

Next-Generation Model-based Variability Management: Languages and Tools

Mathieu Acher, Patrick Heymans, Philippe Collet, Philippe Lahire

▶ **To cite this version:**

Mathieu Acher, Patrick Heymans, Philippe Collet, Philippe Lahire. Next-Generation Model-based Variability Management: Languages and Tools. MODELS 2012 - 15th International Conference on Model Driven Engineering Languages & Systems, Sep 2012, Innsbruck, Austria. hal-00767170

HAL Id: hal-00767170

https://inria.hal.science/hal-00767170

Submitted on 19 Dec 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Next-Generation Model-based Variability Management: Languages and Tools

Mathieu Acher[1], Patrick Heymans[1,2], Philippe Collet[3], and Philippe Lahire[3]

[1] PReCISE Research Centre, University of Namur, Belgium
{mac,phe}@info.fundp.ac.be
[2] INRIA Lille-Nord Europe, Université Lille 1 – LIFL – CNRS , France
[3] I3S – CNRS UMR 6070 Université Nice Sophia Antipolis, France
{lahire, collet}@i3s.unice.fr

Abstract. Variability modelling and management is a key activity in a growing number of software engineering contexts, from software product lines to dynamic adaptive systems. Feature models are the defacto standard to formally represent and reason about commonality and variability of a software system. This tutorial aims at presenting next generation of feature modelling languages and tools, directly applicable to a wide range of model-based variability problems and application domains. Participants (being practitioners or academics, beginners or advanced) will learn the principles and foundations of languages and tool-supported techniques dedicated to the model-based management of variability.

Keywords: Feature Models, Model Management, Variability Modelling, Automated Reasoning, Language, Model Composition and Differences

The estimated length of the tutorial is 3 hours. Specific details about the topic, goal, schedule and supporting material of the tutorial are given in Section 1. Target audience and any pre-requisite background required by attendees are described in Section 2. A brief biography of presenters is given in Section 3.

1 Description of the Tutorial

1.1 Topic

In many domains, systems heavily rely on software and have to be efficiently extended, changed, customized or configured for use in a particular context (e.g., to respond to the specific expectations of a customer) [1, 2]. The challenge for software practitioners is to develop and use the right *models*, *languages*, and *tool-supported techniques* to produce and maintain multiple similar software products (variants), exploiting what they have in common and managing what varies among them. A lack of flexibility in the reusable artifacts or a scope that is too large may have severe consequences on the model-based engineering process. Therefore the modelling and management of *variability* in software intensive systems, for example *Software Product Lines* (SPLs), is a critical task [3].

Feature Models (FMs) are the defacto standard for modelling and reasoning about commonality and variability of a software system (e.g., an SPL). FMs are widely used to compactly define the valid combinations of features supported by an SPL, each valid combination of features (called configuration) corresponding to a given product. Numerous model-based SPL approaches rely on FMs for various purposes: generation of

product configurators, customisation of other artifacts (e.g., code, models) to derive members of an SPL [4, 5], reasoning about properties of an SPL [6, 7], etc. FMs have been intensively studied by academics during the last two decades. The research output is notably a formal semantics, the development of various automated reasoning operations and benchmarks, tools and languages [8]. Industrial tools (pure::variants, Gears) have also been developed and support FMs to fulfill the growing interest of practitioners. FMs will also be part of the Common Variability Language (CVL) standard[4].

The effort still continues since FMs are becoming increasingly complex (e.g., 6000+ features for Linux), handled by several stakeholders or organizations, and used to describe features at various levels of abstraction. Yet, barriers hindering a wider adoption of FMs, for example in industrial contexts, remain and have been reported (e.g., scalability [8], inadequate mechanisms to achieve separation of concerns [9], lack of relationships among languages [10]). To overcome current limitations, we have developed a set of complementary languages (TVL and FAMILIAR) and tool-supported techniques.

1.2 Goal

During the tutorial, participants will learn solutions applicable to a wide range of model-based variability problems and application domains while being supported by concrete tools and languages. We will notably expose new capabilities offered to modellers and show that, without the advances made, some model-based analysis would not be made possible in several case studies. The languages and tools have already been applied in practical contexts (industrial and academic), in different application domains (printers, semi-conductors, medical imaging, video surveillance, etc.) and for various purposes (extraction of FMs from legacy software artifacts [11], SPL evolution [12], management of multiple models [9, 13], model-based validation of SPLs [14], large scale configuration of FMs [15], etc.)

The tutorial aims at introducing new feature modelling languages, tools and their underlying environments, describing their theoretical foundations and showing how their combined use can practically handle the whole process from variability modelling to product configuration. Firstly, we give the participants an overview of TVL, a text-based feature modelling language. We emphasize its concise, natural, yet expressive syntax that subsumes most existing FM dialects [16]. Secondly, we present FAMILIAR (for FeAture Model scrIpt Language for manIpulation and Automatic Reasoning) a language dedicated to the management of multiple FMs [17]. We introduce practical techniques to import, export, compose, decompose, edit, configure, reason about several FMs and combine these operations to realise complex model-based variability management tasks. As effort has been made to support interoperability between the two languages, their combine use can be achieved. At the end of the tutorial, participants of the tutorial will be able to go further with the languages and modelling techniques, and to reuse them in practical or academic contexts.

