Computer Languages, Systems & Structures ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Computer Languages, Systems & Structures

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

Leveraging variability modeling to address metamodel revisions in Model-based Software Product Lines ☆

Jaime Font [a,b,*], Lorena Arcega [a,b], Øystein Haugen [c], Carlos Cetina [a]

[a] Universidad San Jorge, SVIT Research Group, Spain
[b] University of Oslo, Department of Informatics, Norway
[c] Østfold University College, Department of Information Technology, Norway

ARTICLE INFO

Article history:
Received 1 January 2016
Accepted 8 August 2016

Keywords:
Model-based Software Product Lines
Variability Modeling
Model and metamodel co-evolution

ABSTRACT

Metamodels evolve over time, which can break the conformance between the models and the metamodel. Model migration strategies aim to co-evolve models and metamodels together, but their application is currently not fully automatizable and is thus cumbersome and error prone. We introduce the Variable MetaModel (VMM) strategy to address the evolution of the reusable model assets of a model-based Software Product Line. The VMM strategy applies variability modeling ideas to express the evolution of the metamodel in terms of commonalities and variabilities. When the metamodel evolves, changes are automatically formalized into the VMM and models that conform to previous versions of the metamodel continue to conform to the VMM, thus eliminating the need for migration. We have applied both the traditional migration strategy and the VMM strategy to a retrospective case study that includes 13 years of evolution of our industrial partner, an induction hobs manufacturer. The comparison between the two strategies shows better results for the VMM strategy in terms of model indirection, automation, and trust leak.

1. Introduction

Model-Driven Development aims to shift the focus of software development from coding to modeling. Metamodels are used to formalize a set of concepts and the relationships among those concepts. A model conforms to a metamodel if it is expressed by the terms that are encoded in the metamodel.

Model-based Software Product Lines enable a planned reuse of software components in products that are within the same scope [1]. Commonalities and variabilities among the products are formalized into a set of models (and metamodels) using a variability language: either feature models [2,3] (the de facto standard for variability modeling) or Common Variability Language (CVL) [4], (recommended for adoption as a standard by the Architectural Board of the Object Management Group). Although the details are different, all share the idea of modeling commonalities and variabilities among the different products.

☆ This work has been partially supported by the Ministry of Economy and Competitiveness (MINECO) through the Spanish National R+D+i Plan and ERDF funds under the project Model-Driven Variability Extraction for Software Product Line Adoption (TIN2015-64397-R).
* Corresponding author at: Universidad San Jorge, SVIT Research Group, Autovía A-23 Zaragoza-Huesca Km. 299. 50.830 Villanueva de Gállego, Zaragoza, Spain.
E-mail addresses: jfont@usj.es (J. Font), larcega@usj.es (L. Arcega), oystein.haugen@hiof.no (Ø. Haugen), ccetina@usj.es (C. Cetina).

http://dx.doi.org/10.1016/j.cl.2016.08.003
1477-8424/

Similar to other software components, metamodels evolve over time [5]; however, changes that are introduced in the new metamodel revision can invalidate the models that conform to the previous revision of the metamodel. To address this issue, migration strategies [6–10] propose co-evolving models and metamodels together in order to maintain consistency.

However, even though migration strategies have proven to be successful in model-based approaches, their application is not fully automatizable and can be cumbersome and error prone in large systems. Evolution is particularly critical for a successful adoption of model-based Software Product Lines (SPLs) [11].

We believe that the ideas of variability modeling can also be applied at the metamodel level to address the evolution of SPLs and at the same time avoid the issues involved with migration strategies. Our contribution is the Variable MetaModel (VMM) strategy, which enables the evolution of the metamodel without breaking model conformance. In VMM, each metamodel evolution is expressed in terms of metamodel commonalities and variabilities. As a result, already existing models continue to conform to the created VMM, thus eliminating the need for migration and its related issues.

First, we build a retrospective case study of the evolution undergone by our industrial partner (BSH) over the last 13 years regarding the evolution of their models and metamodels. BSH is the leading manufacturer of home appliances in Europe and its induction department produces induction hobs (explained in Section 2) following an MDD approach [12].

We then apply a migration strategy to the case study, manually migrating the models (as described in Section 4) whenever a metamodel change that breaks the conformance between models and metamodels arises. Migration strategies involve the following three issues: (1) model migration introduces indirection to the models; (2) some of the steps of the migration strategy need human assistance; and (3) the trust gained by models (over years of use) is lost when they are migrated.

Finally, we also apply the VMM strategy to the retrospective case study and compare both strategies (VMM and migration). The comparison shows that the VMM strategy achieves better results than migration in terms of the three issues related to migration: (1) VMM eliminates the need for migration (and the indirections introduced); (2) some of the steps of the migration strategy require human assistance while in the VMM strategy those steps are automatic; (3) the trust gained by models remains the same in the VMM strategy (since the model does not need to change).

This paper is an extended and revised version of our paper published at GPCE 2015 [13]. Apart from revisions throughout the article, in this version we have improved the motivation of the approach and included details about the core operations of the VMM approach (InitVMM and addGen). We have also added some lessons learned from the application of the approach to our industrial partner, information which may be valuable for practitioners that want to apply the ideas of VMM to manage metamodel revisions.

2. Background

This section presents the Domain Specific Language (DSL) used by our industrial partner to formalize their products, the IHDSL. It will be used throughout the rest of the paper to present a running example. Then, the Common Variability

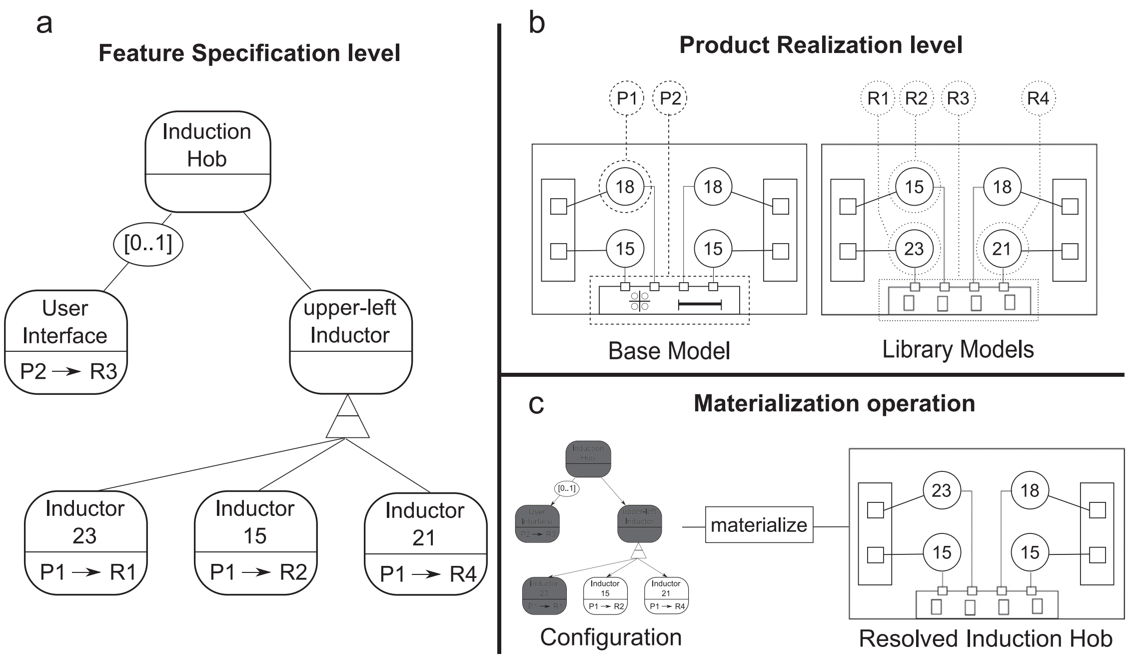

Fig. 1. CVL applied to IH-DSL.

Please cite this article as: Font J, et al. Leveraging variability modeling to address metamodel revisions in Model-based Software Product Lines. Computer Languages, Systems & Structures (2016), http://dx.doi.org/10.1016/j.cl.2016.08.003