Jan Jürjens, Kurt Schneider (Hrsg.): Software Engineering 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 11

Compositional Analyses of Highly-Configurable Systems with Feature-Model Interfaces

Reimar Schröter[1] Sebastian Krieter[2] Thomas Thüm[3] Fabian Benduhn[4] Gunter Saake[5]

Abstract: Today's software systems are often customizable by means of load-time or compile-time configuration options. These options are typically not independent and their dependencies can be specified by means of feature models. As many industrial systems contain thousands of options, the maintenance and utilization of feature models is a challenge for all stakeholders. In the last two decades, numerous approaches have been presented to support stakeholders in analyzing feature models. Such analyses are commonly reduced to satisfiability problems, which suffer from the growing number of options. While first attempts have been made to decompose feature models into smaller parts, they still require to compose all parts for analyses. We proposed the concept of a feature-model interface that only consists of a subset of features and hides all other features and dependencies. Based on a formalization of feature-model interfaces, we proved compositionality properties. We evaluated feature-model interfaces using a three-month history of an industrial feature model with 18,616 features. Our results indicate performance benefits especially under evolution as often only parts of the feature model need to be analyzed again.

Keywords: Configurable Software, Software Product Line, Product-Line Analysis, Product-Line Evolution, Variability Modeling, Feature Model, Modularity, Compositionality

Overview

In this talk, we give an overview on our past and on-going research efforts towards compositional analyses of software product lines. We argue that our notion of feature-model interfaces serves as a foundation for many future analyses of product lines. Feature models are used in most product-line analyses, and especially in family-based analyses [Th14]. Hence, compositional feature modeling is the key towards better scalability.

The main results on feature-model interfaces have been presented at the International Conference on Software Engineering in Austin, Texas [Sc16]. We proved that a set of feature-model analyses can be modularized using feature-model interfaces in a sound and complete manner. Furthremore, we evaluated the benefit on the three-month history of an automotive feature model with 18,616 features, which we decomposed into 40 submodels. We measured large speed-ups with feature-model interfaces, which are due to the fact that feature-model analyses are reduced to boolean satisfiability problems and feature-model interfaces effectively reduce the number of boolean variables.

[1] University of Magdeburg, Germany
[2] University of Magdeburg, Germany
[3] TU Braunschweig, Germany
[4] University of Magdeburg, Germany
[5] University of Magdeburg, Germany

Besides pure analysis of feature models, we also proposed how to incorporate feature-model interfaces into the verification of domain artifacts. At the International Workshop on Variability Modeling of Software-Intensive Systems in Salvador, Brazil, we have presented benefits for theorem proving with KeY [Th16].

While we measured speed-ups for analyses using feature-model interfaces, the computation of feature-model interfaces turned out to be a bottleneck. Existing approaches to eliminate features [Th11, Ac11] did not scale to our large industrial models. We improved those algorithms and evaluated which parameters overcome this bottleneck, as presented at the International System and Software Product Line Conference in Beijing, China [Kr16].

At the International Workshop on Feature-Oriented Software Development in Amsterdam, The Netherlands, we presented another application of feature-model interfaces. In particular, we derived and explained implicit constraints for any given subtree of the feature model, which are otherwise unknown in large feature models [An16].

In our on-going work, we evaluate whether other analyses, such as type checking, can also be improved with feature-model interfaces. It seems that we identified a further bottleneck, which is the composition of several feature-model interfaces. The composition is needed if a query contains features of several feature-model interfaces and the conjunction of satisfiability solvers or binary decision diagrams are quite expensive operations.

References

[Ac11] Acher, Mathieu; Collet, Philippe; Lahire, Philippe; France, Robert B.: Slicing Feature Models. In: Proc. Int'l Conf. Automated Software Engineering (ASE). IEEE, Washington, DC, USA, pp. 424–427, 2011.

[An16] Ananieva, Sofia; Kowal, Matthias; Thüm, Thomas; Schaefer, Ina: Implicit Constraints in Partial Feature Models. In: Proc. Int'l Workshop Feature-Oriented Software Development (FOSD). ACM, New York, NY, USA, pp. 18–27, 2016.

[Kr16] Krieter, Sebastian; Schröter, Reimar; Thüm, Thomas; Fenske, Wolfram; Saake, Gunter: Comparing Algorithms for Efficient Feature-Model Slicing. In: Proc. Int'l Software Product Line Conf. (SPLC). ACM, New York, NY, USA, pp. 60–64, September 2016.

[Sc16] Schröter, Reimar; Krieter, Sebastian; Thüm, Thomas; Benduhn, Fabian; Saake, Gunter: Feature-Model Interfaces: The Highway to Compositional Analyses of Highly-Configurable Systems. In: Proc. Int'l Conf. Software Engineering (ICSE). ACM, New York, NY, USA, pp. 667–678, 2016.

[Th11] Thüm, Thomas; Kästner, Christian; Erdweg, Sebastian; Siegmund, Norbert: Abstract Features in Feature Modeling. In: Proc. Int'l Software Product Line Conf. (SPLC). IEEE, Washington, DC, USA, pp. 191–200, August 2011.

[Th14] Thüm, Thomas; Apel, Sven; Kästner, Christian; Schaefer, Ina; Saake, Gunter: A Classification and Survey of Analysis Strategies for Software Product Lines. ACM Computing Surveys, 47(1):6:1–6:45, June 2014.

[Th16] Thüm, Thomas; Winkelmann, Tim; Schröter, Reimar; Hentschel, Martin; Krüger, Stefan: Variability Hiding in Contracts for Dependent Software Product Lines. In: Proc. Int'l Workshop Variability Modelling of Software-Intensive Systems (VaMoS). ACM, New York, NY, USA, pp. 97–104, 2016.