Chapter 17
Model-Driven Requirements Specification for Software Product Lines

Mauricio Alférez
Universidade Nova de Lisboa, Portugal

Ana Moreira
Universidade Nova de Lisboa, Portugal

Vasco Amaral
Universidade Nova de Lisboa, Portugal

João Araújo
Universidade Nova de Lisboa, Portugal

ABSTRACT

Model-driven methods for requirements specification in Software Product Lines (SPLs) support the construction of different models to provide a better understanding of each SPL feature and intended use scenarios. However, the different models must be composed to show the requirements of the target applications and, therefore, help to understand how features will be integrated in a new product of a software product line. Although well-established standards for creating metamodels and model transformations exist, there is currently no established foundation that allows practitioners to distinguish between the different modeling and composition approaches for requirements models. This chapter provides an overview of different approaches for specifying requirements models and composing models for specific products of an SPL. In particular, it emphasizes one of the most recurring specification techniques: model-driven and use case scenario-based specification. This technique, in combination with feature models and the Variability Modeling Language for Requirements (VML4RE), integrates our approach for model-driven requirements specification for SPLs.

DOI: 10.4018/978-1-61692-874-2.ch017

INTRODUCTION

Software Product Lines are increasingly being adopted by major and medium-sized industrial players to quickly address change requests and improving time to market. SPLs enable modular, coarse-grained reuse through a set of core and varying software elements addressing a particular application domain (Clements & Northrop, 2002). Software Product Line (SPL) engineering is a promising approach to increase software quality and productivity. It encompasses the creation and management of families of products for a particular domain, where each product in the family is derived from a shared set of core assets, following a set of prescribed rules (Clements & Northrop, 2002).

An SPL product shares, with other systems, properties or functionalities that are relevant to some stakeholders. These are usually called "features" and express not only commonalities, but also variabilities that allow us to distinguish among products. The term "commonalities" refers to features that are mandatory to every product in an SPL. It is used to reference the parts of the requirements that are related to SPL common features. The term "variabilities" refers to the variable (optional, variation points and variants) features of an SPL. *Optional* features are not mandatory and might not be included in some of the products of an SPL. A *variation point* identifies a particular concept within the SPL requirements specification as being variable and it offers a number of *variants*. A *variant* describes a particular variability decision, such as a specific choice among *alternative variants*. Typically, we can model the available features and their dependencies (e.g., if feature *X* is selected, feature *Y* also must be selected) using a feature model (Czarnecki & Eisenecker, 2000; Kang, Cohen, Hess, Novak, & Peterson, 1990), that helps to capture the commonalities and variabilities of a family of products.

To understand each SPL feature and intended use scenarios of target products, Model-driven methods for requirements specification support the construction of different models that design the product behavior. However, to show the requirements of the target products, different models must be composed to help to understand and communicate to users, managers, testers and programmers the intended behavior of the new product to be produced from the SPL. Although well-established standards for creating metamodels and model transformations such as Meta-Object Facility (OMG, 2009a) exist, there is currently no established foundation for specifying requirements models for SPLs and compositing these models for specific products. This chapter introduces a classification of several existing approaches for model-driven requirements specification for Software Product Lines and focuses on exploiting use scenario-based techniques. We make explicit the different ways of specification taking into account the concrete syntax of the requirements models and the separation of three core components needed to specify and compose requirements models:

- the base models that specify requirements. For example, specifications of use scenarios using use cases models complemented with activity diagrams;
- variability information that makes explicit which are the SPL features that are common to all the products and which are the features that are particular to some products of the SPL; and
- configuration knowledge, which establishes the mapping between features and base models that specify requirements, for example, associating feature expressions, in the form of logical propositions, to specific model fragments. Also, in Model-Driven Development (MDD), configuration knowledge may include the specification of the transformations of SPL requirements models to compose models for specific products.