Towards Enabling Reuse in the Context of Safety-critical Product Lines

Barbara Gallina
School of Innovation, Design and Engineering
Mälardalen University
Västerås, Sweden
barbara.gallina@mdh.se

Abstract—In the context of safety-critical product lines, industries have to face an urgent challenge: reduction of time and cost for the creation of a safety case. A safety case is a contextualized structured argument constituted of process and product-based sub-arguments to show that a system is acceptably safe. Its creation is an extremely time-consuming and costly activity. To reduce time and cost, reuse capabilities are being investigated from different perspectives however currently no satisfying approach is available. In this paper, we propose a new methodological framework called Anti-Sisyphus. Anti-Sisyphus is aimed at enabling reuse by combining process lines, product lines and safety case lines. The systematization of what varies and what remains in common with respect to process as well as product elements in turns enables the systematization of what varies and remains in common in terms of process-based as well as product-based arguments within a safety case line. Anti-Sisyphus thus has the potential to enable a 3D reuse.

Index Terms—Reuse; Safety Cases; Product Lines; Process Lines; Safety Case Semi-automatic Generation.

I. Introduction

In the context of safety-critical product lines, the term reuse is often considered a taboo. The failure that in 1996 caused the explosion of Ariane 5 [1] is usually used as an argument against reuse. As known Ariane 5 contained a piece of code that was appropriate for its predecessor Ariane 4 but that resulted to be catastrophic for Ariane 5. As a consequence of the Ariane 5 accident, the "to-reinvent or not-to-reinvent" dilemma is usually solved by a tacit acceptance that starting from scratch is safer.

Recently, however, European as well as national research projects [2-4] have being investigating reuse possibilities from different perspectives (e.g., component and contract-based systems development focusing on out-of-context and in-context contracts-based components development). Standardization frameworks have also started embracing reuse. In the avionics domain, the notion of Reusable Software Components (RSC) has been introduced [26]. Similarly, in the automotive domain (ISO 26262 [12]), the notion of Safety Element out of Context (SEooC) has been introduced. These projects/standardization frameworks probably originate from the belief that the lesson that should be learnt by the Ariane 5's accident is a different one: reuse should be possible and appropriate approaches should be investigated.

We also believe that not reinventing the wheel should be possible and that methodological approaches aimed at guaranteeing safer reuse should be available. Thus, to contribute to the provision of these approaches, we propose Anti-Sisyphus1[1] a methodological framework aimed at avoiding unnecessary repetitive actions while building safety cases for safety-critical product lines.

The key-idea of Anti-Sisyphus stems from the following observations:

- In certain domains (e.g. automotive, rail, etc.), safety-critical systems can be treated as families of products, better known as product lines. Thus product lines-oriented engineering practices [14] could be adopted and adapted for safety-critical systems.
- In certain domains, safety-critical systems must be engineered in compliance with specific prescriptive safety processes. For example, DO178C [11] defines guidelines/activities to be adopted/performed in the context of the avionics domain; while ISO 26262 defines the safety processes to be adopted in the automotive domain. By reading the (de facto) standards, it is possible to identify similarities in terms of activities, artifacts to be produced, etc. Thus, families of processes can be identified.
- In certain domains (e.g. automotive), for certification purposes, a safety case might be required. A safety case is a contextualized structured argument consisting of process and product-based sub-arguments to show that a system is acceptably safe. In the context of product lines, safety cases exhibit common pieces of argumentation.

To avoid unnecessary and repetitive actions, product lines-oriented practices could be extended and integrated to cover process as well as safety case aspects. This extension and integration would allow safety-critical product line manufacturers to face: 1) the emergent economic challenge concerning time and cost reduction for the creation of safety cases for certification as well as re-certification purposes; 2) the societal challenge concerning safe products. By enabling reuse, more time will be devoted to more crucial actions, i.e., safer

[1] *Anti-Sisyphus* stems from the existential and econimical need of avoiding unnecessary and repetitive actions. Sisyphus [20] is a character of the Greek miths, who was punished by being compelled to roll an immense boulder up a hill, only to watch it roll back down, and to repeat this action forever.

and traceable integration of reusable and certifiable commonalities and variabilities.

The rest of the paper is organized as follows. In Section II, we provide essential background information, followed by Section III, where the main idea of Anti-Sisyphus is presented. Finally, in Section IV, we present some concluding remarks and future work.

