The Journal of Systems and Software 84 (2011) 285–300

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Key activities for product derivation in software product lines

Rick Rabiser[a,1], Pádraig O'Leary[b,*], Ita Richardson[c,2]

[a] Christian Doppler Laboratory for Automated Software Engineering, Johannes Kepler University, Altenberger Str. 69, 4040 Linz, Austria
[b] RiSE – Reuse in Software Engineering and Computer Science Department, Federal University of Bahia, Salvador, BA, Brazil
[c] Lero – The Irish Software Engineering Research Centre, University of Limerick, Ireland

ARTICLE INFO

Article history:
Received 1 February 2010
Received in revised form 17 June 2010
Accepted 26 September 2010
Available online 7 October 2010

Keywords:
Software product lines
Product derivation, Process

ABSTRACT

More and more organizations adopt software product lines to leverage extensive reuse and deliver a multitude of benefits such as increased quality and productivity and a decrease in cost and time-to-market of their software development. When compared to the vast amount of research on developing product lines, relatively little work has been dedicated to the actual use of product lines to derive individual products, i.e., the process of product derivation. Existing approaches to product derivation have been developed independently for different aims and purposes. While the definition of a general approach applicable to every domain may not be possible, it would be interesting for researchers and practitioners to know which activities are common in existing approaches, i.e., what are the key activities in product derivation. In this paper we report on how we compared two product derivation approaches developed by the authors in two different, independent research projects. Both approaches independently sought to identify product derivation activities, one through a process reference model and the other through a tool-supported derivation approach. Both approaches have been developed and validated in research industry collaborations with different companies. Through the comparison of the approaches we identify key product derivation activities. We illustrate the activities' importance with examples from industry collaborations. To further validate the activities, we analyze three existing product derivation approaches for their support for these activities. The validation provides evidence that the identified activities are relevant to product derivation and we thus conclude that they should be considered (e.g., as a checklist) when developing or evaluating a product derivation approach.

1. Introduction and motivation

There is a clear trend away from single systems to product lines in software engineering (Clements and Northrop, 2001; Pohl et al., 2005; van der Linden et al., 2007b). Software product lines (SPL) aim to leverage extensive reuse in software development to address many of the challenges in software development such as increasing quality requirements and competition in a global market. Software product line engineering (SPLE) involves domain engineering (building the product line) and application engineering (building products based on the product line). In domain engineering, reusable assets (e.g., requirements, components, documentation, test cases) are developed and their commonalities and variability are explicitly defined, typically using variability models. A significant body of research is available on approaches and notations for variability modelling and management, for example (Czarnecki and Kim, 2005; Gomaa, 2004; Pohl et al., 2005; Schmid and John, 2004). In application engineering, concrete products are built based on these reusable assets. Product derivation is a key process in application engineering and addresses the selection and customization of assets from the product line (utilizing the provided variability) to satisfy customer or market requirements (Deelstra et al., 2005). It is important to work on minimizing product-specific development in application engineering and maximize reuse.

In practice, a number of publications have shown that product derivation must not be underestimated. For example, (Griss, 2000) identifies the inherent complexity and the required coordination in the derivation process by stating that "*. . .as a product is defined by selecting a group of features, a carefully coordinated and complicated mixture of parts of different components are involved*". As (Deelstra et al., 2005) point out: the derivation of individual products from shared software assets is still a time-consuming and expensive activity in many organizations. Both publications base their statements on experiences made with product derivation in industry. Our own experiences in research industry collaborations also confirm that product derivation is often underestimated. A strong focus in SPLE has to be on domain engineering, i.e., building up the product line. However, product derivation brings the return of investment required for setting up the product line in the first

* Corresponding author.
E-mail addresses: rabiser@ase.jku.at (R. Rabiser), padraig.oleary@rise.com.br (P. O'Leary), ita.richardson@lero.ie (I. Richardson).
[1] Tel.: +43 0 732 2468 8873; fax: +43 0 732 2468 8878.
[2] Tel.: +353 0 61233799; fax: +353 0 61213036.

