Multi-concern Dependability-centered Assurance via Qualitative and Quantitative Co-analysis

Barbara Gallina, Leonardo Montecchi, André Luiz de Oliveira, and Lucas Bressan

***Abstract*—In various safety-critical domains, multi-concern claims, regarding multiple dependability attributes e.g., safety, security and their interplay, have become common and need to be justified based on evidence. At system design-level, evidence may include a combination of mono-concern as well as multi-concern analysis results. In this paper, to contribute to multi-concern assurance, we focus on system design, and we first present a high-level process that builds on top of the synergy between qualitative and quantitative dependability analysis techniques, which have been used for mono as well as multi-concern analysis. Then, we explain how to instantiate it within the automotive domain. Finally, perspectives for future work are sketched.**

***Index Terms*—Multi-concern assurance, Functional safety, Cybersecurity, ISO 26262, ISO 21434, ISO/IEC/IEEE 42010, Multi-concern qualitative and quantitative co-analysis.**

I. Introduction

In various safety-critical domains, such as automotive, rail, and avionics, multi-concern claims regarding e.g., safety, security and their interplay, have become common and need to be justified based on evidence. Process as well as product-related types of evidence play an important role as a foundation for explaining the validity of such claims. At system design-level, evidence may consist of a combination of mono-concern as well as multi-concern analysis results, in addition to the justification that the chosen analysis techniques are in compliance with standards applicable within the domain of interest.

In the automotive domain, for instance, with respect to functional safety, both deductive/top-down and inductive/bottom-up techniques, are recommended (ISO 26262 [1], Part 4, clause 6). Safety analysis is expected to be performed at the appropriate level of abstraction during the concept (i.e., conducted on the item definition) and product development phases (conducted on the different levels of abstractions related to the technical concept). Quantitative analysis methods, such as quantitative Fault Tree Analysis (FTA) and Markov models, predict the frequency of failures (where a failure is defined as the termination of an intended behaviour of an element or an item due to a fault, i.e., abnormal condition, manifestation), while qualitative analysis methods, such as qualitative FTA and HAZOP (HAZard and OPerability) analysis, identify failures but do not predict their frequency. Quantitative analyses are typically used to address random hardware failures; while qualitative analyses are used to address systematic failures. It follows that quantitative safety analyses complement qualitative safety analyses.

For what concerns cybersecurity, threat scenarios (i.e., potential causes of compromise of cybersecurity properties of one or more assets) are expected to be identified (ISO 21434 [2], 15.4). However, no normative requirement is present to indicate a specific technique to be used. The method for threat scenario identification, however, can use group discussion and/or systematic approaches (e.g., STRIDE [3], which stands for Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege). Regarding analysis of the threat scenarios aimed at identifying attack paths, top-down (e.g., attack trees) and bottom-up analysis can be used. It is important to highlight that, while estimating the exposure of an hazardous event within the ISO 26262 scope is doable, it is not doable to estimate the exposure of an attack. Within the scope of ISO 21434 [2], the attack feasibility is rated instead and it is rated qualitatively.

From the perspective of safety and cybersecurity demonstration, both standards require the provision of an argument. Specifically, ISO 26262 requires the creation of a safety case and ISO 21434 requires the creation of a cybersecurity case. From a process and methodological perspectives, several synergies and potential for reuse have been identified in the literature. For this reason, methods, which are typically used in functional safety for safety analysis as well as safety demonstration, have been extended to tackle also the cybersecurity concern as well as other concerns towards a multi-concern perspective. Specifically, model-based methods for conducting mono-concern/multi-concern analysis and argumentation have been proposed and integrated within development processes and environments.

In this paper, to contribute to multi-concern assurance at system design, we present a process that builds on top of the synergy between qualitative and quantitative dependability analysis techniques, which have been used for mono as well as multi-concern analysis. Our process has the potential to be executed on the AMASS platform, the first defacto certification platform [4], tangible result of the AMASS (Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems) project. Finally, perspectives for future work are also sketched. The rest of the paper is organized as follows. In Section II, we recall background information. In Section III, we introduce our multi-concern dependability-centered process, which relies on the cross-fertilization of specific qualitative and quantitative

B. Gallina was with the School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden, e-mail: barbara.gallina@mdu.se.

L. Montecchi was with the Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway. e-mail: leonardo.montecchi@ntnu.no

A. L. de Oliveira was with the Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora, MG, Brazil. e-mail: andre.oliveira@ice.ufjf.br

L. P. Bressan was with the Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora, MG, Brazil, and Arm Ltd., Cambridge, United Kingdom. e-mail: lucas.bressan@arm.com

analysis techniques. In Section IV, we illustrate our proposed process on an automotive example. In Section V, we discuss the related work. Finally, in Section VI, we conclude the paper and present future research directions.

