Introduction to the theme issue on variability modeling of software-intensive systems

Andrzej Wąsowski¹ · Thorsten Weyer²

Received: 5 August 2015 / Accepted: 15 August 2015 / Published online: 13 October 2015
© Springer-Verlag Berlin Heidelberg 2015

1 Introduction

Variability in software-intensive systems (such as engine control or driver assistance systems in modern vehicles, flight control systems in the aviation sector) is a major challenge when developing high quality systems efficiently and in short release cycles. New technologies and architectural paradigms will make the development and operation of variable-products even more difficult and risky in future. For instance, the Internet of Things poses the requirement of inter-system collaboration in dynamically build networks in a (partially) unknown environment. In this context the management of variability during the entire life cycle of such systems (not only during design time) becomes indispensable.

It is generally accepted that the explicit consideration of variability information complemented by the separate modeling of such information is an important basis as well as a powerful means for coping with variability in an engineering process. On the other hand, new technologies and system paradigms create opportunities to apply techniques from variability management and variability modeling in order to solve the related upcoming engineering challenges.

This theme issue focuses broadly on innovative work in the area of variability modeling and management. We have particularly invited contributions with a strong variability modeling aspect, but also addressing the wider area of variability management, e.g., requirements, architecture, analysis, implementation, evolution and teaching of vari-

ability modeling. The purpose is to present new results for mastering variability throughout the whole lifecycle of systems, system families, and product lines.

2 In this issue

The issue includes six papers discussing diverse use cases for models and modeling in variability intensive systems. We received a total of 25 manuscripts. All submitted manuscripts were rigorously reviewed by an international group of experts in the fields of Variability Modeling and Software Product Line Engineering. Finally, the following six papers were accepted to become published in the journal.

Sánchez et al. [6] propose to use the Drupal content management system as a benchmark for evaluating variability testing methods. Drupal is a highly configurable system with a large number of pluggable features (components). The authors construct a feature model for Drupal in order to enable meaningful feature interaction testing. The feature model is augmented with attributes summarizing the development process properties for each features. This is likely the largest attributed feature model published so far. The authors demonstrate that the use of these non-functional attributes increases effectiveness of test selection against the case where only configuration models are used. We believe that the researchers in the variability testing and in the feature modeling communities will find the paper interesting.

Variability modeling typically links high-level models (feature models) to low-level implementation artifacts. Thus evolution of variability models closely follows evolution of systems. Dintzner et al. [3] study fine grained evolution of the feature model and feature implementations in the Linux kernel project. The model is specified in Kconfig, a domain specific language developed by the kernel

✉ Andrzej Wąsowski
wasowski@itu.dk

¹ IT University of Copenhagen, Copenhagen, Denmark

² University of Duisburg-Essen, Essen, Germany

project. The study encompasses analyzing detailed model changes over 16 releases of the Linux kernel for 20 architecture specific models. The main contribution of the paper is the FMDiff tool that allows computing differences between feature models. The tool is shown to be robust and applicable to large analysis studies like the presented one. To the best of our knowledge, this study constitutes the largest and the most realistic model differencing study available so far.

Dynamic software product lines is a paradigm for developing adaptive configurable software systems that allow developers and users to stage the variability resolution process in several phases, including late reconfiguration at run-time. In dynamic software product lines, feature models become much more expressive, as they need to include binding times for individual constraints, along with specification of what are the legal reconfiguration trajectories of a given system. Lochau et al. [4] present the necessary modeling language extension using requirements from an industrial case study and then implement a verification procedure for analysis of reconfiguration scenarios against constraints on binding times and dependencies of features. The work bridges communities working on verification of behavioral models and analysis of feature models.

Software Product Line Testing is a challenging task, in particular, because of the combinatorial explosion of the number of derivable products. In the experience report “Statistical Prioritization for Software Product Line Testing” [2] Devroey, Perouin, Cordy, Samih, Legay, Schobbens, and Heymans describe a testing method for product line testing which is based on feature models, behavior models and statistical testing. The approach aims at reducing the combinatorial explosion of the state space in product line testing by applying statistical test ideas. The authors also report the benefits they gain from applying their approach in the context of a software product line of assistance systems for helicopter pilots.

Queiroz, Passos, Valente, Hunsen, Apel and Czarnecki (“Empirical Assessment of Two Approaches for Specifying Software Product Lines Use Case Scenarios” [5]) report the results of an empirical investigation of two different specification approaches for software product line scenarios, namely PLUSS and MSVCM. These two techniques are similar with regard to their inputs but different in respect of the compositional (MSVCM) and invasive (PLUS) style for variability management. The authors performed a controlled experiment to investigate the suitability of the approaches to clearly modularize software product line specifications

and concerning the effort for bootstrapping and evolve the software product line specifications.

