Support for complex product line populations

Sascha El-Sharkawy, Christian Kröher, Klaus Schmid
University of Hildesheim, Institute of Computer Science
Marienburger Platz 22, D-31141 Hildesheim, Germany
{elscha, kroeher, schmid}@sse.uni-hildesheim.de

ABSTRACT

In this paper, we describe EASy-Producer, a prototypical tool for complex and large-scale Software Product Line (SPL) development. The tool enables SPL engineers to reduce complexity by combining derivation and composition techniques to manage one large SPL as a combination of individual, but interrelated SPLs.

1. THE TOOL ENVIRONMENT

EASy-Producer (EASy stands for Engineering Adaptive Systems) utilizes the main principles of multi-level staged configuration [1], multi-dimensional variability modeling [4] and multiple software product lines [2, 3]. These approaches are state of the art approaches to address the challenge of handling complex and large-scale SPLs.

The current prototype is a new implementation of the earlier EASy-Producer prototype [5]. It uses decision modeling as described in [6] as basis for variability modeling. Like its predecessor it is embedded within the Eclipse environment.

In EASy-Producer each SPL is managed in its own Product Line Project (PLP). The tool does not fundamentally differentiate between new, partially or fully configured SPL and an instantiated product. A PLP can be part of a derivation chain that results from one of the three different scenarios depicted in Figure 1:

1. A generic PLP can be (partially) instantiated to form a more specific PLP. In Figure 1 a specialized PLP for picture handling *PL_Picture* is derived from a generic one *(PL_Graphics)* that supports general graphic processing.
2. Products can be directly derived from a PLP with open decisions. An example for this is *Pd_1*.
3. Multiple PLPs can be combined as basis for an infrastructure or a product. Examples for this are *PL_Picture* and *Pd_3*.

The decisions made within the configuration step of one PLP lead to the instantiation of the related artifacts of the PLP within the derivation step. In case of *PL_Picture* mentioned above, this could mean that all video processing classes are deleted, whereas decisions relevant to picture functionality (e.g., jpg or gif file handling) would remain open and associated parts of the artifacts would not be instantiated

Each PLP stores information about its pre- and successors, but it is not mandatory to have all associated projects locally available to allow collaborative work. However, if the predecessors can be accessed, it facilitates updating the infrastructure and the variability model.

SPLC'11, August 21–26, 2011, Munich, Germany.
ACM 978-1-4503-0789-5/11/08.

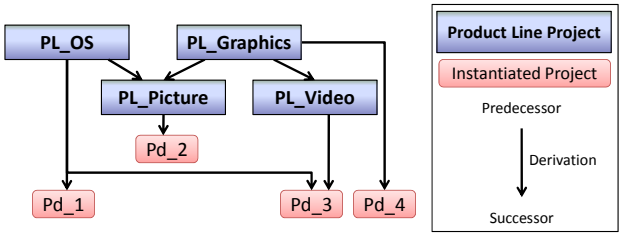

Figure 1: Example project structure

2. CONCLUSION AND FUTURE WORK

EASy-Producer is a prototype combining several state of the art techniques like multi-level staged configuration and multiple software product lines to meet the needs of large-scale product line development. However, there are still open issues such as more sophisticated approaches to model interdependencies among PLPs. We will further research on a better modularization for instantiating and combining assets of aggregated PLPs as well as on the modeling and implementation of multiple binding times.

3. ACKNOWLEDGMENTS

This work is partially supported by the INDENICA research project, funded by the European Commission grant 257483, area Internet of Services, Software & Virtualisation (ICT-2009.1.2) in the 7th framework programme.

4. REFERENCES

[1] K. Czarnecki, S. Helsen, and U. Eisenecker. Staged configuration through specialization and multi-level configuration of feature models. In *Software Process: Improvement and Practice*, volume 10, pages 143–169, 2005.

[2] G. Holl, M. Vierhauser, W. Heider, P. Grünbacher, and R. Rabiser. Product line bundles for tool support in multi product lines. In *5th International Workshop on Variability Modeling of Software-Intensive Systems*, pages 21–28, 2011.

[3] M. Rosenmüller and N. Siegmund. Automating the configuration of multi software product lines. In *4th International Workshop Variability Modelling of Software-intensive Systems*, pages 123–130, 2010.

[4] M. Rosenmüller, N. Siegmund, T. Thüm, and G. Saake. Multi-dimensional variability modeling. In *5th International Workshop on Variability Modelling of Software-intensive Systems*, pages 11–20, 2011.

[5] K. Schmid and H. Eichelberger. Model-based implementation of meta-variability constructs: A case study using aspects. In *2nd International Workshop Variability Modelling of Software-intensive Systems*, pages 63–71, 2008.

[6] K. Schmid and I. John. A customizable approach to full-life cycle variability management. *Science of Computer Programming*, 53(3), pages 259–284, 2004.