

Constraints: The Heart of Domain and Application Engineering in the Product Lines Engineering Strategy

Raúl Mazo, University of Antioquia & CRI Panthéon Sorbonne University, France

Camille Salinesi, CRI Panthéon Sorbonne University, France

Daniel Diaz, CRI Panthéon Sorbonne University, France

Olfa Djebbi, CRI Panthéon Sorbonne University, France

Alberto Lora-Michiels, Baxter International Inc, Belgium

ABSTRACT

Drawing from an analogy between features based Product Line (PL) models and Constraint Programming (CP), this paper explores the use of CP in the Domain Engineering and Application Engineering activities that are put in motion in a Product Line Engineering strategy. Specifying a PL as a constraint program instead of a feature model carries out two important qualities of CP: expressiveness and direct automation. On the one hand, variables in CP can take values over boolean, integer, real or even complex domains and not only boolean values as in most PL languages such as the Feature-Oriented Domain Analysis (FODA). Specifying boolean, arithmetic, symbolic and reified constraint, provides a power of expression that spans beyond that provided by the boolean dependencies in FODA models. On the other hand, PL models expressed as constraint programs can directly be executed and analyzed by off-the-shelf solvers. This paper explores the issues of (a) how to specify a PL model using CP, including in the presence of multi-model representation, (b) how to verify PL specifications, (c) how to specify configuration requirements, and (d) how to support the product configuration activity. Tests performed on a benchmark of 50 PL models show that the approach is efficient and scales up easily to very large and complex PL specifications.

Keywords: Computer Science, Constraint-Based Product Lines, Information Systems, Product Line Analysis, Product Line Configuration, Product Line Integration, Product Line Specification, Product Line Reasoning, Product Line Verification

INTRODUCTION

Many experiences in the industry have shown that Product Lines engineering is an effective

way to deal efficiently with reuse during analysis, design, development, test or even delivery of series of products that contain similar and varying features. Starting from 3 products, the Product Line engineering strategy has positive effects on time to market, product quality and

DOI: 10.4018/jismd.2012040102

customer satisfaction (Clements & Northrop, 2001). Product Line success stories gathered in the SPLC “Hall of Fame”²¹ show companies such as Boeing, Cummins, HP, Nokia, Philips Medical Systems, or Toshiba benefited from the Product Line strategy in many ways, spanning from a drastic reduction of time to market, number of defects per product, engineering effort to deploy and maintain products, combined with a substantial increase in the number of products that can be deployed. The business benefits are multiple: reduced time to revenue, higher profit margins, improved ability to aim at market windows, higher profit margins, reduced risk in product deployment, and even improved reputation of the company due to better product quality².

The Product Line engineering strategy entails two activities: domain engineering and application engineering.

Application engineering consists in analysing, designing, building, customizing, or testing one product *by reuse*. Different kinds of artefacts can be reused: requirements, design fragments, architecture, code, test cases, etc. The reuse strategy is based on the exploitation of Product Line models built during domain engineering.

Domain engineering consists in specifying artefacts *for reuse*. This means specifying the artefacts to make them readily reusable, as well as specifying their reuse conditions. Many different specification languages have been proposed to support this. The most well known is probably FODA and its dialects (Kang *et al.*, 1990). However OVM (Pohl *et al.*, 2005), UML extensions (Ziadi, 2004; van der Maßen, Lichter, 2002), Dopler (Dhungana *et al.*, 2010), the text-based variability language (TVL) (cf. Boucher *et al.*, 2010; Classen *et al.*, 2011), and the DSL proposed in Mannion (2002) are noteworthy alternatives as they allow specifying Product Lines with complementary viewpoints such as marketing, architecture, logistics, maintenance, etc.

In our view, the approaches that exploit these formalisms have in common that they emphasize the role of constraints at both the level of domain engineering and application

engineering. Indeed, domain engineering can be seen in these approaches as the specification of variables (“features”, “attributes”, “variation points”, etc.) and constraints (“dependencies”). Application engineering then consists in defining values for these variables, while ensuring that the constraints are satisfied. Therefore, variables specify what can vary from a product to the other, in other terms the characteristics of artefacts that can be reused. On the other hand, constraints specify the reuse conditions, i.e., when artefacts can (or should) be reused or not.

Our approach is grounded on former research works that proposed to transform traditional Product Line Models (PLMs) into propositional logic (Mannion, 2002; Zhang *et al.*, 2004; Batory 2005; White *et al.*, 2008) or boolean constraints programs in order to reason about them (Benavides *et al.*, 2005; Trinidad *et al.*, 2008; Mendonça *et al.*, 2009). As these works showed it, using traditional Product Line specification formalisms raises a series of problems:

- Being different they are necessary to specify multiple views, but at the same time they are difficult to integrate;
- They are often quite limited in the sense that they are not rich enough to specify complex requirements or configuration constraints;
- They are contemplative, and therefore it is difficult to automate verification, analysis and configuration activities.

The approach taken in this paper makes a step beyond the aforementioned ones: we believe that constraints programming should be at the core of product line engineering rather than just a tool or a by-product issued after transforming classical models. Not only this allows specifying domain models that would not be specified when starting with traditional formalisms, but also it allows integrating multiple views, it provides users with a rich language to specify their requirements, and it brings Product Line specifications that are readily available for automated reasoning using constraints solvers.

34 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/article/constraints-heart-domain-application-engineering/65561

Related Content

Software Metrics Evaluation Based on Entropy

R. Selvarani, T.R.Gopalakrishnan Nair, Muthu Ramachandranand Kamakshi Prasad (2010). *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization* (pp. 139-151).

www.irma-international.org/chapter/software-metrics-evaluation-based-entropy/37030

Design Churn as Predictor of Vulnerabilities?

Aram Hovsepyan, Riccardo Scandariato, Maximilian Steffand Wouter Joosen (2014). *International Journal of Secure Software Engineering* (pp. 16-31).

www.irma-international.org/article/design-churn-as-predictor-of-vulnerabilities/118146

Music Melodies Suited to Multiple Users' Feelings Composed by Asynchronous Distributed Interactive Genetic Algorithm

Kota Nomuraand Makoto Fukumoto (2018). *International Journal of Software Innovation* (pp. 26-36).

www.irma-international.org/article/music-melodies-suited-to-multiple-users-feelings-composed-by-asynchronous-distributed-interactive-genetic-algorithm/201483

IT and Software Industry in Vietnam

Yuko Iwasaki (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1498-1506).

www.irma-international.org/chapter/software-industry-vietnam/29460

Prioritizing COVID-19 Vaccine Delivery for the Indian Population

Meet Singh, Subrata Modakand Dhruvasish Sarkar (2022). *International Journal of Software Innovation* (pp. 1-21).

www.irma-international.org/article/prioritizing-covid-19-vaccine-delivery-for-the-indian-population/301228