Graphical Approach for Modeling of Safety and Variability in Product Lines

Aleksandra Salikiryaki, Iliana Petrova * and Stephan Baumgart †
*Mälardalen University, Västerås, Sweden
Email: asi13004, ipa13001@student.mdh.se
†Dept. Process Development & Change Management, Volvo Construction Equipment, Eskilstuna, Sweden
Email: stephan.baumgart@volvo.com

***Abstract*—Reuse of already developed parts and concepts is a common approach in industry to reduce the time to market and reduce the development efforts. Industrial product lines are often grown over time and structured approaches to support decision-making and manage the complexity are lacking. When developing safety-critical products through product lines, evidence must be provided for all possible product configurations. The lack of a structured product line approach taking the functional safety dimension into consideration makes it challenging for practitioners to provide the required evidence. In this paper we (1) identify requirements that a variability management approach will need to fulfill, (2) discuss existing approaches and their limitations, (3) propose potential extension, (4) apply our approach in an industrial use case and (5) discuss its applicability and future work.**

***Keywords*-Product Line Engineering, Functional Safety, Variability Management, Model-based Development**

I. Introduction

Products developed in the construction equipment domain are heavy vehicles used for mining, digging, road laying and similar. Generally, those products are highly configurable to meet customer needs and one product may have different application scenarios for different customers. Despite the variability, there is a potential of solutions to be reused in one product line but also across product lines. Reusing as many common features as possible is seen to be beneficial, since this decreases the development cost and allows a faster time to market. In literature even an enhanced quality of the products is reported [1].

Software is flexible and has the potential to be easily and less costly adapted to the purposes of the different products compared with electronic hardware solutions. Software Product Lines Engineering (SPLE) is a methodology, which allows systematic reuse of development artifacts through efficient identification and management of commonality and variability of the products [1]. However, the development of efficient reusable software solutions is a challenge since the complexity of the systems introduces plenty of dependencies between the different functions. Furthermore, development of a Product Line (PL) is typically spread among different projects and geographically distributed teams, which complicates the communication and synchronization of the design decisions and possibilities for reuse might be missed [2].

Because of the application of the products in the construction equipment domain, even small malfunctions can lead to accidents with damage to environment and harm to people. Thus, it is crucial to analyze the potential product configurations and ensure their safety. For the purpose of providing guidance and defining a state-of-the-art on how to develop safety-critical products, domain specific functional safety standards are created. The standards define activities and require documentation, in order to collect evidence for the absence of risks and how run-time failures are managed by the system. While ISO26262 [3] is an automotive domain specific standard, the IEC61508 [4] and ISO15998 [5] are relevant for construction equipment machinery. Nonetheless ISO26262 needs to be understood and interpreted for the construction equipment domain, in order to collaborate with suppliers who develop their products according to this standard.

The identification of commonality and variability of the products at early development stages is of a significant importance, since it allows the practitioners to make informed decisions and create configurable and reusable architecture. The functional safety standards require an initialization of the safety work already in early project phases, in order to incorporate hazard mitigation strategies into the system design. Therefore, we focus on the concept phase of development and propose a model-based approach that captures commonality and variability, as well as functional safety-related elements. We the approach in a simplified use case from the construction equipment domain.

In section Section II we present the state of practice in development of PL and discuss the characterizing requirements that a variability management approach for safety-critical PL should fulfill. Section III describes variability management techniques proposed in literature. In section IV we present our approach and explain in detail each of the proposed diagrams. In Section V the proposed approach is discussed with regards to the predefined requirements and its benefits and limitations are presented. Section VI concludes this work and we refer to future work in VII.

II. State of the practice

In a previous empirical study [6] we have studied the challenges that practitioners face when developing safety-

critical products in product lines. In the current work we focus on how variability is managed in industry and what difficulties practitioners experience during the development process [7]. We conducted interviews, studied related literature and derived requirements that a variability management approach needs to fulfill to support the development of safety-critical PL.

A. Empirical study

One of the purposes of the empirical study was to gather objective and complete view of the existing challenges and industrial needs. Informal interviews with experts with different qualifications and responsibilities in the development of safety-critical products were conducted. In particular, practitioners who are involved in the development of more than one safety critical PL as a requirement manager, a software architect, configuration and product managers, as well as software engineers whose focus is on specific functions were interviewed. Furthermore, several discussions were held with a team of safety experts, who are involved in each phase of the process of functional safety certification of the products. In order to gather feedback and discuss the results of the study and possible improvements, we have conducted several presentations and a final workshop.

