Integrating Software Safety and Product Line Engineering using Formal Methods: Challenges and Opportunities

Martin Becker, Soeren Kemmann, K.C. Shashidhar
Fraunhofer Institute for Experimental Software Engineering (IESE)
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
{martin.becker, soeren.kemmann, kodamballi.shashidhar}@iese.fraunhofer.de

Abstract—Product line engineering and safety engineering for software have both become mainstays to address the current challenges in developing software-intensive, safety-critical embedded systems. They address orthogonal concerns and the concepts and methods used by them have naturally evolved independently. A holistic, streamlined approach toward system engineering, however, obviously needs to identify and exploit the opportunities for a beneficial interplay between the two. We believe that appropriate formal models and methods can provide a suitable backbone in realizing such an approach. In this article, we present the challenges that arise while addressing safety in the software product line engineering context; and discuss where opportunities exist for leveraging formal methods and how they can provide the necessary techniques to address them.

Keywords—Software Product Line Engineering, Safety Engineering, Formal Methods, System Engineering

I. INTRODUCTION

Development of software-intensive, safety-critical, embedded systems is facing two major challenges in the recent times [1][2]: (1) ever increasing need for innovation and differentiation of features from software which needs to be developed within ever shrinking development time cycles to remain competitive in the market; and (2) ever increasing product complexity coupled with an ever more stringent demand on the quality, especially safety, requirements. The first challenge, in the face of an increased demand for variant-rich products, is being addressed by product line engineering (PLE) [3][4], while the second challenge, in compliance to standards and certification regimes, is being addressed by safety engineering [5][6].

A. Combined Demand for Safety and PLE in Applications

One of the most prominent domains facing the challenge of handling variant-rich products, which are at the same time safety-critical, is the automotive industry. Due to the fact that the driving factors for innovation and unique selling points of today’s cars are software systems, handling variants and reuse in a systematic way is indispensable. Consider, for example, a car which the customer can order either as a two-wheels or four-wheels driven variant. One can easily imagine that all the control algorithms for the traction-control need to be designed in a different way for each variant. Furthermore, from a safety point of view, different critical situations (or, hazards) may occur when four wheels are driven instead of two wheels. At present, each of these variants must be considered as a separate product and for each variant, the whole development process, including especially the safety engineering activities and artefacts, must be re-applied mostly from scratch. Note that, with the upcoming international standard, ISO 26262 [7] for functional-safety of road-vehicles, the compliance with prescribed safety-engineering process is mandatory for the car manufacturers.

Similar challenges are arising in the commercial vehicle domain. Due to the awareness for safety generated by the ISO 26262 in the key domain of transportation, other domains such as tractors and machinery for agriculture and forestry have also framed an international standard, ISO 25119 [8], for functional safety. This domain is facing the challenge that the maturity of their traditional safety activities is quite low and their process maturity level is often lower than that of the automotive industry. Furthermore, the quotient between number of variants and overall units sold is an order of magnitude higher than in the automotive industry. Hence, a profound PLE together with an integrated safety approach seems to be an obvious response to their challenge. An interesting fact is that, ISO 25119 has already stepped towards this direction by defining reference architectures as part of the standard. Depending on the criticality level, one can select an architecture and reuse the assumptions and calculations needed for its assessment. This is of course just one single aspect, but shows that, implicitly, the safety engineers too would like to have systematic reuse strategies for safety.

As a last one that is worth mentioning, we would like to point to the domain of medical devices. This domain, with an uncompromising demand for safe products, is also facing the software challenge similar to the automotive industry. More and more functionality is based on software and different products often lie only a set of parameters apart from each other. Especially challenging in this area is the normative requirement posed by the IEC 62304 standard [9] that software faults have the probability of 100%. Hence, when software is part of the device, one cannot reason quantitatively about meeting the safety requirements. New approaches for qualitative reasoning via semantics of elements and not via numbers need to be exploited. This challenge is not particular for the medical device domain, but for all products where software is part of the overall system. We mention it explicitly for the medical devices domain because their standard is the most explicit one with respect to this fact. For PLE in this domain, similar challenges and opportunities exist to those of the automotive and commercial vehicle domains.
B. Integration using Formal Methods

In order to meet the described challenges, while keeping costs and efforts manageable, it is clear that a systematic reuse approach is required in both design as well as safety related activities. A solution to this end is to use an integrated approach for software product line engineering and safety engineering, which have both independently proven their benefits in practice. This integration, we believe, need to be essentially based on formalization in order to precisely capture the intended semantics of the various artefacts and provide sound basis for their analyses [10] [11]. The way towards rigorous formal models and methods is of course a continuous process. An essential intermediate step is a semi-formal representation, e.g., the UML. With the rise of model-based development, semi-formal models have become widely known and are the quasi standard in Software Product Line Engineering as well as in Safety Engineering. Thus, a first step is an integration based on the foundation of semi-formal models. The usage of full fledged formal models, typically using logic, state-transition or sequence-causality based representations of various forms with well-defined semantics [12][13], are at present more on the research end. Nevertheless, they have already been used to model the artefacts for both safety and PLE, and analysed using techniques that leverage such industrial strength formal methods as model checking [14], static analyses [15], and satisfiability (SAT/SMT) solvers [16]. While the intermediate step of using semi-formal models for integration is far from being solved, in this article we want to explicitly focus on what is still on the research end – an integration using full fledged formal methods. However, integrating safety engineering and PLE based on formal methods is non-trivial due to challenges that we point out in this article and much work needs to be done before integration can be realized. Among them, for instance, is the fact that systematic variability management of formal safety artefacts is required, but it is not well understood, let alone supported.

C. Scope of Our Discussion

The reuse potential of safety engineering in variant-rich settings so far has only been exploited to a small extent, if at all. There is no holistic approach that integrates safety and product line engineering, but there is obviously a huge demand for such approaches. The goal of this article is to raise awareness for issues with respect to safety in the PLE context; to point to current literature on application of formal methods in safety engineering and PLE; and to discuss issues that arise from variability in safety artefacts. We discuss challenges pertaining to the question of how to merge PLE and safety engineering concepts with a focus on the application of formal methods. This includes an overview on safety engineering, the respective artefacts and engineering steps and specific challenges with respect to variability management. To drive the integration of safety and product line engineering, the article discusses general challenges and approaches for the integration and outlines feasible approaches.

D. Outline

The article is structured as follows. Section II provides an overview of functional safety, its parts and challenges with respect to the changes, variations, and reuse. Section III briefly introduces PLE concepts and discusses current literature on application of formal methods for PLE. Section IV discusses different approaches to integrate safety and product line engineering, and presents an approach for handling variability in safety artefacts. Finally, Section V concludes the article.

II. FUNCTIONAL SAFETY – AN OVERVIEW

A characteristic that every product has in common when talking about safety is that it is a system property. There is nothing like component safety, or even the term “software safety” is quite questionable. This is due to the fact that hazards and risks arise at the border of a physical system only. Safety is a property which is concerned with the prevention or mitigation of these system-boundary hazards and risks. Therefore, safety considerations always start with the system in its intended environment and further build a continuous and traceable picture of artefacts and their overall concept, in order to prevent or mitigate hazards.

Most people have a personal understanding of safety. Especially when considering safety-critical products, one has an idea what is meant by being safe. The common understanding of a safe system is to ensure that no accidents happen. Aligned with this is the definition of safety in the world of formal methods, exemplified by the temporal logic [17] formula (G ¬ p), meaning “something bad never happens”. From a practical point of view this is of course desirable, but in general not achievable, at least not with reasonable effort and cost. Therefore, international standards for functional safety weaken the notion of safety in two points. First, “something bad” could be rephrased to “something unacceptably bad”; second, the phrase “never happens” is extended by a probability threshold, that is, “happens with a probability lower than a maximum acceptable value.” These extensions are of course not new.

From a formal methods point of view one could argue that unacceptably bad is just a matter of defining “bad” in the first place, and the second extension is covered by probabilistic formal methods, such as probabilistic model checking [18]. Nevertheless, for the norm “safety” these are important facts and all activities and artefacts produced by safety engineering target at identifying unacceptably bad things and their causes and of course reducing the probability of their occurrence to a necessary minimum. Because product development is a complex task, faults may be introduced at all stages of development and therefore the whole development is subject to requirements in international functional safety standards. Therefore, we present the basic outline of safety engineering with respect to international safety standards in what follows.

A. Parts of Safety

A schematic outline of the parts of international safety standards is shown in Figure 1. Important first observation is that safety-standards are a kind of corset where the different types of activities and artefacts fit into and form/influence the other. In general, the traditional development comprises process and project management activities and artefacts, and of course, the product development related ones. These are influenced and formed by requirements in the safety standards; furthermore, a new set of activities and artefacts comes into play - the ones from safety engineering itself.
way to reusable safe-systems, or a safe product-line, all of these areas need to be considered, it is important to understand in more detail the implications of safety standards to the three areas of interest.

1) Process and project management requirements

In the safety standards, this part addresses all things in a development which influence a product, but do not depend on a concrete product. These requirements last from qualification of persons involved in the development and end in requirements on the decommissioning process. All of these generic requirements must be fulfilled and evidence must be provided. More interesting demands are the configuration management and the change management requirements. The configuration management in safety standards jargon refers to versioning and tracking of everything which is used in the development. Linked with the configuration management and versioning is the process of changing things. The change management process in a nutshell is that if someone needs to change something, an impact analysis must be performed. The purpose of this impact analysis is to identify which artefacts are influenced by the change.

![Figure 1. Safety standards and entailed engineering activities.](image)

2) Safety-related product engineering requirements

The requirements in this area more or less demand the development with methods and techniques which represent the current state of the art. Therefore, one for example needs to have (written) requirements, has to have architecture and so on. The rigor and formality the different activities and artefacts need to be engineered with strongly depend on the criticality of the product (or the function). Besides this important fact, which will be the subject of the following sections, the product engineering is concerned with the traditional/known phases and artefacts and is therefore not addressed in more detail here.

3) Safety Engineering

The safety engineering areas comprises roughly of everything which needs to be done or produced especially for ensuring safety. As mentioned in the introduction, safety is a system property. Therefore, safety engineering starts at the very beginning of systems engineering. The first step is to understand the part “unacceptably bad”. In order to do this, one needs to define what belongs to the system under consideration and what does not belong to the safety-critical development. This kind of boundary definition in safety jargon is the item definition. Here not only the system boundary itself is defined, but essential part is also the definition of the expected operation environment. It is, for example, possible that for one environment a system is considered safe, for another one it is not. A hairdryer, for example, is not intended to be used while taking a bath in a bathtub. Such a combination of system in a specific environment is considered unsafe. Therefore, a precise definition of possible environmental conditions is crucial for safety and needs to be considered systematically as part of the engineering process.

Once the item and its environment are defined, one has to think about possible hazards, which may occur in the lifetime of the product. In safety jargon this step is called hazard analysis and risk assessment. A hazard is a “potential source of harm”, thus, there are further conditions necessary to produce the harm (or the “bad” thing). Here the notion of probability comes into play. The question of how unacceptable a hazard is, is addressed in the second part - the risk assessment. As risk is defined as “combination of the probability of occurrence of harm and the severity of that harm,” the assessment rates the probability of the necessary environmental situation and the interaction scenario, that is, one has to think about situations and scenarios typical to the use of the system. Additionally, the severity of the expected harm (or accident) is assessed. The combination results in a kind of integrity level. Depending on the concrete standard, one can find different numbers of levels. For automotive systems, one has to assign one of five levels of integrity to the hazard and its associated risk. The above mentioned rigor and formality of the development is derived by this identified level of safety-integrity. For this whole process of item definition, hazard analysis and risk assessment, no consideration or support of formal methods is known to the best of the knowledge of the authors.

Once the hazards have been identified, one must perform a root cause analysis in order to understand the failure propagation through the system. This can be used afterwards to find suitable locations in the system to stop this propagation and hence prevent a failure at system boundary. Several techniques exist to support the modelling of cause-effect relationships of faults. Most of them are built on a formal model in order to calculate probabilities for certain events or to display the necessary combination of causes to trigger these events. Prominent representatives are fault tree analysis (FTA) [19], failure mode effect analysis (FMEA) [20], and generalized stochastic Petri-nets (GSPNs) [21].

The information of possible faults and failures throughout the system can then be used to construct a safety concept. A safety concept constitutes a strategy how and where a failure propagation path is cut off. Because this is an important fact in safety engineering and often misunderstood, we want to explain this in more detail. Our definition of “something bad” relates to system boundary failures, that is, if and only if a failure occurs at system boundary, it is considered to be “bad”. Hence, an internal failure or fault is not a problem, if one can be certain that this is not propagated to the boundary of the system. Furthermore, it is hard to talk about modular or component safety because the essential system boundary is missing in this notion! Consider a component A which is faulty
and another component $B$ implementing a strategy for detection and mitigation of the fault. It is clear that reusing component $A$ without a mitigation/protection component such as $B$ may be unsafe. The purpose of safety concepts is to build this global picture of the system’s safety.

Strongly related to the safety concept is the safety case. A safety case is built as one of the last activities after the product development is done. A safety case is an argument for “why an item is safe supported by evidence compiled from work products of all safety activities during the whole lifecycle” [7]. Thus, in addition to the strategy and plan for the system’s safety in the safety concept, the safety case provides evidence that the plan is fulfilled and that the development complies with the requirements of the respective safety standard. In the safety case, all mentioned areas aggregate to a holistic view of the system.

B. Formal Methods in Safety Engineering

Formal methods are being increasingly applied in safety engineering to provide a precise semantics to safety artefacts and analyze them with guarantees of soundness, similar to the way they are being applied to enhance model-driven engineering [22][23]. This has led to the development of techniques for problems ranging from capturing safety requirements [24] to formalizing [25], synthesis [26][27][28] and analysis [29] of safety artefacts like fault trees, FMEA tables, etc. The general trend here is to combine these techniques to evolve tool-supported methodologies for model-based safety analysis [30] that can ease integration of the design and safety activities in the development lifecycle. The central idea is to use a common formal (nominal) behavioural model of the system across the two activities and maintain a repository of iteratively updated artefacts. Apart from the typical use of the model to provide formal verification against the requirements of the system, it is extended based on information about the failure modes of the components and used to automatically synthesize the safety artefacts, enabling safety assessment. Such an integration of design and safety is yet to mature for industrial use, but the idea has been realized and demonstrated, for example, in the ESACS/ISAAC framework [31][32].

III. PRODUCT-LINE ENGINEERING – AN OVERVIEW

The article assumes that the reader is familiar with the terms, methods, and the process of product line engineering; nevertheless, we briefly introduce the main concepts and approaches to build a common understanding for PLE and the usage of formal methods in this context.

A. PLE Concepts and Approaches

Product line engineering aims at the systematic development of a set of similar software systems by understanding and controlling their common and varying characteristics. In principle, this is achieved with (i) a thorough planning of reuse along a finite number of concrete (existing, planned, or future) products called scoping; (ii) the development and evolution of reusable and thus generic artefacts in a separate family engineering lifecycle, which are provided in a product line artefact base; (iii) the development of products, based on the provided reusable artefacts in the application engineering; and (iv) an overall variability management that allows to exploit the reuse potential in an efficient and effective way, for example, by providing necessary knowledge about supported variability and managing the inter-relations between the various artefacts resulting from the actual and the intended reuse.

B. Formal Methods in Product Line Engineering

The very conception of the FMSPLE workshop bears testimony to the opportunity that exists for enhancing product line engineering by infusion of formal methods. A recent survey [33] compiles state-of-the-art in attempts to exploit this opportunity and identifies the research challenges that still need to be addressed for further adoption. We refer the reader to the survey for a brief tour of the particular formal methods, and the recent progress there in, that are relevant for software product line engineering practice. The survey concludes that while formal methods have aided modelling and composing feature models with some success, new techniques and extensions need to be developed for scalable analyses and configuration of feature, as well as behavioral model. Recently, there has been further progress in this direction, as presented by [34][35] in the setting of feature diagrams (FDs) [36], by [37] in terms of a variant of modal transition systems for behavioral modeling and toward formal verification, as discussed in [38]. The topic is at present a very active area of research, and has also become the central focus of a recent Europe-wide coordinated research project [39].

IV. TOWARDS INTEGRATION OF SAFETY INTO PLE

Developing safety-critical systems with PLE raises the question if and to what extent safety related artefacts can be reused. Reuse potential ranges from safety models, such as Fault Trees (FT) over concrete analysis results, to complete safety cases. In this section, first, we discuss different strategies to integrate Safety into PLE, second, we point out existing related work of safety and PLE, and finally, we present one of our own approaches to manage variability in Component Fault Trees.

A. Integration Strategies

In order to exploit the reuse potential, different strategies to integrate Safety with PLE can be followed. Ideally, with maximum reuse, the entire product line could be certified as a
whole by a certification authority, instead of certifying each single product. However, we are not sure at this point if the scenario is realistic. Safety is a property inherent to a concrete product. Usually, a product line has a broad scope; therefore it seems unlikely to characterize the product line itself as being safe. Furthermore, the assessment if a product is safe or not, requires that all safety-relevant facts for that product are known and this would be required for all products to be derived from the product line. This is typically not the case, as it would mean, that the safety-related part of all products is completely available in the family engineering and only minor, non-safety-relevant modifications/extensions are conducted during application engineering.

At the other end of the spectrum is the typical case: product lines are engineered, with no variability management in the safety engineering and related artefacts, i.e. the core assets comprise only product related artefacts and not safety-related ones. Products are derived from the core assets and the safety-related artefacts are built anew for each product. Obviously, this leaves a considerable reuse potential unexploited, although we have to mention here, that the reuse potential in safety engineering might be less than expected, as for instance, minor changes in the product architecture can result in huge changes in the safety artefacts, if both are not aligned well. A solution strategy can be to align both the safety artefacts with the product artefacts along the envisioned changes or variations. The systematic planning and realization of variability, for example, with scoping, will be a substantial enabler to this end.

Another strategy is to set up a product line for safety engineering, called a safety product line, for a set of products which themselves are not engineered as a product line. With this, the reuse potential in the safety activities and artefacts could be exploited. However, safety artefacts do not support variability per se and continuity is not assured. Some approaches exist for designing variability in safety models [41][42], but to our best knowledge there is no systematic and planned (overall) approach today and from PLE we have learnt that unsystematic variations, unlimited parameterization and deliberate generalization for reuse are almost certain facts for inefficient and failing product lines. Learning the lesson for a second time with safety engineering clearly needs to be avoided. Nevertheless, it is imaginable that parts of a safety analysis (for example, those made for commonalities) may be brought into in the communication with assessors. We think it is at least possible to treat safety models, such as FTs, as another type of product line artefact. That is, we keep artefacts generic by introducing variation points, which are then resolved on instantiation for a specific product. We have to consider the whole chain in PLE! Otherwise “safety” is invalidated.

The highest chance to succeed with the integration of Safety and PLE seems to be a holistic approach, where the product engineering and the safety engineering are set up together in one product line and where the safety related issues are addressed from the very beginning and throughout the complete lifecycle. For instance, the item definition and hazard analysis could be included in the scoping process. Reuse becomes possible if hazards stay the same or hazard changes can be analysed, i.e. one must be able to compare hazards and their parameters. Using formal models, e.g. formalized feature models, to capture essential parts of the item definition and hazard analysis yields the opportunity to systematically reuse hazards or at least reason about changing hazards. Noticeably, the focus of scoping needs to be extended from software with some system information to system operating modes with explicit environmental models, as safety is a property about a system in its environment. A central enabler for the reuse in a family of safe systems will be their joint architecture. The interplay of architecture is essential for a single product, but for considering product lines supporting the mutual influencing character of architecture and safety it becomes indispensable. It is well suited to trace changes in architecture or architectural elements to elements of safety engineering such as safety analysis and safety concepts, and vice versa. Methods to formalize these traces would be a great help to automatically reason on the effect of change. Safety brings in at least one more variability - the safety integrity level. It is reasonable that a component which is valid in one architecture is not valid to be used in another even if the functional specification fits 100%, because its integrity level is less than demanded. Obviously, a family engineering decision has to be made whether to exclude architecture versions, where a component gets different integrity level requirements or decide to develop the component with the highest expectable integrity level, which could lead to enormous additional effort. What is true for one component is true for the whole architecture. Information, constraints and restrictions from the family of safety analyses and safety concepts expected for the product family need to be evaluated to identify viable architectural configurations. Exploring the design space and validity of combinations or architectures, components, and safety artefacts needs automation, which of course requires a formal representation of the process and their models as well.

B. State of the Art - Safety and Product Line Engineering

In the following, we give a short overview of the most important work that has been done in order to integrate safety and product line engineering.

Dehlinger, Lutz et al. [40][41], with their safety analysis of software product lines, concentrate on the reconciliation of reuse, variability and safety assurance. For this purpose, they combine traditional safety models, describing combinational failure logic, with state charts that model component behaviour and are associated with FT leaves. By state charts, as opposed to classical basic events or Markov chains, functional and failure dependences can be expressed by the model. The software FTA (SFTA) model, supported by software FMECA (SFMECA), describes the failure combinations for the whole product family need to be evaluated to identify viable architectural configurations. Exploring the design space and validity of combinations or architectures, components, and safety artefacts needs automation, which of course requires a formal representation of the process and their models as well.
changes on safety. The relationships and dependences of hazards, causes and design assets are visualised by way of a dependency matrix that can be used to trace the safety impact of variations in a product. From the matrix, fault trees are derived whose variations can be traced back to the impact of design changes. Similar to Dehinger, Lutz et al., “fault tree structure is relatively independent of design structure”, a problem they explicitly note, such that “[changes] to the design can have an impact on any part of the fault tree” [42].

Giese, Tichy et al. [44][45] address the issue of local structural variance of the failure model by using a CFT-like concept, where Boolean component failure models are combined according to the system data flow structure. Component ports, representing incoming and outgoing failures (failure modes) are typed using a failure type lattice, which enables type-checking the compatibility of connected ports. By mapping more abstract onto more refined failure modes, possible inconsistency between failure specifications on different abstraction/refinement levels can be detected and avoided.

Additionally, Trigaux and Heymans [46] present a research agenda with a focus on the simplification of existing software PLE methods and on including product safety as a main concern. The objective is to particularly support smaller companies in adopting PL techniques. To our knowledge, no definitive results have been published so far.

C. Variability Management in Safety Artefacts

In the following, we investigate an approach that has been developed as part of an InnoDNT project (cf. Acknowledgement) in more detail and put a focus on how variability can be handled in the safety artefacts, which also comprise of formal models.

The integration of safety artefacts with software product lines inevitably brings up new challenges. First, we need a concept that allows us to make a statement about a system’s safety whenever its functionality changes (what impact does a change have on safety artefacts and on product or product line safety, respectively). Second, we are interested in answers to the questions: which concepts of (formal) safety models can vary, how can they vary, how to cope with generic artefacts (i.e. incomplete or including several variants) during the family engineering and how can we efficiently establish traceability from variability to the variation points in functional and safety models.

From a technical perspective, making variability explicit in the integrated product line, establishing traceability from variability to variation points, and supporting efficient and effective resolution of variation points in the various artefacts the established approaches in PLE can be followed for the safety engineering and safety-related artefacts. For instance, to make variability explicit and establish traceability from variability to variation points in functional models and safety models, for example, component fault trees (CFTs), and also to establish traceability among the models, decision models (DM) can be used [47][48]. Basically, a decision model serves as a container for decisions, which are some kind of variation point. To operationalize the associated variability, the decision defines a question and a set of possible answers (resolution set). The question is then to be answered during resolution (the process of making all decisions and thus deriving an actual product). Decisions are connected by resolution constraints (logical expressions), stating how resolutions of decisions depend on each other.

To bridge the gap between decision models and actual artefacts, which include variation points, an artefact model is used (Figure 3). Each domain concept is represented by a corresponding artefact element which may again refer to other artefacts. If an artefact can vary, it is represented by a variant-artefact element which additionally states a variation point. This allows connecting decisions to variant artefact elements via resolution constraints, because they are Variation Points.

![Figure 3. Metamodel for product line artefacts.](image)

Each artefact making up a component in the comprising system owns a decision model that captures the knowledge about why and how the artefact varies and how that artefact is instantiated for a specific variant of the nesting component. The resolution of the artefacts’ decision models is then driven by another decision model attached to the parent component. It specifies how the component can vary from an external point of view and hence guides the instantiation of that component. This makes the component systematically reusable, because it is made explicit how a component can vary and what the actual impacts on the component’s artefacts are. Finally, DMs of different components (for example, parent and child components) are connected via constraints. Constraints are basically logical statements expressing how the resolution of a decision depends on the resolution of another decision. In this way, the DM contains the information about the impact of a change to a component’s functional model on the corresponding safety model and vice versa.

An example of a component, Proxy(C2), and the corresponding CFT is shown in Figure 4. The rectangles represent elements from the artefact model, viz., artefact elements (AEs) and variant artefact elements (VAEs), while decisions are represented by circles. For each VAE, a decision exists, and via constraints these (simple) decisions are aggregated to more complex decisions. Thus, the variability in the functional model, which is interrelated with the variability in the safety model, is explicitly connected and supports the traceability of changes. To methodically identify and model variability, a component based variability management approach as proposed in [49] can be used.

Regarding the realisation and resolution of variability in the safety artefacts in general and formal safety models in special two different approaches can be followed: intrinsic vs. extrinsic variability realization. The intrinsic approach requires that the language of the artefact supports the realization of variation
the realized variability in the artefact, which would be a substantial help during family engineering.

![Diagram](image.png)

Figure 4. Relating functional and safety models.

In the extrinsic approach, some kind of preprocessor is used to realize and resolve the variation points. With this, no specific language elements are required. If the preprocessor code is included in the artefact, this results in the problem, that the generic artefact, including the preprocessor directives is often syntactically and semantically invalid. Typical problems comprise (i) missing artefact elements for some variants, as their concrete implementations were not known/provided in advance and (ii) several alternatives are provided for a variation point in the artefact, where only one variant is allowed. Unfortunately, this prevents early analysis, if formal methods are used during family engineering. A possible solution to this end is to provide only the default values of the variation points in the document and replace them via a dedicated preprocessor during the resolution. The variability resolution could be supported via an identification/localization of variation point in combination different deltas that change the affected artefact elements in the variation points. This is fairly similar to the approach of decision models outlined above.

It can be concluded that from a technical perspective, the management of variability in safety artefacts, including formal safety models, raises issues that can be addressed to a large extent with the existing approaches. A viable strategy to extend the usage of formal (safety) methods in the PLE context could be to follow the extrinsic approach until the intrinsic approach would definitively provide additional support in managing the variabilities. The prospective advantages then could drive the development of formal models that support the intrinsic variability realization.

V. CONCLUSION

The concepts and methods used in product line engineering and safety engineering have evolved largely independently. A holistic, streamlined approach towards system engineering needs to identify and exploit the opportunities for interplay between the two and we believe that formalization with formal methods as the extreme provides a suitable backbone to achieve this. In this article, we have presented the challenges that arise while addressing safety in the software product line engineering context and have discussed where/how formal methods can provide the necessary techniques to address them.

The integration of safety artefacts with software product lines inevitably brings up new challenges. However, from a technical perspective in order to make variability explicit in the integrated safe product line, the established approaches in PLE can be followed. And it should be mentioned here that the handling of the variation in the safety-related artefact appears to be a minor issue. The major challenge is to integrate safety engineering and PLE into a holistic approach, where the product engineering and the safety engineering are set up together in one product line, and where the safety related issues are addressed from the very beginning and throughout the complete lifecycle. The benefits will be twofold: first, safety artefacts can be reused in an organized and consistent manner; and second, by analyses of safety models that can deal with variability, development risks and certification effort can be reduced, ensuring safer and more competitive products.

The present article intended to raise the awareness for the specific issues involved and, as such, only outlined promising approaches to align, and exploit, the safety and product-line related artefacts. We plan to elaborate and validate the outlined approaches in our future work.

VI. ACKNOWLEDGEMENT

Our work was partially supported by the Fraunhofer Innovation Cluster for Digital Commercial Vehicle Technology (DNT/CVT Cluster). We would like to especially acknowledge the work of our colleagues, Daniel Pech and Marc Förster, on variability in CFTs. K.C. Shashidhar was supported by the ERCIM "Alain Bensoussan" Fellowship Programme.

REFERENCES

References:


