Software Product Line Engineering for Safety-critical Systems

Philipp Lohmüller and Bernhard Bauer
Institute of Computer Science, University of Augsburg, Universitätsstr. 6a, 86159 Augsburg, Germany

Keywords: Software Product Line, Safety-critical Systems, Complexity Reduction, Dependability, Multi-concerns.

Abstract: Nowadays, modern cars can be configured by means of a wide range of software configuration options. In this context, we speak about Software Product Lines (SPLs). In almost every modern automotive vehicle safety-critical components like, e.g., an Adaptive Cruise Control (ACC) are installed. Some SPLs have different safety-critical requirements whereas other SPLs have similar requirements. This paper proposes an approach for the reduction of the complexity of SPLs without loss of safety (aspects) for all participants. For this purpose, a concept has been developed, which clusters products of SPLs with similar safety-critical requirements, i.e., the set of products of an SPLs, which must still be tested, can be reduced immensely. The paper also provides an application example how the reduced set can be used in order to perform a Safety, Security and Timing (SST) based trade-off analysis.

1 INTRODUCTION

In the past few years, safety and security is playing an increasingly important role in various domains like, the automotive domain. Neglecting safety and security issues may have fatal consequences. Looking back at the end of the 1960s, the Ford Motor Company had the objective to develop a fuel-efficient and cost-effective car, which should be introduced onto the market at the beginning of the 1970s. Due to the short time frame, safety checks were skipped and the resulting damages were much higher than the supposed cost savings (Ordóñez et al., 2009). In order to avoid such problems, corresponding counteractive measures were introduced. However, individual safety and security issues or counteractive measures can exclude each other. In this regard, it is necessary to determine the best trade-off. This approach has already been presented in (Lohmüller et al., 2018). Moreover, modern vehicles can be configured by means of a construction kit. In this context, there are several millions of configuration sets for the configuration of a new car. Each configuration set complies with a product of an SPL and thus different characteristics. SPLs have advantages and disadvantages: An advantage of an SPL is variability management (Buchmann and Greiner, 2018) and that you can configure an automotive vehicle individually. (Pohl et al., 2005) However, in theory each configuration set is accompanied with immense hazards with respect to safety and security issues, thus it is necessary to check each configuration/product with respect to its safety and security. Therefore this paper presents an approach how to reduce complexity of SPLs taken safety, security and timing issues into account. This is achieved by eliminating non safety-critical issues as well as building equivalence classes on semantically equivalent elements from the SPL and thus to narrow the number of configuration sets.

First, Sec. 2 gives an overview over the necessary basics. In the following section a case study is introduced in order to exemplify the approach of this paper. The approach dealing with the reduction of SPLs complexity is presented in Sec. 4. The evaluation proves that the approach is a coherent and versatile concept. Sec. 6 points out the novelty of the approach and its relation to existing work. Finally, the paper summarizes the results of this paper and gives an outlook for future work.

2 BASICS

Since this paper aims at safety-critical systems we will have a closer look on the necessary theoretical foundations. Both, safety and security must be avoided preventively, not to endanger human life. In general, safety means accident prevention whereas security means crime prevention. However, security failures can lead to safety issues, e.g. a hacked autopilot. As already mentioned in Sec. 1 safety and secu-

Lohmüller, P. and Bauer, B.
Software Product Line Engineering for Safety-critical Systems.
DOI: 10.5220/0007246102090216
In *Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2019)*, pages 209-216
ISBN: 978-989-758-358-2
Copyright © 2019 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

rity aspects are often contradictory. On the one side, safety ensures that human life and environment is protected against potentially hazardous machines. On the other side, security makes sure that human beings cripple the machinery. (Springer, 2016) Let's take an example from the automotive environment: Modern vehicles are equipped with one or more airbags in order to preserve human life (safety). Moreover, it must be assured that the functionality of an airbag can not be triggered remotely by third parties (security). In order to protect releasing the airbag, other measures are necessary, like, e.g., a secure encryption algorithm. However, if the encryption takes to much time, this can cause new problems, e.g., delayed triggering (timing) of an airbag which endangers human life and thus safety. As it can seen, safety, security and timing may be mutually exclusive.

