Information and Software Technology 62 (2015) 78–100

Contents lists available at ScienceDirect

Information and Software Technology

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

Supporting distributed product configuration by integrating heterogeneous variability modeling approaches

José A. Galindo [a,e,*], Deepak Dhungana [b], Rick Rabiser [d], David Benavides [a], Goetz Botterweck [c], Paul Grünbacher [d]

[a] Department of Computer Languages and Systems, University of Seville, Spain
[b] Siemens AG Österreich, Corporate Technology, Vienna, Austria
[c] Lero–The Irish Software Engineering Research Center, University of Limerick, Ireland
[d] Christian Doppler Laboratory MEVSS, Johannes Kepler University, Linz, Austria
[e] Diverse Team, INRIA Rennes, France

ARTICLE INFO

Article history:
Received 28 April 2014
Received in revised form 1 February 2015
Accepted 4 February 2015
Available online 11 February 2015

Keywords:
Software product lines
Product configuration
Automated analysis

ABSTRACT

Context: In industrial settings products are developed by more than one organization. Software vendors and suppliers commonly typically maintain their own product lines, which contribute to a larger (multi) product line or software ecosystem. It is unrealistic to assume that the participating organizations will agree on using a specific variability modeling technique—they will rather use different approaches and tools to manage the variability of their systems.
Objective: We aim to support product configuration in software ecosystems based on several variability models with different semantics that have been created using different notations.
Method: We present an integrative approach that provides a unified perspective to users configuring products in multi product line environments, regardless of the different modeling methods and tools used internally. We also present a technical infrastructure and a prototype implementation based on web services.
Results: We show the feasibility of the approach and its implementation by using it with the three most widespread types of variability modeling approaches in the product line community, i.e., feature-based, OVM-style, and decision-oriented modeling. To demonstrate the feasibility and flexibility of our approach, we present an example derived from industrial experience in enterprise resource planning. We further applied the approach to support the configuration of privacy settings in the Android ecosystem based on multiple variability models. We also evaluated the performance of different model enactment strategies used in our approach.
Conclusions: Tools and techniques allowing stakeholders to handle variability in a uniform manner can considerably foster the initiation and growth of software ecosystems from the perspective of software reuse and configuration.

1. Introduction and motivation

Software product lines (SPL) are increasingly developed beyond the boundaries of single organizations [1]. For instance, in software ecosystems distributed organizations and teams create software products in a collaborative effort. Variability management and product configuration in such contexts need to reconcile the different modeling approaches, notations, and tools in use. Due to diverse practices in different domains it is unrealistic to assume the use of a single and standardized variability modeling approach, despite ongoing standardization efforts.[1] However, the increasing number of "island solutions" to variability modeling and product configuration hinders communication and collaboration between product line engineers. Especially in software ecosystems [1] it is infeasible to assume one kind of modeling approach for all units of the ecosystem. The required coordination between the participating organizations (e.g., along a supply chain) further complicates this issue. Hence,

* Corresponding author at: Department of Computer Languages and Systems, University of Seville, Spain.
E-mail addresses: jagalindo@us.es, jagalindo@inria.fr (J.A. Galindo), deepak.dhungana@siemens.com (D. Dhungana), rick.rabiser@jku.at (R. Rabiser), benavides@us.es (D. Benavides), goetz.botterweck@lero.ie (G. Botterweck), paul.gruenbacher@jku.at (P. Grünbacher).

[1] OMG Common Variability Language [2], http://www.omgwiki.org/variability/doku.php.

http://dx.doi.org/10.1016/j.infsof.2015.02.002
0950-5849/

there is a strong need for an integrative infrastructure enabling the collaboration between different organizations developing product lines. The approach needs to support different variability modeling languages, notations, and tools. It also needs to support variability at different levels of granularity to model, e.g., customer-facing features, architectural elements, or configuration decisions.

We propose the *Invar* approach, which facilitates the integration of heterogeneous variability models[2] potentially created by different teams. In this paper we focus on the product configuration aspects of our integrative infrastructure. *Invar* deliberately hides the internal technical aspects of using different variability models for configuration from the stakeholders performing the configuration. The specific tools or data formats (see [3–5]) used for defining the variability models are not relevant for the end users who primarily focus on the available configuration choices and their implications. *Invar* unifies configuration operations on variability models and allows modelers to freely choose a data representation by accessing variability models through web services. Our approach does *not* force organizations to *integrate* their configuration tools by adapting the internals of the tools. Instead, we allow them to *compose* their configuration mechanisms using wrappers and interface definitions. We validate our approach by integrating three different variability modeling "dialects", i.e., feature modeling, orthogonal variability modeling (OVM), and decision modeling. We also show how typical scenarios in software ecosystems can be supported with *Invar* and assess the performance of *Invar* regarding the different model enactment strategies of the approach, e.g., different orderings and settings of choices during product configuration.