1.3 Plan

The tutorial is based on slides and supporting demonstrations. An archive will be provided to participants, including tutorial slides, related articles, TVL models, FAMILIAR scripts and packaged tools to interactively play with the models during the tutorial. A concrete, end-to-end demonstration is used for modeling FMs with TVL and performing management operations of FMs with FAMILIAR. The demonstration is based on the re-engineering of a car configurator, that includes the modelling of configuration options

[4] http://www.omgwiki.org/variability/doku.php

using TVL and the management of the TVL model(s) using FAMILIAR. The estimated length of the tutorial is typically 3 hours:

1. (1 hour) The tutorial will start with a short explanation of variability modelling and FMs. We will present TVL, exposing its textual nature and formal semantics. Using the illustrative example, we will explain the main constructions (feature, group cardinalities, attributes) and modularity capabilities of the language. The resulting TVL models will then be analysed by FAMILIAR.
2. (1,5 hours) In the second part, we will present some relevant analyses and operations for the illustrative example and show how to perform them using FAMILIAR. These analyses will include usual satisfiability checking, detection of dead features and wrong cardinalities, but also more advanced operations (aggregation, differencing, slicing and merging of FMs).
3. (0,5 hour) The session will end up with a short summary of what have been done, a broader overview of the applications already realised by our tool suite and a report on our experience with the design and development of the languages. It should strengthen discussions with the audience (tools information, applicability of the techniques, etc.)

1.4 Material and Experience

Numerous presentations of the languages and tools have already taken place in the past. An empirical evaluation of TVL on four real industrial projects has been published and presented at SLE'10 [18]. We plan to reuse introductory slides of TVL used in industrial contexts. TVL has also been presented in VaMoS'10. FAMILIAR has been presented in numerous venues: VaMoS'11 (tool demonstration), SAC'11 (formal presentation), french SPL day (tool demonstration), and ASE'11 (tool demonstration track). It is also currently being used in the french national project YourCast (`http://yourcast.unice.fr`, started in 2012) which aims at building a complex industrial strength SPL of model-based information broadcast systems.

Furthermore, the languages and tools are extensively used during an SPL course given to Masters' students at University of Namur. As a running project, students have to re-engineer car configurators using TVL, FAMILIAR and modelling techniques.

The web page `http://www.info.fundp.ac.be/spltutorials/` gathers some available material we plan to reuse for the tutorial. It includes the slides and screencasts used during the aforementioned demonstrations or the teaching material and illustrative example used in the SPL course.

2 Audience

Target audience. The tutorial targets *software practitioners* – professionals working in industrial contexts: requirements engineers, domain analysists, software architects, or product line managers – that look for languages and tools to model and manage variability. For example, TVL has already been applied in four industrial case studies [18] while FAMILIAR can tackle numerous variability problems reported in the industry [9]. The tutorial also targets an *academic audience*. For example, the recent research results as well as state-of-the-art existing techniques (e.g., reasoning operations reported in [8]) are integrated in FAMILIAR. During the tutorial, academics will learn the design and developement of novel model-based operations (e.g., merging, slicing, differencing techniques for FMs) [17, 9].

Pre-requisite background. Attendees, being industrial practitioners or academics, may have basic knowledge about software product lines and variability, but it is not required. The tutorial targets both beginners and advances.

Target participants of MODELS'12. More specifically, PhD students or modellers unaware of variability modelling techniques will have the opportunity to learn a widely used modelling formalism (FM) as well as existing languages and tools they can directly use. Furthermore, advanced modellers will learn novel techniques that may be reused in their academical or industrial contexts. We particularly target people involved or interested in the development of *model management tools*. Our experience in several case studies has shown that syntactical mechanisms are likely to be insufficient for operations like the computation of differences of FMs [12] or the composition/decomposition of FMs [19, 9]. We have rather developed a set of semantical operations. We will present their foundations and report on our experience, which can benefit for the attendees.

3 Presenters

Dr. Mathieu Acher is a post-doctoral researcher at University of Namur, Belgium. His research focuses on the modelling and management of variability in software intensive systems. He is the main developer of FAMILIAR for which he has designed and implemented novel automated operations. He is teaching a comprehensive SPL course at University of Namur, relying extensively on TVL, FAMILIAR and models in general. He has authored more than 20 peer-reviewed papers and has presented them in numerous international venues. He has performed demonstrations of FAMILIAR at VaMoS'11, SAC'11 and ASE'11.