II. Background

In this section we present the background information on which we base our work. In particular, we provide essential information concerning safety-critical product lines (in Section II.A), safety-oriented process lines (in Section II.B), and safety case lines (in Section II.C).

A. Safety-critical Product Lines

A (*Software) Product Line* is a set of (software-intensive) systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way [15]. The set of core assets is constituted of commonalities and variabilities. A *commonality* is the ability of an asset to be maintained as a constant; while a *variability* is the ability of an asset to be changed (customized) for use in a particular context. Variabilities are defined through variation points and variants. A variation point is the place within an artefact where a decision can be made. Variants represent the alternatives/optionalities associated to a variation point. Desired single products can be derived from the product line by selecting and composing appropriate reusable product elements. Thus, time can be spared by eliminating unnecessary and repetitive actions. These actions consist of re-inventing the wheel by re-modeling/re-engineering (fully or partially) reusable product elements. Certainly, as discussed by Lutz et al. [23], safe reuse is not straightforward and previously learned lessons should be assimilated. However, as recently discussed by Schulze et al., product line engineering in the context of functional safety is feasible.

In safety-critical systems engineering, a product line might be introduced at several levels: 1) to model intra-domain products at a micro-level (e.g. fuel level estimation and display systems within heavy road vehicles in the automotive domain [8]), 2) to model intra-domain products at a macro-level (e.g. sets of cars, sets of trucks, etc in the automotive domain), and 3) to model inter-domain products (e.g. runtime and middleware components). Typical runtime components that may cross domains are: operating systems, communication stack software (e.g. TCP/IP, CAN, etc), device drivers for sensors, device drivers for actuators, math's packages for control systems, etc. Databases constitute an example of middleware components that might be used across domains).

Product Lines are commonly described by using feature models. A Feature Diagram is a graphical representation of a feature model, which is a hierarchically arranged set of features (properties of a system that are relevant to some stakeholder and are used to capture commonalities or discriminate among systems in a family). The semantics of Feature Diagrams has been extensively discussed by Schobbens et al. [18]. Feature Diagrams however do not represent the only description means. Depending on the development stage other modeling capabilities supporting variability modeling can and should be used. In the context of safety-critical systems, it is of utmost importance that commonalities and variabilities take into consideration assets related to safety. It is crucial, for instance, to know that by making an asset vary, the failure behavior changes and thus the system-level safety integrity level changes. For these reasons, product line-oriented safety analysis techniques should be used.

B. Safety-oriented Process Lines

As recalled by Gallina et al. [6] a *development process* identifies a structure that is imposed on the development of a system. More precisely, a *process* can be defined as a set of partially ordered tasks that have to be executed to develop systems. To tasks, additional process elements can be associated: work-products, roles, guidelines, templates, tools, etc. Tasks can be grouped to form an activity and activities in turn can be grouped to form a phase. A *process line* [5] is a family of highly related processes that are built from a set of core process assets in a pre-established fashion. As defined by Gallina et al. [6], a *safety-oriented process line* is a process line focusing on processes for engineering safety-critical systems/product lines. In the framework of safety critical systems engineering, a process line might be introduced at several levels: to model intra-domain processes [24] that slightly vary due to different safety integrity levels (e.g., ASIL A, ASIL B, etc in the automotive domain); to model cross-domain processes [25] that vary due to different safety integrity level types (e.g., ASIL, DAL-Design Assurance Level, in the automotive and avionics domains). Desired single processes can be derived from the process line by selecting and composing appropriate reusable process elements. Thus, time can be spared by eliminating unnecessary and repetitive actions, which consist of modeling (fully or partially) reusable process elements.

To document a process line, languages conceived for documenting single processes have being investigated ([6]and [16]). From this investigation results that SPEM 2.0 [13] is promising due to its support for variability modeling but it is not intuitive and expressive enough. vSPEM [16], which represents a SPEM2.0 extension, has been recently proposed and seems to be more promising and easy to grasp than SPEM2.0. vSPEM, however, is currently not supported by any tool. In [28-29] a methodological approach based on SPEM 2.0 to engineer safety-oriented process lines is proposed. To document safety-oriented processes, S-TunExSPEM [7] has been recently proposed.

