Chapter 3
Modeling and Documenting Aspect–Oriented Mobile Product Lines

Camila Nunes  
Pontifical Catholic University of Rio de Janeiro, Brazil

Uirá Kulesza  
Federal University of Rio Grande do Norte, Brazil

Roberta Coelho  
Federal University of Rio Grande do Norte, Brazil

Carlos Lucena  
Pontifical Catholic University of Rio de Janeiro, Brazil

Flávia Delicato  
Federal University of Rio de Janeiro, Brazil

Paulo Pires  
Federal University of Rio de Janeiro, Brazil

Thais Batista  
Federal University of Rio Grande do Norte, Brazil

ABSTRACT

Aspect-Oriented Software Development (AOSD) has evolved as a software development paradigm over the last decade. Recent research work has explored the use of Aspect-Oriented Programming (AOP) to modularize variations in product lines. This chapter presents a strategy for modeling and documenting aspect-oriented variations by integrating two existing approaches: (1) use cases are used to express the crosscutting nature of the variations of a mobile product line; and (2) crosscutting interfaces help the definition of the relevant variation join points that are raised by the mobile product line core and are extended by its respective variations. The synergy and benefits of the integration between these approaches are demonstrated by modeling and documenting MobileMedia, a software product line that provides support to manage different media (photo, music, and video) on mobile devices. Evolution scenarios of the MobileMedia are used to illustrate the benefits of the integrated usage of use cases and crosscutting interfaces in order to identify and analyze the change impact on the mobile product line.

DOI: 10.4018/978-1-4666-4054-2.ch003
INTRODUCTION

Aspect-Oriented Software Development (AOSD) (Kiczales et al., 1997) is a promising paradigm to improve the separation of concerns. Some concerns have impact in several modules of a software system and bring problems such as scattering, tangling and replicated code. Such concerns are known as \textit{crosscutting concerns}. The existing object-oriented mechanisms and techniques are not able to suitably modularize these crosscutting concerns. Examples of typical crosscutting concerns are: exception handling, persistence, distribution, monitoring and security. The goal of AOSD is to support advanced separation of concerns by modularizing crosscutting concerns through smaller units, called \textit{aspects}. The aspects introduce new abstractions and composition mechanisms, such as join point, pointcut, advice and inter-type declarations. AOSD has been used as a technique for improving the system reusability and maintainability.

Over the last years, AOSD techniques have been widely explored in both research and industry. Recent research work has explored the use of AOSD at the design and implementation levels in different contexts, such as, object-oriented application frameworks (Kulesza et al., 2006), design patterns (Hannemann & Kiczales, 2002), multi-agent systems (Nunes et al., 2009), model-driven development (Alvez et al., 2008; Greenfield, 2004; Sánchez et al., 2008) and software product lines (Alves et al., 2005). Many of these research works indicate that the synergy between Software Product Lines (SPLs) and AOSD is very promising to boost the reuse in software systems development. Software Product Lines (SPLs) (Clements & Northrop, 2001) aim to improve the software reuse through the modeling and implementation of the commonalities and variabilities of artifacts (Clements & Northrop, 2001). In the context of SPLs, there are some works using AOSD to modularize variations in the domain of mobile devices applications (Figueiredo et al., 2008; Alves et al., 2005). Examples of known variations in the mobile domain are: screen size, number of colors, available memory, and types of devices. In addition, there are non-functional features such as portability, performance, security. Most of these kinds of variabilities have a fine-grained and usually tend to exert a crosscutting behavior. In order to overcome this problem, AOSD techniques have been a suitable mechanism for modularizing these variations features.

As previously mentioned, there are many recent works that evaluate the (dis)advantages of AOSD in the implementation level. Although the literature have focused at the implementation level, the problem of using aspects in the context of product lines should be handled in the earliest stages of the software development. Current modeling approaches (Gomaa, 2004; Anthonysamy & Somé, 2008; Lopez-Herrejon & Batory, 2006) do not offer a complete solution to address the modeling of crosscutting variations in both domain analysis and design development stages. Besides, these approaches do not deal with change impact analysis during the evolution of SPLs. In other words, they do not maintain a synergy between the requirements and design models, and the respective artifacts responsible for the implementation of these variations. For example, the approach proposed by (Gomaa, 2004) does not take into account crosscutting variations that are part of the mobile SPL. In addition, when this approach is used alone, it does not allow an adequate evolution and management of the models.

This work proposes an integrated approach to allow the modeling and documentation of crosscutting variations in aspect-oriented mobile product lines. Our approach integrates: (1) the product line development approach based on Extension Joint Points (EJPs) (Kulesza et al., 2006) and Crosscutting Interfaces (XPIs) (Sullivan et al., 2005) – which addresses the modularization of crosscutting and non-crosscutting features in SPL architectures; and (2) the AO use cases (Jacobson & P.-W, 2004) in conjunction with the