An earlier version of this work appeared in [6] and the tool prototype was presented in a short tool demonstration paper [7]. In this article *we provide the following extensions* to this earlier work: (i) we discuss requirements for configuration based on multiple variability models (in different notations) in a software ecosystem context and relate the requirements to the literature; (ii) we describe the integration of a third variability modeling technique based on OVM [8], thus broadening the scope of our work; (iii) we extend *Invar* with different model enactment strategies allowing different orders in a configuration process based on multiple models; (iv) we present an experiment assessing the performance of the enactment strategies; (v) we demonstrate the feasibility and flexibility of our *Invar* approach by applying it to a realistic scenario, i.e., the configuration of the permission system in the Android ecosystem[3] based on multiple variability models; and (vi) we discuss the benefits of *Invar*, e.g., by comparing it to manual configuration based on multiple variability models.

The main contributions of this paper are:

- a set of *configuration* primitives for integrating arbitrary variability modeling approaches to support product configuration;
- a *method for defining dependencies between multiple variability models*;
- support for *composing and integrating heterogeneous variability modes* such as feature models, decision models, and orthogonal variability models;
- support for *different enactment strategies* during product configuration with multiple variability models to customize configuration orderings;
- *evidence regarding the feasibility of Invar* for different configuration scenarios by using it in an enterprise resource planning (ERP) context and for the Android ecosystem; and
- a *discussion of the benefits of Invar* compared to manual configuration based on multiple variability models.

The remainder of this paper is structured as follows: In Section 2 we outline our research questions and approach. In Section 3 we discuss the background of this work, i.e., software product lines and the three different variability modeling approaches we later integrate with our approach. In Section 4 we present an example of a multi product line as a further motivation for our work. Section 5 describes how *Invar* enables the use of heterogeneous variability models during product configuration. Section 6 presents the *Invar* prototype for three different variability modeling tools. Section 7 presents a validation of the flexibility of our approach using different scenarios of applying the approach derived from industrial experience in the ERP domain and from the Android ecosystem. We also present a study of the performance of our approach regarding its model enactment strategies. We discuss the benefits of *Invar* (e.g., when compared to manual configuration based on multiple models) and explicate the threats to validity of our research results. We discuss related work in Section 8 and conclude the paper with a summary and discussion of future work in Section 9.

2. Research approach

We explore the following four research questions:

RQ1: How can different variability modeling approaches be integrated to support product configuration in the context of software ecosystems?
RQ2: Is *Invar* sufficiently extensible and flexible to allow the integration of different variability modeling approaches?
RQ3: Does *Invar* support realistic configuration scenarios in software ecosystems?
RQ4: What is the impact of the *Invar* model enactment strategies on configuration performance?

In order to address these questions we followed the research process shown in Fig. 1, comprising the stages of analysis, implementation, and evaluation.

2.1. Analysis (RQ1)

Our main hypothesis is that different existing variability modeling flavors can be integrated to support product configuration in a software ecosystem context. To validate this idea, we first discussed it with experts from the SPL community and performed a literature review. Later, we extracted the commonality and variability among the different product line modeling approaches. There exist several families of variability modeling approaches that have been designed for different research scenarios [9], resulting in a large number of tools, languages, and operations. The variety of variability model flavors leads to obstacles when different organizations collaborate to configure products. For instance, the relationships among product lines and their variability must be defined, configuration front-ends must be integrated, collaborative and distributed configuration must be supported, and different configuration scenarios must be taken into account. We analyzed these issues further to understand the key challenges and defined requirements related to product lines across organizational boundaries. We discuss these challenges and requirements based on a motivating example in Section 4.

2.2. Implementation

We implemented a solution enabling the joint use of different variability modeling languages by providing a usable front-end

[2] Throughout this paper, we use the term "variability model" to refer to product line models regardless of the specific approach and notation used, e.g., feature models, decision models, OVM models.
[3] android.google.com.