Transformation of Software Product Lines: A Generalizing Framework Based on Category Theory

Gabriele Taentzer[1], Rick Salay[2], Daniel Strüber[3], Marsha Chechik[4]

Keywords: software product lines; variability; model transformation; graph transformation; category theory

1 Abstract

We present our paper from the proceedings of the 2017 edition of the IEEE/ACM International Conference on Model-Driven Engineering Languages and Systems (MODELS) [Ta17].

A *software product line* (SPL) is a portfolio of software products created from a set of common core assets. SPLs enable enterprises to produce custom-tailored products according to their customers' needs. However, a large amount of variability in an SPL can lead to great complexity: From n configuration options or *features*, up to 2^n products can be produced. To cope with this complexity, systematic methods for managing an SPL are required.

The systematic management of an SPL can be supported by expressing potential changes as model transformations. Three kinds of SPL transformations has been considered in the literature: 1. *Changes to feature models* aim to support reasoning about feature additions, deletions, and modifications. Thüm et al. [TBK09] distinguish four categories of changes based on their impact: refactorings, generalizations, specializations, and arbitrary edits. 2. *Lifting of model transformations* [Sa14] aims to make transformation rules from the single-product setting applicable to an entire SPL, represented as a feature-annotated domain model. The intended effect of the transformation is the same as considering each product separately. 3. *SPL refinement* [BTG12] aims to support safe evolution: modifications of the SPL may be restricted so that all or a well-defined subset of all products remain unchanged. None of these approaches allows specifying the *combined* transformation of feature models

[1] Philipps-Universität Marburg, Hans-Meerwein-Str. 1, 35032 Marburg, Germany, taentzer@mathematik.uni-marburg.de

[2] Department of Computer Science, 10 King's College Road, University of Toronto, Toronto, Ontario, Canada, M5S 3G4, rsalay@cs.toronto.edu

[3] Universität Koblenz-Landau, Universitätsstr. 1, 56070 Koblenz, Germany, strueber@uni-koblenz.de

[4] Department of Computer Science, 10 King's College Road, University of Toronto, Toronto, Ontario, Canada, M5S 3G4, chechik@cs.toronto.edu

and domain models. However, such combined transformations are important, since the addition or deletion of features usually entails the corresponding changes in the domain model. Developers need to know whether the overall result is a well-formed SPL again.

To address this need, in this paper, we present a formal framework for rule-based transformations of SPLs based on category theory. Specifically, we make the following contributions:

- We formalize the *category of software product lines*. Based on this category, we obtain a framework that abstracts from the type of model being considered, which makes it applicable to a great variety of models, including UML models and Petri nets.
- We formally define *transformations over SPLs* and show their soundness, i.e., for a given matching site of a rule to an input SPL, each rule application yields a well-defined and unique SPL. We discuss how our framework paves the way for new kinds of SPL analyses with sound tool support.
- We demonstrate the *applicability* of our formalization in three running examples. The flexibility of our framework follows from its capability to support almost all modifications proposed in the aforementioned works. In addition, we support entirely new kinds of modifications resulting from the combinations of the existing ones.

In the future, we will further extend our formal framework so we can use relevant formal results to enable new kinds of product line analyses. Specifically, to show that our framework is amenable to conflict and dependency analysis, we intend to show that our category is $\mathcal{M}$-adhesive, as would then give rise to a theory of product line transformations. Furthermore, we will apply our framework to a set of real-life product lines and provide tool support to make its benefits available to developers.

References

[BTG12] Borba, Paulo; Teixeira, Leopoldo; Gheyi, Rohit: A Theory of Software Product Line Refinement. Theor. Comput. Sci., 455:2–30, 2012.

[Sa14] Salay, Rick; Famelis, Michalis; Rubin, Julia; Di Sandro, Alessio; Chechik, Marsha: Lifting model transformations to product lines. In: International Conference on Software Engineering (ICSE). ACM, pp. 117–128, 2014.

[Ta17] Taentzer, Gabriele; Salay, Rick; Strüber, Daniel; Chechik, Marsha: Transformations of Product Lines: A Generalizing Framework based on Category Theory. In: International Conference on Model Driven Engineering Languages and Systems (MODELS). ACM, pp. 101–111, 2017.

[TBK09] Thüm, Thomas; Batory, Don; Kästner, Christian: Reasoning About Edits to Feature Models. In: International Conference on Software Engineering (ICSE). IEEE, pp. 254–264, 2009.