II. Background

In this section, we present the background consisting of a general overview on 1) multi-concern assurance at design time; 2) qualitative and quantitative multi-concern assurance within the AMASS platform; and 3) Highly Automated Driving Vehicle, the motivating example used to illustrate our process.

A. Multi-concern assurance: focus at design time

In this subsection, first, we recall fundamental definitions regarding concern, assurance, and multi-concern assurance. Then, we recall how assurance is typically expected to be done at design time. **Concern** [5] is defined as "interest in a system relevant to one or more of its stakeholders". **Assurance** [6] is defined as "grounds for justified confidence that a claim has been or will be achieved". Quoting John Rushby [7], "Safety certification assures society at large that deployment of a given system does not pose an unacceptable risk of harm. There are several ways of organizing and conducting certification, but all are conceptually based on scrutiny of an argument that certain claims about safety are justified by evidence about the system." If we consider a multi-concern perspective, multi-concern assurance and certification would mean providing grounds for justified confidence that multi-concern claims have been or will be achieved, as well as arguments that those claims about multi-concerns are justified by the evidence about the system.

Within the automotive domain, multi-concern assurance and certification may mean providing grounds for justified confidence that compliance with applicable standards, including ISO 26262 for functional safety, ISO 21434 for cybersecurity, and their interplay, is guaranteed whenever required based on the item characteristics. This in turn means that recommended analysis methods are used at the expected abstraction level, in order to develop appropriate evidence regarding absence of unreasonable risk as well as sufficient protection of assets against threat scenarios. ISO 26262, specifically, prescribes safety life-cycles for: 1) top-down development of an item, which comprises item definition, hazards analysis and risk assessment (HARA), safety goals and functional safety concept, and technical safety concept; and 2) top-down-and-bottom-up development of an item via integration of safety elements developed out of context. ISO 21434 prescribes a cybersecurity life-cycle expected to be aligned with the safety life cycle. Once the item definition is available, methods to determine the extent to which a road user can be impacted by a threat scenario need to be applied. As stated in ISO 21434, these methods and their work products are collectively known as a threat analysis and risk assessment (TARA) and are performed from the viewpoint of affected road users. TARA includes the asset identification, threat scenario identification, impact rating; attack path analysis, attack feasibility rating, risk value determination, and risk treatment decision.

B. Multi-concern Assurance within the AMASS Platform

The AMASS [4] platform integrates a set of commercial and open-source tools to address multi-concern assurance (i.e., functional safety, cybersecurity) and certification across different safety-critical domains. As reviewed in [8], different qualitative as well as quantitative analysis techniques can be used to develop grounds of justified confidence regarding absence of unreasonable risk. In this paper, we focus on two specific analysis techniques: 1) ConcertoFLA [9], for conducting qualitative failure logic analysis (FLA) based on the Failure Propagation and Transformation Calculus (FPTC) formalism, and 2) State-Based Analysis (SBA) [10], for quantitative analysis based on Stochastic Petri Nets. Both techniques are implemented within the CHESS[1] framework, which in turn has been integrated within the AMASS platform.

Both analyses allow engineers to: 1) decorate component-based system specifications (i.e., functional models given in CHESS Modelling Language, compatible with the SysML[2] standard) with dependability-related behavioural specifications; 2) execute the analysis; and 3) back-propagate the analysis results into the original system specification. In FLA, the qualitative dependability-related behavioural specification is given as a set of FPTC rules, representing local component failure propagation paths. FLA calculates the qualitative failure behaviour at system level based on the failure behaviour at component level, and based on an assumed behaviour to be injected. FLA combines and automates traditional safety analysis techniques (FTA and FMEA). In SBA, the quantitative dependability-related behavioural specification consists of probabilistic information, including failure and repair distribution of components, propagation delays and probabilities, fault tolerance and maintainability concepts.

