Advances in Quantitative Software Product Line Analysis

Clemens Dubslaff

Institute for Theoretical Computer Science
Technische Universität Dresden, Germany
dubslaff@tcs.inf.tu-dresden.de

Abstract: The quantitative analysis of software is important, e.g., for energy-aware systems having constraints on energy consumption while guaranteeing a certain degree of utility. Analyzing software product lines is challenging due to the possibly exponential number of feature combinations. This paper sketches new approaches using probabilistic model checking for a quantitative analysis of software product lines and the integration of such into the software development process.

Software product lines (SPLs, see, e.g., [CN01]) comprise several related software products defined through their commonalities in terms of features. This allows for software configurations which align best to customer needs. Within *dynamic SPLs* [GH03], software products may change during runtime by activating or deactivating features, e.g., by in-app purchases or incremental updates. As the number of products in an SPL might be exponential in the number of supported features, possible interactions between features and arising side-effects are hard to predict for developers. Thus, testing and verification is highly desirable but also has to tackle the exponential blowup in the number of features. *Family-based approaches*, where the behavior of all software products in the SPL are represented by a single operational model, have been successfully applied in SPL verification [CHS$^+$10]. Besides asking for functional correctness of software, quantitative requirements are gaining more and more attention also in the context of SPLs, e.g., imposing restrictions on the energy consumption depending on the chosen features.

Compositional Framework for Dynamic Probabilistic Software Product Lines

Markov decision processes (MDPs) are a standard state-based operational model for systems which contain both, action-based and probabilistic transitions. Annotated with quantitative information, MDPs can be subject of many kinds of quantitative analysis.

Most of the modeling approaches for SPLs use monolithic models instead of compositional ones (see, e.g., [CHS$^+$10]). However, the intuitive concept of features directly yields a modular structure of the SPL: features are encapsulated in modules and composed towards a final software product. The compositional framework for SPLs presented in [DKB14] comprises MDP-formalisms which represent the behaviors of features (called *feature modules*) and an MDP-based component implementing the rules for dynamic feature switches during runtime (called *feature controller*). Specialized composition operators compose feature modules and feature controllers towards a standard MDP, which contains all the feature-aware operational behaviors of the dynamic SPL and allows for a family-based analysis.

Probabilistic Model Checking

A quantitative analysis using probabilistic model checking has a broad tool support for systems modeled by MDPs, e.g., by the prominent probabilistic model checker PRISM [HKNP06]. Whereas other approaches towards SPL verification require specialized algorithms (see, e.g., [CHS$^+$10]), the above framework with its MDP semantics hence enables an SPL verification which benefits from well-established standard methods. Applicability of this approach has been presented in [DKB14] for variants of the energy-aware network device EBOND [HDVH13]. An extension also supporting large-scaled dynamic SPLs with multi-features will be published in [DBK15]. Non-standard probabilistic model-checking techniques to investigate the interplay between energy and utility [BDK14], e.g., using the concepts of *quantiles*, have also been successfully applied for SPLs. Such techniques enable, e.g., to solve a *strategy synthesis problem* for dynamic SPLs asking for schedulers how feature switches can optimize the energy consumption of an energy-aware server system while guaranteeing a minimal bandwidth served in 99.9% of its executions.

Integration into the Development Process

Whereas existing case-studies for the framework presented in [DKB14] employed a direct implementation of feature modules and its feature controller into the input language of the model checker PRISM, an ongoing project called PROFEAT provides an extension of PRISM to support models for dynamic probabilistic SPLs. Besides simple static analysis and feature-model consistency checking, PROFEAT also eases the integration of quantitative analysis during the development process for SPLs.

References

[BDK14] C. Baier, C. Dubslaff, and S. Klüppelholz. Trade-off Analysis Meets Probabilistic Model Checking. In *Proc. of the 23rd CSL, 29th LICS*, pages 1–10. ACM, 2014.

[CHS$^+$10] A. Classen, P. Heymans, P.-Y. Schobbens, A. Legay, and J.-F. Raskin. Model Checking Lots of Systems: Efficient Verification of Temporal Properties in Software Product Lines. In *Proc. of the 32rd ICSE*, pages 335–344. ACM, 2010.

[CN01] Paul Clements and Linda Northrop. *Software Product Lines: Practices and Patterns*. Addison-Wesley Professional, 2001.

[DBK15] C. Dubslaff, C. Baier, and S. Klüppelholz. Probabilistic Model Checking for Feature-oriented Systems. In *TAOSD XII*, LNCS, to appear, 2015. Springer.

[DKB14] C. Dubslaff, S. Klüppelholz, and C. Baier. Probabilistic Model Checking for Energy Analysis in Software Product Lines. In *Proc. of the 13th MODULARITY*, pages 169–180. ACM, 2014.

[GH03] H. Gomaa and M. Hussein. Dynamic Software Reconfiguration in Software Product Families. In *Proc. of the 5th PFE*, LNCS, pages 435–444. Springer, 2003.

[HDVH13] M. Hähnel, B. Döbel, M. Völp, and H. Härtig. eBond: Energy Saving in Heterogeneous R.A.I.N. In *Proc. of the 4th (e-Energy)*, pages 193–202. ACM, 2013.

[HKNP06] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A Tool for Automatic Verification of Probabilistic Systems. In *Proc. of the 12th TACAS*, LNCS, pages 441–444. Springer, 2006.