Co-engineering of Safety and Security Life Cycles for Engineering of Automotive Systems

Robert Bramberger and Helmut Martin

Virtual Vehicle Research GmbH, Graz, Austria, {Robert.Bramberger, Helmut.Martin}@v2c2.at

Barbara Gallina

Mälardalen University, Västerås, Sweden; Barbara.Gallina@mdh.se

Christoph Schmittner

AIT Austrian Institute of Technology GmbH, Vienna, Austria; Christoph.Schmittner@ait.ac.at

Abstract

Nowadays systems are becoming more and more connected. Consequently, the co-engineering of (cyber)security and safety life cycles becomes paramount. Currently, no standard provides a structured co-engineering process to facilitate the communication between safety and security engineers. In this paper, we propose a process for co-engineering safety and security by the explicit systematization and management of commonalities and variabilities, implicitly stated in the requirements of the different standards. Our process treats the safety and security life cycles as members of a security-informed safety-oriented process line and so it forces safety and security engineers to come together and brainstorm on what might be considered a commonality and what might be considered a variability. We illustrate the usage of our process by systematizing commonalities and variabilities at risk analysis phase in the context of ISO 26262 and SAE J3061. We then draw lessons learnt. Finally, we sketch some directions for future work.

Keywords: Security-informed Safety, ISO 26262, SAE J3061, Security-informed Safety-oriented Process Line Engineering (SiSoPLE), HARA, TARA

1 Introduction

Nowadays, systems are becoming more and more connected and offer advanced functionalities. In the automotive domain, for instance, with the advent of Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), and even vehicle to cloud (V2C) communication, road vehicles are playing an active and major role within the Internet of Things (IoT), offering new communication-centred functionalities aimed at increasing safety by e.g., decongesting traffic via roadworks-related communication. However, connectivity may threaten safety, due to numerous security threats, which, as recently surveyed by ENISA [3], are emerging.

Consequently, the co-engineering of (cyber)security and safety life cycles becomes paramount. Currently, no standard provides a structured co-engineering process to facilitate the communication between safety and cybersecurity engineers. ISO 26262 [5] introduces a standardized safety life cycle, which needs to be complemented by requirements stemming from cybersecurity standards (e.g. the upcoming cybersecurity standard ISO/ SAE 21434 [6]) and/or guidelines (e.g. SAE J3061 [8]). SAE J3061 is the only published guidebook that provides suggestions for considering both concerns. Specifically, SAE J3061 proposes a life cycle for handling cybersecurity which is based on the ISO 26262 safety lifecycle. The reason for this analogous life cycle is to allow organizations with safety processes based on ISO 26262 to use a **common framework for cybersecurity and safety** to facilitate the development of a tailored cybersecurity process by capitalizing on aspects of an organization's existing safety process that are common to both cybersecurity and safety, for example, the supporting process procedures and templates.

Thus, the co-engineering of safety and (cyber)security life cycles and, more broadly, the co-engineering of different mono-concern life cycles can be facilitated by the explicit systematization and management of commonalities and variabilities, implicitly stated in the requirements of the different standards. This leads to the engineering of a Security-informed Safety-oriented Process Line (SiSoPL) [13].

In this paper, we extend our initial thoughts published in [16], [29] and we engineer an automotive SiSoPL by using the tool-chain constituted of Eclipse Process Framework Composer (EPF-C) [33]. EPF-C permits users to engineer processes in compliance with a SPEM (Software & Systems Process Engineering Metamodel) 2.0-like language [30], and BVR Tool [31], which permits users to orthogonally manage variability at process level in compliance with the Base Variability Resolution (BVR) language [18]. The toolchain (shown in [21]) obtained via the integration between EPF-C and BVR Tool is part of the AMASS tool platform, delivered by the AMASS project [1], [26], [28], [29] and hosted by

OpenCert [32]. The engineered SiSoPL embraces the automotive regulations comprising ISO 26262 and SAE J3061 and focuses on the risk analysis phase, as initially done in [12], where the automotive Security-informed Safety terminological framework for retrieving the implicit commonalities was proposed. Finally, from our engineered SiSoPL, we derive a single security-informed safety-oriented process targeting the security-informed safety concept of a car2car communication management unit.

The rest of the paper is organized as follows. In Section 2, we recall essential background. In Section 3, we clearly state the problem. In Section 4, we explain our methodology for SiSoPL engineering. In Section 5, we apply our methodology and report about our lessons learned. Finally, in Section 6, we draw our concluding remarks and sketch future work.

