

Towards a Safety-oriented Process Line for Enabling Reuse in Safety Critical Systems Development and Certification

Barbara Gallina, Irfan Sljivo and Omar Jaradat
 MRTC, School of Innovation, Design and Engineering
 Mälardalen University,
 Västerås, Sweden
 {barbara.gallina, irfan.sljivo, omar.jaradat}@mdh.se

Abstract— Safety standards define development processes by indicating the set of partially ordered tasks that have to be executed to achieve acceptably safe systems. Process compliance constitutes a fundamental ingredient in safety argumentation for certification purposes. Certification is a very expensive, time-consuming and quality demanding activity. To increase quality and reduce time and cost, reuse-based approaches are being investigated. In this paper, we adopt process line approach in the framework of safety processes. This means that we treat a family of processes as a product line, and we identify commonalities and variabilities between them. The resulting information guides developers in reusing parts of the process, the system and safety case, e.g. which parts to make more generic, isolating changes in others to avoid ripple effects etc..

Keywords- *Safety processes, Safety standards, Safety certification reuse, Process lines, qualitative standards comparison*

I. INTRODUCTION

In some application domains, safety critical systems must be certified. To achieve the “certified” stamp, a system has to be shown to be acceptably safe. The proof is provided through a structured argument that links evidence with claims. This structured argument is known as safety case. A safety case is expected to include two types of arguments: process and product-based arguments. The first type is devoted to show that a product has been developed in compliance with the process defined in the domain-specific standard. The second type is devoted to show that the product satisfies the safety requirements derived during the hazards analysis. Producing a safety case is a very expensive, time consuming and safety-critical activity. To reduce time and cost as well as to increase quality, it would be beneficial to have at disposal systematic reuse-based approaches allowing for avoidance of wheel reinvention in terms of system development and certification efforts. Similarly, to increase the confidence of safety cases, it would be beneficial to have at disposal a means of evaluating and comparing safety standards to show, for instance, that one type of regulation is better than another one [6]. Clearly, this comparison for quality evaluation purposes would make sense in the case of multiple standards for a single application domain as well as in the case of different versions of a single standard related to a single application domain.

Safety standards define the processes that must be followed to develop safety critical systems. Thus compliance with standards is used to formulate process-based arguments in safety cases. Different standards exist. Their diversity is aimed at providing more appropriate norms with respect to the application domain that they target. In some cases, however, a system can be integrated as a sub-system in different domains (namely railway and automotive) and thus, to be able to reuse, it is important to understand if what is mandatory in the railway domain is also mandatory in the automotive domain. Comparison is not only required between standards. Besides inter-standards comparison, intra-standard comparison is fundamental whenever a new product is introduced with slightly different safety requirements as well as whenever a new version of the standard is introduced. To perform these comparisons, since no efficient approach is currently available, a careful reading of the standards is required each time a cross-domain or intra-domain comparison is needed. This continuous re-inventing of the wheel is time consuming, expensive and error-prone since it provides room for different interpretations. Systematic approaches enabling reuse and comparison are needed.

In the framework of the ARTEMIS EU SafeCer project [1], reuse-based approaches are being investigated. In particular, within this project, we are interested in exploring the possibility to adopt and integrate a product line approach on three dimensions: development processes, systems development, safety cases. This 3-dimension integrated approach, SCPPL (Safety-centred Case, Product, Process Line), would represent the extension of the current 2-dimension integration called SPPL (Software Product and Process Line), which was presented in [16]. SCPPL is justified by observing that the product goals (reified through functionalities) are achieved as a function of executing some process under certain project characteristics. The product goals (including safety goals) are certified as a function of examining some safety case.

In this paper, we focus on the exploration of a process line approach in the context of safety. By analysing rather deeply two standards, namely ISO26262 [4] and EN50126 [5], we observe several similarities. Since they both stem from a third standard, namely IEC61508 [3], the presence of several similarities is not surprising, it is instead a clear witness of their common root. Since these standards are part of a family, to enable reuse and comparison, we propose to treat the processes that they define as a process line. For sake

of precision, it should be said that the notion of ecosystem [15] of processes could also be envisaged since the family of processes expands outside the boundary of a single organization/domain.

Besides motivating the benefits of a process line approach, in this paper, we focus on a specific phase and we identify a phase line, by pointing out its reuse and comparison possibilities. Then we model it by using the current best practices in the process line community and we discuss advantages and disadvantages of these practices with respect to our purposes.

The rest of the paper is organized as follows. In Section II, we provide essential background information. In Section III we present the proposed reuse-based approach, by focusing on a specific phase. In Section IV, we discuss advantages and disadvantages with respect to the modelling languages as well as current limitations of our work. In Section V we discuss related work. Finally, in Section VI 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, in Section II.A we provide essential information concerning two safety standards. In Section II.B, we give an idea of the burden that characterizes a certification process. In Section II.C, we briefly recall the definition, benefits and development process of a process line. Finally, in Section II.D we recall the modelling approaches that are currently at disposal and we focus on domain-specific approaches.

A. Safety standards

In this subsection, we focus on two safety standards, namely ISO26262 and EN50126. The rationale behind the selection of these two standards is that they are siblings, they both stem from the same parent standard and thus reuse in the case of cross-domain certification should be a concrete possibility. The entire set of standards stemming from IEC61508 (e.g. IEC61511 which is defined as being process industry specific) should be considered but for space reasons and illustration purposes we limit the focus to the couple constituted of ISO26262 and EN50126. For each standard, we provide a brief overview and then we focus on a specific process phase: the one that involves the item/concept definition and hazard analysis, named Concept phase including hazard analysis in Section III.B. For this phase, we recall the normative and non-normative parts that are necessary to understand what is presented in Section III.