Another purpose of the conducted study was to extract and analyze information related to safety and variability in PL. However, the distributed environment and the complexity of products makes the process of elicitation complicated. In order to handle with this situation, the practitioners guided us through the development process, the tools and documents used. Variability is hidden in multiple textual-based documents and it can be represented differently in the tools. Furthermore, dependencies between features are not visible and it is difficult analyze their impact on functional safety.

In parallel with the industrial study, we have examined several literature resources that present the state of the art relevant to the development of PL and safety-critical systems [8] [9] [10]. They discuss important aspects that should be analyzed and appropriately handled, as well as challenges and issues that practitioners face.

B. Requirements

As a result of our study, we have synthesized the following requirements that a potential variability modeling technique should fulfill.

R1. *Unified graphical representation*. A common unified representation of the information handled during the analysis and design of PL help to achieve traceability and avoid inconsistencies. In distributed development, where system characteristics need to be communicated and understood in the same way at different global development sites, the use of a graphical modeling approach may improve the self-descriptiveness and ease of interpretation.

R2. *Safety-related elements*. The results of the conducted safety analysis (safety-critical features, hazards, mitigation strategies, criticality levels) should be explicitly represented and traced in the modeling approach to help the practitioners to take informed design decisions. The practitioners we talked to mentioned the relation between safety and non-safety-critical features to be important to capture since it may introduce additional constraints on the architecture. This information may also help practitioners to derive credible evidence that the system is developed according to the functional safety standards.

R3. *Commonality and variability*. Common and variable elements and especially those related to safety-critical features are important to be captured to allow the practitioners to make decisions at further development stages. From a safety perspective, the visual representation of commonality and variability will facilitate the identification and documentation of common mitigation strategies, which can be reused in different products in the PL. Violations of safety goals or missing mitigation strategies in product configurations shall be possible to identify.

R4. *Direct and indirect dependencies*. A failure in one feature might propagate and cause side effects in other features, which are directly or indirectly related to it. A variability management concept shall represent dependencies already at early stages of the development, in order to prevent expensive late design changes. Rules for inconsistency checking may be defined to reveal indirect dependencies.

R5. *Multiple modeling views*. The development of complex embedded systems requires an thorough analysis from different perspectives such as static structure, dynamic behavior, hardware elements and interaction with the environment. In the context of model-based approaches, they are represented in multiple modeling views, in order to allow separation of concerns [11]. Multiple modeling views will help to analyze functional safety in safety-critical PL from different perspectives and enable practitioners to identify potential safety goal violations.

R6. *Traceability*. The modeling approach should allow practitioners to trace the system‘s elements through different views, in order to ensure consistency [11]. This will enhance the process of tracking and considering the impact of possible changes, as well as identification of possibilities for reuse. Traceability to and among safety-related features will support practitioners in the process of safety analysis. For example, the relation between a feature and its related hazards and strategies for their mitigation should be explicitly represented, in order to extract credible evidence for certification.

III. Related work

Variability management techniques proposed in the literature are focused on different abstraction levels and represent variability in different ways [8] [9] [12]. The purpose of

this study is to find an approach that is able to cover all the identified requirements, presented in Section II. In this section, we present the variability management methods identified during the research and discuss their applicability.

- *Feature-based techniques* represent variability in PL in terms of common and variable features, which present "user-visible aspects or characteristics of the domain" [13]. Kang et al. [13] proposed the Feature-Oriented Domain Analysis (FODA) method that uses a hierarchical tree structure to present features. The Feature-Oriented Reuse Method (FORM) is an extension of FODA prosed by Kang et al. [14], which classifies the features in the following categories: capabilities, operating environment, domain technologies, and implementation techniques. A matrix-based approach [15] extends the classical feature models with Unified Modified Language (UML) stereotypes. Its idea is to manage the features in a feature tree view and a feature dependency view. The method provides the possibility for inconsistency checking of the relations at earlier development stages. Yuqin Lee et al. [16] propose a graph structure for representing not only static, but also dynamic dependencies among the features. The feature-based techniques are mainly focused on a high level of abstraction. They are appropriate to facilitate the communication among all stakeholders and represent commonality and variability in the initial development phases. However, these techniques represent a PL only by feature views and this reduces the ability to analyze the products from different perspectives and thus violates requirements *R5* and *R6*. Therefore, the feature-based approaches should be combined with other techniques, in order to create a complete and traceable model of the PL.
- *Model-based techniques* such as UML provide an abstract view of a system from different perspectives throughout all development phases and diagrams can be used to describe static and behavioral aspects of the system. The Product Line UML-based Software Engineering (PLUS) introduced by Gomaa [11] extends UML by representing commonality and variability in a software PL. Maga et al. [17] propose an extension to SysML, which represents variability through extension packages and stereotypes on the relations. The model-based techniques are capable to cover most of the identified requirements. However, none of the examined approaches proposes a strategy for integration with the safety analysis and representation of the safety-related elements, which does not fulfill requirement *R2*.
- *Other techniques* such as COVAMOF proposed by Sinnema et al. [18] bring the idea to extract only the variability in a separate view and represent it using variation points and dependencies among them. The target of this approach is to make variability visible, but it can not be used as a stand-alone-solution. Representing the commonality and variability of the development elements in a unified way is not possible, which we see as crucial for managing functional safety. Thus, requirements *R1*, *R2* and *R4* are not satisfied.

Each of the discussed techniques is suitable for modeling of certain parts of a PL. However, none of them is able to satisfy all identified requirements. None of the investigated methods provide a concept for unified representing of commonality and variability in the development and functional safety artifacts. We propose a solution which combines and extends the examined methods in a graphical model-based approach for management of variability in a safety-critical PL.

IV. Approach

In this section we present a graphical model-based approach, which aims to support the system engineering as well as safety assurance processes of a PL. In this work we focus on the conceptual phase of PL development and more specifically we propose Feature, Use case, State machine and Safety configuration diagrams for representing the system characteristics from different perspectives (fig. 1). We illustrate the diagrams by applying them in a simplified industrial example.

The documentation provided during concept phase is used as an input for deriving the proposed diagrams, which are refined by further analysis. Among others the machine specification can be used to find information about common and optional features in the PL and the targeted machine variants. The Preliminary Hazard Analysis (PHA) [19] and high level safety requirements document are used, in order to obtain safety-critical functions, hazards, mitigation strategies and criticality levels. They are represented in the diagrams in order to provide functional safety context. We aim to depict element dependencies which might be crucial from a safety perspective. Furthermore, new hazards and possible lack of mitigation strategies can be identified. Therefore the diagrams can be used as feedback to the safety analysis.

The diagrams provide information to each other, in order to iteratively develop a model of the PL (Figure 1). The machine specification is used as an input for creating usage scenarios for the products in the PL, which are depicted in the use case diagram. Based on them, the states of the systems in the product are identified and the transitions between them are depicted in the state machine diagram. Product features, specified in the machine specification are represented in the feature diagram. During the analysis of the usage scenarios and the behavior of the products, static and dynamic dependencies are identified and added to the feature diagram. The safety configuration diagram is created based on the top level features and their relation with specific machine models, which are derived from the machine specification.

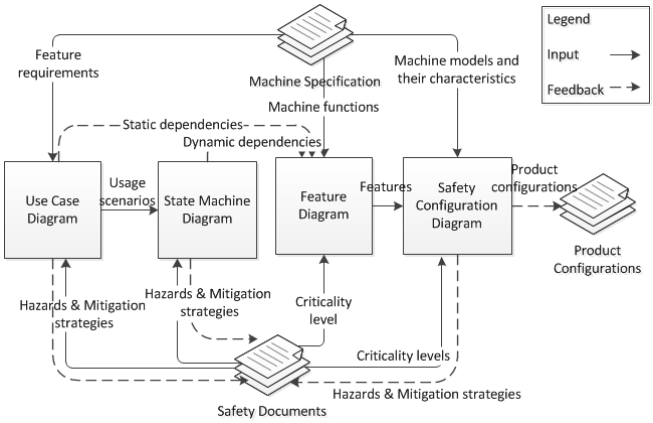

Figure 1. Approach overview.

Each diagram presents the machine features from a different perspective and also the dependencies between them are depicted. This helps practitioners to analyze the current results, estimate possible risks and take informed decisions regarding the next development phases. The proposed approach uses unified representation of commonality, variability and safety elements based on customized UML stereotypes which helps to achieve consistency and enhance the traceability between the views.

- Commonality and variability are represented by elements which show whether a certain machine function is mandatory or optional for the PL, or it has alternative realizations. To achieve this purpose we use the classifying stereotypes mandatory, optional and alternative defined in PLUS [11] to denote features, usage scenarios or machine states in the corresponding diagrams.
- Safety-related elements extracted from the safety documents are represented and examined from different perspective in each diagram. Stereotypes are used to denote safety-critical features, hazards and mitigation strategies. Their corresponding criticality levels are represented as attributes.
- Dependencies represent the interrelations between the elements and they might be static and dynamic dependencies and constraints [15] [16]. The static dependencies depict the hierarchical structure in terms of relations between the features. Dynamic dependencies represent behavioral interactions of the features, which occur during the work of the machine. The constraints allow specification of rules for combination of the different features and potential configurations. The dependencies are explicitly denoted through stereotypes in the feature and safety configuration diagrams, while the use case and state machine diagrams use standard UML relations and transitions for their representation.

In order to explain the notation of the diagrams and their specific purposes, we apply them on a simplified example,

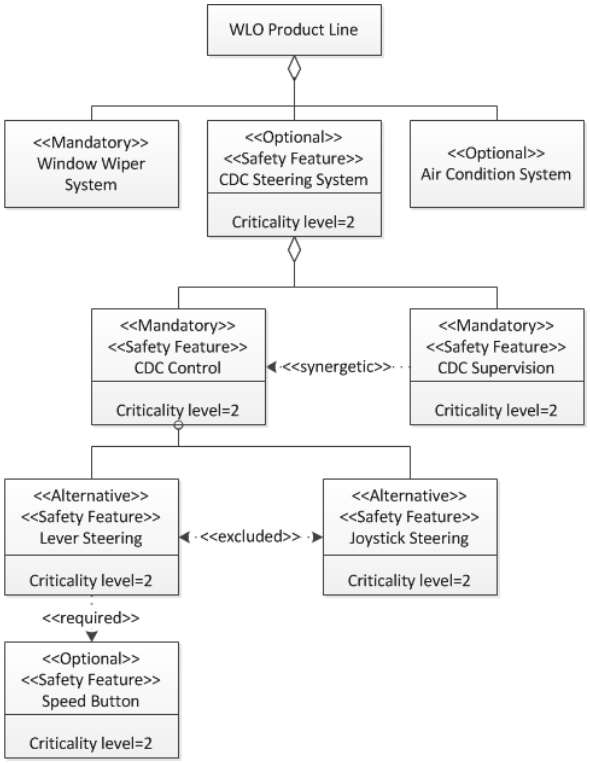

Figure 2. Feature diagram for the WLO PL example.

which is acquired during the conducted industrial study [7]. We have examined the steering functionality of the *Wheel loader (WLO)* PL with two alternatives a) through a mechanical steering wheel and b) through *Comfort Drive Control (CDC)*, which uses a steer-by-wire technology. CDC is specified as an optional feature in the machine specification and safety-critical, because its failure may lead to accidents if the operator looses the steering ability. In order to identify potential failures, the calculations of the CDC are monitored by *CDC Supervision* feature. In case of failures of the CDC, the driver is informed and the machine is entering a safe state.

A. Feature Diagram

The feature diagram provides the possibility to build a logical view of a PL on a higher level of abstraction. It consists of all features in the PL with their possible variations and relations. The diagram is based on the ideas presented in Matrix-based [15] and Graph-based [16] variability modeling techniques and it is extended, in order to explicitly depict safety-critical features. In Figure 2, a part of the feature diagram of the WLO Product Line is shown. Commonality and variability of the features are represented through the predefined stereotypes. For example, *Window Wiper System* is a mandatory feature for all products in the WLO PL and *CDC Steering System* is optional. The hierarchical structure of the features is represented through

a decomposition relation, which shows that CDC Steering System is realized by *CDC Control* and *CDC Supervision* features. If the optional CDC Steering System is configured to be included in the machine, then both CDC Control and CDC Supervision will also be included in order to ensure the correct operation, therefore they are depicted as mandatory features. Both features work in parallel and at specified intervals of time, they are synchronized. This behavior is depicted by the dynamic dependency *synergetic*. We consider also other types of dynamic dependencies - those that show sequential order of execution of the features, denoted with stereotype *serial* and also those that show features working in parallel, denoted with stereotype *collateral* [16].

The CDC Control feature can be realized either by a *Lever* for left-right steering or a *Joystick* for left-right steering and forward-backward driving. Both features are alternative, therefore they represented with *Alternative* stereotype and are related with the static constraint *excluded*. If the *Lever* feature is chosen the machines shall have a *Speed button* installed and this is depicted by the static constraint *required*.

In order to emphasize the safety-critical features we introduce the stereotype *Safety Feature* and add an attribute representing their criticality level. This allows practitioners to see how safety- and non-safety-critical the features related to each other and to analyze the impact of variability on functional safety.

B. Use Case Diagram

The use case diagram provides a structured representation of the possible usage scenarios of the machine functionality. The proposed diagram is based on the use case diagram proposed in PLUS [11]. We add the following elements to explicitly represent the safety-related information identified during the safety assurance process.

- *Hazard* and *Unresolved Hazard* represent possible hazards that may occur during the machine operation. Unlike for the hazards, there is no mitigation strategy introduced for the unresolved hazards at the current development stage. This informs the practitioners that further development is necessary to add new solutions for hazard avoidance.
- *Safety mechanism* corresponds to a mitigation strategy that is used to decrease the risk of certain hazards.
- *Criticality level* is an attribute for hazards and safety mechanisms to capture the criticality of hazards and the criticality level a safety mechanism is able to reduce.

Each scenario is analyzed in detail as a sequence of steps executed during the interaction of the operator with the developed system. Typical inputs to derive the use case scenarios are machine specifications and interviews with responsible system owners.

Figure 3 shows an example of a use case diagram illustrating the application of the elements. Generally we consider 2 possible machine scenarios a) *Pallet handling*

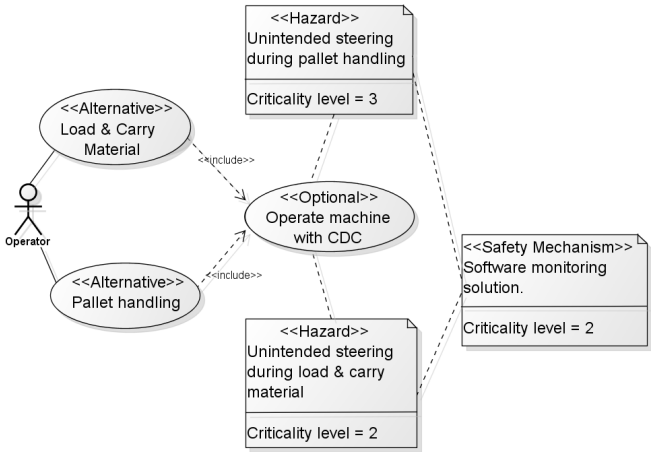

Figure 3. Use case diagram from the WLO PL example.

where lifting forks are attached and b) *Loading and carrying materials* when stones or gravel are moved. In the use case diagram, the usage of CDC in both scenarios is depicted by the *include* relationship. Because of the characteristics and purpose of the CDC Steering System, several hazards might occur in case of its malfunction, for example *Unintended steering*. Since there are differences in the operational environment in the two usage scenarios, two different criticality levels are identified for the hazard. Thus, we separate the representation of the hazard based on the usage scenario: *Unintended steering during pallet handling* with criticality level 3 and *Unintended steering during load and carry material* with criticality level 2, which are depicted in the diagram in elements with stereotype *Hazard*. Furthermore the already available risk reduction strategy *Software monitoring solution*, which is capable to decrease the criticality level with 2, is added. It is depicted in the diagram in element with stereotype *Safety Mechanism*. It is now visible that further safety mechanisms need to be added for the pallet handling case. This information can be used as a feedback for the performed safety analysis and in particular can help to review and improve the performed PHA.

C. State Machine Diagram

The state machine diagram enhances the analysis of the behavior of the systems in the PL in terms of system and sub-system states. We base our approach on the state machine diagram presented in PLUS [11]. Commonality and variability in the behavior of the different systems from the PL are represented through stereotypes and conditions. The alternative states are grouped in an element with a stereotype *Alternative Group*. This idea is based on the *VariationGroup* element, proposed in EAST-ADL [20], which is used to group variable elements. In order to support the safety analysis and certification, we add the following safety related

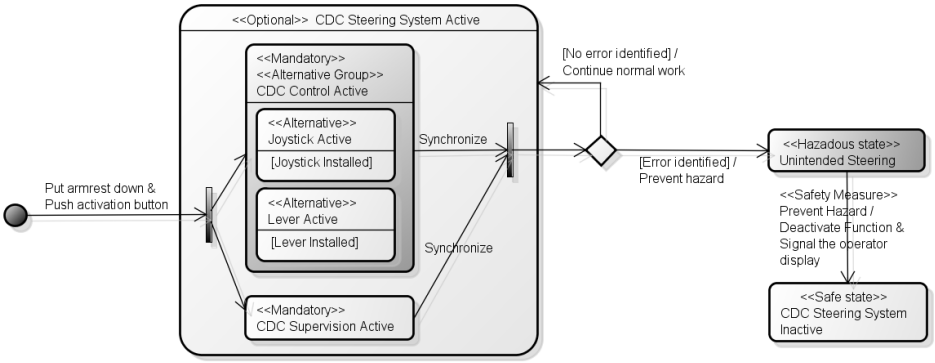

Figure 4. State machine diagram for the CDC Steering System feature from the WLO PL example.

elements:

- *Hazardous State* and *Unresolved Hazardous state* are used to represent the states of the system and its sub-systems which may lead to hazards. The conditions in which these states can be reached are presented on transition relations. Similarly to the use case diagram, the unresolved hazardous state represents hazardous state that has no related safe state.
- *Safe State* represents a state of the system, in which there is no unreasonable risk.
- *Safety Measure* shows the measures taken to transit the system from a hazardous to a safe state.

Through examination of the usage scenarios, described in Section IV-B, the states and transitions between them are determined. Unintended behavior of the system identified during the safety analysis is depicted in the diagram as hazardous states and an analysis of the propagation of the hazards and their impact on the related features can be performed. The diagram is further refined when safe states are identified and possible safety mechanisms are developed. Whether a certain safe state is reachable in a concrete configuration can be tested through a simulation of the constructed model.

In Figure 4, an example of a state machine diagram is shown. It represents the behavioral aspects of a machine which has CDC Steering System installed. The CDC feature can be in active or inactive state, which is depicted in the diagram as *CDC Steering System Active* and *CDC Steering System Inactive* states. In order to activate the CDC feature the armrest shall be put down and the activation button shall be pressed: *Put the armrest down and push the activation button*. When the CDC Steering System is activated, both CDC Control and CDC Supervision are activated in order to ensure the proper functioning. Both features work in parallel and are synchronized at predefined intervals to check the correctness of the calculated outputs. In case of wrong output of the CDC Control identified by the monitoring feature, the CDC is deactivated: *CDC Steering System Inactive*. The measures taken to set the system into a safe state are depicted on the transition, in this case the CDC feature is deactivated and the operator is notified about the error.

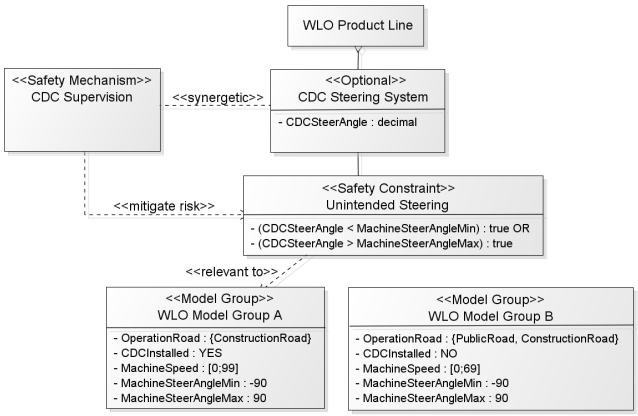

Figure 5. Safety configuration diagram for the CDC Steering System feature from the WLO PL example.

D. Safety Configuration Diagram

The idea of the safety configuration diagram is to represent the commonality and variability aspects in a PL from functional and machine model perspectives. We aim to visualize the conditions in which a hazard may occur for specific machine variants as well as possible safety mechanisms that mitigate the hazard. The features with their commonality and variability and dependencies extracted from the feature diagram are extended with functional parameters. In Figure 5, the functional parameter *CDCSteerAngle* of the optional feature CDC Steering System stores the current steering angle of the machine during operation.

The diagram contains the following elements:

- *Model Group* represents a group of products with similar configurations and add the related parameters, which correspond to the functional parameters of the PL features. We present two WLO model groups in Figure 5. The functional attribute *CDCInstalled* denotes whether a machine model group has CDC installed or not. Functional

attributes *MachineSteerAngle* and *MachineSteerAngleMax* specify the allowed boundaries of the steering angle for each machine model group.

- *Safety Constraint* is used to represent logical conditions, in which a hazard might occur in a relation to the parameters defined for each feature. If the steering angle calculated during the work of the CDC Steering System feature is less than the predefined minimum *MachineSteerAngleMin* or is greater than the predefined maximum *MachineSteerAngleMax* for a certain machine type, the hazard *Unintended Steering* may occur. This relation is denoted with *relevant to* stereotype. The logical conditions may contain rules related to several functions. This element can be used by the practitioners to analyze the conditions which should be taken into consideration when developing mitigation strategies to prevent the hazard. Furthermore, it also helps to derive test scenarios at later stages of development.
- *Safety Mechanism* corresponds to the functions that are introduced as mitigation strategies during development, in order to avoid certain hazards. The relation between safety mechanism functions and hazards is of type many-to-many: one or more hazards might be mitigated by one or more safety functions. The relation is denoted through a stereotype *mitigate risk*. In Figure 5, the safety mechanism CDC Supervision, which is introduced in order to avoid the hazard Unintended Steering, is shown.

V. Discussion of Approach

Based on the results gathered while modeling of the described example and the feedback from practitioners, we can draw the following conclusions for the benefits and limitations of our approach.

A. Benefits of the approach

In the following section we review how our approach is fulfilling the stated requirements.

R1. *Unified graphical representation*. Common stereotypes are used in all diagrams, which eases the creation of different views and enhances the readability of the model. The graphical representation eases the perception of the information and helps in the communication among the different specialists involved in the development.

R2. *Safety-related elements*. The artifacts derived during the development of the use case and state machine diagrams can be used as inputs for improving the safety analysis. Furthermore safety-related elements are captured and traced through all diagrams. Their impact on safety-related and non-safety-related features can be analyzed and corresponding safety mechanisms can be highlighted.

R3. *Commonality and variability*. Commonality and variability is represented in all diagrams and provides the possibility to analyze how a feature impacts the functional safety and whether configurations are valid and safe. Furthermore, the derived information is used as a guideline in the design of configurable architectures during later development stages.

R4. *Direct and indirect dependencies*. Dependencies between features are identified during the development of the Use case and State machine diagrams. This information is necessary for change management, evolution of the PL and considering functional safety. Capturing feature dependencies allows to specify constraints on potential configurations and later design decisions. Feature dependencies provide the opportunity to further establish automated analysis of violations of safety goals. Moreover, feature interactions and dependencies may lead to recognition of specific hazards that are otherwise hard to identify.

R5. *Multiple modeling views*. The approach allows to capture different aspects of the PL and allows separation of concerns using different modeling views. We could observe that this reduces the complexity and is easier to understand by practitioners.

R6. *Traceability*. Traceability between the development elements is achieved through their common names and stereotypes through the different views. The traceability of safety-related elements represented in the different diagrams helps in the process of building evidence for achieving functional safety certification of the configurations.

B. Limitations of the approach

During the case study we identified several possible limitations in applying the modeling approach directly in a real project. In order to properly build the diagrams and produce a correct model, the practitioners should have background knowledge in model-based development. This may not always be the case. Our approach uses different development artifacts, gathered during product specification and safety analysis as an input. This information can be difficult to extract and efforts are necessary to capture the correct information. The example is very simplistic and cannot ensure that the model is scalable and manageable in complex systems. Moreover, the example is taken from a particular safety-critical domain, thus it does not give the possibility to assess whether the approach is applicable in other domains.

VI. Conclusion

The number of functions and variety of technologies increases the complexity of the systems in the construction equipment domain. Customer needs shall be fulfilled at the same time machine safety shall be assured. Several techniques for variability management are proposed in literature. We have created important requirements a variability management method should satisfy to manage functional safety in a safety-critical PL. Furthermore, we describe and evaluate three main categories of variability management methods and none of the examined techniques is able to fulfill all stated requirements. In this work we

focus on modeling of the PL during concept phase as it is important to build a comprehensive concept to guide the further development steps. Use case, state machine, feature and safety configuration diagrams are proposed to capture commonality and variability aspects of PL from different points of view with focus on the safety dimension of the PL. The modeling approach is applied on a realistic example from the construction equipment domain to show its essential properties and how the identified requirements are achieved. We finalize by discussing the benefits and limitations of our approach.

VII. Future work

Currently our concept considers variability in the context of safety-critical PL in the construction equipment domain. In order to evaluate its universal coverage the technique should be applied on multiple examples from different domains. Furthermore, to ensure the scalability of our approach, a more complex industrial case should be considered. The result from these studies might arise the need of adding more elements in the diagrams or considering possible changes in the representation of the model, in order to increase its usability and applicability.

Since the presented approach is focused only on a higher level of abstraction, additional research will be conducted to find an effective way to model the PL for later development stage. Furthermore, there is a possibility to enrich the model with more diagrams on the same level and thus achieve more complete representation of the systems. In order to improve the applicability it is useful to develop a tool, which is able to extract the safety-related elements and analyze the consistency of the models.

Acknowledgment

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement no 295373, Vinnova and the KKS-funded ITS-EASY Post Graduate School for Embedded Software and Systems.

References

[1] K. Pohl, G. Böckle, and F. J. Linden, *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc., 2005.

[2] A. Pretschner, M. Broy, I. H. Kruger, and T. Stauner, "Software engineering for automotive systems: A roadmap," in *2007 Future of Software Engineering*. IEEE Computer Society, 2007, pp. 55–71.

[3] *ISO 26262 Road vehicles – Functional safety*, International Organization for Standardization Std., 2011.

[4] *IEC 61508 functional safety of electrical/ electronic/ programmable electronic safety-related systems*, International Electrotechnical commission Std., 2010.

[5] *ISO 15998 Earth-moving machinery – Machine-control systems (MCS) using electronic components – Performance criteria and tests for functional safety*, International Organization for Standardization Std., 2008.

[6] S. Baumgart, J. Froberg, and S. Punnekkat, "Industrial challenges to achieve functional safety compliance in product lines," in *Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference on*, 2014, pp. 356–360.

[7] I. Petrova and A. Salikiryaki, "Development of a variant-management for safety-critical product lines," Master's thesis, Mälardalen University, To appear 2015.

[8] M. A. Babar, L. Chen, and F. Shull, "Managing variability in software product lines," *Software, IEEE*, vol. 27, no. 3, pp. 89–91, 2010.

[9] M. Sinnema and S. Deelstra, "Classifying variability modeling techniques," *Information and Software Technology*, vol. 49, no. 7, pp. 717–739, 2007.

[10] I. Habli and T. Kelly, "A safety case approach to assuring configurable architectures of safety-critical product lines," in *Architecting Critical Systems*. Springer, 2010, pp. 142–160.

[11] H. Gomaa, *Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures*. Addison Wesley Longman Publishing Co., Inc., 2004.

[12] L. Chen, M. Ali Babar, and N. Ali, "Variability management in software product lines: A systematic review," in *Proceedings of the 13th International Software Product Line Conference*, 2009.

[13] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (FODA) feasibility study," DTIC Document, Tech. Rep., 1990.

[14] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh, "Form: A feature-oriented reuse method with domain-specific reference architectures," *Annals of Software Engineering*, vol. 5, no. 1, pp. 143–168, 1998.

[15] H. Ye and H. Liu, "Approach to modelling feature variability and dependencies in software product lines," *IEEE Proc. Softw.*, vol. 152, pp. 101–109, 2005.

[16] Y. Lee, C. Yang, C. Zhu, and W. Zhao, "An approach to managing feature dependencies for product releasing in software product lines," in *Proceedings of the 9th International Conference on Reuse of Off-the-Shelf Components*, 2006.

[17] C. Maga and N. Jazdi, "An approach for modeling variants of industrial automation systems," in *Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on*, vol. 1. IEEE, 2010, pp. 1–6.

[18] M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch, "Covamof: A framework for modeling variability in software product families," in *Software Product Lines*. Springer, 2004, pp. 197–213.

[19] C. A. Ericson, *Hazard Analysis Techniques for System Safety*. Wiley, 2005.

[20] H. Blom, H. Lönn, F. Hagl, Y. Papadopoulos, M. O. Reiser, C. J. Sjöstedt, D. J. Chen, F. Tagliabò, S. Torchiaro, S. Tucci *et al.*, "EAST-ADL: An Architecture Description Language for Automotive Software-Intensive Systems - White Paper," Maenad Project, Tech. Rep., 2013.