2 Background and related work

In this section, we present the background information on which we base our work.

2.1 Relevant safety and cybersecurity standards

In this sub-section, we provide an overview about the standards for functional safety and cybersecurity targeting non-autonomous road vehicles.

ISO 26262 [5] is the automotive functional safety standard (first release 2011). It describes a safety life cycle for the development of safety-related automotive systems with the purpose of guaranteeing absence of unreasonable risk due to hazards caused by malfunctioning behaviour of electrical/electronic systems. The scope in edition 2018 has been extended from passenger cars to further road vehicles. Now it also deals with trucks, busses and motorcycles. ISO 26262 provides an informative guideline on "potential interaction of functional safety with cybersecurity".

The life cycle is structured into phases. The first phase, called the concept phase, starts with the item definition, i.e., a description of the system with regard to its functionality, interfaces, environment, etc. Once the item is defined, the HARA (Hazard Analysis and Risk Assessment) is performed to identify/categorize/evaluate hazardous events, i.e., the combination of hazard (potential source of harm) and operational situations (scenarios that can occur during vehicle's life). Harm is defined as the physical injury or damage to the health of persons. To minimize harm, unreasonable risk has to be reduced. To support risk evaluation, ISO 26262 has introduced the notion of Automotive Safety Integrity Level (ASIL), which can assume one out of five values, ranging from negligible QM and ASIL A to ASIL D, where D represents the most stringent level. An ASIL is assigned based on the severity, the exposure, and the controllability of the hazardous event. The assignment of the ASIL constrains the stringency of the following activities within the safety life cycle. Another parameter used to influence the stringency is the recommendation level (neutral, recommended, highly recommended), abbreviated as RecL, which is typically assigned in conjunction with the ASIL to provide guidance on method application.

The result of the concept phase is the functional safety concept, represented by the set of safety goals (top-level safety requirements) derived from the HARA findings.

SAE J3061 [8] provides high level guiding principles for cybersecurity for the complete engineering life cycle. It proposes more concrete communication paths between functional safety and cybersecurity engineering.

The cybersecurity life cycle initiates at the concept phase with the feature (i.e., system) definition in which the scope of the feature is specified with respect to physical boundaries, cybersecurity perimeter, and trust boundaries of the feature. After that, TARA (Threat Analysis and Risk Assessment) is performed. TARA is an analysis technique applied to identify potential threats to a feature and to assess the risk associated with the identified threats. Cybersecurity goals are derived and formulated for each of the highest risk potential threats documented in the TARA. SAE J3061 does not introduce a specific notion for cybersecurity level. However, it outlines a sampling of security analysis methods for performing the TARA such as the method used by the E-Safety Vehicle Intrusion Protected Applications (EVITA) program, the Threat, Vulnerabilities, and implementation Risks Analysis (TVRA) method, the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) method, and the HEAling Vulnerabilities to ENhance Software Security and Safety (HEAVENS) method and attack tree information. These methods propose possible security levels.

ISO/SAE 21434 defines requirements related to cybersecurity risk management for road vehicles that include electrical and electronic (E/E) systems. A joint working group of ISO and SAE experts develop ISO/SAE 21434. It will replace SAE J3061 and provides a framework, which supports the establishment of a cybersecurity culture during the complete product life cycle. Since cybersecurity risks can increase during the products lifetime it demands a management system, which is able to monitor changes in the threat landscape, vulnerabilities, etc. and provide updates from postproduction until decommissioning.

The standard recommends the definition of specific metrics for process rigour and risk level. In the presented approach the SecRL (Security Risk Level) has been defined to quantify risk and to perform variability management. In general, different security risk levels are needed for different attributes (privacy, operational, financial, safety). The paper at hand only deals with a risk level for functional safety. The standard is currently under development. The expected date of publishing is Q4 2020.

In addition, standards demand an established engineering process according to state of the art automotive quality standards (e.g. ASPICE, CMMI, IATF 16949).

2.2 Co-engineering life cycle for safety and security

In this subsection, we recall the method for co-engineering used in the core of this paper.

