

## Chapter 7

# Experience with Automatic Product Derivation of Mobile Applications Using Model-Driven Techniques

**Elder Cirilo**

*Pontifical Catholic University of Rio de Janeiro, Brazil*

**Uirá Kulesza**

*Federal University of Rio Grande do Norte, Brazil*

**Mário Torres**

*Federal University of Rio Grande do Norte, Brazil*

**Carlos Lucena**

*Pontifical Catholic University of Rio de Janeiro, Brazil*

### ABSTRACT

*In this work, the authors describe their experience on the adoption of a model-driven product derivation tool to help variability management and automatic product instantiation of a mobile product line, called MobileMedia. MobileMedia is a software product line (SPL) that provides support to manage (create, delete, visualize, play, send) different medias (photo, music, and video) on mobile devices. It was previously implemented as a Java Micro Edition (JME) application, in two different versions: (i) the first one uses conditional compilation implementation mechanisms to modularize their respective variations; and (ii) the second one adopts aspect-oriented programming, specifically AspectJ language, to explore a better modularization and separation of their respective variations. In this chapter, the authors illustrate how GenArch, a model-driven product derivation tool developed at the authors' research labs, can be used to automatically produce the different applications of both versions of MobileMedia SPL. The chapter discusses the impact of using these two different modularization techniques (conditional compilation and aspect-oriented programming) for the product derivation process by emphasizing their benefits and drawbacks and also showing the particular model-driven techniques used to better provide their instantiation.*

DOI: 10.4018/978-1-61520-655-1.ch007

## **INTRODUCTION**

A software product line (SPL) (Clements & Northrop, 2001) can be seen as a system family that addresses a specific market segment. A system family (Parnas, 1976) is a set of programs that shares common functionalities and maintain specific functionalities that vary according to specific systems being considered. Software product lines and system families are typically specified, modeled and implemented in terms of common and variable features. A feature (Czarnecki & Eisenecker, 2000) is a system property or functionality that is relevant to some stakeholder and is used to capture commonalities or discriminate among systems in SPLs.

Current SPL approaches (Clements & Northrop, 2001) propose the organization of the development process in two main stages: domain and application engineering. Domain engineering is concerned with: (i) the modeling of the scope of the SPL; (ii) the design of a flexible architecture that addresses the common and variable features of the SPL; and finally (iii) the implementation of the reusable code assets from the SPL architecture. In application engineering, different products of the SPL can be synthesized from the existing reusable code assets produced in domain engineering. Modern software engineering approaches, such as Generative Programming (Czarnecki & Eisenecker, 2000) and Software Factories (Greenfield, Short, Cook, & Kent, 2004), have motivated the automatic product instantiation in application engineering through the adoption of domain-specific languages and code generators. This process is also called product derivation (Deelstra, Sinnema, & Bosch, 2005). It involves the adequate selection, composition and customization of the code assets implemented for the SPL using appropriate tools. Over the last years, feature-based product derivation tools (Pure::Variants, 2009) (Gears, 2009) (Cirilo, Kulesza, & Lucena, 2008) have been proposed with the aim to synthesize automati-

cally SPL products using the feature model as a configuration domain-specific language.

In this work, we describe our experience on the adoption of a model-driven product derivation tool to help the variability management and automatic product instantiation of a mobile product line, called MobileMedia. MobileMedia is a software product line (SPL) that provides support to manage (e.g., create, delete, visualize, play and send) different medias (e.g., photo, music and video) on mobile devices. It is illustrated how GenArch, a model-driven product derivation tool developed at our research labs, can be used to produce automatically different products of two different versions of MobileMedia SPL: (i) one implemented using Java and conditional compilation; and (ii) the other one codified using Java and AspectJ. We discuss the impact of using these two different modularization techniques (conditional compilation and aspect-oriented programming) on the product derivation process by emphasizing their benefits and drawbacks, and also showing how model-driven techniques and mechanisms can be used to better support their automatic product derivation.

## **BACKGROUND**

In this section, we provide an overview of the MobileMedia product line and the GenArch product derivation tool. Both are fundamental to understand better our experience about the automatic product derivation of a product line in the mobile application domain using model-driven techniques and mechanisms.

### **MobileMedia Product Line**

To describe our experience on the adoption of a model-driven product derivation tool, we use a software product line for mobile devices, called MobileMedia (Figueredo, 2008). It is an evolu-

9 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/experience-automatic-product-derivation-mobile/66463](http://www.igi-global.com/chapter/experience-automatic-product-derivation-mobile/66463)

## Related Content

---

### A Formal Approach for Securing XML Document

Yun Bai (2010). *International Journal of Secure Software Engineering* (pp. 41-53).

[www.irma-international.org/article/formal-approach-securing-xml-document/39008](http://www.irma-international.org/article/formal-approach-securing-xml-document/39008)

### Filter/Wrapper Methods for Gene Selection and Classification of Microarray Dataset

Norreddine Mekour, Reda Mohamed Hamouand Abdelmalek Amine (2019). *International Journal of Software Innovation* (pp. 65-80).

[www.irma-international.org/article/filterwrapper-methods-for-gene-selection-and-classification-of-microarray-dataset/230924](http://www.irma-international.org/article/filterwrapper-methods-for-gene-selection-and-classification-of-microarray-dataset/230924)

### Systems Analysts' Attitudes Toward Information Systems Development

James J. Jiang, Gary Kleinand Joseph L. Balloun (2001). *Strategies for Managing Computer Software Upgrades* (pp. 179-188).

[www.irma-international.org/chapter/systems-analysts-attitudes-toward-information/29919](http://www.irma-international.org/chapter/systems-analysts-attitudes-toward-information/29919)

### A Novel Approach of Load Balancing and Task Scheduling Using Ant Colony Optimization Algorithm

Selvakumar A.and Gunasekaran G. (2019). *International Journal of Software Innovation* (pp. 9-20).

[www.irma-international.org/article/a-novel-approach-of-load-balancing-and-task-scheduling-using-ant-colony-optimization-algorithm/223519](http://www.irma-international.org/article/a-novel-approach-of-load-balancing-and-task-scheduling-using-ant-colony-optimization-algorithm/223519)

### Recommendations for Conducting Software Reviews

Yuk Kuen Wong (2006). *Modern Software Review: Techniques and Technologies* (pp. 268-280).

[www.irma-international.org/chapter/recommendations-conducting-software-reviews/26908](http://www.irma-international.org/chapter/recommendations-conducting-software-reviews/26908)