C. Safety Case Lines

As recalled by Gallina et al. [8], a safety case is a contextualized structured argument containing process and product-based arguments to link evidence with claims. Process-based arguments show that the system has been developed in compliance with the processes mandated by the standards. Product-based arguments show that the product satisfies the

safety requirements derived during the hazards analysis. The purpose of a safety case is to show that a system is acceptably safe. In the context of safety-critical product lines, safety cases exhibit commonalities as well as variabilities, thus they represent a family of safety cases, i.e., a safety case line. Variabilities are classified into intrinsic (a variation due to style variation used for arguing) and extrinsic (a variation due to a product/process variation) [8]. In this paper we focus on the extrinsic ones. Desired single safety case can be derived from the safety case line by selecting and composing appropriate reusable safety case fragments. Thus, time can be spared by eliminating unnecessary repetitive actions consisting of modeling (fully or partially) reusable safety case fragments.

Goal Structuring Notation (GSN) [10] represents a promising and broadly accepted means to document safety cases [28]. An extension to GSN has also been proposed to document safety cases that address product lines [9].

III. Anti-Sisyphus Overview

In this section, we provide an overview of Anti-Sisyphus, our methodological framework proposal to enable reuse in the context of safety-critical product lines while building safety cases. To better convey the main message, we systematically refer to Fig. 1. This figure illustrates a 3D space populated by families of items consisting of Safety Argumentation (safety case), process and product-related specification of commonalities and variabilities. We call these families SAPP lines, which stands for Safety Argumentation, Process and Product-related lines.

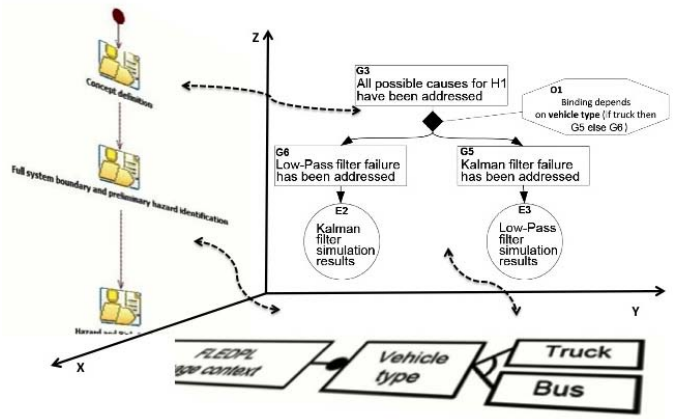

Fig. 1 Anti-Sisyphus key-idea

Quadrant Z,Y illustrates a product-based fragment of a safety case line (taken from [8]) related to an automotive (Scania) safety-critical product line. This fragment of the safety case line exhibits a variation point denoted by the diamond. The octagon associated to the diamond states a condition that constrains the selection based on the usage context. To model systematically the commonalities and the variabilities that exist between safety cases of a safety case line, we use the appropriate GSN extension [9], as done in Fig. 1.

Quadrant X,Y illustrates a fragment of the usage context (taken from [8]) of the product line consisting of Fuel level estimation and display systems. The usage context also exhibits a variation point. To model systematically the commonalities and the variabilities that exist between products of a safety-critical product line, we use different modelling languages depending on the process activity, e.g., feature diagrams (as done in Fig. 1, quadrant X, Y) for requirements modelling as explained by Gallina et al. [8].

Finally, quadrant Z,X illustrates a fragment of a safety-oriented process line related to ISO 26262 (adapted from [6]). More specifically this fragment, given at a high level, indicates that Concept/item definition, Full system boundary and preliminary hazards identification, Hazard and risk assessment are common activities in all ISO 26262-compliant processes. To model systematically the commonalities and the variabilities that exist between processes of a safety-oriented process line, we use SPEM2.0 (quadrant X, Z, shows an Eclipse Process Framework (EPF) [19] model).

Once intra-quadrant commonalities and variabilities are modeled systematically it is time to model systematically inter-quadrant dependencies. For instance, the variation point within the fragment of the safety case line depends on the variation point within the usage context. In accordance with the GSN extension for product lines [9], an octagon is used to denote this extrinsic dependency.

In Fig. 1, dotted lines are used to indicate that quadrants are related. Process elements are indeed used within the safety case line to claim compliance with the process mandated by the standard. Thus process-based fragments (even though not indicated for space reasons) populate quadrant Z,Y. The artifacts that are produced during the execution of a process constitute the product line artifacts. Some of these product line artifacts are then used to claim that the product line behaves acceptably safe.

Thus, Anti-Sisyphus' key-idea consists in exploring the specification of SAPP lines (resulting from the combination of process lines, product lines and safety case lines) in order to benefit from its advantages. At a first glance this initial laborious specification work involving the systematization of commonalities and variabilities may seem to be discouraging and too expensive. In the long run, however, by assuming that that all the dependencies, especially the safety-critical ones, can be traced and managed, it offers appealing advantages. Single safety cases (related to single products developed according specific single processes), for instance, can be derived in a rather straightforward manner, by applying MDSafeCer [17]. After having configured a product and its corresponding process, the corresponding safety case can be obtained by pruning the safety case line in accordance with the choices performed during the product as well as process configuration. In principle, these single safety cases could be semi-automatically generated

IV. Conclusion and future work

In this paper, we have presented a new methodological framework called Anti-Sisyphus. This new approach is aimed at solving the "to-invent or not-to-reinvent" dilemma, which animates daily discussions between safety-critical systems manufacturers. Anti-Sisyphus proposes a combined 3D reuse-

based approach to avoid useless repetitions. Anti-Sisyphus consists of the combination of process lines, product lines and safety case lines. The rational behind this combination is that safety cases are constituted of product as well as process-based arguments, and thus to enable reuse while building safety cases it is necessary to systematize commonalities and variabilities with respect to product as well as process elements.

To make Anti-Sisyphus concrete, a long-term research work is required. In the short-term future, with respect to Fig. 1, we initially plan to focus on quadrant X, Z. We intend to devote our attention to specification means for safety process lines. Our intention is to combine S-TunExSPEM [7], vSPEM [16] and our methodological proposal [27] to specify safety-oriented process lines. The idea is to build on top of the experience gained while using our methodological approach to engineer the automotive safety-oriented process line [27] as well as the cross-domain safety-oriented process line [29].

In the mid-term future, we plan to broaden our focus and consider also the quadrant Z,Y. The idea is to build on top of MDSafeCer [17] and THRUST [27] and provide support for semi-automatic generation of pieces of safety case lines from process lines. We also plan to consider quadrant X,Y, and based on what presented by Sljivo et al. [21], generate safety case fragments from product-based information, more specifically contract-based architectural models.

In the long-term future, we aim at considering all three quadrants and their interdependencies (cross-cutting constraints among process/product/argumentation elements), and thus achieve tool-supported capabilities for the semi-automatic generation of safety case lines from process and product lines.

Acknowledgment

This work has been partially supported by the European Project ARTEMIS SafeCer [2] and by the Swedish SSF SYNOPSIS project [3].

References

[1] Design by Contract: The Lessons of Ariane: http://archive.eiffel.com/doc/manuals/technology/contract/ariane

[2] ARTEMIS-JU- 269265 SafeCer - Safety certification of software-intensive systems with reusable components.

[3] SYNOPSIS- SSF- RIT10-0070. Safety analysis for predictable software intensive systems. Swedish Foundation for Strategic Research.

[4] [4] FP7- 289011 OPENCOSS - Open platform for evolutionary certification of safety-critical systems.

[5] T. Ternite, ”Process lines: a product line approach designed for process model development”. Software Engineering and Advanced Applications, Euromicro Conference, pp. 173-180, 2009.

[6] B. Gallina, I. Sljivo, and O. Jaradat, ”Towards a safety-oriented process line for enabling reuse in safety critical systems development and certification”. 35th IEEE Software Engineering Workshop (SEW), Heraclion, Crete (Greece), 2012.

[7] B. Gallina, K. R. Pitchai, and K. Lundqvist, ”S-TunExSPEM: towards an extension of SPEM 2.0 to model and exchange tuneable safety-oriented processes”. 11th International Conference on Software Engineering Research, Management and Applications (SERA), SCI 496, Springer, Prague, Czech Republic, August 7-9, 2013.

[8] B. Gallina, A. Gallucci, K. Lundqvist, and M. Nyberg, ”VROOM & cC: a method to build safety cases for ISO 26262-compliant product lines”. Workshop on Next Generation of System Assurance Approaches for Safety-Critical Systems (SASSUR), CNRS report (HAL/Arxiv), Toulouse, France, 24 Sept. 2013.

[9] I. Habli and T. Kelly, ”A safety case approach to assuring configurable architectures of safety-critical product lines”. International Symposium on Architecting Critical Systems (ISARCS), Prague, Czech Republic, Springer, pp. 142–160, Jun. 2010.

[10] GSN: GSN Community Standard Version 1 (Nov. 2011)

[11] RTCA DO-178C (EUROCAE ED-12C), ”Software consider-ations in airborne systems and equipment certification”. RTCA Inc., Washington, DC, Nov. 2011.

[12] ISO26262. Road vehicles – Functional safety. International Standard, Nov. 2011.

[13] Software & systems process engineering meta-model (SPEM), v 2.0. Full Specification formal/08-04-01, Object Management Group, 2008.

[14] K. Pohl, G. Böckle, and F. J. van der Linden, ”software product line engineering: foundations, principles and techniques”. ISBN: 3540243720, Springer-Verlag, I edition, 2005.

[15] P. Clements and L. Northrop, ”Software product lines: practices and patterns”. Addison Wesley, Reading, MA, USA (2001)

[16] T. Martínez-Ruiz, F. García, M. Piattini, and J. Münch, ”Modeling software process variability: an empirical study”. IET Software, vol. 5, no. 2, pp. 172-187, 2011.

[17] B. Gallina, ”A model-driven safety certification method for process compliance”. 2nd IEEE International Workshop on Assurance Cases for Software-intensive Systems (ASSURE),Italy, pp. 204-209, 2014.

[18] P.-Y. Schobbens, P. Heymans, J-C. Trigaux, and Y. Bontemps, ”Generic semantics of feature diagrams”. Comput. Netw. 51, 2, pp. 456-479, Feb. 2007.

[19] Eclipse Process Framework http://www.eclipse.org/epf/

[20] http://en.wikipedia.org/wiki/Sisyphus

[21] I. Sljivo, B. Gallina, J. Carlson, and Hansson, ”Generation of safety case argument-fragments from safety contracts”. 33rd International Conference on Computer Safety, Reliability, and Security (SafeComp). Springer, 2014.

[22] J. Bosch, ”From software product lines to software ecosystems”. 13th International Software Product Line Conference (SPLC). Carnegie Mellon University, Pittsburgh, PA, USA, pp. 111-119, 2009.

[23] R. R. Lutz, G. G. Helmer, M. M. Moseman, D. E. Statezni, and S. R. Tockey, ”Safety analysis of requirements for a product family”. 3rd International Conference on Requirements Engineering: Putting Requirements Engineering to Practice (ICRE). IEEE Computer Society, Washington, DC, USA, 1998.

[24] B. Gallina, S. Kashiyarandi, H. Martin, and R. Bramberger, ”Modeling a safety- and automotive-oriented process line to enable reuse and flexible process derivation”. 8th International Workshop on Quality-Oriented Reuse of Software (QUORS), IEEE Computer Society, Västerås, Sweden, 2014.

[25] B. Gallina, S. Kashiyarandi, K. Zugsbrati, and A. Geven, ”Enabling cross-domain reuse of tool qualification certification artefacts”. 1st International Workshop on DEvelopment, Verification and VAlidation of cRiTical Systems (DEVVARTS), Springer, Florence, Italy, 2014.

[26] AC 20-148. ”Reusable software components”. Federal Aviation Administration, Dec. 2004.

[27] B. Gallina, K. Lundqvist, and K. Forsberg, ”THRUST: a method for speeding up the creation of process-related deliverables”. IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC), Colorado Springs, CO, USA, Oct. 5-9, 2014.

[28] R. Berthold, E. Denney, M. Fladeland, G. Pai, B. Storms and M. Sumich, ”Assuring Ground-based Detect and Avoid for UAS Operations”. IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC), Colorado Springs, CO, USA, Oct. 5-9, 2014.

[29] M. Schulze, J. Mauersberger, and D. Beuche, ”Functional safety and variability: can it be brought together?”. 17th International Software Product Line Conference (SPLC). ACM, New York, NY, USA, 236-243, 2013.