Security-informed Safety-oriented Process Line Engineering (SiSoPLE) [13] is a co-engineering method, which represents the extension of SoPLE, Safety-oriented Process Line Engineering [14], [15]. Similar to SoPLE, SiSoPLE consists of a two-phase method for engineering families of safety life cycles/processes. The first phase is aimed at engineering the domain from a process perspective i.e., identifying and systematizing process-related commonalities and variabilities, focusing on security-informed safety-related commonalities and variabilities, in order to concurrently engineer a set of processes. The second phase is aimed at deriving single processes via selection and composition of commonalities and variabilities. From a tooling perspective, SiSoPLE as well as SoPLE can be supported by the integration between EPF Composer, recently re-brought to life [20], and BVR Tool [31]. This integration was qualitatively evaluated as promising in [11] and its implementation was presented in [20]. To make the paper self-contained, we recall basic information regarding EPF Composer and BVR Tool.

EPF Composer implements a metamodel which exhibit a satisfactory overlapping with the SPEM (Software & Systems Process Engineering Metamodel) 2.0 language [30]. EPF Composer enables authoring, tailoring and deploying engineering life cycles and processes. This means that process structures containing all necessary process elements (e.g., activities, tasks, roles, work products, etc.) can be specified.

The BVR Tool implements the BVR (Base Variability Resolution) [18] language, built on top of CVL (Common Variability Language) [19] enable variability modelling in the context of the engineering of families of safety-critical systems. BVR enables orthogonal variability management for any model (called Base model), instance of a Meta-Object Facility (MOF)-compliant metamodel. Via the BVR Tool, variability engineers create three kinds of models:

VSpec models specify Feature-Oriented Domain Analysis (FODA) [22] -like models. To specify cross-branches constraints, which limit inclusion/exclusion within a subtree based on choices on other subtrees, Basic Constraint Language (BCL) is used.

Resolution models define the desired inclusion/exclusion choices for the specific configuration/resolution.

Realization models specify the placement fragments (i.e., sets of elements forming conceptual holes in a base model, which may be replaced by replacement fragments) and replacements within the fragment substitutions. A Fragment substitution is an operation that, if executed, substitutes a model fragment (placement fragment) with another (replacement fragment).

2.3 Safety and security co-analysis

In this subsection, we recall the method for co-analysis used in the core of this paper.

EVITA [4] is used to quantify the risk of potential cyberattacks. A risk level is derived based on "attack potential", "attack probability", "severity" and "controllability". It is a criterion that indicates the risk that functional safety can possibly be levered out by an attacker in certain circumstances.

Based on HARA in [24] SAHARA (Security-Aware Hazard Analysis and Risk Assessment) was introduced. It combines HARA from the safety and the STRIDE approach from the security domain. The intention of SAHARA is to identify security issues which can have an impact to safety concepts on system level. It also considers impacts which can occur because of safety issues.

FMVEA (Failure Mode, Vulnerabilities and Effect Analysis), [27], extends the FMEA and performs a combined safety and security analysis. It considers threat modes and failure modes. Threat modes describe possible ways how the security attribute of a component may fail caused by vulnerabilities. FMVEA determines the probability of a threat mode based on identified attack scenarios and vulnerabilities.

In [23] the relationship between HARA and TARA was investigated with regard to a joint assurance case.

3 Problem statement

Since vehicles provide highly interconnected system functions realized in software, the systems are no longer isolated. They become cyber-physical and cybersecurity has to be part of the centre of interest. Existing safety-related processes have to be expanded with methods like threat analysis and risk assessment and attack tree analysis.

An important aspect is the identification of relationships between cybersecurity and safety because freedom of interference has to be guaranteed. It is possible that security threats have impact to safety, if safety functions are implemented in software. Whereas safety deals with hazards and mishaps cybersecurity addresses threats resulting from malicious intent from external to the E/E system.

The methodology described in the next chapter is intended to identify all possible ways how functional safety may be violated in the different development lifecycle phases. In a combined process cybersecurity and safety risks will be identified jointly. In this context it has to be considered that there are risks which are only related to safety issues (e.g. hardware failure) and risks which are only related to cybersecurity (e.g. attackers want to capture personal data). Cybersecurity risks without safety relation will be possibly

identified but they are out of scope from the perspective of the paper at hand.

Based on analogies between safety and cybersecurity it is useful to define processes, which are integrating both topics. An integrated point of view is necessary because joint safety and security analysis will lead to measures, which have the objective to mitigate identified risks, which can be caused by both disciplines.

In the initial situation process developers have to work with standards which describe separated topics. Engineering teams in companies need an integrated development process which deals with quality, safety and security on different levels for different projects. Process developers have to harmonize several standards in their processes and provide evidence to all engineering areas.

Highly connected vehicles need a process to track the security status during the whole lifetime because previously unknown attacks may have the opportunity to compromise functional safety.