To decorate the system specifications, appropriate stereotypes can be attached to the components. These stereotypes implements the conceptual metamodels for safety [11] and security [12]. UML State Machine diagrams can also be attached to components to specify more detailed failure behaviour of a component relating external faults (input failures), internal faults, and their effects (output failures). Cybersecurity threat events (external faults/attacks), for instance, which exploit system vulnerabilities (internal faults), as well as their consequences (output failures in the item), can be specified into CHESS 'ErrorModel'-stereotyped state machine diagrams attached to components. Pre-defined enumerations of cybersecurity threats are STRIDE-compliant: *unauthorized access of service*, *unauthorized modification of service*, and *unauthorized denial of service*, and so are vulnerabilities: *missing data integrity scheme*, *inadequate encryption strength*, and *resource allocation without limits*. Security threats stated as external faults in state machine diagrams are linked to component internal failures (vulnerabilities), leading to the occurrence of component safety-related failure modes. The consequences of a security threat correspond to safety-related component failure modes, leading to *loss of confidentiality*, *integrity*, and/or *availability*, i.e., CIA properties.

[1] http://www.polarsys.org/projects/polarsys.chess

[2] https://www.omg.org/spec/SysML/1.4/PDF

For instance, intentional attacks on the target of evaluation (TOE, i.e., an asset, for example a system model or item) can be modeled. Once the threats and attack paths associated with assets and CIA properties have been identified, risk assessment is performed to classify and prioritize the threats to be treated with the introduction of security controls. During risk assessment, the risk posed by each attack path (threat scenario) is then manually calculated based on its feasibility and impact, and a risk matrix. The attack feasibility rating can be defined based on *qualitative* attack potential, attack-vector, Common Vulnerability Scoring System (CVSS)[3], or OWASP[4] risk models. In OWASP, the risk of a threat is determined by likelihood * impact (severity) levels. The higher is the likelihood and impact, higher is the risk posed by a threat.

C. Highly Automated Driving Vehicle

In this paper, we customize a pre-existing example [13] concerning a Highly Automated Driving (HAD) Vehicle. In our customization, the HAD vehicle comprises four cameras, automated powertrain, and steering actuators connected to the vehicle computer. The Vehicle Computer calculates longitudinal and lateral movements from the data provided by cameras, and it sends commands to control powertrain and steering actuators. The vehicle contains an on-board tester component that requires a flash USB stick to update the vehicle computer firmware and/or operating system. Our customization aims at including current recommendations regarding risks associated with sensor vulnerabilities and potential sensor signal manipulation efforts such as camera blinding. For further details, the interested reader may refer to the Cybersecurity Best Practices for the Safety of Modern Vehicles drafted by National Highway Traffic Safety Administration.

III. Cross-Fertilization of Analysis Techniques: the Multi-Concern Process

As mentioned in the introduction, qualitative and quantitative techniques can complement each other to increase assurance. In this section, we explain how the qualitative FLA and the quantitative SBA techniques can benefit from each other. Their cross-fertilization, which was briefly mentioned in [11], can occur in two directions, which basically reflect the top-down and bottom-up system design processes.

A. Top-down cross-fertilization

A top-down system design process essentially consists in a progressive decomposition: an overview of the system is first formulated, by specifying, but not detailing, its subsystems. Further detail is added at each iteration, until the entire specification is completed. The ISO 26262-prescribed life-cycle for the item development is a top-down system design process.

In the context of a top-down process, as soon as an overview of the system is available, FLA can help assessing whether the current architecture exhibits unreasonable risk and/or is sufficiently protected against threat scenarios (mono and or multi-concern analysis). At this stage, some assumptions are made on the failure behavior of components; at the same time constraints on what should be their failure behavior are devised as result of the analysis. The necessity of opting for mono or multi-concern analysis depends on the system in focus. In both types of analysis, the results may indicate that, in order to avoid a certain risk, a key component X must avoid failing with a specific failure mode f_Y. For example, if mono-concern and e.g., safety-focused analysis is executed, analysis results may indicate that, in order to avoid catastrophic and harmful consequences, a key component X must avoid failing with a specific failure mode f_Y. When dealing with multiple concern simultaneously, in the context of FLA, the failure modes are further specialized with concern-specific failure/vulnerability information and their propagation and impact at system level is traced, allowing for the identification of the interplay of multi-concern failures modes. For instance, an attack to a component X (incoming external fault from the perspective of X and outgoing failure mode from the perspective of the environment) can exploit an internal vulnerability and lead to a safety-related failure mode f_Y.

In the subsequent stages, details on the concrete architecture and its components are introduced. At this point, SBA can be applied in order to: i) verify that the constraints mandated by the previous analysis are satisfied (e.g., the occurrence of f_Y is indeed a very rare event); or ii) assess other quantitative metrics on the system, like availability or performance.