ISO26262 regulates the automotive domain and more specifically it is intended to be applied to safety-related systems that include one or more electrical and/or electronic systems and that are installed in series production passenger cars with a maximum gross vehicle mass up to 3 500 kg. In ISO26262, the Concept phase consists of the following four chained normative clauses:

- **Item Definition:** the main objective of this clause is to de-fine and describe the item, its dependencies on and interaction with the environment and other items. The expected deliverable is Item definition.

- **Initiation of the safety lifecycle:** the objectives of this clause are: to distinguish between a new item development and a modification to an existing item, and to define the safety lifecycle activities in case of modification. The expected deliverables are: Impact analysis and Safety plan.
- **Hazard analysis and risk assessment:** the objective of this clause is to identify and to categorise the hazards that mal-functions in the item can trigger and to formulate the safety goals related to the preventions of mitigation of the hazardous events, in order to avoid unreasonable risk. The expected deliverables are: Hazard analysis and risk assessment report, Safety goals, Verification review.
- **Functional safety concept:** the objective of this clause is to derive the functional safety requirements from the safety goals, and to allocate them to the preliminary architectural elements of the item, or to external measures. The expected deliverables are: Functional safety concept and Verification report of the Functional safety concept.

Each of these clauses consists of a number of tasks that need to be performed in a specific order.

In this subsection, we focus on the *Hazard analysis and risk assessment* clause and we detail its tasks, deliverables (work-products), and guidelines. The Hazard analysis and risk assessment clause includes:

- **Tasks (normative):**
 1. Initiation of Hazard analysis and Risk assessment,
 2. Situation analysis,
 3. Hazard identification,
 4. Hazard classification,
 5. Automotive Safety Integrity Levels determination,
 6. Determination of Safety goals, and
 7. Verification of Hazard analysis, risk assessment, and safety goals.
- **Deliverables (normative):**
 - Hazard analysis and risk assessment report (output of tasks 1-5),
 - Safety goals (output of task 6)
 - Verification of hazard analysis and risk assessment and safety goals (output of task 7).

For some of these tasks the standard provides guidelines (which can be normative or non-normative) on how they can and should be executed. In the case of Hazard identification, several techniques for systematic hazard identification have been suggested e.g. brainstorming, checklists, quality history, FMEA (Failure Mode and Effects Analysis) and field studies.

EN50126 regulates the railway domain, and more specifically it is intended to be applied in the area of signalling and control. EN50126 is a European standard provided by the European Committee for Electrotechnical standardization (CENELEC). It defines a process to manage Reliability, Availability, Maintainability and Safety (RAMS) in the railway sector. This process is constituted of fourteen

normative phases. In this paper, we focus on the first three phases, which are:

- **Concept Phase:** the objective of this phase is to develop a sufficient level of understanding of the system. The expected deliverables are: verification report and management structure adequate to implement the RAMS requirements of the subsequent phases in the lifecycle.
- **System Definition and Application Conditions Phase:** the objective of this phase is to define the mission profile, boundary of the system and the scope of system hazard analysis. Moreover, this phase must establish the application conditions, RAMS policy, and safety plan for the system. The expected deliverables are: verification report, RAMS policy for the system and the safety plan.
- **Risk Analysis Phase:** the objective of this phase is multi-fold: 1) empirically or creatively identification of the hazards associated with the system and the events leading to them; 2) estimation of the risk associated with the hazards; 3) development of a process for risk management. The expected deliverables are: Hazard log and verification report.

In this subsection, we focus on the Risk analysis phase and we detail its tasks and deliverables:

- **Tasks (normative):**
 1. Hazard identification,
 2. Hazard classification,
 3. Risk evaluation,
 4. Determination and classification of acceptability of the risk,
 5. Establishing the Hazard Log,
 6. Assessment of all tasks in the phase.
- **Deliverables (normative):**
 - Hazard log (output of task 5),
 - Verification report (output of task 6).

To help achieving the above mentioned objective of *Risk Analysis Phase*, the standard provides guidelines [20] in terms of methods to be used to identify hazards either empirically (e.g. checklists, structured walkthrough, FMEA, and Task Analysis for man-machine interfaces) or creatively, e.g. brainstorming and - HAZard and OPerability Studies (HAZOP).

B. Certification burden

As presented in Subsection II.A, standards provide normative as well as non-normative parts. All these parts written in natural language are often rather abstract and in some cases even unclear. Thus, interpretation, adaptation with respect to the systems to be developed as well as with respect to the organizational needs and refinement are fundamental steps to be performed to achieve a well-defined development process from the standard description. In the case of cross-domain certification, it is also necessary to identify the similarities and the differences that exist between the standards. This requires the provision of terminological mappings. An additional aspect that contributes in making the certification a real burden is the evolving nature of

standards. Each time a new version is published, changes must be identified.

Even though usually standards like EN50126 have a life that is counted in decades (typically 2), and evolution is quite slow, it must be noted that the last few years have seen the emergence of ISO26262 and the issuing of a major revision of EN50126 (as well as, in the avionics domain, the issuing of DO178C [28]). This means that in the coming years industry will face the problem of introducing the new standards, running in parallel projects along the old and new standards, adding therefore interest to reuse-based approaches.

In the literature, efforts aimed at contributing to cross-domain standards comparison [12, 17] are available. Efforts aimed at contributing to intra-standard cross-version comparison [19] are also available. Despite these efforts, since they are mainly provided in natural language, they do not offer an effective and efficient solution. Due to the absence of systematic approaches in the context of the safety community, the actors (stakeholders) involved (systems producers as well as systems certifiers) waste time and money since they have to re-invent the wheel each time. Moreover, the risk of failure is higher.