4 Methodology to define a process flow

To define a joint safety and security process, based on available but separated processes, it is necessary to have a systematic procedure to identify commonalities and variabilities. The proposed way is to use SiSoPLE, which is able to define joint processes. This chapter describes the development of two independent standard compliant processes for safety and security. These processes are the base for variants with project specific ASIL, SecRL and quality. The safety and security co-analysis delivers ASIL and SecRL, which are parameters for the variability management. The underlying workflow is illustrated in Figure 1.

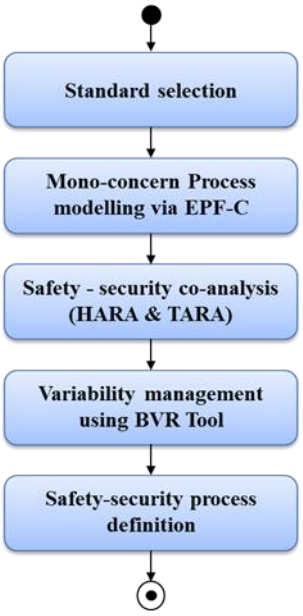

Figure 1 Workflow for safety-security co-engineering process

Activities in cross concern applications, which have to be executed in any case, are called safety security co-engineering activities (instead of the single concern "commonality"). This term definition allows the extension of indicated activities, because it is the intention to "maximize" co-engineering activities and reduce variabilities if it is possible. Co-engineering capable methods can deal with both areas and they do not need to be a commonality in a strict sense. In this generalized view, it is sufficient that the methods head to the same goal.

4.1 Standard selection

The first step to create a process is to define which requirements it must fulfil. At least standards, which are demanded by legislator and customers, have to be considered.

4.2 Process modelling

The base process and the related model contain all activities, which can possibly be part of the development process and are directly related to the underlying standards. This means that all activities which may be needed for any ASIL and any SecRL are modelled. Later on, company specific activities and realisations will be added in the process definition step. Finally, for all concerns process models (mono concern models) are available. They are the basis for the following variability management.

4.3 Safety and security co-analysis

As recalled in Section 2.1, safety and security co-analysis is an important step in the concept phase, which has major impact to the following engineering activities. The described approach uses the resulting ASIL and SecRL as parameters to manage variability and define the co-engineering process.

Co-analysis in the concept phase has to make sure that interaction between different concerns is considered, because it should ensure that cybersecurity issues are considered as well as safety. The approach should guarantee that any additional potential hazards will be identified, which would stay undiscovered if only one discipline is examined in an isolated way. HARA and TARA must be performed in parallel but interweaved and consider potential dependencies between safety and security. The management of interaction between safety and security in an assessment is addressed in specific research papers (see section 2).

Identification of hazards and potential causes is an indispensable prerequisite for a safe and secure system. Hazards and threats from both areas need to be identified because unknown issues can lead to unsafe control actions, independent whether the cause is related to a hardware fault (classic safety-oriented view) or to a security issue. The goal is to define measures that are appropriate to mitigate any identified risks. To make sure that measures from competitive disciplines do not influence each other in a non-admissible way, a trade-off in the risk reduction measures has to be

considered. The impact of each single safety and security measure needs an evaluation to find a balance.

Finally, arguments have to be collected in the assurance case, which covers the integrated and harmonized safety and security case, to show that the implemented measures are conform with underlying standards.

4.4 Variability management

Variability management is based on the defined ASIL and SecRL parameter set (see Figure 2). These two parameters have a major impact to the extent of the minimum required process activities. Tailoring of the base processes to a project specific multi-concern process means that unneeded activities

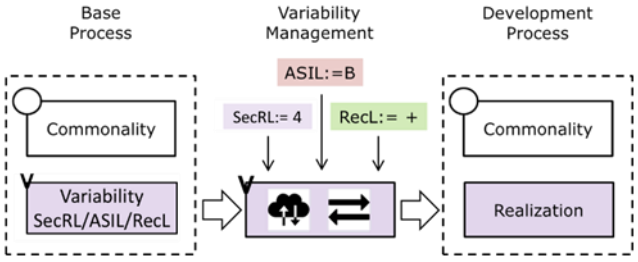

Figure 2 Variability management based on parameters [2]

are removed, and new project specific activities are added. Standards or company specific regulations demand for the application of a defined set of methods for a particular ASIL or SecRL. The development process must deal with variability because ASIL and SecRL varies for different items and in different projects.