B. Bottom-up cross-fertilization

In a bottom-up design process, the individual base elements of the system are first specified in detail, and then connected together to form larger subsystems. The ISO 26262-prescribed life-cycle for the safety-element-out-of-context development is a bottom-up design process.

In bottom-up design, combining different analyses can also help to cope with the system's complexity. Performing any analysis on a full specification of the system can be very difficult; for example, the well-known state-space explosion problem is a challenge for applying SBA on large models. On the other hand, performing more lightweight analyses like FLA requires a good understanding of the failure behavior of components, which is typically difficult to obtain without first analyzing the internal behavior of components with other techniques. In this perspective, the application of SBA on selected parts of the system (i.e., subsystems) can help to obtain aggregated information on their failure behavior, to be used as input for subsequent higher-level analyses. For example, such information can then be used to justify the assumptions that are needed for performing FLA at a higher abstraction level.

In general, during the design and development of modern systems, designers typically adopt an iterative and incremental but hybrid process, resulting from a mix of the top-down-and-bottom-up processes. In the automotive domain, for instance, the item-centered top-down process is merged with the element-centered bottom-up process.

[3] http://www.first.org/cvss/cvss-guide.pdf

[4] https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

IV. The Multi-Concern Assurance Process in Action

In this section, we exemplify our multi-concern dependability-centered assurance process, which relies on the cross-fertilization of the qualitative (FLA) and quantitative (SBA) analyses. The exemplification is done within the automotive domain, considering a top-down process for the item development related to a hypothetical (but realistic) Highly Automated Driving (HAD) vehicle. The focus is on functional safety and cybersecurity.

Our process, illustrated in Fig. 1, encompasses the following iterative phases.

1) Item Definition: functions are allocated to systems, subsystems, and components, and their dependencies and interactions with the environment are specified in a component-based model. At system level, designers specify the components, their ports, and relationships between components (Fig. 1a shows the definition of the HAD vehicle).

2) Qualitative HARA and TARA: at system level, via FLA, engineers assess the risk. Here we consider a multi-concern risk resulting from hazardous events (i.e., hazard + operational situation), attack paths (threat scenarios) and their interplay. **Concerning functional safety**, in Fig. 1b, we only illustrate the FPTC rule that captures the input-output behaviour of Vehicle Computer, where the occurrence of incorrect values of *cameraFrontIn* and *cameraRear* input ports lead to *too high torque* in the Vehicle Computer output port. We only focus on the propagation leading to the *loss of vehicle control* hazard, with too high torque, too high longitudinal movement, too low longitudinal movement, too high lateral movement, and too low lateral movement. **Concerning cybersecurity**, we consider a scenario, where incorrect values of *cameraLeftIn* and *cameraRightIn* inputs (Fig. 1b) are due to a *data spoofing attack*, and not actually originated from *left/right* Camera components (Fig. 1a). The specification of the cybersecurity scenario is modeled with a state machine diagram for Camera component (Fig. 1c). This model shows that a *Data Spoofing* attack along with the *Missing Data Integrity Schemes* vulnerability are the causes of an *Unauthorized Modification of Service* threat. Once the failure behavior is modeled, the input ports of *leftCamera* and *rightCamera* (Fig. 1a) instances of *Camera* component are injected with *valueCoarse* faults and failure logic analysis is executed. The injected *valueCoarse* failure modes into *left* and *right* Camera's *lensSensor* inputs are propagated throughout their *video* output ports (Fig. 1a).