In “The Shape of Feature Code: An Investigation of Twenty C-preprocessor-based System” [1] Bonifácio, Borba, Ferraz, and Accioly investigate twenty projects using C preprocessor directives for variability management. The authors analyze the projects in respect to three metrics: scattering degree, tangling degree, and nesting degree. The authors come to the conclusion that feature scattering follows a power-law distribution, whereas the other two metrics can rely on central statistics such as median or mean values. Based on the identified benchmarks, the authors derived several thresholds which can be applied in projects when using preprocessor-based annotations.

The six papers contribute in diverse ways to the field of variability modeling. They demonstrate the importance of variability models in software engineering through empirical studies and through new model-based techniques. We trust that the readers will find them interesting and timely.

Acknowledgements We would like to thank the Editors-in-Chief of the Journal on Software and Systems Modeling for agreeing to publish this theme issue on Variability Modeling. We thank all the authors who have submitted their work to this highly competitive theme issue. As many as 75 reviewers have contributed substantial time to reviewing the works. Finally, we thank Martin Schindler of SoSyM for patient support during the lengthy editorial process of this theme issue.

References

1. Bonifácio, R., Borba, P., Ferraz, C., Accioly, P.: Empirical assessment of two approaches for specifying software product line use case scenarios. *Softw. Syst. Model.* doi:[10.1007/s10270-015-0471-3](https://doi.org/10.1007/s10270-015-0471-3) (2015)
2. Devroey, X., Perouin, G., Cordy, M., Samih, H., Legay, A., Schobbens, P.-Y., Haymans, P.: Statistical prioritization for software product line testing: an experience report. *Softw. Syst. Model.* doi:[10.1007/s10270-015-0479-8](https://doi.org/10.1007/s10270-015-0479-8) (2015)
3. Dintzner, N., van Deursen, A., Pinzger, M.: Analysing the linux kernel feature model changes using FMDiff. *Softw. Syst. Model.* doi:[10.1007/s10270-015-0472-2](https://doi.org/10.1007/s10270-015-0472-2) (2015)
4. Lochau, M., Bürdek, J., Hölzle, S., Schürr, A.: Specification and automated validation of staged reconfiguration processes for dynamic software product lines. *Softw. Syst. Model.* doi:[10.1007/s10270-015-0470-4](https://doi.org/10.1007/s10270-015-0470-4) (2015)
5. Queiroz, R., Passos, L., Valente, M.T., Hunsen, C., Apel, S., Czarnecki, K.: The shape of feature code: an investigation of twenty c-preprocessor-based systems. *Softw. Syst. Model.* doi:[10.1007/s10270-015-0483-z](https://doi.org/10.1007/s10270-015-0483-z) (2015)
6. Sánchez, A.B., Segura, S., Parejo, J.A., Ruiz-Cortés, A.: Variability testing in the wild: the Drupal case study. *Softw. Syst. Model.* doi:[10.1007/s10270-015-0459-z](https://doi.org/10.1007/s10270-015-0459-z) (2015)

Andrzej Wąsowski is Associate Professor at the IT University of Copenhagen. Earlier, he worked also at Aalborg University in Denmark, and as visiting professor at INRIA Rennes and University of Waterloo, Ontario. His interests are in semantic foundations and tool support for model-driven development, especially for software product lines and component-based systems. Many of these projects involve commercial or open-source partners, primarily in the

domain of safety-critical embedded systems. Wąsowski holds a PhD degree from the IT University of Copenhagen, Denmark (2005) and a M.Sc. Eng. degree from the Warsaw University of Technology, Poland (2000). He is a recipient of the Sapere Aude research leadership award from The Danish Council for Independent Research (2012).

Thorsten Weyer is Research Group Leader at the University of Duisburg-Essen and Head of 'Requirements Engineering and Conceptual Design' at paluno (The Ruhr Institute for Software Technology). For more than 15 years, he has been working as a researcher and consultant with regard to the areas requirements engineering, system analysis, variability management, and model-based software engineering. He is a member of organization committees, pro-

gram committees and reviewer for scientific conferences, journals and international research funding. Thorsten Weyer is vice chair of the council of the International Requirements Engineering Board (IREB) and member of the editorial board of the 'Requirements Engineering Magazine'.