BVR provides a mechanism to change activities and methods for various items according to different parameter sets. This feature is implemented by the usage of choices and constraints in the VSpec and the Resolution diagram (see Figure 3) of the

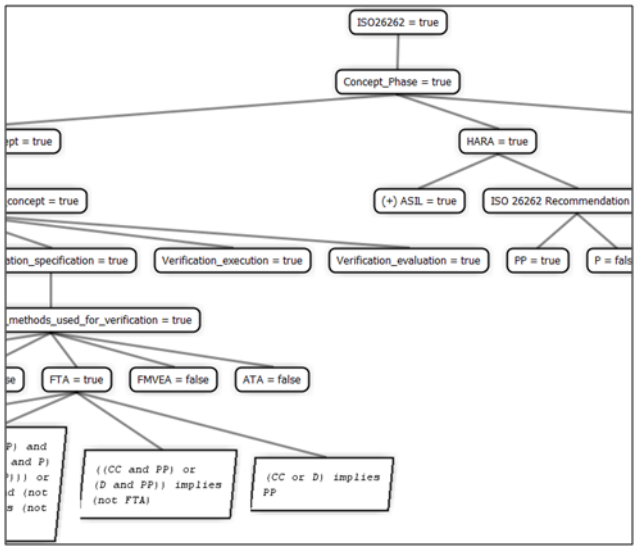

Figure 3 BVR Resolution diagram

BVR tool. The procedure how to build a process model and how BVR works in detail is described in the case study and in chapter "Management of families/lines" in Deliverable 6.3 [2].

An important feature that allows compliance checking is the verification function of the BVR Tool. This function uses constraints to evaluate the process model to make sure that it is compliant with the underlying standards. If the constraints are defined (this work is done only once) BVR can verify the model and all its alternative variants to identify modelling mistakes, which prevent a model from being standard compliant.

4.5 Definition of joint process

Process designers can use the parametrized model as starting point to integrate company and project specific requirements to get the joint process that implies demanded quality aspects and provides the wanted level of safety and security.

5 Case study

In this section, we report about the case study. More specifically, we illustrate the application of our approach, focusing on the concept phase, to a collaborative security and safety-critical system.

5.1 System and scenario description

The case study uses a fleet of autonomous (model) cars that communicate at runtime via car2car communication to form a platoon (the interested reader may refer to the AMASS Deliverable 1.6 [2] for further details). The fleet constitutes a safety- and security-critical system of systems. The focus is on safety and security aspects of the radio connection, which is enabled by the car2car communication management unit. Precisely, the scenario in focus is as follows: an attacker threatens the fleet's integrity by adding unauthorised code to the communication manager unit. The execution of this code increases the CPU load to a forbidden level. As result the communication breaks down and the platooning function is not any longer available (hazardous event). In a real life scenario this hazardous event may cause harm to people.

Thus, in our scenario, the communication management unit loses its functionality (safety issue) triggered by a cybersecurity attack. Once the cybersecurity issue is identified, the software must be updated and also the hazard analysis needs a reverification to guarantee that it is still valid. Engineers have to check that there are no unwanted side-effects of the security update on any safety aspects.

5.2 Objectives

The objective of the case study is the definition and evaluation of a joint process for co-engineering safety and cybersecurity.

5.3 Application of process flow

For the cybersecurity- and safety-critical system under consideration, ISO 26262 and SAE J3061 are identified as relevant. Then, the process modelling in EPF-C begins. The

obtained process model is a direct representation of the underlying standards and contains all addressed activities. It is called base model and is used to perform process tailoring, where unwanted activities are removed and new project specific ones are added.

According to the concept phase of ISO 26262, item definition and HARA needs to be performed. SAE J3061 demands a feature definition and a security analysis. In the system under consideration a combined hazard and threat analysis was performed with the tool ANSYS medini analyze [1]. The analysis was done using EVITA, which is one of the supported methods. The outcome was ASIL=B related to functional safety and SecRL=4 related to cybersecurity. ASIL B demands a minimal set of activities to achieve compliance with ISO 26262 and leads to safety measures to undercut the allowed failure rate.

Currently, standards do not define strict process requirements for (cyber)security, but it is demanded to have a defined process and a consistent line of arguments, when the product is brought to the market. Based on the standard compliant minimal set and the project specific requirements, the process variability management via the BVR Tool is started. As recalled in the background section, this requires the creation of three models: VSpec, Resolution, and Realization.