For this purpose, (Lohmüller et al., 2018) developed an approach how to calculate an optimal preventative trade-off between (partly) conflicting (multi-) concerns including safety, security as well as timing. A quick view on the approach delivers the following:

1. Devise potential alternative solutions, i.e., individual decision options by which the trade-off is calculated.
2. Identify failure modes and transfer it into a structured Safety Goal Hierarchy Model (SGHM) as described in (Lohmüller et al., 2018). The top-level goal represents a safety goal since it is the main objective to obtain a system, which is acceptably safe. The leaves represent the Single Point of Failures (SPOFs), which are essential for the next step.
3. Perform a Failure Mode and Effects Analysis (FMEA) for all SPOFs.
4. Perform a Multi-Criteria Decision Analysis (MCDA) to calculate the optimal trade-off.

As already indicated in Sec. 1, SPLs enable many ways of configurations, e.g., configurations of a car. Since there are different definitions of an SPL, a configuration set or configuration option is meant in this paper, when speaking about SPLs in this paper. Usually, the different features of SPLs are modeled by a Feature Model (FM). FMs, which have been first introduced in the Feature-Oriented Domain Analysis (FODA) (Kang et al., 1990) contain all necessary features, i.e., all configuration options. Thereby, an FM is modeled and structured hierarchically. For each (sub-)feature there are several characteristics: {*abstract*, *concrete*}, {*mandatory*, *optional*}, {*or*, *xor*} and {*requires*, *excludes*}. With exception of the latter tuple, one characteristic of each tuple must be selected for each (sub-)feature. A detailed description of the individual characteristics can be found in (Lee et al., 2002). Fig. 1 shows an exemplary FM for configuring assistance systems (legend: *abstract feature*, concrete feature). Thereby, you can choose between five assistance systems whereas Emergency Brake Assist (EBA) is mandatory. If an ACC is configured, the Cruise Control (CC) may not be selected. The ACC provides two options: either with a maximum speed of 160 km/h or 210 km/h. Furthermore, if the Lane Assist (LA) is selected, the Lane Departure Prevention (LDP) must be selected as well. The LDP provides two options of which at least one must be selected: an acoustic and visual warning.

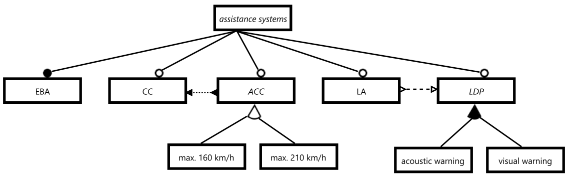

Figure 1: Exemplary FM.

3 CASE STUDY

In this section, a case study is presented to show the approach of the paper, which will be elaborated in the next section. Modern automotive vehicles are equipped with assistance systems, like an ACC or a LDP. An ACC operates like a cruise control, but adapts the speed depending on distance of the vehicle driving ahead. Thereby, modern sensor types like radar (Abou-Jaoude, 2003) or video based systems (Murray and Jackson, 2006) are used. An LDP system adjusts the steering behavior as well as ac-/decelerating to keep the car in its lane. Therefore, different sensors like an infrared sensor or video based system are used in order to detect lane markers (Litkouhi, 2012). From a functional point of view, ACC and LDP are independent from each other. Thus, one could assume that ACC as well as LDP could be tested and analyzed independently from each other since they are not linked in any way. However, this assumption has not been scientifically proven yet, that's why further profound research is necessary. First, an FM is needed covering the entire SPLs of the configurable car. Thus, this FM contains an abstract feature *assistance systems* including the concrete subfeatures *ACC* as well as *LDP*. Since the feature *assistance systems* is optionally, one sub-feature must be selected at least. Fig. 2 shows an excerpt from the FM.

All safety-critical features are tagged with a boolean flag for SST (legend: safety, security, tim-

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

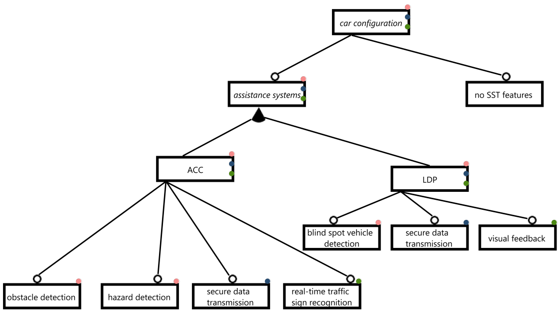

Figure 2: Excerpt from an FM of a car configuration.

ing) in order to trace the individual concerns during the whole algorithm. Thus, the FM serves as a central model of control, in which the detailed SST information of the individual *assistance systems* are stored. Since the assumption was made that the *ACC* as well as the *LDP* could be tested separately, an accurate review is mandatory checking which system components of the features *ACC* and *LDP* are concerned in each case. For this purpose, it is essential to look into the underlying System Model (SM), which is depicted in Fig. 3. It shows dependencies between the different *sensor types* and *assistance systems*.