C. Process lines

A 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, can be associated work-products (which comprise artifacts, deliverables and outcomes), 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 [7] is a family of highly related processes that are built from a set of core process assets in a pre-established fashion.

Comparisons among processes characterize the main activity in a process line development approach, more specifically during the domain engineering phase. Through comparisons, it is possible to retrieve common core assets (commonalities), which represent the family identifiers and variable assets (variabilities), which are assets that present the ability to be changed. Variable assets include: 1) optionality; 2) alternatives (exclusive or, i.d. only one variant can be selected). As stated in [23] for product lines, to characterize variability in more detail, it is useful to answer the following questions: "What varies?" and "How does it vary?". These questions are fundamental also for process lines as surveyed in [26]. The answers to these questions lead to the definitions of variability subject and variability object, which are recalled here:

- a *variability subject* is a variable item of the real world or a variable property of such an item (e.g. hazard analysis technique);
- a *variability object* is a particular instance of a variability subject (e.g. FMEA, HAZOP, etc).

A *variation point* is a representation of a variability subject. A *variant* is a representation of a variability object.

Common core assets imply reusable assets, while variable assets imply process evolution. Once the core assets

are defined, process tailoring can take place. Each process in the process line can be derived by taking advantage of all the assets in the process line. More specifically, a desired process can be derived by performing two steps: 1) selection of all the commonalities plus the desired variants at variation points; 2) composition of what was previously selected.

To achieve a beneficial process line that will pay off the initial cost, a three-step development process [13] should be followed. First of all a sound scoping must be performed. The scoping is necessary to define the boundary that ensures commonalities maximization and minimization of ad-hoc variabilities. After scoping, it is time to identify commonalities and variabilities and then, finally, the process line can be architected by creating a process reference architecture, i.e. a process model that contains all generic process assets and a decision model governing which assets to put together under which circumstances, forming a specific process instance.

D. Process lines modelling

To model a process-line, three different types of approaches have been investigating so far (see [24] and [25]): product line-oriented, domain-specific (processes-specific) and hybrid approaches. To the first type, belong for instance feature diagrams and all its variants. To the second type belong SPEM 2.0 and vSPEM. Finally, to the third type belongs a combination of the first two. An effective modelling approach for process lines is not yet at disposal. In our work, since we believe that they better suit the needs of process engineers, we focus on domain-specific approaches and we recall essential modelling constructs, necessary to understand the models presented in Section III.

SPEM (Software Process Engineering Meta-model) 2.0 [8] is the OMG's standard for software process modelling. SPEM 2.0 offers static as well as dynamic modelling capabilities, the latter achieved by including links to other modelling languages (e.g. UML activity diagrams and state machines). Static modeling elements are at disposal to model the structure of a process, more precisely its reusable content (e.g. phases, activities, tasks, etc.) and how this content is statically related (e.g. work breakdown structure). In addition to the static modeling elements, SPEM2.0 provides the so called *in use* elements, which represent proxies for a work definition in the context of activity diagrams to model the behavior of one specific process or template of a process.





SPEM 2.0 also offers modelling capabilities to address process variability. SPEM 2.0 advocates for a single process model that includes variability [24]. In SPEM 2.0, a process element (e.g. an activity) can be a variability element and to it the process engineer can associate separate objects representing the differences (additions, changes, omissions) relative to the original (called base). The variability element has an attribute that characterizes its variability type. The Variability Type enumeration class defines the different types of variability, briefly recalled in Table 1. The reader may refer to [8] for further details.

Table 1 Variability types in SPEM 2.0

Variability type	Description
na	Not assigned
contributes	Provides a way to contribute to attribute values and association instances of the base, without altering it. The base is logically replaced with an augmented variant.
replaces	Defines a replacement of a base. The replacement consists of either a complete new variant or a change concerning fundamental relationships.
extends	Provides a kind of inheritance. The variant has the same properties as the base, but might override the inherited properties with its own values.
extends-replaces	Combines the effects of <i>extends</i> and <i>replaces</i> , i.d. only replaces the values that have been redefined and leaves all other values of the base unchanged.

In Table 2, we recall some of the SPEM 2.0 graphical modelling elements that can be interrelated to achieve a process (line) model. We only recall those elements that we use in Section III.C.

Table 2 Process modelling elements in SPEM 2.0

Concept Language	Activity	Task	Work-product ¹	Guideline
SPEM2.0				
¹ In the context of this paper, we focus on a specific class of work-products. We focus on deliverables, which are work-products that may be delivered to an internal or external party.				









More specifically, the graphical modelling elements shown in Table 2 are those used to model the process behaviour. Activities and tasks, for instance, represent the actions in the activity diagrams used to model the dynamics.

vSPEM [9] is the extension of SPEM 2.0 offering more effective variability modelling mechanisms. vSPEM focuses on a separate specification of common and variable process elements [24]. Modelling effectiveness is increased in a twofold way: 1) by providing modelling capabilities offering fundamental constructs (such as variation point and variant) in accordance with typical product line practices; 2) by supporting a graphical representation of process variability. In vSPEM, the variants that can be selected at variation points are linked to the variation points by using a special relationship (called *canBeOccupied*) that indicates that the variation point must be occupied by a single variant. The graphical concrete syntax that represents this relationship is an arrow with a filled circle at the rear tip.

In Table 3, we recall some of the vSPEM graphical modelling elements that can be interrelated to achieve a

process (line) model. We only recall those elements that we use in Section III.D.

