(Hrsg.): Software Engineering und Software Management 2018,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2018 1

Is There a Mismatch between Real-World Feature Models and Product-Line Research?

Alexander Knüppel[1] Thomas Thüm [2] Stephan Mennicke [3] Jens Meinicke [4] Ina Schaefer [5]

Abstract: This work has been presented at the joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering in Paderborn, Germany [Kn17].

Feature modeling has emerged as the de-facto standard to capture variability of a software product line in a compact and understandable fashion. Multiple feature modeling languages that evolved over the last decades to manage industrial-size product lines have been proposed. However, less expressive languages, solely permitting require and exclude constraints, are permanently and carelessly used in product-line research. We address the problem whether those less expressive languages are sufficient for industrial product lines. We developed an algorithm to eliminate complex cross-tree constraints in a feature model, enabling the combined usage of tools and algorithms working with different feature model dialects in a plug-and-play manner. However, the scope of our algorithm is limited. Our evaluation on large feature models, including the Linux kernel, gives evidence that require and exclude constraints are not sufficient to express real-world feature models. Hence, we promote that research on feature models needs to consider arbitrary propositional formulas as cross-tree constraints prospectively.

Keywords: Software product lines, feature modeling, cross-tree constraints, model transformation, expressiveness, require constraints, exclude constraints

Overview

In this talk, we discuss the role of arbitrary propositional formulas as cross-tree constraints in feature modeling. We argue that the two most prominent kinds of cross-tree constraints, namely *require* and *exclude constraints*, are not enough to capture product lines created for real-world projects. Unfortunately, even nowadays, part of product-line research gives the impression that those simple constraints suffice to describe industrial software product lines.

The main results on this mismatch between newly proposed product-line research on the one hand and feature models of industrial product lines on the other hand have been

[1] TU Braunschweig, Germany
[2] TU Braunschweig, Germany
[3] TU Braunschweig, Germany
[4] University of Magdeburg, Germany; Carnegie Mellon University, USA
[5] TU Braunschweig, Germany

presented at the joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering in Paderborn, Germany [Kn17]. We showed that, to this day, a significant portion of novel contributions on feature models (e.g., algorithms or empirical evaluations) do not discuss complex constraints. The consequence is that applicability of these contributions on large-scale feature models remains questionable. Our evaluation on real-world feature models of considerable size portraits that simple constraints are not enough.

Although Schobbens et al. [Sc07] were the first to indicate that tree-based feature modeling languages with simple constraints are expressively incomplete, our work extends their prior theory and highlights the differences in expressive power with real numbers. To help the community to overcome this limitation, we developed a semantics-preserving transformation between languages with complex constraints and languages with simple constraints and gave proof of its correctness. The algorithm is based on the notion of abstract features [Th11]. Moreover, we successfully evaluated the benefits of this algorithm on a total of 127 real-world feature models: four monthly snapshots of an obfuscated automotive product line from our industrial partner with up to 18,616 features and 1,369 cross-tree constraints, eight models translated from the KConfig variability language (including the linux kernel), and 116 models translated from the component definition language (CDL).

On the upside, our algorithm is able to eliminate complex constraints in a feature model while preserving the represented set of products. The downside of our algorithm is that, for large feature models, our algorithm may render feature model applications (e.g., SAT analysis) infeasible due to a potential blow-up in the numbers of new features and constraints. However, the elimination of complex constraints is indispensable for practical product-line engineering. We thus advocate that product-line research should consider complex constraints as default in the future. We further think that a community effort is needed to evaluate which and how approaches tailored to basic feature models can be applied to complex constraints.

References

[Kn17] Knüppel, Alexander; Thüm, Thomas; Mennicke, Stephan; Meinicke, Jens; Schaefer, Ina: Is there a mismatch between real-world feature models and product-line research? In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering. ACM, pp. 291–302, 2017.

[Sc07] Schobbens, Pierre-Yves; Heymans, Patrick; Trigaux, Jean-Christophe; Bontemps, Yves: Generic semantics of feature diagrams. Computer Networks, 51(2):456–479, 2007.

[Th11] Thum, Thomas; Kastner, Christian; Erdweg, Sebastian; Siegmund, Norbert: Abstract features in feature modeling. In: Software Product Line Conference (SPLC), 2011 15th International. IEEE, pp. 191–200, 2011.