Subdomain-oriented Implementation of Model-driven Software Product Lines

Christoph Elsner

Friedrich-Alexander-University Erlangen-Nuremberg, Germany
Christoph.Elsner@informatik.uni-erlangen.de
http://www4.cs.fau.de/~elsner/

1 Introduction

Software product line approaches for domain analysis—like, for example, feature-oriented domain analysis (FODA) [3]—suggest to assign the features of the product line domain to distinct subdomains to approach the complexity of large software systems. In ideal case, not only the conceptual feature model, but also the architecture and the implementation of a subdomain is held strictly separate from the others. This vision requires separation of concerns (SoC) on implementation level and, when having a model-driven presentation of the architecture, also on the level of models. If a multi-staged modeling approach is adopted (e.g., when having a computational-independent, a platform-independent and a platform-dependent modeling layer) the recomposition of the subdomains has to be possible on all of these abstraction layers. This does not only include the composition of models, but also the composition of model-to-model transformers and of model-to-code generators.

There already exist various approaches for implementing SoC on code level and model level (e.g., patterns, feature-oriented, and aspect-oriented technologies). The model-driven development framework openArchitectureWare [4], for example, supports aspect-oriented SoC on model transformer and generator level. However, all these are stand-alone technologies. An integrating concept to facilitate the development of product line subdomains in a strictly separate way is still missing.

2 Subdomain-oriented Implementation of Model-driven Software Product Lines

Our approach for subdomain-oriented composition of model-driven software product lines packages all artifacts related to one sub-domain (i.e., models, meta models, transformers, generators, and code) together with the knowledge *how* to integrate with the other subdomains. For model-driven product lines, this know-how of subdomain integration corresponds to the knowledge on the so-called *model transformation workflow*, in especially (1) when to load an addtional (aspectual) model, (2) when to execute an additional (aspectual) model transformer or code generator or (3) when to include additional manually-written (aspectual) code

during the generation of a specific product. Specifying the subdomain workflows to facilitate their easy composition to cover a specific, compound domain is therefore one of our main objectives. [2]

3 Composition of Model Transformation Workflows

We currently analyze and implement two different approaches for composing model transformation workflows, all based on the XML-like openArchitectureWare [4] workflow language, which we already have extended with a model-based foundation using the Xtext framework [5]. This makes the workflow language a domain-specific *modeling* language (DSML) and allows for easier processing. The first approach will extend the language itself with an aspect-oriented syntax and semantics and a corresponding weaver with additional superimposition capabilities to specifically suit the needs of workflow weaving. The second approach will employ the VML tool suite, which is the generic successor of the VML4RE [1] variability modeling language. The VML tool suite facilitates rapid development of an additional DSML defining composition operators for combining domain-specific models models; we will use it to develop our own VML dialect VML4oAW for composing workflow models.

4 Contribution

The presentation will motivate the need for subdomain-oriented product line development and highlight the specific implications when implementing model-driven product lines. It will explain how model transformation workflows can support subdomain-oriented composition and discuss the two implementation approaches with respect to conciseness and expressiveness.

References

1. Mauricio Alferez, Uira Kulesza, Nathan Weston, Joao Araujo, Vasco Amaral, Ana Moreira, Awais Rashid, and Michael C. Jaeger. Ample deliverable 1.3: A metamodel for aspectual requirements modelling and composition. `http://ample.holos.pt/gest_cnt_upload/editor/File/public/AMPLE_WP1_D13.pdf`.
2. Christoph Elsner, Daniel Lohmann, and Wolfgang Schröder-Preikschat. Towards separation of concerns in model transformation workflows. In Steffen Thiel and Klaus Pohl, editors, *12th Software Product Line Conf. (SPLC '08), Second Volume*. Lero, 2008.
3. K. Kang, S. Cohen, J. Hess, W. Novak, and S. Peterson. Feature-oriented domain analysis (FODA) feasibility study. Technical report, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, November 1990.
4. OpenArchitectureWare homepage. `http://www.openarchitectureware.org/`.
5. Eclipse Modeling - Textual Modeling Framework - Xtext. `http://www.eclipse.org/modeling/tmf/`.