAB/Simulink models, to reduce their overall complexity to allow for their efficient comparison. Utilizing SCMA and the CMs for comparison respectively, we compare all input models and identify all similar hierarchical structures between them, allowing for a preliminary assessment of the portfolio in its entirety as well as identifying relations between individual variants prior to their SPL migration. We compare all abstracted models, hence all CMs to identify common structures even when respective model parts have been completely relocated during clone-and-own. The CM describes MATLAB/Simulink models, and precisely, each encapsulated sub-model, here *subsystem*, within, by exploiting their inherent graph-structure, and therefore blocks connected via signals. Approximating any such subsystem, the CM reflects which two block functions are directly connected and how often. By that, our CM aims to be easy to extract from the original model and to exhibit a low probability of mismatch, that is, two different models resulting in the same descriptor. With the CM being always of identical dimensions, hence exhibiting the same row and column construction, our SCMA calculates the similarity for any two CMs based the extent of their entries, and thus, similar connections between pairs of functional blocks. Preserving information on the abstracted subsystems' parent-child relation, compared CMs form *trees*, which group together CMs from multiple models within *nodes*, which connect if comprised CMs exhibit a parent-child relation. The output of our SCMA is a *forest*, comprising multiple trees, each reflecting a similar hierarchical structure between multiple input models. Practitioners can tailor results, and for instance, focus on specific variant subsets or define thresholds to evaluate the most similar variants in isolation.

For our evaluation, we apply SCMA to an industrial case-study of MATLAB/Simulink models from the automotive domain to assess performance as well as precision and recall of our approach. Moreover, we investigate the suitability of our descriptor, the CM, to abstract MATLAB/Simulink models for their efficient comparison. Our results show SCMA to exhibit a reasonable runtime given the assessment of 18 model variants, comprising  $\approx 15.000$  blocks in total. Moreover, our evaluation reveals SCMA to exhibit a high precision, total recall and shows our CM to be a suitable descriptor for MATLAB/Simulink software systems. Overall, we argue that SCMA allows for an efficient comparison of an entire model portfolio at large, recreating information about commonalities across several models to assist practitioners with the daunting task of migrating embedded systems towards an SPL fashion.

## Bibliography

- [SSS18] Schlie, Alexander; Schulze, Sandro; Schaefer, Ina: Comparing Multiple MATLAB/Simulink Models Using Static Connectivity Matrix Analysis. In: Proc. of the Intl. Conference on Software Maintenance and Evolution (ICSME). IEEE, pp. 160–171, 2018.