Analysing Self-Adaptive Systems as Software Product Lines

Juliane Päßler
University of Oslo
Oslo, Norway
julipas@ifi.uio.no

Maurice H. ter Beek
CNR-ISTI
Pisa, Italy
maurice.terbeek@isti.cnr.it

Ferruccio Damiani
University of Turin
Turin, Italy
ferruccio.damiani@unito.it

Einar Broch Johnsen
University of Oslo
Oslo, Norway
einarj@ifi.uio.no

Silvia Lizeth Tapia Tarifa
University of Oslo
Oslo, Norway
sltarifa@ifi.uio.no

Abstract

This one-page document summarises a paper published in JSS [1].

CCS Concepts

• **Software and its engineering** → **Software product lines**; **Formal methods**; **Model checking**; • **Computer systems organization** → **Embedded and cyber-physical systems**; **Robotics**; • **Mathematics of computing** → **Probabilistic representations**; • **Theory of computation** → **Verification by model checking**.

Keywords

Dynamic software product line, Self-adaptive system, Feature model, Featured transition system, Probabilistic model checking, Robotics

ACM Reference Format:
Juliane Päßler, Maurice H. ter Beek, Ferruccio Damiani, Einar Broch Johnsen, and Silvia Lizeth Tapia Tarifa. 2025. Analysing Self-Adaptive Systems as Software Product Lines. In *29th ACM International Systems and Software Product Line Conference - Volume A (SPLC-A '25), September 01–05, 2025, A Coruña, Spain.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3744915.3748483

Self-adaptive systems (SASs) are often modelled by a two-layered structure to attain a separation of concerns between the application logic, modelled in the *managed subsystem*, and the adaptation logic, modelled in the *managing subsystem*. We propose to use techniques introduced for software product lines (SPLs) to formally model and analyse SASs while maintaining this separation of concerns.

The managed subsystem of an SAS is often component-based, where at each point of time a subset of components is chosen to fulfil the given task. We propose to model this variability of the managed subsystem in a feature model such that the managed subsystem can be viewed as a family of systems, each family member, i.e., product, corresponding to a valid feature configuration, i.e., a valid combination of components, of the managed subsystem. These valid configurations can then be viewed as different modes of operation of the managed subsystem. We propose to use probabilistic featured transition systems (FTSs) to model the behaviour of the products of the managed subsystem and the environment. We extend the standard FTSs with probabilities and costs/rewards to accommodate for uncertainties in the managed subsystem and the environment and for quantitative analyses, respectively.

The managing subsystem of a component-based SAS can adapt the managed subsystem, causing a switch between different modes of operation, by activating and deactivating components of the managed subsystem depending on input from the managed subsystem and the environment. Thus, if the managed subsystem is modelled using SPL techniques as described above, the managing subsystem can be modelled as a feature controller, activating and deactivating features of the managed subsystem while adhering to the constraints imposed by the feature model. To reflect the switches between configurations, the FTS of the managed subsystem can be extended with transitions between the behaviour of different products. Thereby, the behaviour of all possible (re-)configurations of the managed subsystem is modelled in a single model.

We use a small-scale evaluation [2] featuring a simplified model of an autonomous underwater robot with the mission to find and inspect a pipeline located on a seabed to showcase how the proposed approach can be used to model and analyse an SAS. The small-scale evaluation is modelled in the tool ProFeat [3], which extends the probabilistic model checker PRISM with capabilities for family-based modelling and analysis. ProFeat allows to model the SAS as described above: a feature model whose features can be used as guards in the probabilistic model of the managed subsystem can be specified, and the managing subsystem can be modelled as a separate *feature controller*, enabling it to activate and deactivate features which enables and disables transitions of the managed subsystem. Using ProFeat, we showcase how (quantitative) properties, including the correctness of the adaptation logic, can be analysed. Furthermore, we include an evaluation of the proposed approach of modelling and analysing SASs with techniques introduced for SPLs.

References

[1] Juliane Päßler, Maurice H. ter Beek, Ferruccio Damiani, Einar Broch Johnsen, and S. Lizeth Tapia Tarifa. 2025. Analysing Self-Adaptive Systems as Software Product Lines. *Journal of Systems and Software* 222 (2025), 112324. https://www.sciencedirect.com/science/article/pii/S0164121224003686

[2] Juliane Päßler, Maurice H. ter Beek, Ferruccio Damiani, Einar Broch Johnsen, and Silvia Lizeth Tapia Tarifa. 2024. Analysing Self-Adaptive Systems as Software Product Lines (Artifact). Zenodo. doi:10.5281/zenodo.14230735

[3] Philipp Chrszon, Clemens Dubslaff, Sascha Klüppelholz, and Christel Baier. 2018. ProFeat: Feature-Oriented Engineering for Family-Based Probabilistic Model Checking. *Formal Aspects of Computing* 30, 1 (2018), 45–75. doi:10.1007/s00165-017-0432-4

SPLC-A '25, A Coruña, Spain

ACM ISBN 979-8-4007-2024-6/25/09
https://doi.org/10.1145/3744915.3748483