Our created VSpec model focuses on activities that vary in relation to ASIL and SecRL. Alternatives (XOR relation) and optionality (0/1) are also specified in the VSpec model. Once the VSpec is created, we can generate the Resolution model, as shown in Figure 3. Having set ASIL=B and SecRL=4, we are able to resolve the variability within our resolution model by choosing the appropriate features, where the variability parameters ASIL and SecRL decide whether process activities and specific methods have to be executed or not. More precisely, we assign "true" xor "false" to each activity of the model to define the process model, which will only include the features with true-value assignment and constraints satisfaction. Constraints make sure that all necessary activities are part of the model. They use logic operations to link elements. In the example shown in Figure 3 "(CC or D) implies PP" means that if ASIL C or D is selected also PP (++) has to be selected to receive a valid validation result. Once constraints are defined, they can be evaluated as often as needed if the BVR-function "Validate" is selected. If the validation is "true" the created model complies with the defined constraints and the requirements of the underlying standard.

Finally, we are ready to create the realization model, where the binding between the abstract representation of the desired/re-configured/resolved process, representing the joint process, and the concrete representation, expected to be rendered by EPF-C, is specified via a set of substitution rules.

In the realization model, partly shown in Figure 4, we specify that FTA, shown in Figure 3, shall be removed because it is a deductive analysis method, which is not highly recommended for ASIL B. Specifically, placement (FTA) and replacement (null) are specified.

Since FMEA is required for all ASILs, no substitution is included. Similar considerations are valid for FMVEA.

When the process and possible substitutions are executed/realized, the final process model is exported to an EPF-C processable XMI format. The tailored and standard compliant model is now available in EPF-C. BVR Tool supports variability management and makes sure that all relevant activities and methods are part of the final model.

In our elaborated joint process model, FMEA is used in combination with FMVEA to perform a joint safety and security analysis to specify requirements for functional safety and cybersecurity.

5.4 Discussion and lessons learned

Besides safety issues the joint process has to cover security aspects as well. The behaviour of security issues is different to safety. From the safety point of view, it is sufficient to analyse the item and implement measures which make sure that the intended ASIL will be achieved. If no safety issue has been missed, developers can assume that implemented ASIL is valid without a time limit. From the security point of view the situation is different because malicious attacks have to be considered. In the example above, the communication management unit has lost its functionality (safety issue) triggered by a cybersecurity attack. It is also important to investigate the possibility that safety issues enable attackers to find new attack paths. SAHARA is a methodology which is able to support an analysis towards this direction.

The SecRL covers only the risk but it is not an adequate metric for process rigour and the related engineering effort. **Process rigour** might be a key indicator used as argument that during the engineering phase a sufficient combination of activities has been taken into account.

Metrics are not covered in normative parts of the security standards. Therefore, developers have to define some kind of process specific process rigour. Discussions to develop a

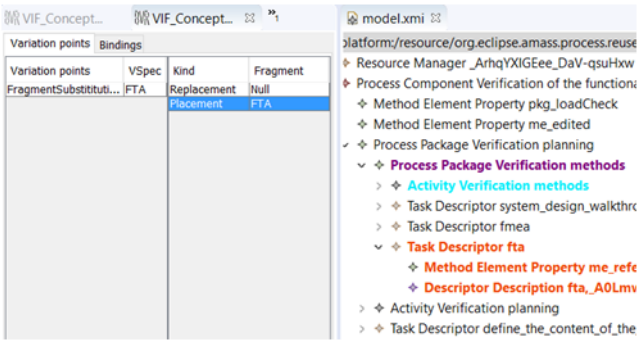

Figure 4 BVR Realization diagram

framework concerning a cybersecurity assurance level is ongoing in standardisation working groups.

ISO 26262 demands processes to maintain functional safety during operation. Related to this requirement, a **field monitoring and update procedure** has to be available in the development process to ensure functional safety **until decommissioning**. Safety and especially security monitoring increase the effort, but it is absolutely essential with regard to automated driving functions.

Determination of risk in the early phases (e.g. TARA in concept phase) is based on parameters which **will change during the development** phase because "public" tools and methodologies to perform attacks will be improved and can influence parameters in the analysis. **Regular updates of the threat analysis** have to be planned (e.g. once a year) to check the validity of assumptions. A sole threat assessment before SOP is not sufficient in all cases because new attack paths may be developed.

In particular, the phase starting with post-production until decommissioning is very important for the security engineering life cycle. Cybersecurity monitoring during the use of items in the field will bring up information about new threats and vulnerabilities which are basis for a response plan.