Dr. Patrick Heymans is full professor of software engineering at University of Namur, Belgium and visiting researcher at INRIA/University of Lille/CNRS. He is founding member and co-director of the PReCISE research centre (50 researchers) where he leads the requirements engineering and software product line efforts. He has supervised 9 PhD theses and authored 85 peer-reviewed papers. He is a regular referee for top SE journals and conferences, and associate editor of IEEE TSE. Patrick was recently the program chair of RE'11. He is principal investigator on various international software engineering research projects and regularly acts as an advisor and trainer for IT companies.

Dr. Philippe Lahire is a full professor at the Université Nice Sophia Antipolis, France. He is involved in research on model-driven engineering, aspect-oriented modeling and software product line engineering. He has supervised 6 PhD theses and authored more than 30 peer-reviewed papers. Philippe regularly acts as member of program committees (TOOLS'07-'09, ECMFA'11, MODELS'12, etc.). He has been involved in several european and french research projects and also led Tempus collaborative projects with eastern countries.

Dr. Philippe Collet is an associate professor (HDR) at the Université Nice Sophia Antipolis, France. He is involved in research on applying model-driven techniques to large scale software architectures with focus on self-adaptive systems and software product lines. He has supervised 4 PhD theses and authored more than 40 peer-reviewed papers. In 2012, Philippe is program co-chair of the main french software engineering conference (CIEL). From 2002, he has been principal investigator on various industrial and national software engineering research projects.

References

1. Svahnberg, M., van Gurp, J., Bosch, J.: A taxonomy of variability realization techniques: Research articles. Softw. Pract. Exper. **35**(8) (2005) 705–754
2. Pohl, K., Böckle, G., van der Linden, F.J.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag (2005)
3. Chen, L., Babar, M.A., Ali, N.: Variability management in software product lines: a systematic review. In: SPLC'09. (2009) 81–90
4. Czarnecki, K., Antkiewicz, M.: Mapping features to models: A template approach based on superimposed variants. In: GPCE'05. Volume 3676 of LNCS. (2005) 422–437
5. Czarnecki, K., Pietroszek, K.: Verifying feature-based model templates against well-formedness ocl constraints. In: GPCE '06: Proceedings of the 5th international conference on Generative programming and component engineering, ACM (2006) 211–220
6. Classen, A., Heymans, P., Schobbens, P.Y., Legay, A., Raskin, J.F.: Model checking lots of systems: efficient verification of temporal properties in software product lines. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10). (2010) 335–344
7. Classen, A., Heymans, P., Schobbens, P.Y., Legay, A.: Symbolic model checking of software product lines. In: Proceedings of the 33rd International Conference on Software Engineering (ICSE'11). (2011) 321–330
8. Benavides, D., Segura, S., Ruiz-Cortes, A.: Automated analysis of feature models 20 years later: a literature review. Information Systems **35**(6) (2010)
9. Acher, M., Collet, P., Lahire, P., France, R.: Separation of Concerns in Feature Modeling: Support and Applications. In: AOSD'12, ACM (2012)
10. Hubaux, A., Classen, A., Mendonça, M., Heymans, P.: A preliminary review on the application of feature diagrams in practice. In: VaMoS'10. (2010) 53–59
11. Acher, M., Cleve, A., Collet, P., Merle, P., Duchien, L., Lahire, P.: Reverse engineering architectural feature models. In: ECSA'11. Volume 6903 of LNCS. (2011) 220–235
12. Acher, M., Heymans, P., Collet, P., Quinton, C., Lahire, P., Merle, P.: Feature Model Differences. In: 24th International Conference on Advanced Information Systems Engineering (CAiSE'12). LNCS, Springer (2012) to appear.
13. Acher, M., Collet, P., Gaignard, A., Lahire, P., Montagnat, J., France, R.: Composing multiple variability artifacts to assemble coherent workflows. Software Quality Journal (special issue: Quality Engineering for Software Product Lines) (2011) 1–46
14. Acher, M., Collet, P., Lahire, P., Moisan, S., Rigault, J.P.: Modeling variability from requirements to runtime. In: ICECCS'11, IEEE (2011) 77–86
15. Hubaux, A., Heymans, P., Schobbens, P.Y., Deridder, D., Abbasi, E.: Supporting multiple perspectives in feature-based configuration. Software and Systems Modeling 1–23
16. Classen, A., Boucher, Q., Heymans, P.: A text-based approach to feature modelling: Syntax and semantics of TVL. Science of Computer Programming, Special Issue on Software Evolution, Adaptability and Variability **76**(12) (2011) 1130–1143
17. Acher, M., Collet, P., Lahire, P., France, R.: A domain-specific language for managing feature models. In: SAC'11, PL Track, ACM (2011) 1333–1340
18. Hubaux, A., Boucher, Q., Hartmann, H., Michel, R., Heymans, P.: Evaluating a textual feature modelling language: Four industrial case studies. In: SLE'10. Volume 6563 of LNCS. Springer (2011) 337–356
19. Acher, M., Collet, P., Lahire, P., France, R.: Comparing approaches to implement feature model composition. In: ECMFA'10. Volume 6138 of LNCS. (2010) 3–19