Author’s Accepted Manuscript

Generative Software Product Line Development using Variability-Aware Design Patterns

Christoph Seidl, Sven Schuster, Ina Schaefer

PII: S1477-8424(15)30060-9
DOI: http://dx.doi.org/10.1016/j.cl.2016.08.006
Reference: COMLAN233

To appear in: Computer Language

Received date: 21 December 2015
Accepted date: 24 August 2016

Cite this article as: Christoph Seidl, Sven Schuster and Ina Schaefer, Generative Software Product Line Development using Variability-Aware Design Patterns, Computer Language, http://dx.doi.org/10.1016/j.cl.2016.08.006

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting galley proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
Generative Software Product Line Development using Variability-Aware Design Patterns

Christoph Seidl, Sven Schuster, Ina Schaefer

Technische Universität Braunschweig, Germany

Abstract

Software Product Lines (SPLs) are an approach to reuse in-the-large that models a set of closely related software systems in terms of commonalities and variabilities. Design patterns are best practices for addressing recurring design problems in object-oriented source code. In the practice of implementing an SPL, instances of certain design patterns are employed to handle variability, which makes these “variability-aware design patterns” a best practice for SPL design. However, there currently is no dedicated method for proactively developing SPLs using design patterns suitable for realizing variable functionality. In this paper, we present a method to perform generative SPL development with design patterns. We use role models to capture design patterns and their relation to a variability model. We further allow mapping of individual design pattern roles to (parts of) implementation elements to be generated (e.g., classes, methods) and check the conformance of the realization with the specification of the pattern. We provide definitions for the variability-aware versions of the design patterns Observer, Strategy, Template Method and Composite. Furthermore, we support generation of realizations in Java, C++ and UML class diagrams utilizing annotative, compositional and transformational variability realization mechanisms. Hence, we support proactive development of SPLs using design patterns to apply best practices for the realization of variability. We realize our concepts within the Eclipse IDE and demonstrate them within a case study.

1. Introduction

Design patterns are templates for standard solutions to recurring design problems, predominantly in object-oriented languages [1], e.g., Java, C++ or UML class diagrams (see Sec. 2.2). Software Product Lines (SPLs) are an approach to reuse in-the-large where a family of closely related software systems is represented in terms of commonalities and variabilities—often referred to as features (see Sec. 2.1). SPLs utilize, among others, object-oriented programming and modeling languages to realize features.

Email address: c.seidl@tu-braunschweig.de (Christoph Seidl)

دریافت فوری متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات