Safety, Security, and Configurable Software Systems: A Systematic Mapping Study

Andy Kenner
METOP GmbH & Otto-von-Guericke University Magdeburg, Germany
andy.kenner@metop.de

Richard May
Harz University Wernigerode, Germany
rmay@hs-harz.de

Jacob Krüger
Ruhr-University Bochum & Otto-von-Guericke University Magdeburg, Germany
Jacob.Krueger@rub.de

Gunter Saake
Otto-von-Guericke University Magdeburg, Germany
saake@ovgu.de

Thomas Leich
Harz University Wernigerode & METOP GmbH, Germany
tleich@hs-harz.de

ABSTRACT

Safety and security are important properties of any software system, particularly in safety-critical domains, such as embedded, automotive, or cyber-physical systems. Moreover, particularly those domains also employ highly-configurable systems to customize variants, for example, to different customer requirements or regulations. Unfortunately, we are missing an overview understanding of what research has been conducted on the intersection of safety and security with configurable systems. To address this gap, we conducted a systematic mapping study based on an automated search, covering ten years (2011–2020) and 65 relevant (out of 367) publications. We classified each publication based on established security and safety concerns (e.g., CIA triad) as well as the connection to configurable systems (e.g., ensuring security of such a system). In the end, we found that considerably more research has been conducted on safety concerns, but both properties seem underexplored in the context of configurable systems. Moreover, existing research focuses on two directions: Ensuring safety and security properties in product-line engineering; and applying product-line techniques to ensure safety and security properties. Our mapping study provides an overview of the current state-of-the-art as well as open issues, helping practitioners identify existing solutions and researchers define directions for future research.

CCS CONCEPTS

• **Software and its engineering → Software product lines**; **Software safety**; • **Security and privacy → Software security engineering**; • **General and reference → Surveys and overviews**.

KEYWORDS

Safety, Security, Software Product Line Engineering, Configurable Systems, Mapping Study

SPLC '21, September 6–11, 2021, Leicester, United Kingdom

ACM ISBN 978-1-4503-8469-8/21/09.
https://doi.org/10.1145/3461001.3471147

ACM Reference Format:
Andy Kenner, Richard May, Jacob Krüger, Gunter Saake, and Thomas Leich. 2021. Safety, Security, and Configurable Software Systems: A Systematic Mapping Study. In *25th ACM International Systems and Software Product Line Conference - Volume A (SPLC '21), September 6–11, 2021, Leicester, United Kingdom.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3461001.3471147

1 INTRODUCTION

Safety and Security (S&S) are important quality attributes of any software system, particularly in the context of safety-critical domains in which harms to the system, its users, or the surrounding environment must be avoided. For instance, automotive systems (e.g., self-driving cars) must prevent accidents and injuries [23], cloud-computing systems must ensure data availability and privacy [28], or cyber-physical systems must assure the safety of involved humans [18, 56]. Consequently, various software-engineering process models and standards (e.g., ISO 26262) in such domains explicitly involve S&S concerns.

Moreover, many safety-critical systems are configurable (i.e., they exhibit variability) to adapt them to specific customer requests, hardware restrictions, or legal regulations. Configurability adds a layer of complexity to a software system, allowing its developers to derive a family of similar, yet customized variants by enabling or disabling features (i.e., user-visible functionalities) [7, 81]. Since each variant comprises individual features and potential interactions between these, ensuring S&S becomes even more challenging. As a consequence, research on configurable systems and especially product-line engineering has also been concerned with S&S concerns. For instance, Acher et al. [3] study how configurability may reveal confidential information, Abal et al. [1] investigate potential security threats originating from variability bugs, and we [46] proposed using variability models to assess the threat potential of configurable systems.

Despite the extensive research on S&S in the context of configurable systems, we are not aware of a systematic overview of the current state-of-the-art. Thus, it is unclear what S&S concerns have been studied, what concerns may have been neglected, and what directions are important for research to tackle. In this paper, we aim to address this problem by presenting the results of a systematic mapping study [47]. For this purpose, we employed an automated

search on Scopus[1] and included 65 publications from the last decade (2011–2020). By studying the properties of the reported research (e.g., addressed security concerns, covered product-line engineering spaces), we aim to summarize the current state-of-the-art and highlight neglected S&S concerns. Note that we focus on a more qualitative analysis, and thus do not report the typical publication statistics of systematic literature studies. In detail, we contribute:

- A systematically elicited overview of recent research on S&S in the context of configurable systems.
- A discussion of what properties have been covered, and what properties ask for more research.
- A replication package that comprises lists of all retrieved and included publications as well as our raw data.[2]

The results of our study help researchers and practitioners to obtain an overview understanding of the recent state-of-the-art. Furthermore, our analysis can guide practitioners while identifying and selecting techniques or methods for development projects, and researchers while scoping future research.

2 BACKGROUND

In the following, we provide key background on *safety*, *security*, and *configurable systems*.

2.1 Safety

Safety is a desired attribute of a software system to ensure that the system behaves as intended and does not pose harm to itself, its users, and the environment. A system is safety-critical, if bugs or failures in its behavior threaten the life or health of humans, pose danger to the environment, or could cause significant property damage [48]. Such systems are developed in several prominent domains, such as automotive, avionics, railways, or medical devices. Typically, specifically designed techniques, mechanisms, and standards (e.g., ISO 26262, DO-178C) are employed to engineer safety-critical systems, assuring their correct behavior, and avoiding harms.

Since software has become a major part of any safety-critical system (e.g., from controlling features of a car up to autonomous driving), software safety also increasingly impacts the safety of the whole system. In particular, software is an essential component for industrial-control systems, embedded systems, or cyber-physical systems—used to monitor and manipulate the behavior of such systems [50]. In addition, such systems increasingly interact with (and adapt to) the surrounding environment, processes, humans, or other systems, posing additional interrelations, dependencies, and safety concerns [18, 28, 56, 103].

2.2 Security

Security (also called cyber-security or IT security) involves different properties (e.g., confidentiality, authorization) that ensure that a system is protected, among others, against unauthorized access to its features, data, or hardware to avoid their theft or misuse [36, 37]. So, while safety aims to protect the environment from misbehavior of the system, security aims to protect the system against potential threats of that environment. For this purpose, security involves preventing any kind of planned action (e.g., exploiting a vulnerability) that could expose the system and potentially harm its users or the environment (e.g., by stealing intellectual property). Vulnerabilities to a system's security can be caused by faults in its design (e.g., unintended feature interactions), development (e.g., buffer overflows), or operation (e.g., configuration error) [65, 80].

Typically, security goals are scoped and assessed based on the CIA triad, which involves [38, 86]:

Confidentiality: The data in a system is accessible only for authorized users.

Integrity: The system's features and data can only be modified with authorized access.

Availability: The system ensures timely and reliable access to its features and data.

In the context of information security, these security goals have been extended by the following:

Accountability: Any action executed in the system or on its data can be traced to a unique entity (e.g., an individual or system).

Authenticity: The identity of an entity can be unambiguously proven to be the one claimed.

Non-repudiation: For every action, it is possible to prove that it was executed and what entities were involved—preventing that any action can be disputed later on.

These security goals are widely established in practice, leading to several guidelines and standards, such as ISO 2700X or BSI 200.

2.3 Configurable Systems

A configurable system builds upon a software platform that comprises a set of features (i.e., user-visible functionalities) that can be customized (i.e., configured) to specific user requirements, regulations, or hardware limitations. Typically, advanced configurable systems employ concepts, methods, and techniques of software product-line engineering [7, 81]. Namely, variability models (particularly feature models) have become an established means to organize, structure, and document features as well as their dependencies [7, 22, 40, 73, 89]. In the source code, a variability mechanism (e.g., preprocessors, parameters) is used to implement variation points that control the configurable features [30, 94]. Most commonly, features are optional, meaning that they can only be enabled or disabled (i.e., Boolean). Developers can then define a valid configuration by setting a value for each feature (e.g., true or false), so that none of the specified dependencies (e.g., in the feature model) is violated. This step is usually supported by configurator tools that ensure the validity of a configuration, propagate configuration decisions [7, 53], and automatically derive the configured variant.

For this paper, we are mainly concerned with two concepts of product-line engineering that we use as classification criteria for identified publications. First, product-line engineering can be described as a *projection* on problem space, solution space, and a mapping between both [6, 7]. The problem space covers the domain abstraction of a configurable system, for instance, as specified in a feature model. In contrast, the solution space covers the concrete implementation of the configurable platform. Both are connected through a mapping (e.g., feature names defined in the variability mechanism and a configuration) that allows to derive a configured system.

Second, a configurable system can be *verified* against different attributes, for which Thüm et al. [95] distinguish three strategies:

[1] https://www.scopus.com
[2] bitbucket.org/akenner/splc2021-study_data

(1) An analysis is product-based if it operates on one specific configuration of the system—using the code itself or an abstraction (e.g., a model or configuration).
(2) An analysis is family-based if it operates on domain artifacts (i.e., the platform) only, and involves knowledge about valid configurations (e.g., the feature model).
(3) An analysis is feature-based if it operates on domain artifacts only, and analyzes each feature in isolation (i.e., not considering valid configurations and feature interactions).

These two concepts provide high-level criteria that we use to structure our data, which we explain in the next section.

3 STUDY DESIGN

Our research objective was to identify, classify, and discuss existing research on S&S in the context of configurable software systems. To achieve this objective, we employed a systematic mapping study following the guidelines of Kitchenham et al. [47]. In this section, we describe the individual steps of our study based on these guidelines.

3.1 Initial Screening

We conducted an initial screening to ensure the need for our mapping study (i.e., that there is a relevant body-of-knowledge and no recent study similar to ours). For this purpose, we employed the following search string (without time constraints, cf. IC_2):

```
("safety" OR "security") AND
("software product line")
```

During this initial screening, we used the literature database Scopus, from which we obtained 2,600 publications. The large number of publications confirms a sufficient research interest in S&S in the context of configurable systems. Moreover, the recent tertiary study of C and Chandrasekaran [21] did not reveal such a literature review or mapping study. Consequently, we considered our research objective valuable and initiated our systematic mapping study.

3.2 Search String

We employed an automated search on Scopus, which ensures a certain quality by indexing only peer-reviewed publications and by reviewing the venues, too. Also, Scopus covers various publishers, reducing the threat that we miss highly relevant publications. Building on our initial screening (i.e., identifying regular synonyms used in titles), we defined the following search string:

```
(("config* product*" OR "config* system*" OR "product
line*" OR "product famil*" OR "software famil*" OR
"system famil*" OR "variant*rich system*") AND
("security" OR "safety"))
```

Note that we used Scopus' default search settings, which is why we covered titles, abstracts, and keywords.

3.3 Selection Criteria

To identify and select relevant publications, we defined the following inclusion criteria:

IC_1 The publication is written in English.
IC_2 The publication has been published between 2011 and 2020.
IC_3 The publication is a peer-reviewed conference paper or journal article (e.g., excluding journal-first or keynote summaries).
IC_4 The publication is longer than three pages.
IC_5 The publication is concerned with the intersection of configurable systems with safety and/or security.

We used IC_4 to ensure that a publication provides enough details to comprehend the addressed problem. Also, we did not perform a quality assessment. Instead, we relied on Scopus' review of publication venues, IC_3, and IC_4 to ensure the quality of publications. This represents an established adaptation [47], since we are structuring previous findings that are based on different research methods. We focus on the last decade (IC_2) to cover the most recent advancements in research and practice, arguing that older important findings on S&S usually become well-established practices and standards.

3.4 Data Extraction

Depending on the main concern of each publication (i.e., safety or security), we adapted what data we extracted to tackle our research objective. Namely, we extracted additional data for all publications on security, since we could rely on established classifications of security goals (cf. Section 2.2). Unfortunately, we are not aware of similar classifications of safety goals. So, we extracted the following data:

- For safety and security
  - Standard bibliographic data to elicit a consistent data basis.
  - The **perspective** of the publication, which is divided into two sub-criteria (read the → as "employed to"): *Safety/security → SPL* covers publications that are concerned with the safety/security of a configurable system. *SPL → safety/security* covers publications that aim to address a safety/security concern with the help of product-line techniques.
  - The **domain** that a publication is concerned with (e.g., automotive, embedded systems).
  - The safety/security **standard** a publication refers to, or to which it can be aligned (e.g., ISO 27001).
  - The product-line **projection** of a publication, which specifies whether it covers problem space, solution space, or the mapping between both [7].
  - The **verification** method reported, namely whether the configurable system is analyzed based on products, features, or the whole product family [95].
  - Whether a publication considers the **evolution** of a configurable system or not.
  - Whether the publication reports on specific **tool support** (e.g., FeatureIDE [66], pure::variants [17], own prototypes).
- Only for security
  - The **security goals** covered in a publication. We consider the *CIA triad* (i.e., confidentiality, integrity, availability) and additional *information-security* concerns, namely authorization, accountability, and non-repudiation.
  - How security concerns are documented or managed in the publication in terms of their **specification**, for instance, as security goals or non-functional requirements.
  - Whether the publication focuses on **security threats**, namely vulnerabilities or the disclosure of trade secrets, or defines strategies to remediate those threats.

We derived these criteria from concepts, research, and guidelines on S&S or product-line engineering. So, our data spans a diverse and relevant set of criteria that helps to connect both areas. To

manage our data, we used spreadsheets that were available to all authors, and which we contribute in our replication package.

3.5 Conduct

The first author of this paper conducted the automated search on March 15th 2021, retrieving 367 publications. Afterwards, the first and second authors independently inspected all publications to identify those relevant for our research objective. Disagreements between the two authors were resolved during discussions (partly involving other authors as independent advisors) until they achieved consensus on their decision.

After removing duplicates and analyzing titles as well as abstracts, we kept a total of 146 publications. We then read each publication in detail and employed all of our inclusion criteria. Finally, we considered 65 publications as relevant for addressing our research objective.

4 STUDY RESULTS

In this section, we present the results of our systematic mapping study, omitting typical publication statistics. For simplicity, we assigned each publication that is concerned with safety as well as security to its predominant concern (e.g., analysis techniques that could be used for either, but are demonstrated for one only). To assess the criteria we defined for the data extraction, we use three options: *completely* (●), *partly* (◐), and *not* (○) fulfilled.

4.1 Safety

Regarding the intersection of safety and configurable systems, we included 41 publications. We provide an overview of these publications and their criteria in Table 1. Next, we describe our results based on these criteria.

Perspective. The number of publications addressing each of our two perspectives is quite balanced. 18 publications are concerned with the safety of a configurable system (S&S → SPL). For instance, Pett et al. [79] apply a risk-based change-impact analysis on an automotive architecture, combining product sampling, risk-based testing, and configuration prioritizing. 23 publications are concerned with employing product-line techniques to assure safety concerns. For example, motivated by the automotive industry, Ali et al. [5] present a model-based reasoning framework for systematically managing hazards.

Domain. We can see in Table 1 that safety-related publications span a variety of domains. Note that we identified some of the domains by considering the standards that are referenced, for instance, ISO 26262 is concerned with the automotive domain [20]. Other domains were explicitly mentioned or we could clearly derive them from the publication's context, for example, if it is concerned with medical systems [16]. Most of the publications (19) focus on the automotive domain or safety-critical systems in general (12). The remaining 10 publications cover six different domains, namely cyber-physical systems (3), avionics (2), medical systems (2), emergency systems, mechatronics, and railways.

Standard. 18 of the 41 publications explicitly mention to address a standard. Most frequently, ISO 26262 of the automotive domain is mentioned (15), with other standards occurring sparsely. Namely, DO-178B/C occurs twice [19, 24], whereas IEC 65108 is named only once [29]. Only five of the 18 publications tackle a specific phase of the mentioned standard, namely the concept phase of ISO 26262. Two publications have an extended focus that involves additional phases, namely the actual (software) development [41, 102]

Table 1: Classification of safety-related publications.

	Perspective			Standard		Projection			Verification				
Reference	S&S → SPL	SPL → S&S	Domain	Standard addressed	Phase indicated	Problem space	Solution space	Mapping	Product-based	Family-based	Feature-based	Evolution	Tool support
[2]	○	●	R	○	○	●	●	●	○	○	○	○	○
[5]	○	●	A_1	○	○	●	○	●	○	○	○	○	○
[9]	○	●	A_1	●	○	◐	◐	○	○	○	○	◐	○
[10]	○	●	S	○	○	○	○	○	○	○	○	○	○
[11]	●	○	A_1	●	●	●	●	●	◐	○	○	◐	○
[12]	○	●	S	○	○	●	○	○	○	○	○	◐	◐
[14]	○	●	S	○	○	●	●	●	◐	○	○	●	◐
[16]	○	●	M_1	○	○	●	○	○	●	○	○	○	●
[19]	○	●	A_2	●	○	●	○	○	○	○	○	○	○
[20]	●	○	A_1	●	●	●	●	●	○	○	○	○	●
[24]	●	○	A_2	●	○	●	●	●	●	○	○	○	●
[25]	●	○	A_1	●	○	●	●	●	●	○	○	○	○
[26]	●	○	S	○	○	●	●	●	●	○	○	●	●
[27]	○	●	A_1	●	○	●	○	○	●	○	○	○	○
[29]	○	●	S	●	○	●	●	●	●	○	○	○	●
[31]	●	○	A_1	●	○	●	●	●	○	○	○	○	◐
[41]	○	●	A_1	●	●	●	○	○	○	○	○	○	◐
[42]	○	●	A_1	●	○	●	●	●	○	○	○	●	●
[39]	●	○	A_1	●	●	●	●	○	○	○	○	○	○
[43]	○	●	A_1	●	○	●	●	●	○	○	○	●	○
[44]	●	○	A_1	●	○	●	●	●	●	○	○	○	●
[49]	○	●	A_1	●	○	○	○	○	○	○	○	○	◐
[51]	●	○	A_1	○	○	●	○	○	●	○	○	○	○
[56]	○	●	C	○	○	●	○	○	○	○	○	○	○
[57]	●	○	S	○	○	●	○	○	●	○	○	○	●
[59]	●	○	M_1	○	○	●	○	○	○	○	○	○	●
[60]	●	○	A_1	○	○	●	○	○	●	○	○	○	○
[62]	○	●	C	○	○	●	○	○	●	○	○	○	●
[63]	○	●	C	○	○	●	○	○	●	○	○	○	●
[64]	○	●	S	○	○	●	○	○	●	○	○	○	●
[74]	●	○	S	○	○	●	○	○	○	○	○	○	●
[75]	●	○	A_1	●	○	●	●	●	●	○	○	●	◐
[77]	○	●	S	○	○	●	○	○	●	○	○	○	●
[79]	●	○	A_1	○	○	●	○	○	●	○	○	◐	●
[82]	○	●	S	○	○	●	○	○	○	○	○	○	●
[83]	●	○	E	○	○	●	●	●	○	○	○	○	◐
[84]	●	○	S	○	○	◐	○	○	○	○	○	◐	◐
[85]	●	○	A_1	●	●	●	●	○	○	○	○	○	○
[90]	○	●	A_1	●	○	●	●	●	○	○	○	●	●
[92]	○	●	M_2	○	○	●	◐	●	●	○	○	○	●
[101]	○	●	S	○	○	●	●	●	○	○	○	○	●

●: Completely fulfilled; ◐: Partly fulfilled; ○: Not fulfilled
A_1: Automotive; A_2: Avionics; C: Cyber-physical systems;
E: Emergency systems; M_1: Medical systems; M_2: Mechatronics;
R: Railways; S: Safety-critical systems

Projection. A majority of the publications (39) is concerned with the problem space, for instance, Kowal et al. [51] aim to reduce test suites by explicitly modeling information about shared resources and communications in a feature model. Almost half (20) of the publications address the solution space, covering different safety-related artifacts, such as fault trees [44] or safety cases [31]. 18 publications are concerned with some kind of mapping between problem and solution space, for example, Kelly et al. [44] tailor different artifacts for hazard analyses and risk assessments based on a specific configuration. We marked this criterion as partly fulfilled for some publications to indicate that these mention a projection, but do not explain it in detail. However, such cases occurred sparsely: twice for the problem [9, 84] and twice for the solution space [9, 92].

Verification. Regarding the verification, we found that no publication proposes a family-based or feature-based method. In contrast, 19 publications refer to product-based verification methods (i.e., the safety of a configured variant is verified). For two of these publications, we considered this criterion as partly fulfilled, because they discuss this verification without explaining details [11, 14]. Interestingly, all 19 publications rely on some sort of optimization to enhance the product-based verification, for example, Lachmann et al. [57] reduce the number of tests by using delta-oriented test-case prioritization for an individual system configuration.

Evolution. We found 11 publications that (at least partly) consider the evolution of a system, for instance, Schulze et al. [90] propose a technique for the automotive domain to reflect how a system change impacts the functional safety. The five publications fulfilling this criterion partly indicate awareness for evolution, but do not explain how it is or could be dealt with. Still, most publications do not even mention evolution in any context.

Tool Support. We identified 19 publications that report on the integration of safety concerns in established tools (either directly or based on extensions), such as FeatureIDE [59], pure::variants [77], or Enterprise Architect [43]. Some publications involve other tools or prototypical implementations that have been developed particularly for the described research. Eight publications partly fulfill this criterion, since they highlight tool support as immediate future work, for instance, to model safety costs [84] or specify safety-case lines [31].

4.2 Security

Regarding the intersection of security and configurable systems, we included 24 publications. We provide an overview of the publications and their criteria in Table 2. In the following, we describe our results based on these criteria.

Perspective. 18 of the 23 security-related publications propose techniques for analyzing or ensuring security goals of a configurable system (S&S → SPL). The remaining six publications aim to employ product-line techniques to support security goals (SPL → S&S). For instance, Mellado et al. [68] extend SecureTropos to cover the specifics of configurable systems. Similarly, Krieter et al. [52] propose to utilize concepts of dynamic software product lines to ensure the security of cloud systems based on Intel SGX.

Domain. Interestingly, we can see in Table 2 that compared to the safety-related publications far fewer techniques have been designed for or evaluated in a specific domain. Namely, 18 publications are concerned with product-line engineering in general, and two with any type of software system. Actual domains are only mentioned once each: automotive [91], cloud computing [52], embedded systems [97], and internet of things [96].

Standard. We found that security standards are considered only in two publications. First, Wilson and Young [104] refer to NIST 800-160, which builds upon ISO 15288, to propose an architecture for resilient systems. Second, Villela et al. [100] rely on ISO 25010 to define quality attributes, but these do not represent security goals.

Projection. Again, a huge majority of the publications covers the problem space (20). For example, Mellado and Mouratidis [67] extend Tropos to incorporate support for secure product-line engineering on a formal level. Additionally to the problem space, 11 publications are also concerned with the solution space as well as the mapping between both. For instance, Peldszus et al. [78] propose how to perform model-based (i.e., problem space) security analyses of product lines (i.e., solution space), and require a consequent mapping between both. We did not assign four publications to any projection, mostly because they exemplify or analyze potential security problems, but do not aim to address them, yet.

Verification. Regarding verification, we could define criteria only for six publications. Most (5) are again concerned with product-based verification, while one publication proposes a family-based verification [78]. Again, we found no publication that aimed at verifying the security of individual features.

Evolution. In seven publications, the evolution of a configurable system and its security goals is at least partly considered. For example, Murashkin et al. [71] propose to automatically optimize different quality attributes (e.g., security) for a certain configuration. To support developers, they suggest visualizations that help understand the evolution of optimal configurations.

Tool Support. Eight publications extend existing tools or propose own prototypes. For instance, Peldszus et al. [78] build upon Eclipse to implement their family-based verification technique. Again, we considered that publications partly fulfilled this criterion, if they specified tool support as immediate future work.

Security Goal. Surprisingly, only 10 publications explicitly specify one ore more security goals that they are concerned with. So, as we can see in Table 2, no security goal is consistently or extensively addressed in research. Nine publications focus on confidentiality and integrity, seven on availability, four on accountability, and three on authorization as well as non-repudiation. This shows that most security goals are based on the CIA triad, while far fewer stem from concerns of information security. Moreover, only three publications consider how to fulfill all security goals at the same time [67, 68, 91]. For instance, Mellado and Mouratidis [67] describe a holistic security framework with which they aim to facilitate the development of secure configurable systems. In contrast, Varela-Vaca et al. [99] describe a framework for verifying whether a system configuration complies with cyber-security policies—but without considering any of the security goals.

Specification of Security Goals. We identified a set of publications that manages security goals as non-functional requirements or functional quality attributes. In either case, the publications propose different ways of handling and managing these specifications during development. Especially, it seems that there is no agreement on the purpose and use of security-goal specifications for configuring.

Table 2: Classification of security-related publications.

	Perspectives				Projection			Verification					Security goals: CIA Triad			Security goals: Information security				
Reference	S&S → SPL	SPL → S&S	Domain	Standard	Problem space	Solution space	Mapping	Product-based	Family-based	Feature-based	Evolution	Tool support	Confidentiality	Integrity	Availability	Authorization	Accountability	Non-repudiation	Specification	Security threats
[3]	●	○	S_1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	◐
[13]	●	○	S_1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	◐
[33]	●	○	S_1	○	●	●	●	○	○	○	○	○	◐	◐	○	○	○	○	●	○
[34]	●	○	S_1	○	●	●	●	●	○	○	○	●	◐	◐	○	○	◐	○	●	○
[35]	●	○	S_1	○	●	●	●	●	○	○	○	○	○	○	○	○	○	○	●	○
[45]	○	●	S_2	○	●	○	○	○	○	○	●	◐	○	○	○	○	○	○	○	●
[52]	○	●	C	○	●	●	●	●	○	○	●	○	●	●	○	○	○	○	○	○
[67]	○	●	S_1	○	●	○	○	○	○	○	○	◐	●	◐	◐	◐	●	◐	○	○
[68]	○	●	S_1	○	●	○	○	○	○	○	○	◐	◐	◐	◐	◐	◐	◐	○	○
[69]	●	○	S_1	○	○	○	○	○	○	○	○	○	◐	◐	◐	○	○	○	○	●
[70]	●	○	S_1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
[71]	●	○	S_1	○	●	○	○	○	○	○	●	●	○	○	○	○	○	○	●	○
[72]	●	○	S_1	○	●	●	●	○	○	○	○	●	○	○	○	○	○	○	○	●
[78]	●	○	A	○	●	●	●	○	●	○	◐	●	●	●	●	○	○	○	○	○
[87]	●	○	S_1	○	●	●	●	○	○	○	○	◐	○	○	●	○	○	○	●	○
[91]	●	○	S_1	○	●	●	●	○	○	○	○	◐	●	●	●	●	●	●	○	○
[93]	●	○	S_1	○	●	○	○	○	○	○	◐	◐	○	○	○	○	○	○	●	○
[96]	●	○	I	○	●	○	○	○	○	○	○	○	●	●	●	○	○	○	●	○
[97]	○	●	E	○	●	●	●	○	○	○	○	●	○	○	○	○	○	○	○	○
[98]	○	●	S_2	○	●	○	○	○	○	○	○	●	○	○	○	○	○	○	○	●
[99]	●	○	S_1	○	●	●	●	●	○	○	◐	●	○	○	○	○	○	○	○	○
[100]	●	○	S_1	●	●	○	○	○	○	○	○	◐	○	○	○	○	○	○	●	○
[104]	●	○	S_1	●	●	●	●	●	○	○	●	○	○	○	○	○	○	○	○	●
[105]	●	○	S_1	○	●	○	○	○	○	○	○	●	○	○	○	○	○	○	●	○

●: Completely fulfilled; ◐: Partly fulfilled; ○: Not fulfilled
A: Automotive; C: Cloud computing; E: Embedded systems; I: Internet of things;
S_1: Software product lines; S_2: Software systems

In nine publications, security goals are managed as quality attributes of the configurable system, using different methods to do so. For example, Hammani et al. [33] propose to incorporate security goals as feature attributes or additional part in a feature model. Other publications rely on goal models or other external representations, such as tables [93, 105]. To specify security goals and their dependencies (e.g., to each other, to additional quality attributes), most publications rely on the knowledge of system experts.

Security Threats. Lastly, we identified six publications that are fully concerned with security threats. Three intend to identify and understand potential vulnerabilities of a configurable system. For instance, Mesa et al. [69] investigate the web-based plug in system WordPress and Muniz et al. [70] study `#ifdef` annotations to understand whether vulnerability are related to the system's architecture or variability mechanism. In contrast, Kenner [45] proposes to rely on product-line techniques and existing vulnerability databases to document and manage potential vulnerabilities of established software systems. Three other publications discuss solutions for tackling vulnerabilities of configurable systems.

Regarding remediation strategies, Myllärniemi et al. [72] propose to enrich configurable systems with counter measures for tackling security threats. Such extensions would allow a system to prevent, detect, and counter potential attacks. Similarly, Wilson and Young [104] aim to incorporate resilience into a configurable system by defining specific variation management for secure assets.

Finally, we identified threats that can originate from the disclosure of trade secrets. For instance, Bécan et al. [13] analyzed an online-video generator to deduce its general behavior, and extracted the structure as well as variation points. In the end, they were able to implement an own generator with improved features. Identically, Acher et al. [3] reason that it can be simple to elicit confidential information about a configurable system from its behavior.

5 DISCUSSION

Next, we summarize core insights we obtained from our results, based on which we discuss 14 **literature gaps (LG)** we identified. We identified these gaps through collaborative analyses and discussions among all authors, building on our results and practical experiences in the domains of S&S and configurable systems. Note that the literature gaps indicate directions for future research, but require a more extensive validation.

5.1 Safety

Not surprisingly, we can see in Table 1 that safety research on configurable systems is mostly driven by safety-critical **domains**, such as embedded and automotive systems. Consequently, safety **standards** seem to be a dominant driver of most research, mostly for the automotive domain. More precisely, almost all publications on that domain reference a standard. Unfortunately, only a third of these publications specify which **phase** of the standard they address. In addition, most publications refer to **tool support**, providing an extensive set of implementations practitioners can rely on. Overall, we can see that *the intersection of safety and configurable systems seems to be heavily driven by industry*, with few publications presenting domain-independent work. We consider this a highly valuable situation, since it highlights the importance of this research direction and its potential to immediately impact practice. Moreover, the close connection to industry provides the opportunity to employ new research in real-world settings.

As mentioned, the automotive domain is dominant and also exemplary in terms of applying safety standards. We are aware of other established safety standards, such as IEC 61508 for functional safety on which many others built, for instance, ISO 26262 (automotive) or EN 50128 (railway). While a larger number of standards exists, those are rarely referenced in the publications we identified, except in case of the automotive domain. For this reason, we argue that **(LG_1)** *research on configurable systems should be connected to existing safety standards to address domain specific artifacts, tasks, or processes.* So, research can be more closely aligned to the specifications against which particular artifacts, resulting variants, or processes are certified. Moreover, it seems helpful to clarify the extent to which a standard is covered by research to make it easier to understand how well they are aligned and what gaps are still open.

Safety standards are not available for every domain of safety-critical systems. However, there may be similar problems in each domain (e.g., hazard analysis, fault trees). So, it seems interesting to **(LG_2)** *investigate to what extent standards and research results on the safety of configurable systems can be transferred between domains.* For instance, we are not aware of specific safety standards for cyber-physical systems, which may be guided by those for embedded systems. Similarly, techniques and methods proposed for one domain may be transferable to another one. However, this requires researchers to investigate whether and how findings in one domain can be transferred to other domains.

In most cases, those publications referring to a safety standard and indicating a particular phase focus on the conception. This poses additional problems, since it may hide how variability and safety relate during other development phases. For instance, it seems unclear how safety is supported in the remaining phases, how it is impacted by feature interactions, or how it is traced throughout the whole development process. We argue that **(LG_3)** *research should cover and investigate all phases defined in safety standards to unveil and tackle problems that may occur after the initial conceptualization of safety-critical configurable systems.* Note that some domains (e.g., automotive, avionics) have detailed process definitions and standards that partly address such problems. Consequently, findings in such domains can serve as a basis to scope future research and adopt it to other domains.

Despite appearing industry-driven, we found that few publications report on employing the proposed techniques on real-world systems. Mostly, this seems to be caused by the fact that only the concept phase (i.e., focusing on the problem space) is addressed. Consequently, we argue that **(LG_4)** *more research on the intersection of safety and configurable systems must be evaluated on real-world systems.* Precisely, we require studies on systems that exhibit a large number of features and variation points to see whether proposed techniques scale. We are aware that this may be problematic, due to confidentiality issues or missing safety-related artifacts (e.g., requirements) for similar open-source systems.

5.2 Security

Interestingly, several of our findings regarding security differ heavily from those for safety. Namely, we identified far fewer concrete **domains** that would imply collaborations between research and industry. Instead, most findings and techniques on the intersection of security and configurable systems seem to be driven by more fundamental research. As a result, we identified only two **standards**, which are not even directly concerned with security. However, we again found considerable existing or planned **tool support**. Overall, we can see that *the intersection of security and configurable systems seems to be driven by fundamental research.* While this means that most techniques and findings are likely transferable to a variety of domains (i.e., they are not domain-specific), this also means that they may not tackle actual needs of practitioners.

That few concrete domains are addressed in security-related publications is an interesting observation. For instance, this could imply that security is neglected in industry or that research tackles the wrong concerns. We suggest that **(LG_5)** *more research on configurable systems' security should be conducted in collaboration with practitioners* to understand real-world problems and needs. Precisely, we see the need to improve and facilitate the knowledge transfer to practice and elicit actual problems in industry. Otherwise, it remains unclear to what extent this research is valuable for real-world use cases and configurable systems in a domain.

Surprisingly, the two standards that are explicitly mentioned in the security-related publications are not concerned with security. We were surprised by this fact, since there is a variety of security standards. For instance, ISO 27001 is concerned with the secure management of a system, ISO 15408 defines common security criteria, IEC 62443 specifies network and system security, and ISO 21434 involves security of vehicles. Apparently, such standards are far less established compared to safety standards. Reflecting on this fact, we consider it valuable to **(LG_6)** *analyze and compare existing security standards concerning their application in the context of configurable systems.* This may help to understand how those standards

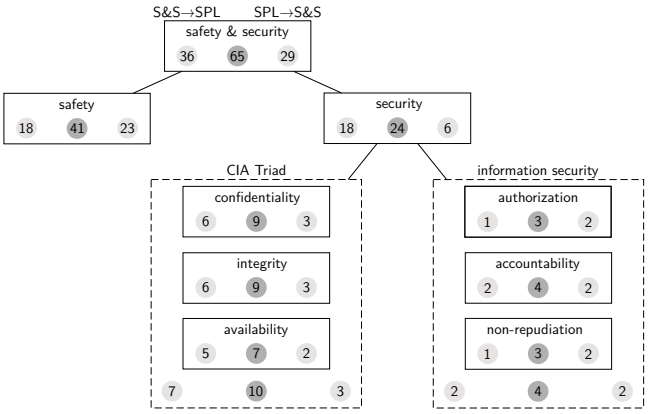

Figure 1: Overview of the publications we identified, separated by goals and perspectives (numbers indicate publications: left S&S → SPL; middle total; right SPL → S&S).

can be used to systematically ensure the security of configurable systems. The consequent insights can help to scope future research, for instance, on consistently aligning development methods with a specific security standard, improving the transfer into practice, or facilitating the certification of variants.

It is interesting that most publications do not even address a concrete **security goal**. Moreover, if security goals are stated, these are usually only a part of the CIA triad (i.e., only a subset or an individual goal partly). Also, goals for information security in configurable systems have rarely been investigated. The goals themselves are documented in a **specification**, typically as non-functional requirements or as functional quality attributes.

Our findings clearly highlight a general need to **(LG_7)** *explore security (and particularly information security) in the context of configurable system in more detail.* Interestingly, we can see a close connection between security goals and configurability: The degree to which a security goal must be fulfilled depends on the concrete use case, and thus can vary for different domains and contexts. For instance, some domains may require alternative implementations for individual features to fulfill security goals for the internal or external use of the system (e.g., secrecy) [3, 58]. Consequently, a configurable system can be immediately beneficial for an organization to fulfill specific security goals regarding a required level of protection. Unfortunately, it seems unclear how a secure configurable system can be developed based on product-line engineering. So, it seems worth exploring existing standards and frameworks for secure software engineering and aligning them to process models for product-line engineering [7, 55, 81]. This could lead to concrete frameworks, guidelines, and recommendations that support organizations while implementing security goals in their configurable systems.

The existing research proposes methods for modeling, documenting, and analyzing security specifications. However, eliciting the actual specification usually involves a qualitative analysis based on expert knowledge, and is only useful in combination with other quality attributes, such as performance, energy consumption, or response time. Namely, achieving a security goal immediately impacts such other quality attributes (e.g., higher security may reduce the response time or lower a system's performance). A qualitative analysis is needed, since it is currently problematic to impossible to measure security—particularly in complex scenarios involving a large number of features and quality attributes (e.g., in embedded systems). We see the need to **(LG_8)** *investigate ways to quantify the level of security or the fulfillment of certain security goals for configurable systems.* This can help to provide a more specific and objective assessment, ideally taking into account how using a configurable system influences an organization's security in general.

Security threats are usually caused by existing vulnerabilities that result from failures within a system's life cycle. Despite the enormous danger that vulnerabilities pose to a system, or even to the entire organization in which it operates, security threats are sparsely addressed in the context of configurable systems. The publications we identified suggest that it is valuable to consider vulnerability information during the development and evolution of a configurable system. Moreover, some studies identified that configurability can cause new types of vulnerabilities. For instance, Acher et al. [3] describe how trade secrets may be revealed, depending on how variability has been implemented. Unfortunately, mitigation strategies (e.g., resilience) are barely explored.

Seeing the lack of research on security threats, we propose to **(LG_9)** *investigate vulnerabilities caused by the configurability of a system as well as strategies to mitigate such vulnerabilities and prevent them for all variants.* For instance, similar to Acher et al. [3] and Abal et al. [1], existing configurable systems should be explored to uncover potential security threats and understand their relation to the configurability in the system. By investigating the causes of security threats, it is possible to understand at what phase in a system's life cycle they were introduced and how such threats affect later phases. For example, they may be based on faulty requirements or an incorrect implementation (e.g., wrong `#ifdefs`). Understanding such causes could help researchers and practitioners to focus on the most important phases and causes. Finally, we need strategies to at least mitigate security threats in existing systems and to prevent them in the future, for instance, by reducing the attack surface of a configurable system.

5.3 Safety and Security

To provide a more concise overview regarding the **perspectives** in our data set, we display an overview of how many publications cover what S&S goal and what perspective in Figure 1. We can see that the safety-related publications are comparably balanced. In contrast, security-related publications focus heavily on ensuring security of a configurable system (S&S → SPL). Moreover, we can see the focus of security-related publications on the CIA triad.

It is interesting that far fewer publications are concerned with employing product-line techniques to address security goals (SPL → S&S). This could indicate that such techniques are not as helpful for supporting security as they are for safety. Additionally, existing research focuses on adding variability to requirements management, neglecting other phases, such as designing, implementing, or testing the security of a system. Seeing this discrepancy, we argue that **(LG_{10})** *further research on established processes and standards to ensure a system's security is required to understand how product-line techniques could support these.* Particularly, it is important to identify to what extent configurability is covered by or impacts processes.

Regarding the **projection**, we already mentioned that most publications on S&S focus on the problem space. The limited research that has been conducted on the actual solution space or mapping causes several of the problems we highlighted before. Particularly, we are missing a detailed understanding of how S&S concerns are propagated from the problem to the solution space, and from where potential violations of S&S properties of a configurable system stem from. For instance, the domain may be specified incorrectly, the implementation can comprise bugs, or the mapping (e.g., configuration) could be faulty. Such issues can potentially cause risks or threats to the S&S of a configurable system.

We propose to **(LG_{11})** *improve our understanding of how S&S of a configurable system can be consistently managed throughout all projections.* Namely, empirical studies (e.g., similar to the one of Abal et al. [1]) can help to reveal the most prominent sources for S&S threats. Then, recommendations and new techniques can be defined to ensure that S&S are consistently assured in every projection—limiting the potential for introducing threats.

Interestingly, we found only one publication that proposes a family-based **verification**. All other publications that mention a verification refer to a product-based one. Consequently, existing research on S&S faces the same problems of other analysis techniques for configurable systems that are product-based [95]. Precisely, each configuration of the system must be verified individually, increasing costs and preventing assurance of the underlying platform.

In our opinion, it would be valuable to **(LG_{12})** *explore how existing techniques for S&S of configurable system could be lifted to family-based or feature-based verification.* As for other analysis techniques, such a lift would allow to assure S&S not only for a specific configuration, but the system as a whole. This would increase the confidence in the S&S of the system, can help unveil and address problems in feature interactions, and avoid costly adaptations and re-certification (e.g., because S&S requirements must be manually tailored to and assured for each configuration).

We can see that little research is concerned with the simultaneous **evolution** of S&S in combination with the configurability of a system. Regarding safety, this may be caused by the focus on the concept phase, and poses the problem that it seems unclear how to manage S&S concerns during a system's evolution. For instance, it seems unclear what parts of a safety/security analysis must be repeated after a change, how evolution impacts S&S, and how to trace as well as manage S&S during system evolution.

Based on this insight, we argue for **(LG_{13})** *conducting more research on S&S during the evolution of a configurable system.* Interestingly, the evolution of configurable systems has recently gained more attention in research [15, 54, 76], and we underpin the need for this direction with a focus on S&S. In combination with family-based or feature-based verification, we argue that this research direction could significantly reduce the costs of re-certifying a system configuration after a change has been introduced.

5.4 Safety and Security in Concert

During our analysis, we identified various relations and dependencies between the individual criteria we elicited. This is not surprising, but we want to take this opportunity to also reflect on the relation between S&S. Precisely, we argue that they should not be considered in isolation, particularly not in the context of configurable systems. For instance, security threats allow attackers to potentially change the behavior of a system, and thus threaten the safety of interacting individuals or the environment. This becomes a particularly important issue for configurable systems, since they operate in safety-critical domains. For instance, automotive and many cyber-physical systems cannot be isolated from individuals (which is typically done to assure safety), and thus must consider S&S simultaneously.

Interestingly, none of the publications we identified was concerned with S&S at the same time. So, based on our sample, we argue that we need to **(LG_{14})** *establish techniques and methods that have an integrated view on the S&S of configurable systems.* A particular challenge in this regard is to understand and manage the dependencies between S&S. Simply put, effective security is necessary to ensure the safety of a system, but it can cause side effects (e.g., authorization mechanisms may affect availability).

Focusing on the automotive domain, which seems to drive safety-related research on configurable systems, we found only one publication addressing primarily security. The same problem may exist in other domains, such as cyber-physical systems and the internet of things. Such domains exhibit significantly different systems compared to more traditional software systems (e.g., connected and autonomous cars), and involve novel processes, for instance, during their engineering, development, or operation. Systems with different properties are combined, increasingly interconnected, and evolved; including novel systems that are only created for new types of interactions (e.g., with individuals or other systems). At the same time, such systems must be configurable to various requirements, restrictions, or regulations, up to the point of self-adapting at run-time to environmental changes. The consequent interaction between physical and cyber components poses new challenges for combining safety, security, and configurable systems.

Basically, safety is a necessary property to ensure the dependability of a system, especially in safety-critical domains. As we found during our study, safety research can build upon well-defined standards and industrial use cases, which is not the case for security research. Considering the automotive domain again, which is also a driver for standards and their alignment with research, our results suggest that a particular standard (i.e., ISO 21434) is in development to address cyber security for road vehicles. The content of this standard is still under construction, but the sketched outline implies a high similarity to the safety standard ISO 26262. This offers the opportunity to analyze security threats, describe their impact on a system, and propose measurements to achieve certain security goals. At this point, we see two challenges: First, standards do not link the processing of S&S, which is why research must aim to align them to advance towards a more integrated view on S&S. Second, as already practiced in the domain of safety-critical systems and in research with industrial focus, the new standard requires adaptations to novel development processes. Similarly to ISO 26262, we have to focus on extending the standard development process based on concepts of systematic reuse and product-line engineering. This opens a variety of research opportunities for researchers to investigate, for instance, regarding to what extent it is possible to generalize and transfer from safety to security—which would help to harmonize both properties in the future.

6 THREATS TO VALIDITY

We are aware of a few threats that could impair the internal and external validity of our study. First, our current search strategy does not cover all publications related to the S&S of configurable systems. For instance, our own work [46] of using variability models for analyzing vulnerabilities of a software system was not part of our data set. This threat to the internal validity is caused by our search strategy, which does not cover all possible concepts established in product-line engineering (e.g., variability models). Second, some publications describe their research in great detail, while others provide only sparse details—showing a lack of consistency and completeness regarding the available information. Although we already applied several quality criteria, we cannot ensure that this did not led to misinterpretations on our side. Third, there were some issues regarding the fulfillment of specific criteria. For instance, it was sometimes unclear if the authors present a concept or a concrete implementation in the context of safety. This issue was caused by authors providing ambiguous or insufficient information to classify them without the risk of misinterpretation. Finally, there are two threats regarding the external validity of our results. Due to our search strategy, we reduced the number of included publications to 65 based on a single literature database (Scopus). Still, the smaller the number of included publications, the higher the potential that a misclassification impacts the overall results.

While these issues threaten our findings, we aimed to mitigate them by employing multiple researchers in the literature analysis. Moreover, we systematically elicited a large number of relevant and peer-reviewed publications, limiting the threat of missing important publications in this research area. In this context, Scopus offers the possibility to examine publications of various publishers so that a broad range of the relevant literature in the research area is covered. As a result, we argue that our current mapping study is valuable and provides detailed insights. Still, we plan to extend it considerably in future work, namely by conducting a more extensive systematic literature review. We plan to refine our search strategy, involve more literature databases (e.g., IEEE Xplore or the ACM Guide to Computing Literature), and perform even more in-depth analyses to confirm the identified trends and gaps regarding S&S for configurable software systems.

7 RELATED WORK

During our search, we found five publications that focus on an analysis of safety or security for configurable software systems. However, none of these works provides a comprehensive and systematic overview, for example, in terms of a mapping study. Beside these, we are aware of a tertiary study in which S&S are considered as quality attributes [21]. Next, we briefly discuss more closely related works.

Safety. Sandim Eleutério et al. [88] report a systematic mapping study (2000–2016) on dynamic software product lines. While security is not addressed, one out of nine publications is related to safety. Since it is only a single publication, no detailed analysis is carried out beyond this. Baumgart and Fröberg [8] describe a systematic mapping study on the functional safety of product lines. While this is related to our own study, we do not focus on functional safety to the same extent. Instead, we provide complementary insights, for instance, on security, projections, and verification.

Security. In their systematic literature review (2005–2016), Ahmed et al. [4] focus on the domain of constraint interaction testing. They analyzed 103 publications, of which two involve security concerns. However, these two publications are not related to configurable software systems. Hammani [32] survey security concerns as non-functional requirements in the context of modeling and verifying software product lines. The survey provides a high-level overview on non-functional requirements, whereas details on security concerns are missing—and safety is completely excluded. Finally, Mahdavi-Hezavehi et al. [61] report a systematic literature review (2000–2011) on the variability of quality attributes (including security) of service-based software systems. Similar to our findings, fewer publications (five out of 46) were concerned with security. Again, Mahdavi-Hezavehi et al. analyze security only on a high level together with other quality attributes. With our study, we provide more detailed insights into S&S of configurable systems.

8 CONCLUSION

In this paper, we presented a systematic mapping study regarding S&S research in the context of configurable systems. With our study, we provide an overview understanding of what research has been conducted in the intersection of these domains. To this end, we built on 65 publications and covered a time period of ten years (2011-2020). Based on these publications, we identified 14 literature gaps, which indicate potential directions for future research. Regarding safety, we identified several gaps related to standards, including an insufficient alignment of research to standards as well as a missing transfer of standards between domains. Additionally, there seems to be a lack of (reported) real-world evaluations. Regarding security, we recommend to conduct research on vulnerabilities caused by configurability as well as security standards, goals, and potential ways to quantify their level of fulfillment. Furthermore, the collaboration with practitioners should be improved. Regarding S&S, we see the need to investigate how product-line techniques could support established processes or standards, and how they could be lifted to family-based or feature-based verification. Moreover, research on S&S during the evolution of configurable systems and their management through all projections should be a goal for future research. Since none of the publications addressed S&S simultaneously, we propose to establish techniques and methods to provide an integrated view on S&S in configurable systems.

In future work, we plan to expand on our mapping study by conducting a more detailed systematic literature review, allowing us to obtain more in-depth insights and improving their validity. Moreover, we are working on adopting product-line techniques for assessing the security of configurable systems. We plan to advance this research based on the insights we obtained in this mapping study, for instance, by incorporating security standards we identified and considering different projections. Finally, it would be interesting to discuss our insights with practitioners to understand their needs.

ACKNOWLEDGMENTS

This research has been supported by the German Federal Ministry of Education and Research (16KIS0526), the German Federal Ministry of Transport and Digital Infrastructures (19H17006D) as well as the German Research Foundation (LE 3382/3-1, SA 465/49-3).

REFERENCES

[1] Iago Abal, Jean Melo, Ştefan Stănciulescu, Claus Brabrand, Márcio de Medeiros Ribeiro, and Andrzej Wąsowski. 2018. Variability Bugs in Highly Configurable Systems: A Qualitative Analysis. *ACM Transactions on Software Engineering and Methodology* 26, 3 (2018).

[2] Muhammad Abbas, Robbert Jongeling, Claes Lindskog, Eduard Paul Enoiu, Mehrdad Saadatmand, and Daniel Sundmark. 2020. Product Line Adoption in Industry: An Experience Report from the Railway Domain. In *SPLC*. ACM.

[3] Mathieu Acher, Guillaume Bécan, Benoit Combemale, Benoit Baudry, and Jean-Marc Jézéquel. 2015. Product Lines Can Jeopardize Their Trade Secrets. In *ESEC/FSE*. ACM.

[4] Bestoun S. Ahmed, Kamal Z. Zamli, Wasif Afzal, and Miroslav Bures. 2017. Constrained Interaction Testing: A Systematic Literature Study. *IEEE Access* 5 (2017).

[5] Shaukat Ali, Paolo Arcaini, Ichiro Hasuo, Fuyuki Ishikawa, and Nian-Ze Lee. 2019. Towards a Framework for the Analysis of Multi-Product Lines in the Automotive Domain. In *VaMoS*. ACM.

[6] Sofia Ananieva, Sandra Greiner, Thomas Kühn, Jacob Krüger, Lukas Linsbauer, Sten Grüner, Timo Kehrer, Heiko Klare, Anne Koziolek, Henrik Lönn, Sebastian Krieter, Christoph Seidl, S. Ramesh, Ralf Reussner, and Bernhard Westfechtel. 2020. A Conceptual Model for Unifying Variability in Space and Time. In *SPLC*. ACM.

[7] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. 2013. *Feature-Oriented Software Product Lines*. Springer.

[8] Stephan Baumgart and Joakim Fröberg. 2016. Functional Safety in Product Lines – A Systematic Mapping Study. In *SEAA*. IEEE.

[9] Stephan Baumgart, Joakim Fröberg, and Sasikumar Punnekkat. 2012. Towards Efficient Functional Safety Certification of Construction Machinery Using a Component-Based Approach. In *PLEASE*. IEEE.

[10] Stephan Baumgart, Joakim Fröberg, and Sasikumar Punnekkat. 2014. Industrial Challenges to Achieve Functional Safety Compliance in Product Lines. In *SEAA*. IEEE.

[11] Stephan Baumgart, Joakim Fröberg, and Sasikumar Punnekkat. 2015. Enhancing Model-Based Engineering of Product Lines by Adding Functional Safety. In *MASE*. CEUR-WS.org.

[12] Stephan Baumgart, Xiaodi Zhang, Joakim Fröberg, and Sasikumar Punnekkat. 2014. Variability Management in Product Lines of Safety Critical Embedded Systems. In *ICES*. IEEE.

[13] Guillaume Bécan, Mathieu Acher, Jean-Marc Jézéquel, and Thomas Menguy. 2015. On the Variability Secrets of an Online Video Generator. In *VaMoS*. ACM.

[14] Benjamin Behringer, Martina Lehser, and Steffen Rothkugel. 2014. Towards Feature-Oriented Fault Tree Analysis. In *COMPSAC*. IEEE.

[15] Thorsten Berger, Marsha Chechik, Timo Kehrer, and Manuel Wimmer (Eds.). 2019. *Software Evolution in Time and Space: Unifying Version and Variability Management*. Schloss Dagstuhl.

[16] Sara Bessling and Michaela Huhn. 2012. Formal Safety Analysis and Verification in the Model Driven Development of a Pacemaker Product Line. In *Dagstuhl-Workshop MBEES*. Schloss Dagstuhl.

[17] Danilo Beuche. 2012. Modeling and Building Software Product Lines with Pure::Variants. In *SPLC*. ACM.

[18] Stefan Biffl, Matthias Eckhart, Arndt Lüder, and Edgar Weippl (Eds.). 2019. *Security and Quality in Cyber-Physical Systems Engineering*. Springer.

[19] Rosana T. V. Braga, Onofre Trindade, Kalinka R. L. J. Castelo Branco, and Jaejoon Lee. 2012. Incorporating Certification in Feature Modelling of an Unmanned Aerial Vehicle Product Line. In *SPLC*. ACM.

[20] Lucas Bressan, André L. de Oliveira, and Fernanda Campos. 2020. An Approach to Support Variant Management on Safety Analysis Using CHESS Error Models. In *EDCC*. IEEE.

[21] Marimuthu C and K. Chandrasekaran. 2017. Systematic Studies in Software Product Lines: A Tertiary Study. In *SPLC*. ACM.

[22] Krzysztof Czarnecki, Paul Grünbacher, Rick Rabiser, Klaus Schmid, and Andrzej Wąsowski. 2012. Cool Features and Tough Decisions: A Comparison of Variability Modeling Approaches. In *VaMoS*. ACM.

[23] Yanja Dajsuren and Mark van den Brand (Eds.). 2019. *Automotive Systems and Software Engineering: State of the Art and Future Trends*. Springer.

[24] André L. de Oliveira, Rosana T. V. Braga, Paulo Masiero, David Parker, Yiannis Papadopoulos, Ibrahim Habli, and Tim Kelly. 2019. Variability Management in Safety-Critical Systems Design and Dependability Analysis. *Journal of Software: Evolution and Process* 31, 8 (2019).

[25] André L. de Oliveira, Rosana T. V. Braga, Paulo C. Masiero, Yiannis Papadopoulos, Ibrahim Habli, and Tim Kelly. 2014. A Model-Based Approach to Support the Automatic Safety Analysis of Multiple Product Line Products. In *SBESC*. IEEE.

[26] Andreas Demuth, Roberto E. Lopez-Herrejon, and Alexander Egyed. 2014. Automatic and Incremental Product Optimization for Software Product Lines. In *ICST*. IEEE.

[27] Simon Diemert, Laure Millet, and Jeff Joyce. 2020. Safety Properties of Hybrid System Product Lines. In *SysCon*. IEEE.

[28] Tharam Dillon, Chen Wu, and Elizabeth Chang. 2010. Cloud Computing: Issues and Challenges. In *AINA*. IEEE.

[29] Dominik Domis, Rasmus Adler, and Martin Becker. 2015. Integrating Variability and Safety Analysis Models Using Commercial UML-Based Tools. In *SPLC*. ACM.

[30] Cristina Gacek and Michalis Anastasopoules. 2001. Implementing Product Line Variabilities. In *SSR*. ACM.

[31] Barbara Gallina. 2015. Towards Enabling Reuse in the Context of Safety-Critical Product Lines. In *PLEASE*. IEEE.

[32] Fatima Z. Hammani. 2014. Survey of Non-Functional Requirements Modeling and Verification of Software Product Lines. In *RCIS*. IEEE.

[33] Fatima Z. Hammani, Maryem Rhanoui, and Bouchra El Asri. 2014. Towards a Variable Non-Functional Requirements Integration for Component-Based Product Line: A Generic Approach. In *WCCS*. IEEE.

[34] Jose-Miguel Horcas, Mónica Pinto, and Lidia Fuentes. 2016. An Automatic Process for Weaving Functional Quality Attributes Using a Software Product Line Approach. *Journal of Systems and Software* 112 (2016).

[35] Jose-Miguel Horcas, Mónica Pinto, and Lidia Fuentes. 2017. Green Configurations of Functional Quality Attributes. In *SPLC*. ACM.

[36] ISO/IEC 25010:2011-03 2011. *Systems and Software Engineering – SQuaRE - System and Software Quality*. Standard. ISO.

[37] ISO/IEC 27000:2018 2018. *Information Technology – Security Techniques – Information Security Management Systems*. Standard. ISO.

[38] Joint Task Force Transformation Initiative. 2012. *Guide for Conducting Risk Assessments*. Technical Report NIST SP 800-30r1. NIST.

[39] Hermann Kaindl, Roman Popp, and David Raneburger. 2015. Towards Reuse in Safety Risk Analysis Based on Product Line Requirements. In *RE*. IEEE.

[40] Kyo C. Kang, Sholom G. Cohen, James A. Hess, William E. Novak, and A. Spencer Peterson. 1990. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Technical Report CMU/SEI-90-TR-21. Carnegie Mellon University.

[41] Michael Käßmeyer, Peter Bazan, Markus Schurius, Rüdiger Berndt, and Reinhard German. 2016. A Formal Model for Stateful and Variant-Rich Automotive Functions. In *ISSREW*. IEEE.

[42] Michael Käßmeyer, David Santiago Velasco Moncada, and Markus Schurius. 2015. Evaluation of a Systematic Approach in Variant Management for Safety-Critical Systems Development. In *EUC*. IEEE.

[43] Michael Käßmeyer, Michael Schulze, and Markus Schurius. 2015. A Process to Support a Systematic Change Impact Analysis of Variability and Safety in Automotive Functions. In *SPLC*. ACM.

[44] Tim Kelly, Ibrahim Habli, Paulo Cesar Masiero, André L. de Oliveira, Rosana T. V. Braga, and Yiannis Papadopoulos. 2016. Model-Based Safety Analysis of Software Product Lines. *International Journal of Embedded Systems* 8, 5/6 (2016).

[45] Andy Kenner. 2020. Model-Based Evaluation of Vulnerabilities in Software Systems. In *SPLC*. ACM.

[46] Andy Kenner, Stephan Dassow, Christian Lausberger, Jacob Krüger, and Thomas Leich. 2020. Using Variability Modeling to Support Security Evaluations: Virtualizing the Right Attack Scenarios. In *VaMoS*. ACM.

[47] Barbara A. Kitchenham, David Budgen, and O. Pearl Brereton. 2015. *Evidence-Based Software Engineering and Systematic Reviews*. CRC Press.

[48] John C. Knight. 2002. Safety Critical Systems: Challenges and Directions. In *ICSE*. ACM.

[49] Sahar Kokaly, Rick Salay, Mehrdad Sabetzadeh, Marsha Chechik, and Tom Maibaum. 2016. Model Management for Regulatory Compliance: A Position Paper. In *MiSE*. ACM.

[50] Andrew J. Kornecki and Janusz Zalewski. 2010. Safety and Security in Industrial Control. In *CSIIRW*. ACM.

[51] Matthias Kowal, Sandro Schulze, and Ina Schaefer. 2013. Towards Efficient SPL Testing by Variant Reduction. In *VariComp*. ACM.

[52] Sebastian Krieter, Jacob Krüger, Nico Weichbrodt, Vasily A. Sartakov, Rüdiger Kapitza, and Thomas Leich. 2018. Towards Secure Dynamic Product Lines in the Cloud. In *ICSE*. ACM.

[53] Sebastian Krieter, Thomas Thüm, Sandro Schulze, Reimar Schröter, and Gunter Saake. 2018. Propagating Configuration Decisions with Modal Implication Graphs. In *ICSE*. ACM.

[54] Jacob Krüger, Sofia Ananieva, Lea Gerling, and Eric Walkingshaw. 2020. VariVolution. In *SPLC*. ACM.

[55] Jacob Krüger, Wardah Mahmood, and Thorsten Berger. 2020. Promote-pl: A Round-Trip Engineering Process Model for Adopting and Evolving Product Lines. In *SPLC*. ACM.

[56] Jacob Krüger, Sebastian Nielebock, Sebastian Krieter, Christian Diedrich, Thomas Leich, Gunter Saake, Sebastian Zug, and Frank Ortmeier. 2017. Beyond Software Product Lines: Variability Modeling in Cyber-Physical Systems. In *SPLC*. ACM.

[57] Remo Lachmann, Sascha Lity, Sabrina Lischke, Simon Beddig, Sandro Schulze, and Ina Schaefer. 2015. Delta-Oriented Test Case Prioritization for Integration Testing of Software Product Lines. In *SPLC*. ACM.

[58] Robert Lindohf, Jacob Krüger, Erik Herzog, and Thorsten Berger. 2021. Software Product-Line Evaluation in the Large. *Empirical Software Engineering* 26, 30 (2021).

[59] Malte Lochau, Johannes Bürdek, Stefan Hölzle, and Andy Schürr. 2017. Specification and Automated Validation of Staged Reconfiguration Processes for Dynamic Software Product Lines. *Software and Systems Modeling* 16, 1 (2017).

[60] Philipp Lohmüller and Bernhard Bauer. 2019. Software Product Line Engineering for Safety-Critical Systems. In *MODELSWARD*. SCITEPRESS.

[61] Sara Mahdavi-Hezavehi, Matthias Galster, and Paris Avgeriou. 2013. Variability in Quality Attributes of Service-Based Software Systems: A Systematic Literature Review. *Information and Software Technology* 55, 2 (2013).

[62] Niloofar Mansoor, Jonathan A. Saddler, Bruno Silva, Hamid Bagheri, Myra B. Cohen, and Shane Farritor. 2018. Modeling and Testing a Family of Surgical Robots: An Experience Report. In *ESEC/FSE*. ACM.

[63] Ethan T. McGee and John D. McGregor. 2015. Composition of Proof-Carrying Architectures for Cyber-Physical Systems. In *SPLC*. ACM.

[64] Ethan T. McGee and John D. McGregor. 2016. Using Dynamic Adaptive Systems in Safety-Critical Domains. In *SEAMS*. ACM.

[65] Gary McGraw. 2006. *Software Security: Building Security In.* Addison-Wesley.

[66] Jens Meinicke, Thomas Thüm, Reimar Schröter, Fabian Benduhn, Thomas Leich, and Gunter Saake. 2017. *Mastering Software Variability with FeatureIDE*. Springer.

[67] Daniel Mellado and Haralambos Mouratidis. 2012. Towards the Extension of Secure Tropos Language to Support Software Product Lines Development. In *WOSIS*. SCITEPRESS.

[68] Daniel Mellado, Haralambos Mouratidis, and Eduardo Fernández-Medina. 2014. Secure Tropos Framework for Software Product Lines Requirements Engineering. *Computer Standards & Interfaces* 36, 4 (2014).

[69] Oslien Mesa, Reginaldo Vieira, Marx Viana, Vinicius H. S. Durelli, Elder Cirilo, Marcos Kalinowski, and Carlos Lucena. 2018. Understanding Vulnerabilities in Plugin-Based Web Systems: An Exploratory Study of Wordpress. In *SPLC*. ACM.

[70] Raphael Muniz, Larissa Braz, Rohit Gheyi, Wilkerson de L. Andrade, Baldoino Fonseca, and Márcio de Medeiros Ribeiro. 2018. A Qualitative Analysis of Variability Weaknesses in Configurable Systems with #ifdefs. In *VaMoS*. ACM.

[71] Alexandr Murashkin, Michał Antkiewicz, Derek Rayside, and Krzysztof Czarnecki. 2013. Visualization and Exploration of Optimal Variants in Product Line Engineering. In *SPLC*. ACM.

[72] Varvana Myllärniemi, Mikko Raatikainen, and Tomi Männistö. 2015. Representing and Configuring Security Variability in Software Product Lines. In *QoSA*. ACM.

[73] Damir Nešić, Jacob Krüger, Ștefan Stănciulescu, and Thorsten Berger. 2019. Principles of Feature Modeling. In *ESEC/FSE*. ACM.

[74] Damir Nešić and Mattias Nyberg. 2018. Verifying Contract-Based Specifications of Product Lines Using Description Logic. In *DL*. IEEE.

[75] Damir Nešić, Mattias Nyberg, and Barbara Gallina. 2019. Constructing Product-Line Safety Cases from Contract-Based Specifications. In *SAC*. ACM.

[76] Michael Nieke, Lukas Linsbauer, Jacob Krüger, and Thomas Leich. 2019. Second International Workshop on Variability and Evolution of Software-Intensive Systems (VariVolution 2019). In *SPLC*. ACM.

[77] Markus Oertel, Michael Schulze, and Thomas Peikenkamp. 2014. Reusing a Functional Safety Concept in Variable System Architectures. In *MODELS*. CEUR-WS.org.

[78] Sven Peldszus, Daniel Strüber, and Jan Jürjens. 2018. Model-Based Security Analysis of Feature-Oriented Software Product Lines. In *GPCE*. ACM.

[79] Tobias Pett, Domenik Eichhorn, and Ina Schaefer. 2020. Risk-Based Compatibility Analysis in Automotive Systems Engineering. In *MODELS*. ACM.

[80] Charles P. Pfleeger and Shari Lawrence Pfleeger. 2002. *Security in Computing*. Prentice Hall.

[81] Klaus Pohl, Günter Böckle, and Frank J. van der Linden. 2005. *Software Product Line Engineering*. Springer.

[82] Mona Rahimi, Wandi Xiong, Jane Cleland-Huang, and Robyn Lutz. 2017. Diagnosing Assumption Problems in Safety-Critical Products. In *ASE*. IEEE.

[83] Vítor Rodrigues, Simone Donetti, and Ferruccio Damiani. 2019. Certifying Delta-Oriented Programs. *Software and Systems Modeling* 18, 5 (2019).

[84] Ricardo J. Rodríguez and Sasikumar Punnekkat. 2014. Cost Optimisation in Certification of Software Product Lines. In *ISSRE*. IEEE.

[85] Aleksandra Salikiryaki, Iliana Petrova, and Stephan Baumgart. 2015. Graphical Approach for Modeling of Safety and Variability in Product Lines. In *SEAA*. IEEE.

[86] Spyridon Samonas and David Coss. 2014. The CIA Strikes Back: Redefining Confidentiality, Integrity and Availability in Security. *Journal of Information System Security* 10, 3 (2014).

[87] Luis E. Sanchéz, J. Andrés Díaz-Pace, Alejandro Zunino, Sabine Moisan, and Jean-Paul Rigault. 2014. An Approach for Managing Quality Attributes at Runtime Using Feature Models. In *SBCARS*. IEEE.

[88] Jane D. A. Sandim Eleutério, Felipe N. Gaia, Andrea Bondavalli, Paolo Lollini, Genaína N. Rodrigues, and Cecília M. Fischer Rubira. 2016. On the Dependability for Dynamic Software Product Lines: A Comparative Systematic Mapping Study. In *SEAA*. IEEE.

[89] Ina Schaefer, Rick Rabiser, Dave Clarke, Lorenzo Bettini, David Benavides, Goetz Botterweck, Animesh Pathak, Salvador Trujillo, and Karina Villela. 2012. Software Diversity: State of the Art and Perspectives. *International Journal on Software Tools for Technology Transfer* 14, 5 (2012).

[90] Michael Schulze, Jan Mauersberger, and Danilo Beuche. 2013. Functional Safety and Variability: Can It Be Brought Together?. In *SPLC*. ACM.

[91] Michael E. Shin, Hassan Gomaa, and Don Pathirage. 2018. A Software Product Line Approach for Feature Modeling and Design of Secure Connectors. In *ICSOFT*. SCITEPRESS.

[92] Seppo Sierla, Bryan M. O'Halloran, Heikki Nikula, Nikolaos Papakonstantinou, and Irem Y. Tumer. 2014. Safety Analysis of Mechatronic Product Lines. *Mechatronics* 24, 3 (2014).

[93] Danillo Sprovieri, Nikolaos Argyropoulos, Carine Souveyet, Raúl Mazo, Haralambos Mouratidis, and Andrew Fish. 2016. Security Alignment Analysis of Software Product Lines. In *ES*. IEEE.

[94] Mikael Svahnberg, Jilles van Gurp, and Jan Bosch. 2005. A Taxonomy of Variability Realization Techniques. *Software: Practice and Experience* 35, 8 (2005).

[95] Thomas Thüm, Sven Apel, Christian Kästner, Ina Schaefer, and Gunter Saake. 2014. A Classification and Survey of Analysis Strategies for Software Product Lines. *ACM Computing Surveys* 47, 1 (2014).

[96] Oleksandr Tomashchuk. 2020. Threat and Risk Management Framework for eHealth IoT Applications. In *SPLC*. ACM.

[97] Salvador Trujillo, Iñaki Alonso, Brahim Hamid, David González, Manuel Blanco, and Huaxi (Y.) Zhang. 2011. Towards Variability Support for Security and Dependability Patterns: A Case Study. In *SPLC*. ACM.

[98] Ángel J. Varela-Vaca, Rafael M. Gasca, Jose A. Carmona-Fombella, and María T. Gómez-López. 2020. AMADEUS: Towards the AutoMAteD secUrity teSting. In *SPLC*. ACM.

[99] Ángel J. Varela-Vaca, Rafael M. Gasca, Rafael Ceballos, María T. Gómez-López, and Pedro Bernáldez Torres. 2019. CyberSPL: A Framework for the Verification of Cybersecurity Policy Compliance of System Configurations Using Software Product Lines. *Applied Sciences* 9, 24 (2019).

[100] Karina Villela, Taslim Arif, and Damiano Zanardini. 2012. Towards Product Configuration Taking into Account Quality Concerns. In *SPLC*. ACM.

[101] Lisong Wang, Sijie Li, Ou Wei, Mingyu Huang, and Jun Hu. 2018. An Automated Fault Tree Generation Approach With Fault Configuration Based on Model Checking. *IEEE Access* 6 (2018).

[102] Fredrik Warg, Hans Blom, Jonas Borg, and Rolf Johansson. 2019. Continuous Deployment for Dependable Systems with Continuous Assurance Cases. In *ISSREW*. IEEE.

[103] Danny Weyns. 2019. Software Engineering of Self-Adaptive Systems. In *Handbook of Software Engineering*. Springer.

[104] Beth Wilson and Bobbi Young. 2020. Cyber Secure and Resilient Approaches for Feature Based Variation Management. In *SSS*. IEEE.

[105] Guoheng Zhang, Huilin Ye, and Yuqing Lin. 2014. Quality Attribute Modeling and Quality Aware Product Configuration in Software Product Lines. *Software Quality Journal* 22, 3 (2014).