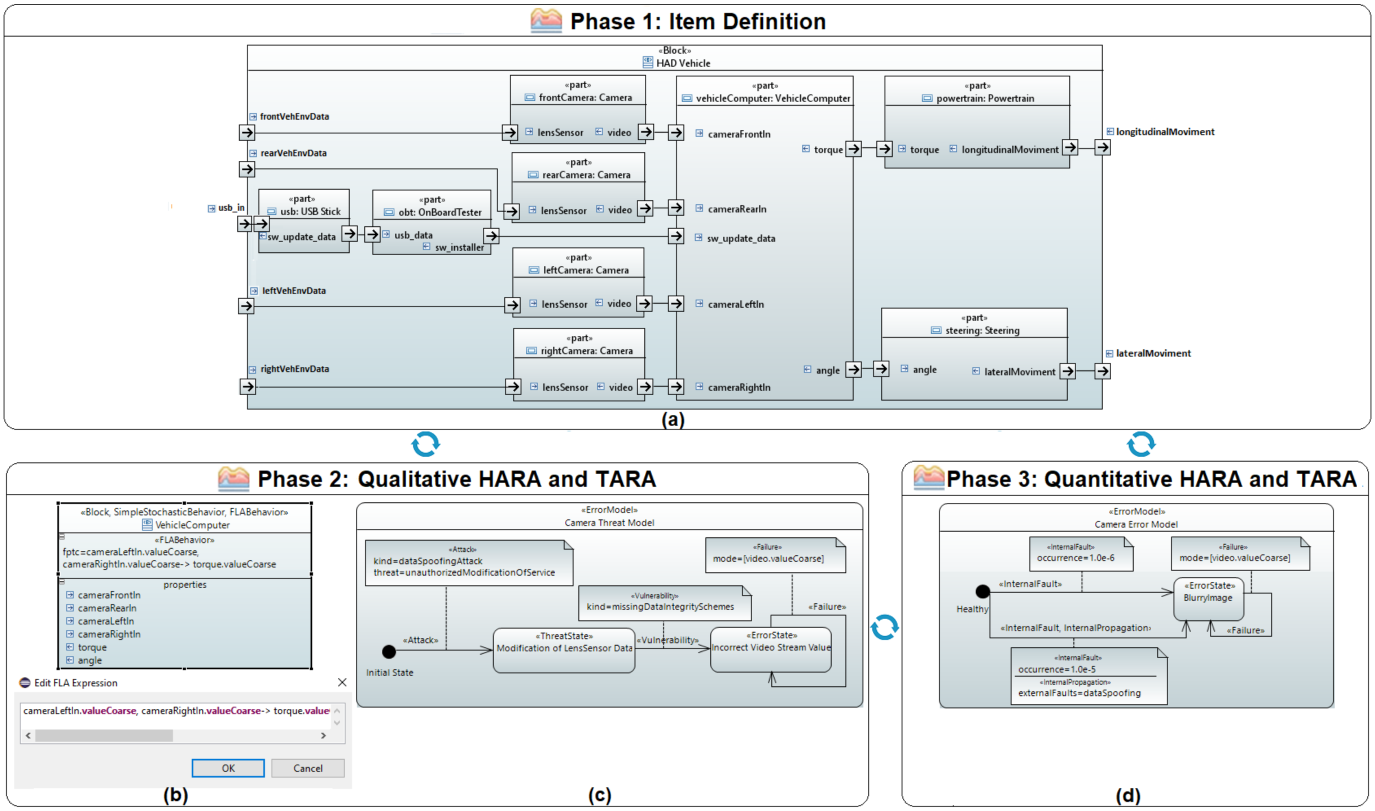
Based on the functional safety and cybersecurity specification, we perform the safety and cybersecurity co-analysis and reveal the failure propagation path, which includes the interplay of attacks and hazardous events: the failure modes into *Camera's lensSensor* inputs as *data spoofing attacks* exploit *missing data integrity scheme* vulnerabilities, leading to *video* output failure modes. *Camera.video* output failure modes are then propagated throughout *VehicleComputer*, *Powertrain*, and *Steering* components input and output ports (e.g., *torque* and *angle*), leading to the occurrence of *loss of vehicle control* hazardous event due to *too high/low Powertrain.longitudinalMovement* or *too high/low Steering.lateralMovement*.

The update of *Vehicle Computer* firmware via *USB stick* is another vulnerability that could be exploited by a tampering attack. This would lead to the corruption of video data (i.e., incorrect values on the *cameraFrontIn* and *cameraRearIn* ports, see Fig. 1c), thus potentially leading to *too high torque*, and ultimately to the occurrence of a hazardous event. After threat identification, we classified their risks according to likelihood and impact (severity) factors. Finally, we derived security goals related to the violation of asset's CIA properties as consequences of attacks, e.g., ensuring the availability of video streaming service related to an *Unauthorized Modification of Service* threat, and ensuring the absence of loss of vehicle control, in the event of *data spoofing* attacks. We derived in total two security goals related to the *confidentiality* and *integrity* of *Camera's video streaming*, and four security goals related to *Powertrain* and *Steering* components. A security-informed fault tree for the *loss of vehicle control* hazard can be synthesized, by executing xSAP[5], from component HAD FLA annotations and state-machine diagrams.