6 Conclusion and future work

In this paper, we proposed a process for co-engineering safety and security by the explicit systematization and management of commonalities and variabilities, implicitly stated in the requirements of the different standards. Our process treats the safety and security life cycles as members of a security-informed safety-oriented process line. It forces safety and security engineers to come together and brainstorm on what might be considered a commonality and what might be considered a variability. We illustrated the usage of our process by systematizing commonalities and variabilities at risk analysis phase in the context of ISO 26262 for functional safety and SAE J3061 for cybersecurity. We obtained a SiSoPL from which we derived the intended process for co-engineering and our lesson learned. However, cybersecurity is an ongoing development towards ISO/SAE 21434, which will extend the process with new activities and steps. While functional safety is more stable there are also developments to extend the consideration from functional safety towards Safety Of The Intended Functionality (SOTIF) [7]. SOTIF describes a situation where hazards can be caused by insufficient performance or insufficient knowledge about the later environment. In a similar direction UL4600 [10] goes towards a guidance document regarding the evaluation of automated driving. A focus is here also on the reduction of unknowns, e.g. a process to generate understanding about the later environment and potential scenarios for complex systems. Recent examples and the new research on adversarial images [25] show that security is also an important consideration for safety of the intended functionality. For such systems, a life cycle targeting the co-engineering of safety and cybersecurity needs to consider the potential adversarial impact on the environment of an automated system. Thus, the extension of our SiSoPL, considering the interplay of the different and relevant standards and guidance within the automotive domain, constitutes part of our future work. We also aim at quantitatively evaluating the tailoring enabled by our SiSoPL, as done in [17], within the space domain.

Acknowledgment. This work is supported by the EU projects AMASS and Secredas [9]. Research leading to these results has received funding from the EU ECSEL Joint Undertaking under grant agreement n° 692474 (project AMASS) and n°783119 (project Secredas) and from Sweden's Vinnova, and Sweden's Knowledge Foundation via the SACSys (Safe and Secure Adaptive Collaborative Systems) project, the COMET K2 - Competence Centres for Excellent Technologies Programme of the Austrian Federal Ministry for Transport, Innovation and Technology (bmvit), the Austrian Federal Ministry of Science, Research and Economy (bmwfw), the Austrian Research Promotion Agency (FFG), the Province of Styria, and the Styrian Business Promotion Agency (SFG). ANSYS supported the AMASS project with academic licences for their commercial tool medini analyze.

References

[1] AMASS (Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems) Project (online), https://www.amass-ecsel.eu.

[2] AMASS Project: Deliverables (online) https://www.amass-ecsel.eu/content/deliverables.

[3] ENISA (European Network and Information Security Agency), *ENISA good practices for security of Smart Cars*, https://www.enisa.europa.eu/publications/enisa-good-practices-for-security-of-smart-cars.

[4] EVITA project, https://www.evita-project.org.

[5] ISO 26262 (2018), *Road vehicles – Functional safety*, International Standard.

[6] ISO/SAE 21434, *Road vehicles – Cybersecurity Engineering - General Overview.* https://www.iso.org/standard/70918.html

[7] ISO/PAS 21448 (2019), *Road vehicles - Safety of the intended functionality*.

[8] SAE - Society of Automotive Engineers, *SAE J3061 - Cybersecurity Guidebook for Cyber-Physical Automotive Systems*.

[9] SECREDAS (Product Security for Cross Domain Reliable Dependable Automated Systems), http://secredas.eu/

[10] Underwriters Laboratories Inc. (UL), *UL 4600 - Standard for Safety for the Evaluation of Autonomous Products*.

[11] I. Ayala, B. Gallina (2016), *Towards Tool-based Security-informed Safety Oriented Process Line Engineering*, 1st ACM International workshop on Interplay of Security,

Safety and System/Software Architecture (ISSA), Copenhagen, Denmark.

[12] J. Castellanos Ardila, B. Gallina (2017), *Towards Efficiently Checking Compliance Against Automotive Security and Safety Standards*, 7th IEEE International Workshop on Software Certification., Toulouse, France.

[13] B. Gallina, L. Fabre (2015), *Benefits of security-informed safety-oriented process line engineering*, Digital Avionics Systems Conference (DASC), IEEE/AIAA 34th (pp. 8C1-1), IEEE.

[14] B. Gallina, I. Sljivo, O. Jaradat (2012), *Towards a Safety-oriented Process Line for Enabling Reuse in Safety Critical Systems Development and Certification*, Post-proceedings of the 35th IEEE Software Engineering Workshop (SEW-35).