0164-1212/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.09.042

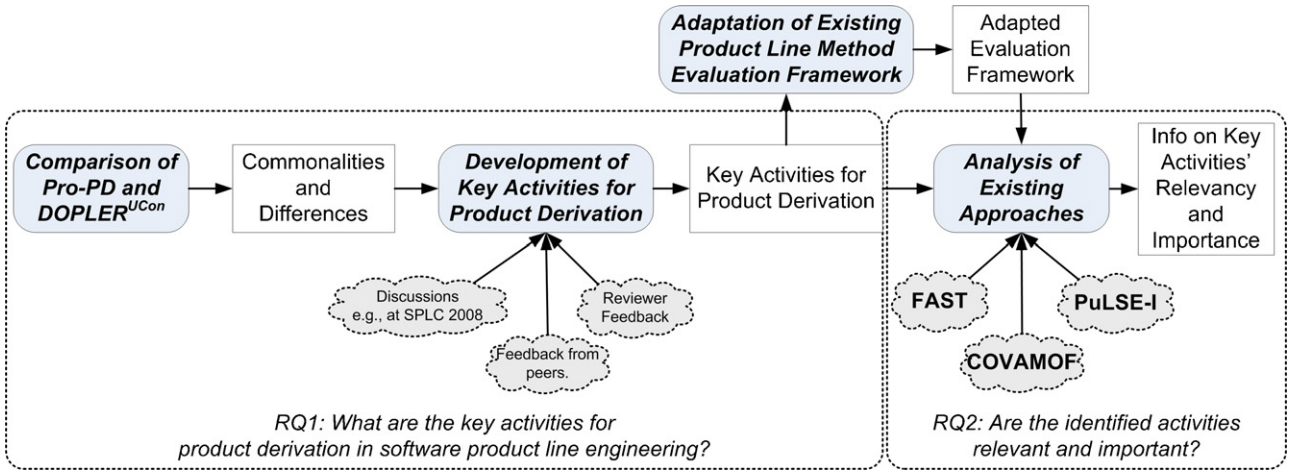

Fig. 1. Research method and research questions.

place by allowing to derive customized products quickly and in an automated way.

Research in SPL has, in the past, focused more on how to scope, define, and develop product lines rather than on how to effectively utilize them in product derivation. A recent systematic literature review (Rabiser et al., 2010) however shows an increasing number of publications, conference tracks, and workshops over the last decade demonstrating the general interest in product derivation. While the requirements for product derivation tool support have been outlined (Rabiser et al., 2010), there is still no clear picture regarding the activities to be supported. Available product derivation approaches and tools have been developed independently to address requirements in different contexts or domains.

Two such approaches are:

(i) Pro-PD (*Pro*cess reference model for *P*roduct *D*erivation) was developed at Lero (the Irish Software Engineering Research Centre) with the goal of defining a process reference model for product derivation as a foundation for situation-specific process approaches to product derivation. Pro-PD focuses on the activities, roles and work artefacts used to derive products from a software product line. Pro-PD uses process patterns that capture solutions to product derivation process challenges (e.g., co-ordinating product-platform synchronisation) as building blocks for creating a product derivation process instance. Pro-PD, its development, and its validation are also described in O'Leary (2010).

(ii) DOPLER^{UCon} (*D*ecision-*O*riented *P*roduct *L*ine *E*ngineering for effective *R*euse: *U*ser-centered *Con*figuration) was developed at the Christian Doppler Laboratory for Automated Software Engineering (Johannes Kepler University (JKU) Linz, Austria) driven by industry needs with the goal to define a user-centred, tool-supported product derivation approach. DOPLER^{UCon} is one of two parts in a decision-oriented product line engineering approach called DOPLER. The other part – DOPLERP^{VM} (Dhungana et al., 2010) – supports variability modelling and management. DOPLERP^{UCon} aims to support both domain experts like sales staff or managers as well as engineers in product derivation based on DOPLER variability models. DOPLERP^{UCon}, its development, and its validation are also described in (Rabiser, 2009).

Both approaches independently sought to identify product derivation activities, Pro-PD through its process reference model and DOPLERP^{UCon} through its tool-supported product derivation approach. Neither approach was designed exclusively for a particular organization or domain but the development of both approaches was driven by industry needs and experiences. The two approaches have already been applied in different cases (cf. Section 2).

In a research collaboration between Lero and JKU we have compared our approaches in detail and identified key activities for product derivation common to both approaches. While the two approaches have been developed in independent projects, with different goals and for different purposes, we still found many interesting parallels. In a previous publication (O'Leary et al., 2009) we presented an overview of our first results, i.e., we described key activities, important issues and lessons learnt for product derivation. In this paper we present details about the comparison and focus on the identification and validation of product derivation key activities. We illustrate the key activities with examples from industry collaborations at both Lero and JKU, and provide evidence for their relevance by systematically analyzing three often-cited and well-known product derivation approaches for their support for these activities.

2. Research method

The goal of this research is to define key activities for product derivation through comparing two product derivation approaches developed by the authors in two different, independent research projects. While a general approach to product derivation might not be possible, we envision that a list of activities that are common in existing approaches will help researchers and practitioners when developing, adapting or evaluating a product derivation approach.

More specifically we are investigating two research questions:

- *What are the key activities for product derivation in software product line engineering?* We elicit the activities by comparing our two approaches in detail and motivate the activities using examples from industry collaborations.
- *Are the identified activities relevant and important?* We systematically analyze existing product derivation approaches regarding their support for the activities using a validation framework.

Fig. 1 depicts an overview of our research method. We being by comparing our two product derivation approaches to elicit commonalities and differences. Based on these, we developed the key activities for product derivation which we refined based on discussions (remote and at conferences) and feedback from peers. Based on an adapted existing product line method evaluation framework, we finally analyzed the key activities to be able to provide evidence