Domain-Specific Languages and Model Transformations for Software Product Line

Javier Troya
Department of Computer Languages and Systems, University of Seville, Spain
jtroya@us.es

Juha-Pekka Tolvanen
MetaCase, Finland
jpt@metacase.com

Sergio Segura
Department of Computer Languages and Systems, University of Seville, Spain
sergiosegura@us.es

ABSTRACT

This tutorial introduces and demonstrates the use of Model-Driven Engineering in Software Product Lines. In particular, it teaches participants about domain-specific languages, metamodeling and modeling, and where these techniques can be best used (and where not). Along with modeling, tutorial teaches various model transformation approaches and how they can be effectively used to bring software product lines to a different domain and to optimize them. The use of models for handling product variation is demonstrated with real-life examples from various industries and product lines.

CCS CONCEPTS

•Software and its engineering → Software product lines; Model-driven software engineering; Domain specific languages; Software configuration management and version control systems; *Abstraction, modeling and modularity;* Architecture description languages; System modeling languages; feature modeling;

KEYWORDS

Tutorial, Software Product Line, Domain-Specific Language, Feature Model, Model Transformation, ATL

ACM Reference format:
Javier Troya, Juha-Pekka Tolvanen, and Sergio Segura. 2017. Domain-Specific Languages and Model Transformations for Software Product Line. In *Proceedings of SPLC '17, Sevilla, Spain, September 25-29, 2017,* 1 pages.
DOI: 10.1145/3106195.3106227

1 INTRODUCTION

Model-Driven Engineering (MDE) is a methodology that advocates the use of models as first-class entities. Models are constructed from the definition of Domain-Specific Languages (DSLs) and they can be manipulated with the use of Model Transformations (MTs). DSLs and MTs are, therefore, key elements in MDE. This tutorial presents how to deal with Software Product Lines (SPLs) following the MDE methodology, for which it explains how to define SPLs with DSLs and how to manipulate them with MTs.

2 PLAN AND CONTENTS

The first part of the tutorial focuses on expressing variability in different ways, including parameter tables, feature models and DSLs. Parameter tables illustrate a basic approach of capturing variation and variants, made by entering or choosing legal parameter values. Domain engineers provide the type of parameters and application engineers give values to them. Feature models (FMs) are a special type of information model applied in product lines. It adds dependencies among variation points. Domain engineers create FMs covering the whole product family. Application engineers then choose which features to include from the family feature tree. With DSLs, the language definition itself defines the variation space and related rules, and language users follow this variation space when using the language to create variants. Code is then generated from the variant models. Domain engineers define the DSLs and application engineers use them. Following the MDE methodology, all languages for variability are defined via metamodels.

The second part of the tutorial focuses on Model Transformations (MTs). They provide the essential mechanisms for manipulating and transforming models, and are defined at the metamodel level. Therefore, all SPLs conforming to a metamodel can be transformed with the same MT. We distinguish between *out-place* transformations, where the target and source domains are different, and *in-place* transformations, where both domains are the same. The former can be used to move FMs to other domains such as UML architectural models, while the latter can be used for evolving or editing FMs. Among the several model transformation languages that exist, the Atlas Transformation Language (ATL) is in a prominent position due to its importance in both the academic and the industrial arenas, and is the one taught in the tutorial.

3 PRESENTERS' BACKGROUND

Juha-Pekka Tolvanen works at MetaCase and has been involved in domain-specific approaches since 1991. He has acted as a consultant world-wide for modeling language and code generation development, and has co-authored more than 60 publications.

Javier Troya and Sergio Segura work in the Department of Computer Languages and Systems of the University of Seville. Javier is an expert in model transformations, having co-authored more than 20 publications in the filed, while Sergio is an expert in feature models and their analysis, having co-authored more than 30 research publications in the field.

ACKNOWLEDGMENTS

The work by Javier and Sergio is partially supported by the European Commission (FEDER) and Spanish Government under CICYT project BELI (TIN2015-70560-R), and the Andalusian Government project COPAS (P12-TIC-1867).