Table 3 Process modelling elements in vSPEM

Concept Language	Activity	Task	Work-product ¹	Guideline
vSPEM variation point				
vSPEM variant				

III. ENABLING REUSE IN SAFETY CERTIFICATION

In this section we build on the background information provided in Section II. First of all we motivate the benefits of adopting a process line approach in the context of safety standards. Then, we concentrate on a single phase and we identify a phase line and then we model it by using two different modelling languages. Finally, we show the reuse capabilities by deriving the two phases that are part of the phase line.

A. Safety-oriented process line: benefits

From the adoption of a process line approach, significant benefits can be derived: cost and time reduction as well as quality increase. These benefits are made possible thanks to the reuse possibilities that inherently characterize this approach. Process commonalities, indeed, represent clearly reusable process elements in safety-critical systems development. These commonalities also represent the source for reusable process-based arguments.

Besides reuse, qualitative comparison is also enabled since safety-critical variabilities become evident. A standard, for instance, may require/suggest the usage of a particular hazards analysis technique that better allows for hazards identification. To achieve these benefits, it is fundamental to understand what can vary between safety processes and what, instead, remains unchanged (common). At a first glance, processes defined in different standards seem to exhibit only variabilities. Terminological differences constitute a barrier to a straightforward identification of commonalities. Moreover, as mentioned in Section II.C, phases are constituted of a set of activities, which in turn are constituted of a set of tasks and which in turns are constituted of a set of steps. Thus, commonalities are unlikely at the root level of this nested structure. Finally, from an execution point of view, phase, activities, tasks, etc. may be performed in a different order. From a pure syntactical comparison, all these differences may be interpreted as variabilities. However, to be able to justify a process line approach we must have more commonalities than variabilities. To solve this problem, we propose to go beyond syntactical differences. To do that, we provide the following definitions, which are helpful to reduce the variabilities and increase the commonalities:

Partial commonality: whenever two process elements of the same type (e.g. two activities) expose at least one common aspect (e.g. at least a task is equivalent).

Full commonality whenever two process elements of the same type (e.g. two activities) expose only common aspects (e.g. all tasks are equivalent).

Moreover, each time two process elements of two different processes are called in a different way but they denote they same concept, we propose to consider them as a commonality. For instance, we interpret the term “clause” used in ISO-26262 as an activity in the process modelling domain. Similarly, we also interpret the term phase used in EN50126 as an activity.

Finally, in case the order of execution differs between different processes, we propose to neglect it if the effect on the work-products is unchanged.

B. Phase line: towards a process line

To present the adoption of the process line approach in the context of safety, in this subsection we focus on the phase named *Concept phase, including hazard analysis* [10], which corresponds to the ISO26262 and EN50126 phases presented in Section II.A.

To achieve the process line, we apply the development process for process lines briefly sketched in Section II.C. The scoping step is implicitly performed since it is a consequence of the choice of the two standards. After scoping, we identify commonalities and variabilities to be able to define a phase line, a family of phases, from which standard-specific phases can be derived. To do that, for each standard, we take the following actions:

- identification of activities, tasks, work-products, guidelines (fundamental to qualitatively detect which standard is more effective);
- identification of the order in which activities and tasks should be performed;
- identification of the way in which tasks are grouped to form activities;
- identification of the way in which activities are grouped to form phases.

Then, we compare activities with activities, tasks with tasks, etc. We also compare the order of execution. From this comparison, as predicted in Section III.A, we experience that common aspects are not easy to map due to:

- irrelevant terminological differences;
- irrelevant ordering differences;
- irrelevant grouping differences.

We overcome terminological differences either by selecting one of the terms already in use or by introducing an additional term to cover those already in use. For instance, in ISO26262 we have the activity called Item definition which focuses on the item, and in EN50126 we have the activity Concept phase which focuses on the system, we conclude that item and system are irrelevant terminological differences. Thus we propose to name the activity *Concept definition* to cover both.

To overcome irrelevant order and or grouping, we decide to introduce new activities that regroup tasks in the same

way and order, when necessary. For instance, we introduce the activity *Full System boundary and preliminary hazard identification* corresponding to activity *System Definition and Application Condition* in EN50126 and to the first two tasks in ISO26262 of the activity *Hazards analysis and risk assessment*. Similarly, we decide to consider activities in different phases when appropriate. For instance, we do not consider the clause *Functional safety concept* as an activity of the phase line *Concept phase including hazard analysis*. That clause has to be included in the following phase line.

In Table 4, we focus on one single activity and we provide the naming conventions that we use in our phase line. As Table 4 summarizes, in EN50126 the activity starts with task *Hazard identification*. In ISO26262, instead, two tasks namely *Instantiation of Hazard analysis and Risk assessment*, and *Situation analysis*, are carried out first. To solve this grouping difference, we consider a new activity, called *Hazard analysis and risk assessment*. This new activity starts with *Hazard identification*. The ISO 26262 tasks that represent an irrelevant grouping difference, are considered as mentioned before at the end of the activity *Full System boundary and preliminary hazard identification*.

Then, to identify commonalities, we use the definitions provided in Section III.A. For instance, we consider as partial commonality the task *Hazard Classification*, which is conducted in a similar way by both standards, as authors in [17] pointed out. More precisely in [17] authors conclude that “the various schemes are not fundamentally different, and could be seen as various instances of a single consistent scheme”. For instance, we consider a full commonality the task *Hazard Identification*.

Table 4 Naming convention at activity/task level

Activity-Hazard analysis and risk assessment	T1: Hazard identification T2: Hazard Classification T3: ASIL- risk determination T4: Determine the safety goal T5: Determine and classify acceptability of the risk T6: Establish a Hazard Log T7: Phase Verification
ISO26262-Hazard analysis and risk assessment	T1: Initiation of Hazard analysis and Risk assessment T2: Situation analysis T3: Hazard identification T4: Hazard classification T5: ASIL determination, T6: Determination of Safety goals T7: Verification of Hazard analysis, risk assessment and safety goals.
EN50126-Risk analysis	T1: Hazard identification T2: Hazard classification, T3: Risk evaluation, T4: Determination and classification of acceptability of the risk T5: Establishing the Hazard Log T6: Assessment of all tasks in the phase

C. Phase line in SPEM2.0

In this subsection, we model the phase line *Concept phase including hazard analysis* using SPEM2.0. Models are created by using Eclipse Process Framework (EPF) [21], which, with respect to our purposes, offers a complete support for SPEM2.0. To model the notion of partial commonality in SPEM 2.0 we have three options: *contributes*, *extends-replaces* and *extends* (see Table 1). In the framework of our phase line, we use the *extends-replaces* relationship. We choose this relationship since it permits modellers to express the fact that some values are replaced and other are left unchanged. To model variabilities, we use the *replaces* relationship.

Figure 1 shows the phase line, which is constituted of three common activities and one activity that might be replaced by an empty activity (optionality).

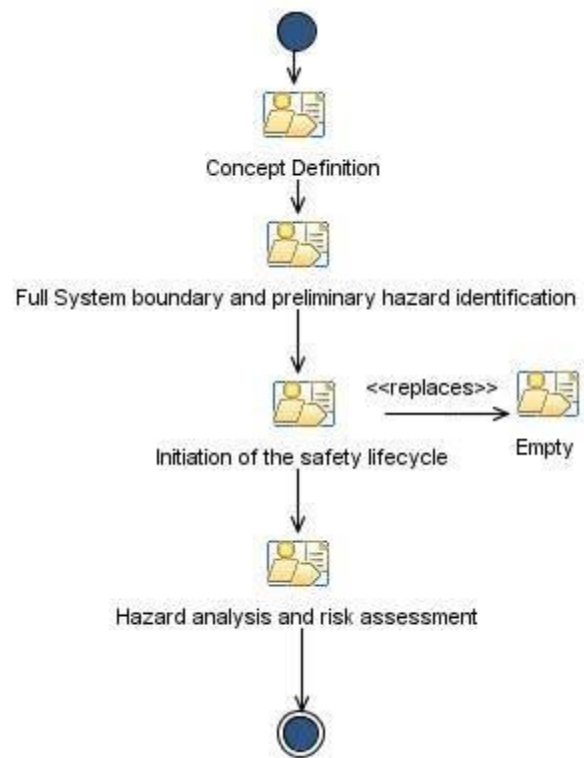


Figure 1 Concept phase including hazard analysis in SPEM2.0

Figure 2 details a partial commonality (the Hazard analysis and risk assessment activity) of the phase line. As Figure 2 shows, this activity is constituted of one full commonality, three partial commonalities and three variabilities (optionality).

For what concerns deliverables and guidelines, variability cannot be shown graphically in EPF. Separate tables containing information related to them can be associated to the model. In Figure 3, for instance, the deliverables of the activity (line) *Hazard analysis and risk assessment* are given.

Work Product Breakdown	
Breakdown Element	
	Hazard analysis and risk assessment report ISO26262
	Hazard Log EN50126
	Safety Goals ISO26262
	Verification report EN50126
	Verification review of hazard and risk analysis and safety goals ISO26262

Figure 2 Work-products (deliverables)

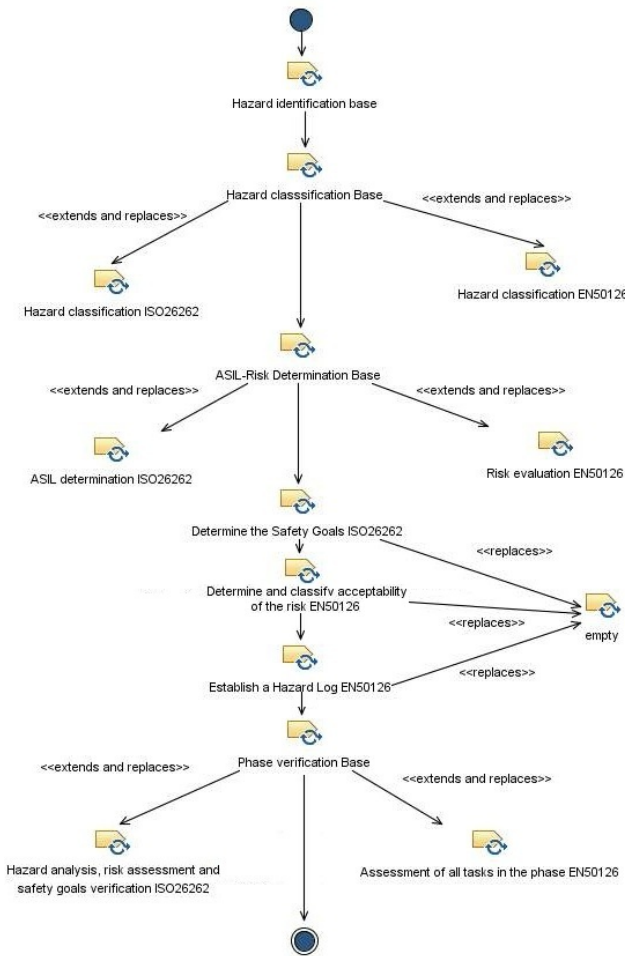


Figure 3 Hazard analysis and risk assessment in SPEM2.0

D. Phase line in vSPEM

In this subsection, we model the phase line Concept phase including hazard analysis using vSPEM Models are created by using StarUML [22]. The change with respect to the modelling tool is motivated by the fact that EPF does not support vSPEM. StarUML supports SPEM2.0 and also offers extension mechanisms which are easy to use and thus convenient for our exploratory purposes. However, since the support for SPEM2.0 is not satisfying from a compliance and completeness point of view, we have preferred EPF than StarUML for the models in Section III.B. More specifically,

in Figure 4 we model the same phase-line that is modelled in Figure 1 and in Figure 5 (available at the end of Section III.E), we model the same zoom-into-activity that is modelled in Figure 2.

In vSPEM, variabilities and partial commonalities are treated in the same way and they are graphically modelled thanks to the presence of variation points and variants. The arrow that relates the variation point with the variant is an instance of the canBeOccupiedBy relationship.

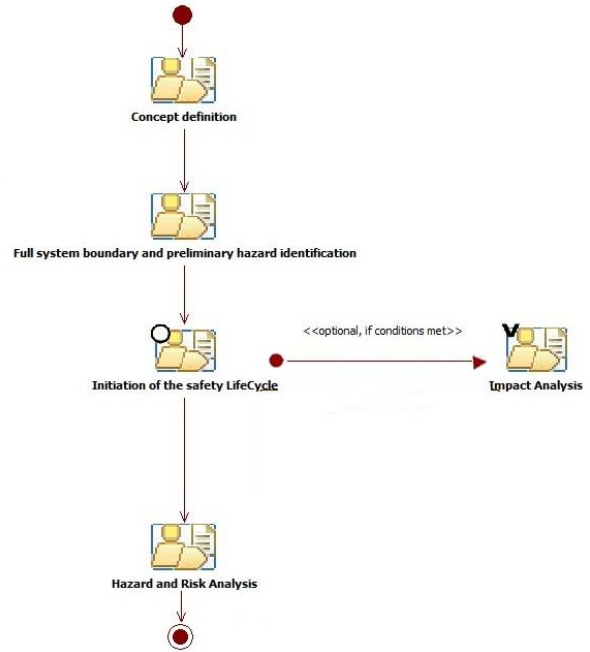


Figure 4 Concept phase including hazard analysis in vSPEM

E. Phase derivation

In this subsection, we show how the two phases can be derived and we emphasize the reuse possibilities. To do this, we limit the discussion to the activity level (see for instance the SPEM 2.0 model shown in Figure 1). By following the same steps that we are about to present, the reader can complete the derivation by zooming into the activity Hazard analysis and risk assessment.

To derive the ISO 26262 phase from the phase line shown in Figure 1, we select all the commonalities and desired variants at variation point. As a result we obtain:

ISO 26262 phase={Concept definition, Full system boundary and preliminary hazards identification, Initiation of the safety life-cycle, Hazards and risk analysis}

To derive the EN50126 phase, we behave in the same way and as a result we obtain:

EN50126 phase={Concept definition, Full system boundary and preliminary hazards identification, Hazards and risk analysis}

From a quick comparison of the two above-given sets, we see that the majority of the activities can be reused (either fully or partially). Thus in case of cross-domain certification

needs, the effort required can be localized and minimized. This result definitively encourages the adoption of a process line approach.

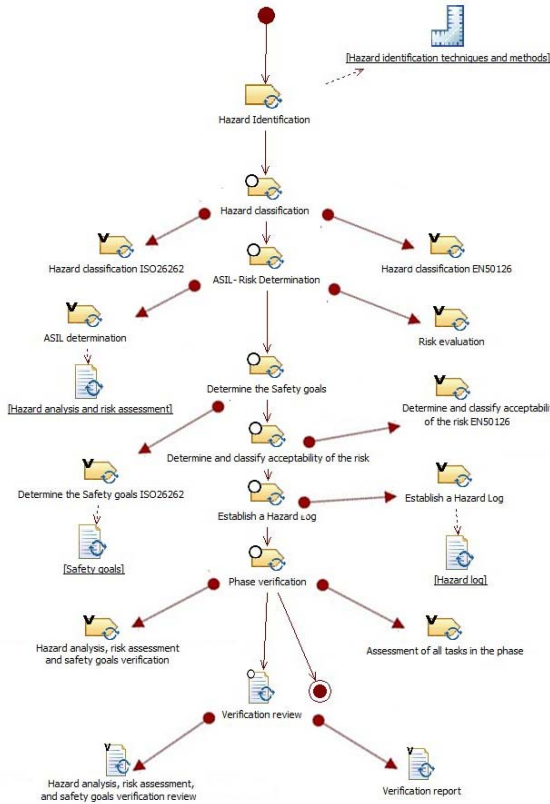


Figure 5 Hazard analysis and risk assessment in vSPeM

IV. DISCUSSION

In this subsection, we discuss with respect to variability and partial commonality modelling the advantages and disadvantages of SPEM 2.0 and vSPeM. Moreover, we also point out the limitations of our current work.

In SPEM 2.0, the variability can be expressed by using 5 different relationships as shown in Table 1 (Section II.D). These relationships also support us in expressing the notion of partial commonality. However, as pointed out in [25], SPEM2.0 lacks graphical modelling support and thus its understandability is hindered.

vSPeM, instead, is more understandable thanks to its graphical modelling capabilities. However, it does not support us to nicely distinguish partial commonality and variability.

Moreover, in both modelling languages, it is not possible to clearly model: 1) the inclusive or relationship; 2) cross-assets constraints. Finally, variability in terms of the process dynamics is not supported at all. The absence of these modelling capabilities hinders the achievement of a successful process line since the processes that can be derived are not compliant with the expected processes. Indeed, it might be useful to be able to model the possibility

of selecting more than one variant at a variation point (inclusive or) for example in the case of roles.

Similarly, it is fundamental to have at disposal the possibility to model that the selection of asset X requires/excludes the selection of asset Y. Finally, process dynamics should be modelled to avoid rigid process execution.

Currently, our work has limited the attention to activities, tasks, deliverables and guidelines. To be able to understand which modelling capabilities are required in the framework of safety processes, this work must be extended to identify and better solve the issues involved.

V. RELATED WORKS

The extremely time-consuming and expensive certification process represents a real burden for industries especially for those that produce slightly different safety critical systems (a product line) or systems that must be compliant to multiple safety standards (ecosystems). Standards comparison (towards the achievement of reuse-based approaches) is being investigated. In [29], authors survey several standards and give useful insights on common aspects of different safety guidelines. In this work, however, reuse is not addressed. In [14], authors present a reuse-based method for tailoring and verifying a software process. Thanks to its formal underpinnings, this method can be considered a powerful pioneer of the process line approach. Authors formally define a process as a tuple constituted of: 1) a finite set of process modules (reusable elements) and 2) configuration, which represents the modules interconnections. To enable process tailoring, a set of useful operations (including e.g. deletion, addition, splitting) is at disposal. The process designer can execute these operations to achieve the desired process model. Syntactic correctness and type conformance checking are then performed to statically verify the process.

In [11], authors sketch their research intention towards the provision of a high-level process model that abstracts away domain-specific details. According to them, the high-level process model should contain abstract but common activities required by the safety standards. This research intention has been partially reified in [10]. The resulting high-level process model, however, is not conceived in a clearly reusable or usable way. This work consists of an aggregation of the activities required by the standards (more specifically a juxtaposition of activities, considered to be conceptually equivalent, is carried out). Our work partially benefits from this work but introduces a different approach. Instead of proposing a single high-level process model, we define a process line and we allow for the derivation of domain-specific processes through selection and composition of process models.

In [18], authors investigate the possibility to produce a generic certification approach. They first conduct a comparative review in terms of commonalities and differences of international safety related standards in three different sectors (railway, automotive and aerospace). Despite the differences at the detailed level, the authors observe common aspects related to the processes for development and assessment of safety critical systems. Then,

based on these identified common aspects, the authors elicit and model the common underlying logic between the processes proposed by the standards. The proposed model consists of three integrated processes (a development process, safety assessment process and safety case process) which provide a basis for reusability in safety assessment and system certification. Despite its interest, this work models a generic and thus too abstract process. Our work with respect to this one differs since, besides the generality (commonalities), models the variabilities and thus the possibility to derive highly domain-specific processes. Moreover, we use standard-based modelling capabilities.

In [30], authors propose an approach based on UML profiles to capture explicitly the relationships between generic standards (e.g. IEC61508) and corresponding sector-specific standards. Similarly to our work, this work also contributes to systematizing the specification of the existing relationships between standards. Our work, however, differs from this one since we focus not only on parent-child relationships but also sibling relationships. Moreover, we use languages conceived within the process modelling community without introducing additional UML-based profiles.

CONCLUSION AND FUTURE WORK

In this paper, we have proposed to adopt a process line approach to systematize the knowledge concerning safety-oriented development processes. We have motivated our proposal by showing that certain standards can be easily considered as a family since several commonalities characterize them. The recognition of a family enables reuse possibilities as well as ease of qualitative comparisons.

Then, we have focused our attention on a specific process phase and we have developed a phase line. To be able to develop a successful phase line by maximizing commonalities and reducing variabilities, we have proposed to eliminate irrelevant variabilities (due, for instance, to terminological differences). We have, then, modelled the phase line by using two different modelling languages. By doing this we have realized that both these languages provide advantages and disadvantages and thus none of them can be considered a modelling solution.

Finally, we have derived the standard-specific phases from the phase line and we have shown the concrete reuse possibilities enabled by the approach.

In the future, we plan to build on our findings. In particular, since we believe that a process line approach is interesting and effective for providing reuse possibilities and ease of comparison in the framework of safety-oriented processes, we intend to further develop this research direction.

First of all with the support of the industrial partners involved in the SafeCer project, we plan to accurately perform: 1) the process line scoping, 2) the identification of commonalities and variabilities and 3) the process line architecting. Our target family will be mainly based on IEC 61508 and its descendants. However, we are also interested in investigating to which extent other transportation-related standard such as D0178B/C [27-28] can be considered part

of the process line. Our intention is also to understand how to manage the foreseeable huge number of variabilities. Should we zoom in until we reach the point in which everything is a variability? Is it possible to reduce the variabilities by identifying semantic equivalences between standards? As stated in [13], empirical work is fundamental to understand if the process line approach is really applicable, besides feasible.

Secondly, we intend to propose an extension of vSPeM to be able to overcome the current shortcomings. Then we plan to define a specific system function and analyse the hazards that are potentially related to it. This function definition and analysis will be carried out according to different phases in the phase line. The intention is to prove that the reuse possibilities are really tangible. Besides reuse possibilities, we also intend to exploit the process line approach as a possibility to ease qualitative comparison between standards. A function defined and analysed according to one phase may result to be safer than the same function defined and analysed according to another phase. More specifically, a phase may recommend guidelines or tools that result to be more effective in identifying threats propagation paths leading to hazards.

ACKNOWLEDGMENT

This work has been partially supported by the European Project ARTEMIS SafeCer [1] and by the Swedish SSF SYNOPSIS project [2]. We thank Iain Bate for fruitful discussions on this paper.

REFERENCES

- [1] ARTEMIS-JU- 269265 SafeCer - Safety Certification of Software-Intensive Systems with Reusable Components. <http://www.safecer.eu/>
- [2] SYNOPSIS- SSF- RIT10-0070. Safety Analysis for Predictable Software Intensive Systems. Swedish Foundation for Strategic Research. <http://www.mrtc.mdh.se/index.php?choice=projects&id=0356>
- [3] IEC61508:2010. Functional safety of electrical/electronic/programmable electronic safety-related systems.
- [4] ISO26262. Road vehicles – Functional safety. International Standard, November 2011.
- [5] EN 50126-1:1999. Railway applications: The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Basic requirements and generic process, including corrigendum May 2010.
- [6] N. Leveson. White Paper on The Use of Safety Cases in Certification and Regulation, updated May 6, 2012.
- [7] T. Ternite. Process Lines: A Product Line Approach Designed for Process Model Development. Proceedings of the 35th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 173-180, Patras, Greece, August 27-29, 2009.
- [8] OMG. Software & systems Process Engineering Meta-model (SPeM), v 2.0. Full Specification formal/08-04-01, Object Management Group, 2008.

- [9] T. Martínez-Ruiz, F. García, M. Piattini, J. Münch. Modeling Software Process Variability: An Empirical Study. *IET Software*, vol. 5, no. 2, pp. 172-187, 2011.
- [10] A Generic Process Model for Integrated Certification and Development of Component-based Systems. SafeCer deliverable D2.1.1 Month 8, 2011.
- [11] M. Åkerholm and R. Land. Towards Systematic Software Reuse in Certifiable Safety-Critical Systems. Position paper presented at the International Workshop on Software Reuse and Safety (RESAFE), Washington, D.C., September, 2009.
- [12] E. Ledinot, J. Gassino, J.-P. Blanquart, J.-L. Boulanger, P. Quéré, B. Ricque. A cross-domain comparison of software development assurance. Proceedings of the 6th European Congress for Embedded Real Time Software and Systems Conference (ERTSC), Toulouse, France, February 2012.
- [13] O. Armbrust, M. Katahira, Y. Miyamoto, J. Münch, H. Nakao, and A. Ocampo. Scoping software process lines. *Software Process: Improvement and Practice - Examining Process Design and Change*, volume 14, issue 3, pp. 181-197, May 2009.
- [14] H.-C. Yoon, S.-Y. Min, and D.-H. Bae. Tailoring and Verifying Software Process. Proceedings of the 8th Asia-Pacific on Software Engineering Conference (APSEC). IEEE Computer Society, Washington, DC, USA, pp. 202-209, 2001.
- [15] J. Bosch. From Software Product Lines to Software Ecosystems. Proceedings of the 13th International Software Product Line Conference (SPLC), Carnegie Mellon University, Pittsburgh, PA, USA, pp. 111-119, August 2009.
- [16] H. D. Rombach. Integrated Software Process and Product Lines. Proceedings of the International Software Process Workshop (SPW), Beijing, China, LNCS 3840/2006, pp. 83-90, M. Li, B. Boehm and L.J. Osterweil (eds.), 2005.
- [17] J.-P. Blanquart, J.-M. Astruc, P. Baufreton, J.-L. Boulanger, H. Delseny, J. Gassino, G. Ladier, E. Ledinot, M. Leeman, J. Machrouh, P. Quéré, B. Ricque. Criticality categories across safety standards in different domains. ERTS, Toulouse, France, 1-3 February 2012.
- [18] Y. Papadopoulos, J. A. McDermid. The Potential for a Generic Approach to Certification of Safety-Critical Systems in the Transportation Sector. *Journal of Reliability Engineering and System Safety*, 63, pp. 47-66, Elsevier Science, 1999.
- [19] Z. Stephenson. Guidance on the application of the revised IEC 61508 to DS 00-56. Technical Report SSEI-TR-000045, Software Systems Engineering Initiative, December 2009.
- [20] EN 50126-2:2007. Railway applications: The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 2: Guide to the application of EN 50126-1 for safety, 2007.
- [21] Eclipse Process Framework <http://www.eclipse.org/epf/>
- [22] StarUML <http://staruml.sourceforge.net/en/>
- [23] K. Pohl, G. Böckle, and F. J. van der Linden. *Software Product Line Engineering: Foundations, Principles and Techniques*. ISBN: 3540243720, Springer-Verlag, 1 edition, 2005.
- [24] J. Simmonds and M. C. Bastarrica. Modeling Variability in Software Process Lines. Universidad de Chile, TR/DCC-2011-10 September 14, 2011.
- [25] J. Simmonds, M. C. Bastarrica, L. Silvestre, A. Quispe. Analyzing Methodologies and Tools for Specifying Variability in Software Processes. Universidad de Chile, TR/DCC-2011-12, November 4, 2011.
- [26] T. Martínez-Ruiz, J. Münch, F. García, and M. Piattini. Requirements and constructors for tailoring software processes: a systematic literature review. *Software Quality Control*, 20, 1, pp. 229-260, March 2012.
- [27] RTCA DO-178B (EUROCAE ED-12B), Software Considerations In Airborne Systems and Equipment Certification. RTCA Inc., Washington, DC, December 1992.
- [28] RTCA/DO-178C Software Considerations in Airborne Systems and Equipment Certification. RTCA, Inc. 2011.
- [29] A. Wabenhurst and B. Atchison. A Survey of International Safety Standards. Software Verification Research Centre, School of Information Technology, The University of Queensland, Brisbane 4072, Australia, Technical report 99-30, November 1999.
- [30] R. K. Panesar-Walawege and M. Sabetzadeh and L. Briand. Using UML Profiles for Sector-Specific Tailoring of Safety Evidence Information. Proceedings of the 30th International Conference on Conceptual Modeling (ER'11), LNCS, Vol. 6998, pp. 362-378, Springer, 2011.