[15] B. Gallina, S. Kashiyarandi, H. Martin, R. Bramberger (2014), *Modeling a safety-and automotive-oriented process line to enable reuse and flexible process derivation*, Computer Software and Applications Conference Workshops (COMPSACW), IEEE 38th International, pp. 504-509.

[16] B. Gallina, M. A. Javed, H. Martin, R. Bramberger (2019), *Co-engineering of security and safety life-cycles for engineering security-informed safety-critical automotive systems in compliance with SAE J3061 and ISO 26262*, 24th International Conference on Reliable Software Technologies-Industrial Presentation Track (Ada-Europe), Warsaw, Poland, June 11-14.

[17] B. Gallina (2019), *Quantitative Evaluation of Tailoring within SPICE-compliant Security-informed Safety-oriented Process Lines*, Journal of Software: Evolution and Process, EuroSPI Special Issue, DOI:10.1002/smr.2212.

[18] Ø. Haugen, O. Øgård (2014), *BVR–better variability results*, International Conference on System Analysis and Modeling (pp. 1-15). Springer, Cham.

[19] Ø. Haugen (2012), *Common Variability Language (CVL)*, Object Management Group, Tech. Rep. ad/2012-08-05 [Online]. Available: http://www.omgwiki.org/variability/doku.php

[20] M. A. Javed, B. Gallina (2018), *Get EPF Composer back to the future: A trip from Galileo to Photon after 11 years*, EclipseCon, Toulouse, France.

[21] M. A. Javed, B. Gallina (2018), *Safety-oriented process line engineering via seamless integration between EPF composer and BVR tool*, Proceedings of the 22nd International Conference on Systems and Software Product Line-Volume 2 (pp. 23-28), ACM.

[22] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, A. S. Peterson (1990*), Feature-oriented domain analysis (FODA) feasibility study (No. CMU/SEI-90-TR-21),* Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.

[23] H. Martin, R. Bramberger, C. Schmittner, Z. Ma, T. Gruber, A. Ruiz, G. Macher (2017), *Safety and security co-engineering and argumentation framework*, International Conference on Computer Safety, Reliability, and Security (pp. 286-297). Springer, Cham.

[24] G. Macher, E. Armengaud, C. Kreiner, E. Brenner, C. Schmittner, Z. Ma, M. Krammer (2018), *Integration of security in the development lifecycle of dependable automotive CPS*, Solutions for Cyber-Physical Systems Ubiquity (pp. 383-423), IGI Global.

[25] N. Morgulis, A. Kreines, S. Mendelowitz, Y. Weisglass (2019), *Fooling a Real Car with Adversarial Traffic Signs*, arXiv preprint arXiv:1907.00374.

[26] A. Ruiz, B. Gallina, J. L. de la Vara, S. Mazzini, H. Espinoza (2016), *Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems*, 5th International Workshop on Next Generation of System Assurance Approaches for Safety-Critical Systems (SASSUR), Trondheim.

[27] C. Schmittner, T. Gruber, P. Puschner, E. Schoitsch (2014), *Security application of failure mode and effect analysis (FMEA)*, International Conference on Computer Safety, Reliability, and Security (pp. 310-325), Springer, Cham.

[28] J. L. de la Vara, E. Parra Corredor, A. Ruiz Lopez, B. Gallina (2019), *AMASS: A Large-Scale European Project to Improve the Assurance and Certification of Cyber-Physical Systems*, Proceedings of the 20th International Conference on Product-Focused Software Process Improvement (PROFES), Barcelona, Spain.

[29] J. L. de la Vara, A. Ruiz, B. Gallina, G. Blondelle, E. Alaña, H. Herrero, F. Warg, M. Skoglund, R. Bramberger (2019), *The AMASS Approach for Assurance and Certification of Critical Systems*, embedded world Conference (ewC), Nuremberg, Germany.

[30] OMG (2008), *Software & systems Process Engineering Meta-model (SPEM), v 2.0*, Full Specification formal/08-04-01.

[31] BVR Tool. https://github.com/SINTEF-9012/bvr

[32] OpenCert - hosting the AMASS platform. https://www.polarsys.org/opencert/about/

[33] Eclipse Process Framework, Eclipse Foundation, Inc., Canada, http://www.eclipse.org/epf/.

[34] ANSYS medini analyse, ANSYS Inc., USA, https://www.ansys.com/products/systems/ansys-medini-analyze