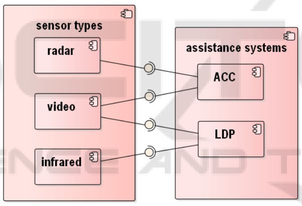

Figure 3: Abstract SM of the case study.

Thereby, one will realize that the *ACC* uses the components *radar* and *video* whereas the *LDP* uses the components *infrared* and *video*. Since the component *video* is used from both, the *ACC* and the *LDP*, a separate testing between *ACC* and *LDP* is not reasonable, i.e., *ACC* depends on *LDP* and vice versa. Hence, it is the main objective to find out these features in the FM having a similar level of SST. For this purpose, a reachability tree is created with the same hierarchical structure as the source FM. The values of the nodes correspond to

1. the SST values of the FM (legend: T = true, F = false) and
2. the Equivalence Class (EC) of the corresponding feature that is equated with the last superordinated abstract feature.

For instance, $\{\{TTT\}, AS\}$ means that safety, security as well as timing is assigned to *true*. The second value of the node is assigned to the EC *assistance systems*.

By means of depth-first search, all the distinct paths of the reachability tree are chosen. The paths may not contain the *FFF* value since *FFF* reflects non safety-critical issues or features. Fig. 4 shows the underlying reachability tree of the case study. After the depth-first search, the green colored paths must be tested and analyzed in further steps. The orange colored nodes are duplicated since they are already included in other paths and are assigned to the same EC than other ones. The red colored items are non safety-critical and must not be considered anymore.

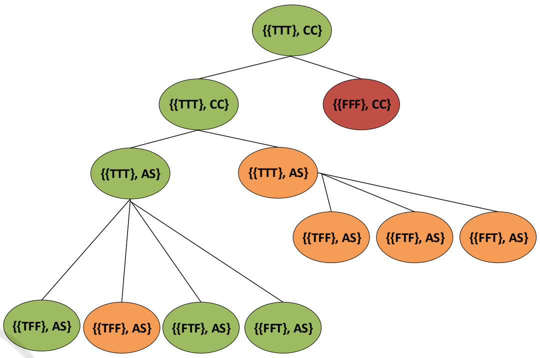

Figure 4: Reachability tree of the application example with SST tags, equivalence classes and paths to be checked.

As can be seen, by means of the reachability tree of Fig. 4 *LDP* has been eliminated for further testings and analyses, i.e., the complexity has been reduced to the green paths. This is justified because both, the *ACC* as well as the *LDP* are assigned to the same EC. Moreover, the *ACC* and the *LDP* have a common used component: the *video* based sensor. Furthermore, the *ACC* uses a *radar* sensor whereas the *LDP* uses an *infrared* sensor. It can thus be concluded that the *radar* sensor has similar SST requirements as the *infrared* sensor, although they are designed and implemented differently. It is hence sufficient to test and analyze the *ACC* paths. In this specific case study 3 of 8 paths need to be elaborated in further steps. Consequently, a complexity reduction of *62,5%* has been guaranteed. Since the reachability tree is structurally identical with the corresponding FM, the relevant features can be determined easily. Fig. 5 shows the reduced FM whereas non relevant (SST) features have been grayed out. This FM can now be used as a basis for further safety and security analysis.

For instance, the reduced FM may be used subsequently in order to perform a trade-off analysis as suggested by (Lohmüller et al., 2018). In this context, there is a linking between the features of the reduced FM and the goals/SPOFs of the SGHM in order to eliminate non relevant goals/SPOFs from the SGHM. Thus, the optimal trade-off can be calculated by concerning less goals/SPOFs but taking all safety-

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

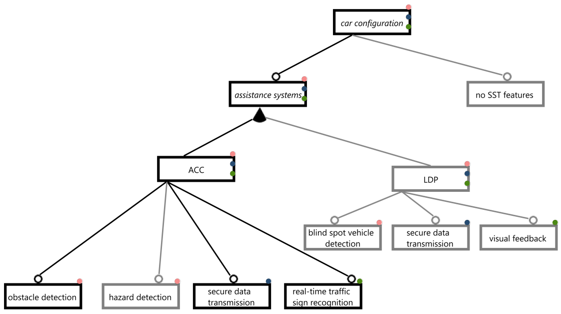

Figure 5: Reduced FM with relevant SST features.

critical requirements into account. Consequently, the complexity can also be reduced since less calculation steps are needed. Fig. 6 shows the matched SGHM. The meaning of the colors is analogous to Fig. 4.

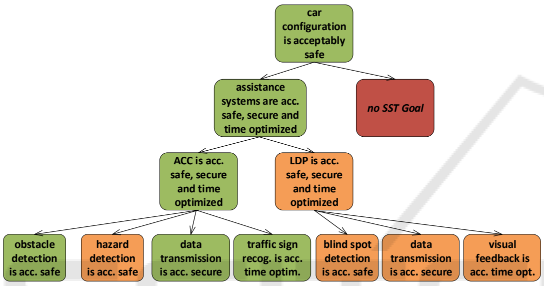
car configuration is acceptably safe
assistance systems are acc. safe, secure and time optimized
no SST Goal
ACC is acc. safe, secure and time optimized
LDP is acc. safe, secure and time optimized
obstacle detection is acc. safe
hazard detection is acc. safe
data transmission is acc. secure
traffic sign recog. is acc. time optim.
blind spot detection is acc. safe
data transmission is acc. secure
visual feedback is acc. time opt.

Figure 6: Simplified SGHM of the application example.

4 APPROACH

It is aim of the current section to present the concepts and theoretical backgrounds generally so that the approach is applicable in various domains and application cases, e.g., automotive, avionics or railway. For this purpose, a concept picture is shown in Fig. 7, which represents the entire approach in an abstract manner .

Step 1. As already mentioned in Sec. 2, FMs have been established for the realization of SPLs since they have a well-defined hierarchical structure by nature. The central FM serves as a basis for all further steps, i.e., each of these steps is performed by means of the FM. FMs usually map entire systems, though only parts of them shall be tested or analyzed normally since they are relevant for individual departments. Due to the fact that in this paper SPLs are applied in safety-critical systems it is mandatory to annotate a boolean SST flag for each feature. This enables providing information about which kinds of safety-critical concerns are involved. However, despite the advantages of feature modeling, the complexity of such FMs concerning safety, security and timing issues is hard to deal with (Pohl et al., 2013). Therefore, the complexity has to be reduced and thus the effort is manageable.

Step 2. In the previous step, features of the FM have been selected, which are relevant for individual departments. It has not been considered that some features have further dependencies which are not apparent at first sight. That's why the approach links the FM and the underlying already existing SM. The latter is realized similar to an UML component diagram. Thus, it is determined whether there are additional subcomponents of the individual system components of the FM, which must be considered as well (Li et al., 2018). The link is provided between the individual (abstract) features and the system components. Since the linking is bidirectional, the selection of the corresponding features in the FM are updated according the dependencies of the system components and their subcomponents in the FM. In summary, step 2 extends the selection of the features (of step 1) since there are commonly used system components.

Step 3. After updating the FM according to the dependencies with the SM the actual reduction of complexity is performed. For this purpose, a reachability tree is created, which maps the same hierarchical structure of the FM to a new tree structure containing all SST annotations including all possible paths of the SPLs. The reachability tree is needed since it contains only relevant information about the FM, which is necessary for further calculations. Thus, overhead is reduced and the upcoming algorithmic processes are facilitated. The values of the nodes are assigned to SST annotations as well as the EC of the corresponding features. As already mentioned in Sec. 3 the individual ECs refer to the corresponding super-ordinated abstract feature in the FM. All elements within an EC, i.e., features are equivalent to each other since these elements comply with reflexivity, symmetry as well as transitivity (Guta and Kiukas, 2015). It is now essential to point out all distinct SST requirements. Essentially, this means that all different paths must be found starting with the root node. Let us assume there are paths in the FM, i.e., SPLs like the following:

1. *car configuration* → *assistance systems* → *ACC* → *detection of maximum permitted speed*
2. *car configuration* → *assistance systems* → *LDP* → *visual warning signal*

For both SPLs, the path in the reachability tree may be the following: $\{\{TTT\}, CC\} \rightarrow \{\{TTT\}, CC\} \rightarrow \{\{TTT\}, AS\} \rightarrow \{\{TFF\}, AS\}$. Therefore, it can

SCIENCE AND TECHNOLOGY PUBLICATIONS

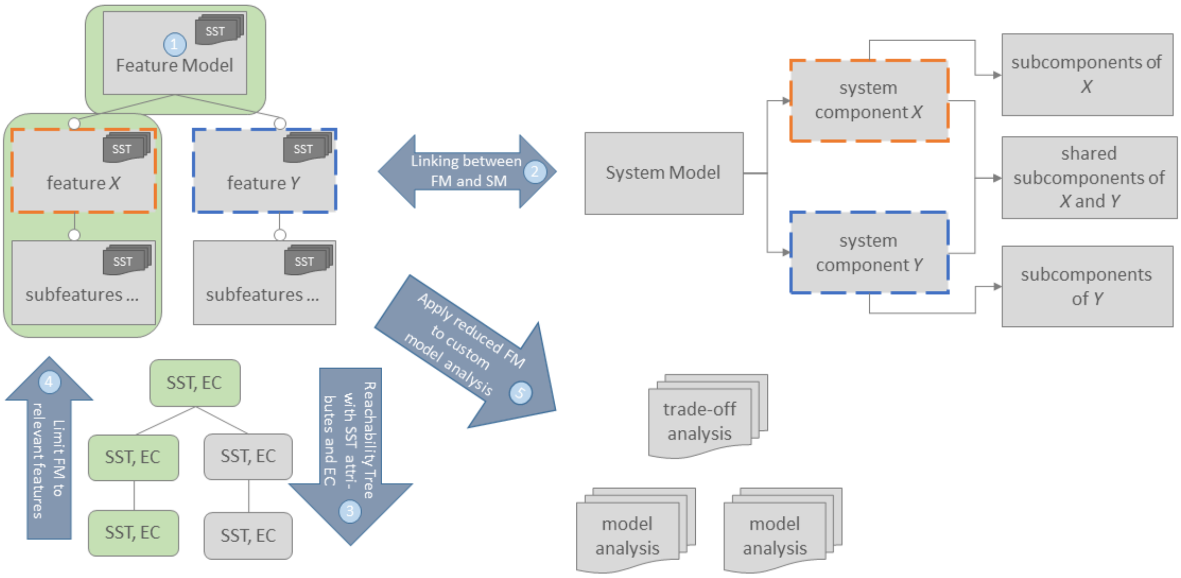

Figure 7: Abstract concept picture of the approach.

be concluded that the *detection of maximum permitted speed* as well as *visual warning signal* have similar requirements since both elements are within the same EC (*assistance systems*) and are just safety-critical, i.e., it is sufficient to test and analyze just one path. In order to provide all distinct paths, a depth-first search algorithm is applied for the entire reachability tree. There are two termination conditions for the calculation of each path: The current node is

1. non safety-critical, i.e., the SST value is assigned to *FFF* or
2. a leaf, i.e., there are no further nodes that must be checked for this path.

By means of the depth-first search it is ensured that all nodes have been considered and duplicate paths are eliminated. Since FMs and thus also reachability trees may be very extensive, it is important not to neglect the complexity of this algorithm. If there is a reachability tree *RT* with $|V|$ nodes and $|E|$ edges the complexity of this algorithm is (Naumov et al., 2017):

$$DFS(RT) \in O(|V| + |E|)$$

Thus, the complexity is linear and acceptable in time, i.e., no optimizations are necessary. In conclusion, it is clarified to which degree the reduction of complexity is performed. It is dependent on two factors:

- total number of leaves: $|\lambda(RT)|$
- number of distinct leaves with different predecessors: $|\delta(RT)|$

Thus, the following applies for $\delta(RT)$:

$$\forall x, y \in \delta(RT) : \pi(x) \neq \pi(y)$$

whereas $\pi(x)$ defines the set of predecessors on the basis of node x. Finally, the degree of complexity reduction *CR*, is defined as follows:

$$CR(RT) = \frac{|\lambda(RT)| - |\delta(RT)|}{|\lambda(RT)|}$$

Step 4 and 5. After step 3, the reachability tree contains all paths, which are safety-critical and must be processed anymore. To facilitate this, it is mandatory to mark the corresponding features in the FM. The complexity of the FM and thus safety-critical SPLs is now reduced and can be used as a basis for model-based calculations, e.g., trade-off optimization as shown in Sec. 3.

5 EVALUATION

The approach is evaluated by means of a scenario based evaluation, i.e., the concept is inspected with respect to some quality attributes on basis of selected scenarios. The approach is measured by means of the following quality attributes:

- *interoperability*: The emphasis is placed on compatibility with other components or systems. When fulfilling interoperability, the main functionality of the concept of this paper, i.e., complexity reduction, is accomplished as well, since it is based on the interaction between different models and components.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

- *maintainability*: The approach of this paper will be investigated in terms of changeability and testability.
- *adaptability*: Finally, it is analyzed whether the concept is expendable and adaptable.

At this point, it must be mentioned that the concept of this paper has been realized in the form of an Eclipse plugin by applying EMF[1] and Sirius[2]. The plugin consists of three graphical editors, which enable modeling of FMs, SMs and reachability trees. Each of them are developed in EMF and Sirius, i.e., there are three underlying *.ecore meta models at all. Due to the same modeling environment it enables linkings or transformations between *FM ↔ SM*, *FM ↔ reachability tree* and *FM → custom models or analyses* (remember Fig. 7). The logic of the real complexity reduction is realized in the editor of the reachability tree.

The evaluation will prove that all the quality attributes, which have been listed above, are fulfilled by the approach of this paper. Thereby, the following scenarios have been selected:

1. requirements will be changed
2. a feature of the FM is added, deleted or modified
3. the structure of the SM will be changed
4. the modeling type of the SM is changed
5. consideration of further or other concerns/attributes

The development in industry is proceeding forward with giant strides, thus the requirements may constantly change. When changing the requirements, new features will be added or existing features will be deleted or modified. Moreover, the SM will be updated in a parallel process as well. Since the reachability tree maps the structure of the FM, it will be updated automatically as well. Thus, the *interoperability* between these models and components is fulfilled. Next, changeability must be proven. Since this scenario make modifications in the FM and the maintenance of the functionality has already been proven for the quality attribute *interoperability*, changeability is guaranteed. Each complex system has to be tested in order to avoid errors. As already indicated, the SM is redundant and the structure of the FM is included in similar but another form in the reachability tree. Thus, the similar models can be tested against each other for validity and correctness. Hence, it has been shown that testability is fulfilled as well. Taking into account that changeability and testability is met, *maintainability* is achieved. The modification of the FM affects just the EMF/Sirius instance model but not the EMF *.ecore model, i.e., the meta model. The FM instance model is arbitrary *adaptable* and expandable by new functionality like, e.g., adding new features. In summary, all quality attributes apply to this scenario.

Scenario #1 implies scenario #2 and #3. In this way, scenario #2 and #3 has already been proven since scenario #1 has been evaluated.

An SM can be represented by various notations like, e.g., component diagram, composite structure diagram or object diagram. The interoperability depends on the interaction between FM, SM and the reachability tree. For this purpose, the linking between the features of the FM and the components or objects of the new SM type must be set. Thus, the complexity reduction is ensured and *interoperability* is fulfilled. When changing the type of modeling an SM, one type of error might occur: The linkings to the corresponding features have not been set correctly. The topic of this scenario is that the type of SM modeling can be changed, i.e., changeability is fulfilled. By analogy with scenario #1, testability is accomplished since the information of the FM are contained in the reachability tree as well. Furthermore, the SM usually provides redundant design. This means that *maintainability* is fulfilled by changeability and testability. In keeping with scenario #1, changing modeling type of the SM only relates to instance model of EMF/Sirius, not to the EMF meta model. That's why, the approach is also expendable and adaptable, i.e., *adaptability* is met. It has been evaluated that this scenario fulfills all quality attributes.

The concept of this paper considered the concerns SST. The following scenario will demonstrate that the approach is also combinable with other (additional) concerns or attributes, e.g., reliability. Based on this new concern, the linkings between the features and the components of the SM may be updated. The new concern reliability is annotated in the FM, this adjustment also updates the reachability tree since it depends on them. Consequently, *interoperability* is fulfilled. Even though, the considered concerns will be modified, it is possible to modify the FM and SM. This point has already been shown in scenario #1-3, thus changeability is ensured. Testability is fulfilled as well since the FM and SM is usually designed redundantly usually. That's why *maintainability* is guaranteed. This scenario covers adapting the concerns, i.e., *adaptability* is fulfilled.

Tab. 1 shows an overview of the quality attributes and the individual scenarios as listed at the beginning of this section. As it can be seen, all quality attributes

[1] https://www.eclipse.org/modeling/emf/

[2] https://www.eclipse.org/sirius/

Table 1: Overview of the evaluation.

	#1	#2	#3	#4	#5
interoperability	✓	✓	✓	✓	✓
maintainability	✓	✓	✓	✓	✓
portability	✓	✓	✓	✓	✓

are fulfilled by each scenario, i.e., the concept of this paper was successfully proven.

6 RELATED WORK

In this section related publications and projects will be presented and compared with the approach of complexity reduction of SPLs on safety-critical systems.

6.1 Safety-critical Concerns

The MERgE project[3], which has been founded by the ITEA enhances interactions between multi-concerns. It focusses on safety and security, especially on maintenance of quality characteristics during the product lifecycle. These include relationships between them, legacy management, cost reduction and robustness. This project considers various domains, among them radio communication, automotive and space industry. (Robinson et al., 2016) However, it has not been considered that also timing plays an important role in the context of safety-critical systems. The MERgE project covers maintenance of safety and security in the product lifecycle but not in SPLs. Consequently, the reduction of SPL complexity has not been explained. There are a large number of standards and norms, which have to be fulfilled in a safety-critical environment. That's why (Brunner et al., 2017) developed an approach how to model, document and integrate safety and security requirements. This approach doesn't consider the timing concern, just safety and security. Furthermore, there is no SPL used. Moreover, there is no complexity reduction in context of equivalent model components. Since SST requirements are often in contradiction to each other, (Lohmüller et al., 2018) developed a methodology how to calculate an optimal trade-off. Thereby, a Multi-Criteria Decision Analysis algorithm as well as the Goal Structuring Notation - a structured argumentation notation - is used to solve the conflicts. However, integrating SPLs as well as the complexity reduction by forming equivalence classes is not part of this work. (Lohmüller et al., 2018) In all the mentioned publications with safety-critical context the complexity reduction of SPLs does not seem to have been evaluated scientifically before.

6.2 Software Product Lines

The work of (Pohl et al., 2005) serves as baseline for SPLs. In this publication, definitions and scopes of SPLs are presented more closer to the reader. However, there is no reference to SST concerns as well as complexity reduction of SPLs. In (Pohl et al., 2018) the complexity of FMs is measured and analyzed by means of worst case execution analyses in order to improve state-of-the-art analysis tools. The approach improves complexity in the context of state-of-the-art analysis tools whereas this paper is focused on complexity reduction in safety-critical environment. Furthermore, there is no proposal for the creation of equivalence classes in order to cluster semantically similar features. (Li et al., 2018) propose an approach to reduce the complexity of FMs. In this paper, the complexity is reduced by linking the FM with the SM. However, in our paper equivalence classes are built in order to reduce complexity. In this way, complexity can be reduced even further. Li et al. achieved a reduction of 40% whereas the approach of our paper could reach over 60% in case of our case study. Moreover, we considered safety-critical requirements, like, e.g., SST whereas Li et al. only considered economical requirements. In (Hitesh and Kumari, 2018) a concept has been proposed to optimize feature selection in order to choose the elements that will be reused most commonly. This procedure is applicable if economic goals, like, e.g., costs play an important role. However, in our paper safety-critical requirements must be preferred. In this context, it may have fatal consequences if only most commonly reused features will be considered.