3) Quantitative HARA and TARA: at a refined level of abstraction, engineers via SBA assess the reliability, availability, maintainability, with respect to random failures of physical and mechanical hardware components that contribute to the top-event (hazard), via calculation of quantitative metrics, e.g., Probabilistic Metric for Hardware Failure (PMHF), failure and repair rates. Specifically, for physical and hardware elements such as *Cameras*, *Powertrain*, and *Steering* actuators, we specified non-deterministic internal faults and assigned values to probabilistic properties (e.g., *failure occurrence*) of these faults in CHESS error state machine diagrams. Fig. 1d illustrates the 'ErrorModel' state machine diagram of the *Camera* component. *Internal faults* (with their activation rates and probabilities) may lead to the *Blurry Image* error state and then finally to the failure of the component. We also specified state machine diagrams for the *Powertrain* and *Steering* components, and used the CHESS 'SimpleStochatsticBehavior' annotations to assign *failure occurrences* to *Vehicle Computer*, *Powertrain*, *Steering*, and *Camera* (Fig. 1d) component instances. These probabilistic properties are inputs to determine the probability of the loss of vehicle control hazard via execution of SBA. *Occurrence rates* can also be assigned to external *attacks*, which can be included into a CHESS-SBA model using the *InternalFault* and *InternalPropagation* annotation (Fig. 1d).

The allocation of safety/security requirements to mitigate the effects of hazards and security threats may modify the item, implying in revaluation of risks. Hazard/Threat analysis and risk assessment should be performed until all hazard/threat related the risks have been reduced to acceptable levels. The introduction of a security control to mitigate a threat may raise different hazards and safety requirements (integrity levels), modifying hardware reliability, availability, and/or performance requirements. Our approach supports iterative and incremental co-analysis of the interplay between safety, reliability, and security properties of CPS, and the production of

[5] https://es-static.fbk.eu/tools/xsap/



cameraLeftIn.valueCoarse, cameraRightIn.valueCoarse-> torque.value

Fig. 1: Phases, models, and tools in the multi-concern assurance process, applied to the HAD vehicle system.

multi-concern certification evidence from a single component-based model enriched with dependability information.

V. Related Work

There exist several frameworks that support model-based analysis of different concerns, e.g., see [14], [15], [16], [17]. Recently, a growing interest in extending safety techniques to integrate security aspects can be observed. Hierarchically Performed Hazard Origin and Propagation Studies (HiP-HOPS) [14] has been extended for security [18]. The HiP-HOPS security extension is built upon the TARA metamodel from Open Dependability Exchange (ODE) metamodel[6], and focuses on the allocation and decomposition of security requirements and controls, and it uses attack trees to model security violations and their propagation. The Architectural Analysis and Design Language (AADL) has since long time supported qualitative and probabilistic safety analysis, through its Error Model Annex [15]; and an AADL *Security* Annex (AADL-SA) has also been proposed, to support the specification of CIA properties of component ports. In [19], authors propose an extension of the Yakindu Security Analyst tool that combines Component Fault Trees (CFT) and attack trees. SPTA-SafeSec [20] proposes an extension of the System Theoretic Process Analysis (STPA) method, to also address security.

More in general, the need for an integrated support for multi-concern analysis has been observed; the report in the AMASS deliverable [8] presents an overview of methods addressing multi-concern assurance. Besides the AMASS project and platform [4], other projects and corresponding frameworks have contributed to the integration of multiple analysis techniques covering different concerns. Notably, COMPASS [16] and AMADEOS [17] are representative frameworks that follow such direction. In this paper, we provided a methodological solution on how such kind of integrated platform can be employed to comply with safety and security standards, with a focus in the automotive domain.

VI. Conclusion and Future Work

The assurance of modern systems requires addressing multiple, possibly conflicting, concerns, and their interplay. This became particularly evident in the automotive domain with the introduction of ISO 26262 and ISO 21434 standards. In this paper we discussed the need for multi-concern co-analysis and assurance, and described a possible process for its application in the automotive domain, focusing on safety and security analyses supported by the AMASS platform.

For a practical multi-concern assurance process, however, further challenges need to be solved. In particular, how to generate evidence spanning multiple properties, integrating results from multiple analyses, is still a challenge that needs to be faced. Furthermore, reusing pieces of evidence will be of increasing importance, when complex interactions among concerns are considered.

[6] https://github.com/DEIS-Project-EU/DDI-Scripting-Tools/tree/master/ODE_Metamodel