Product Line Tool-Chain: Variability in Critical Systems

C. L. Saratxaga* and C. Alonso-Montes*, O. Haugen†, C. Ekelin‡ and A. Mitschke§

*Soft. Syst. Eng. Unit, Tecnalia, Bizkaia, Spain  {cristina.lopez,carmen.alonso}@tecnalia.com
† SINTEF, Pb. 124 Blindern, 0314 Oslo, Norway oystein.haugen@sintef.no
‡ Dept. Electronics and Soft. Volvo Tech. Corp. Sweden cecilia.ekelin@volvo.com
§ EADS Deutschland Gmbh, IW andreas.mitschke@eads.net

Abstract—Competitiveness has thrown industries towards adding more features to existent products increasing their inherent complexity. Indeed, the main challenge is to define mechanisms and tools to control the huge amount of information from requirements towards the final system design. SPL provide mechanisms to control the evolution and design of product families, based on an exhaustive variant analysis. However, critical system industry does not adopt them due to the lack of tool support for the complete life-cycle. In this paper, a product line tool chain is presented based on the analysis of current SPL tools and approaches in order to fit the specific needs within industry partners in the CESAR project. The main goal is to show the benefits of a combination of SPL tools in an industrial scenario which fits their particular needs.

Keywords—SPL, CVL, PLUM, product line tool chain, critical systems, variability management, CESAR

I. INTRODUCTION

Software Product Lines (SPL) have been widely used in the industry to systematically exploit commonalities and variability in a product family to reduce costs and time-to-market. The benefits of using SPL have been extensively studied in the literature [1], [2], including, among others, benefits related with the product quality (e.g. less errors, standardisation, maintenance) or with the business benefits (e.g. reduction of time-to-market, user satisfaction). However, the adoption and implementation of SPL implies an initial high up-front investment that prevents most companies from applying it. Another important drawback, is the complexity of the procedures and the high amount of generated documentation once the SPL is established. Currently, the management of the evolution in an already established SPL is still a research challenge.

In the literature, different SPL approaches can be found in order to represent and manage the variability, like feature modelling [3], decision modelling [4] and the common variability language [5], among others.

Feature Modelling (FM) represents the domain through feature classification within different categories (e.g. mandatory, optional, alternative) using a graph-based model. This approach is useful for high abstraction engineering levels, since it provides a global perspective, serving as a basis for both end-customer and product configuration. However, its main drawback is the increasing complexity in the management of the complete model.

Decision Modelling (DM) focuses only in the variable part of the product. The general performance and efficiency in the variability management is improved. However, this solution hides the global vision of the system. This problem is especially important in those cases where decisions cannot be isolated from the general system understanding. The PLUM tool [6], [7] is a DM based toolbox that follows a MDSD (Model Driven Software Development) approach. The process of solving the variability is generic and domain independent, guided by logical questions/decisions. PLUM is mainly used to manage the variability at a high level of abstraction (business logic decisions, user preferences).

The Common Variability Language (CVL) [8] has been defined to work tightly with Domain Specific Languages (DSLs) of the companies, and it focuses on the specification of variability in a model separate from the base model. The variability management provided by CVL can handle a high variety of possible variable situations based on the concept of substitution. A substitution involves the notions of placement and replacement in the base model. Thus, CVL not only defines the feature model, but also the realisation of the resolutions into the base model. CVL assumes that the base model and the resolved product belong to the same domain. So, this restricts the relationships at different levels of abstraction. CVL tool exploits CVL capabilities and it has been already tested for a specific train language (TCL) [9]. Standardisation activities are being under development with the OMG in order to make CVL as a standard for variability management [10], [11].

Critical systems are intrinsically complex, due to both safety restrictions and the impact of business and market-based decisions into the final system design. The definition of an unique and general SPL solution becomes a real challenge due to the high complexity of the design together with the impact among variants from different engineering phases, regardless the added difficulty of safety constrains. Currently, there is a tendency in industry to increase system functionality, providing more features and functions to customers, in order to improve the competitiveness of the final products. This fact leads towards a high increment