7 CONCLUSION AND OUTLOOK

In this paper, an approach has been presented to reduce complexity of Feature Models with consideration of Safety, Security and Timing concerns. In this context, it has been explained that the complexity reduction depends on system components of the System Model with respect to Safety, Security and Timing concerns. Furthermore, an algorithmic has been developed for detecting substructures of an Feature Model that have similar SST requirements. For that purpose, a reachability tree has been used to find similar substructures. It is also applied for finding suitable equivalence classes in order to cluster related features. In conclusion, the Feature Model is reduced accord-

[3] https://itea3.org/project/merge.html

ing to Safety, Security and Timing constraints and can be used for further model analyses. The evaluation showed that the modification of features is possible. For future work, it might be feasible to integrate a change impact analysis. Thus, it would be possible to determine whether a feature or set of features is profitable with regard to a level of complexity reduction.

REFERENCES

Abou-Jaoude, R. (2003). ACC Radar Sensor Technology, Test Requirements, and Test Solutions. *IEEE Transactions on Intelligent Transportation Systems*, 4(3):115–122.

Brunner, M., Huber, M., Sauerwein, C., and Breu, R. (2017). Towards an Integrated Model for Safety and Security Requirements of Cyber-Physical Systems. In *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 334–340, Prague, Czech Republic.

Buchmann, T. and Greiner, S. (2018). Managing Variability in Models and Derived Artefacts in Model-driven Software Product Lines. In *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development - MODELSWARD*, pages 326–335, Funchal, Madeira, Portugal.

Guta, M. and Kiukas, J. (2015). Equivalence Classes and Local Asymptotic Normality in System Identification for Quantum Markov Chains. *Communication in Mathematical Physics*, 335(3):1397–1428.

Hitesh and Kumari, A. C. (2018). Feature Selection Optimization in SPL using Genetic Algorithm. *Procedia Computer Science*, 132:1477–1486.

Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., and Peterson, A. S. (1990). Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical report, Carnegie Mellon University - Software Engineering Institute.

Lee, K., Kang, K. C., and Lee, J. (2002). Concepts and Guidelines of Feature Modeling for Product Line Software, Engineering. In *Proceedings of the 7th International Conference, ICSR-7*, Austin, USA.

Li, M., Grigg, A., Dickerson, C., Guan, L., and Ji, S. (2018). A Product Line Systems Engineering Process for Variability Identification and Reduction. *ArXiv e-prints*.

Litkouhi, B. B. (2012). Lane Departure Warning and Change Assist System Utilizing Active Materials. United States Patent. Patent No.: US 8,111,147 B2.

Lohmüller, P., Fendt, A., and Bauer, B. (2018). Multi-Concerns Engineering for Safety-Critical Systems. In *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development - MODELSWARD*, pages 504–510, Funchal, Madeira, Portugal.

Murray, P. and Jackson, D. (2006). Camera Technique for Adaptive Cruise Control (ACC) Sensor Adjustment. United States Patent. Patent No.: US 7,121,011 B2.

Naumov, M., Vrielink, A., and Garland, M. (2017). Parallel Depth-First Search for Directed Acyclic Graphs. In *Proceedings of the Seventh Workshop on Irregular Applications: Architectures and Algorithms*, pages 4:1–4:8, New York, NY, USA.

Ordóñez, L. D., Schweitzer, M. E., Galinsky, A. D., and Bazerman, M. H. (2009). Goals Gone Wild: The Systematic Side Effects of Over-Prescribing Goal Setting. Working Paper.

Pohl, K., Böckle, G., and van der Linden, F. (2005). *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer.

Pohl, R., Höchsmann, M., Wohlgemuth, P., and Tischer, C. (2018). Variant Management Solution for Large Scale Software Product Lines. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, pages 85–94, New York, NY, USA.

Pohl, R., Stricker, V., and Pohl, K. (2013). Measuring the Structural Complexity of Feature Models. In *Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering*, pages 454–464, Silicon Valley, CA, USA.

Robinson, C. R., Pequery, J., and Michiels, S. (2016). MERgE: Technology Advancement for Cohesion of Concerns in System Engineering. In *Proceedings of CERTS 2016: The 1st Workshop on Security and Dependability of Critical Embedded Real-Time Systems*, pages 25–30, Porto, Portugal.

Springer, M. (2016). Was ist der Unterschied zwischen Safety und Security? https://www.tuev-nord.de/explore/de/erklaert/was-ist-der-unterschied-zwischen-safety-und-security/. accessed June 8th, 2018.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS