# A Model-Based Safe-by-Design Approach with IP Reuse for Automotive Applications

Morayo Adedjouma
Université Paris-Saclay, CEA, LIST
F-91120, Palaiseau, France
Email: morayo.adedjouma@cea.fr

Nataliya Yakymets
Université Paris-Saclay, CEA, LIST
F-91120, Palaiseau, France
Email: nataliya.yakymets@cea.fr

***Abstract*—The paper presents an approach for design/safety co-engineering and reuse of safety IP Cores. The approach is based on a compositional development process coupling system development and safety processes from a formalization of activities proposed in ISO26262. The co-engineering approach integrates reuse of safety and design artifacts to reduce development efforts. We illustrate the approach on the example of an Adaptive Cruise Control System within a tool called Sophia. We discuss the advantages and limitations that the approach brings for the development of safety critical systems.**

***Keywords*–automotive system; model-based; functional safety; IP Core reuse.**

## I. Introduction

With the growing complexity of systems in the automotive domain, vehicles become more and more safety-critical as failures or hazardous decisions about the environment may lead to accidents that cause human lives. Due to the safety-critical nature of such systems, system and safety engineers are prone to follow safety standards (e.g., ISO26262 [1]), best system development pratices and associated tools. In this context, Model-Based System Engineering (MBSE) and IP Core reuse are promising approaches. MBSE helps to integrate various methods and tools for safety analysis into the common system modeling environment, to customize this environment to the automotive domain and to provide extensive traceability links across the safety analysis process [2]. The IP Core reuse approach allows reducing the system development efforts by using libraries of pre-existing design artifacts [3]. We also integrate the reuse of safety artifacts like libraries of failures modes, hazards, etc. to reduce the safety activities effort [4]. In practice, however, the tool support of the aforementioned approaches is not well integrated.

In this paper, we present an approach based on MBSE, Model-Based Safety Assessment (MBSA) and reuse concepts. Our approach extends such approaches as [5][6] to the automotive domain based on ISO26262 standard, in particular, the recommendations of Part 4 about product development at the system level. Although ISO26262 provides generic recommendations on which safety related workproducts should be issued, it does not specify the particular processes on how to get those workproducts. There may exist dependencies between workproducts recommended by ISO26262, which can slow down the system development. Therefore, an efficient way to implement the standard recommendations is to turn to system and safety co-engineering and parallelize steps of both processes when possible. The advantage of such an approach is that the safety activities do not block the system development activities, and vice-versa.

In addition, we apply well-trusted design principles by exploiting the ability to reuse pre-modeled ( and already pre-analyzed) IP Cores, as well as libraries of safety artifacts. As defined in Part 4 of ISO26262, by IP Cores we consider well-trusted designs for elements (including hardware and software components), as well as well-trusted or standardized interfaces. By safety artifacts we understand well-trusted technical safety concepts and mechanisms for the detection and the control of failures [1]. Although, pre-analyzed design elements and safety analysis results are context-dependent and cannot be reused as-is according to ISO26262, certain artifacts (e.g., definition of generic failure modes, risks and their causes) can be detached from the context and reused in the form of libraries.

We illustrate the approach on an Adaptive Cruise Control (ACC) system [7]. The ACC is an example of safety-critical system that requires engineers to adopt a safety standard. We conduct our analysis of the ACC system using Sophia [6], a modeling tool which offers a graphical development environment for system design and safety analysis. Sophia semi-automates the proposed methodology and improves the traceability of system and safety artifacts at the early phases of the system development lifecycle. Using this case study, we show how to obtain a safe-by-design automotive system and reduce the design and analysis efforts due to the co-engineering and libraries reuse approach.

The rest of the paper is organized as follows. Section I motivates our work. Section II presents the current practices and weaknesses for safe development of automotive systems. We present our approach in Section III and its tool support in Section IV. We outline the approach application on the ACC system in Section V and discuss our concluding remarks in Section VI.

## II. State of the art and practice

The integration of any classical safety analysis method into an MBSE environment requires three main steps [5]: (1) system model creation, (2) safety annotation and modeling, (3) safety analysis and generation of results. Several initiatives, approaches and tools have evolved over time in the field of MBSA. In these approaches, the system model can be created using languages, such as UML (Unified Modeling Language) [8], SysML (System Modeling Language) [9], or domain specific languages like EAST-ADL [10]. Then the system model is extended with the safety concepts and relations either by using safety profiles like [6][11]-[12] or by translating the system model into formal or safety languages for further analysis [13][14]. Some of these approaches come with a methodology to comply with standards, e.g., ISO26262 [10][12]. Once the model has been annotated with safety data, it can be analyzed using MBSA tools that offer one or a few methods for safety analysis. The latter case needs additional efforts to study the semantics of the languages and to implement the bridges between tools. Examples of analytical and simulation tools are xSAP [14] and AltaRica toolset [13]. Despite profound analysis provided by those tools, many of them require professional knowledge of modeling methods (e.g., Markov chains or Petri nets) and formal languages (e.g., AltaRica, SMV, etc.), which is a barrier for widespread utilization. Concerning reuse of pre-existing artifacts, these tools require reverse engineering to build system models. RiskWatch [15] or Pilar [16] are examples of tools implementing risk management methodologies. Those tools are exclusively qualitative, and based on various tabular structures filled by informal description methods. The running system is never explicitly modeled, hence there is no reuse capabilities of the IP Core provided. Some others research and model-based tools like Hip-Hops [17], Visual Figaro [18], CAFTA, Isograph Reliability Workbench [19], ConcertoFLA [20] and Medini Analyze [21] implement features to store and reuse some safety artifacts. Most of them provide libraries or databases of failure modes, their causes and effects, component failure patterns, etc. However, the lack of interoperability between the tools and/or closed data formats make it difficult to reuse and/or export the safety models, libraries and results.

There were also some initiatives and projects working on safety certification platforms, e.g., the European projects OPENCOSS [22] and AMASS [23]). These projects and their associated tools aim at safety certification according to different standards (including ISO26262) based on MBSA. As part of the AMASS platform, our work helps follow ISO26262 recommendations for the system development through a MBSE and MBSA approach in an unified environment. The work relies on open data and open languages (UML/SysML) and provides reusability features of IP Core and safety artifacts libraries.

## III. Approach

Figure 1 presents the co-engineering methodology at a glance. The methodology shows how to conduct safety assessment based on the reference phase model for the development of a safety-related item at system level (Part 4) described in ISO26262. It allows conducting system development in parallel with safety assessment with respect to ISO26262 requirements. By parallelizing both concerns into a co-engineering process, the system development steps are not blocked by the safety steps. The reuse of pre-analyzed IP Cores and safety artifacts reduces development and analysis efforts.

The inputs for the proposed methodology are: 1) system description including requirements, functional and system architecture; and 2) safety analysis results from the FMEA (Failure Mode and Effects Analysis) [24], the FTA (Fault Tree Analysis) [25] and the HARA (Hazard Analysis and Risk Assessment), obtained at the prior phases of system development lifecycle, such as concept definition. To harmonize the methodology with ISO26262 reference model, we prefix its main steps with the appropriate clauses of the standard. The methodology includes the following steps:

- (4.5) Initiation of product development at the system level. We determine and plan the functional safety activities to perform during the development of Automotive System (AS).
- (4.6) Specification of the technical safety requirements synchronized with system requirements. The system requirements are specified and analyzed by the system engineer and safety expert to derive the technical safety requirements. At this stage, one can reuse existing safety mechanisms from the IP Core library for the requirements specification, e.g., fault detection measure, self-monitoring concept, warning concept.
- (4.7) System design synchronized with system safety analyses. The technical safety requirements as result of previous step help define the system design. If the library contains prior pre-modeled and analyzed components included in the system under analysis, they could be reused (with regard to appropriate impact analysis related to the new system context). Safety analyses are conducted on the defined architectural design to avoid systematic failures of the system. System safety analyses may include the HARA in the case of usage of reusable elements and if new hazards are introduced, the FMEA and the qualitative FTA. During safety assessment, various safety artifacts (hazards, failure modes, causes, effects, control mechanisms, etc.) could be reused from and added to the safety artifacts libraries. The newly analyzed elements are also stored into the libraries for their reuse during further iterations or for future usage in other projects.

## IV. Tool support

The co-engineering methodology given in Section III is semi-automated in Sophia tool, a MBSE/MBSA framework [26]. Sophia includes a set of DSLs (Domain Specific Language) dedicated to several aspects of safety assessment methodology. Each DSL is a UML profile having its equivalent viewpoint [27]. As shown in Figure 1, each viewpoint could be used during one or several steps of the methodology. The Safety Requirement DSL describes a taxonomy and properties of safety requirements compliant with ISO26262 (Part 8). The Process Management DSL defines the evolution of system architecture through its lifecycle by introducing such concepts like system, function, hardware, software along with corresponding allocation relationships. The HARA, FMEA, FTA and property verification DSL describe the safety concepts related to these methods recommended in ISO26262.

Figure 2 shows an overview of Sophia architecture. The tool is implemented as Eclipse plugins on top of Papyrus tool, a customizable environment supporting modelling of systems using standardized languages (e.g., UML, SysML). The framework is modular so that dedicated analysis modules can be used either independently or conjointly in a given user-defined process. Sophia provides a fluent and integrated flow supported by ISO26262 Process package that interacts with the safety analyses. These packages 1) refer to the Safety and Reliability package providing generic definitions to dependability concepts; 2) provide a mechanism to save and reuse safety artifacts obtained during safety analyses in the form of annotated elements; 3) provide functionality to save and reuse pre-analyzed safe IP Cores in form of SysML models or SysML Blocks annotated with safety concepts. Sophia also provides bridges to other external tools for further safety analyses like NuSMV [28], AltaRica, FIDES [29]and

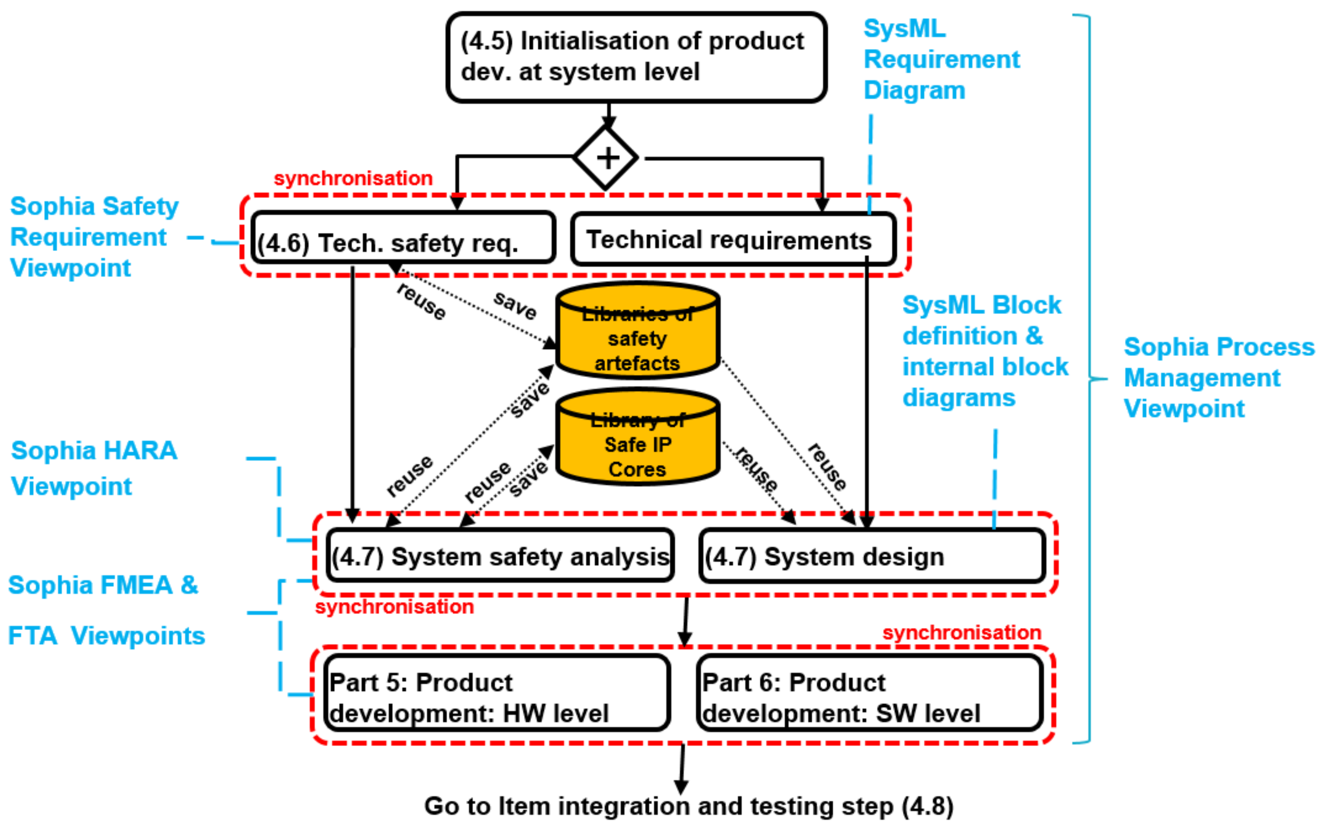

(4.5) Initialisation of product dev. at system level
SysML Requirement Diagram
synchronisation
Sophia Safety Requirement Viewpoint
(4.6) Tech. safety req.
Technical requirements
reuse
save
Libraries of safety artefacts
SysML Block definition & internal block diagrams
Sophia Process Management Viewpoint
save
Library of Safe IP Cores
Sophia HARA Viewpoint
reuse
reuse
save
reuse
reuse
(4.7) System safety analysis
(4.7) System design
Sophia FMEA & FTA Viewpoints
synchronisation
synchronisation
Part 5: Product development: HW level
Part 6: Product development: SW level
Go to Item integration and testing step (4.8)

Figure 1. System/safety co-engineering methodology and safety viewpoints mapping

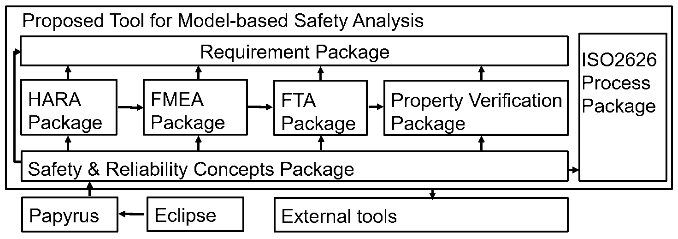

Proposed Tool for Model-based Safety Analysis
Requirement Package
HARA Package
FMEA Package
FTA Package
Property Verification Package
ISO2626 Process Package
Safety & Reliability Concepts Package
Papyrus
Eclipse
External tools

Figure 2. Sophia tool architecture to support ISO26262 recommendations

XFTA [13]. The analyses yield results that are propagated back through the design and display in the models using dedicated profiles, editors, tables, and Papyrus customization toolsets.

## V. Case Study

We apply the proposed methodology (Figure 1) to design a safety-critical Adaptive Cruise Control (ACC) System. The ACC is a well-known automotive system that allows a vehicle's cruise control to adapt the vehicle's speed to the traffic environment. The ACC uses a radar attached to the front of the vehicle to detect whether preceding vehicles are moving in the path of the host car with the ACC. If there is no preceding vehicle, the ACC maintains the driver selected speed. When a preceding vehicle shows up, the system may automatically apply braking, control throttle or shift gear to adapt the vehicle speed and maintain the selected clearance without driver intervention.

### A. *Initialization of Product Development at the System Level*

As we consider the reuse of various elements from IP Core and safety artifact libraries, we perform an impact analysis to assess the effects of the reused artifacts in our context and to determine the applicable safety activities that we will need to conduct for the ACC system development.

### B. *Specification of the Technical Safety Requirements*

We capture and model the system requirements of the ACC system. Hereafter, we consider the requirement REQ_ACC_03 given in Figure 3. This requirement is satisfied by the component ACC module (Figure 4), and its refinement in several sub-requirements REQ_ACC_03a, REQ_ACC_03b, REQ_ACC_03c, are satisfied by the system function Increment speed, Decrement speed and Shift gear, respectively. These requirements are enriched with safety requirements (prefixed by Safety_REQ_ACC) specifying the safety measures/mechanisms identified as we performed the safety analyses of the system (see Section V-D). We store the technical safety requirements into the appropriate library for later usage.

### C. *System Design*

Figure 4 shows the top level design of the ACC system with the interconnecting interfaces between its components. The core part of the ACC system is the ACC module. It processes data information from the Radar.

The ACC module sends a signal to Brake Control in case of braking. The Engine Control and Electronic Throttle Control control the vehicle speed by increasing or decreasing

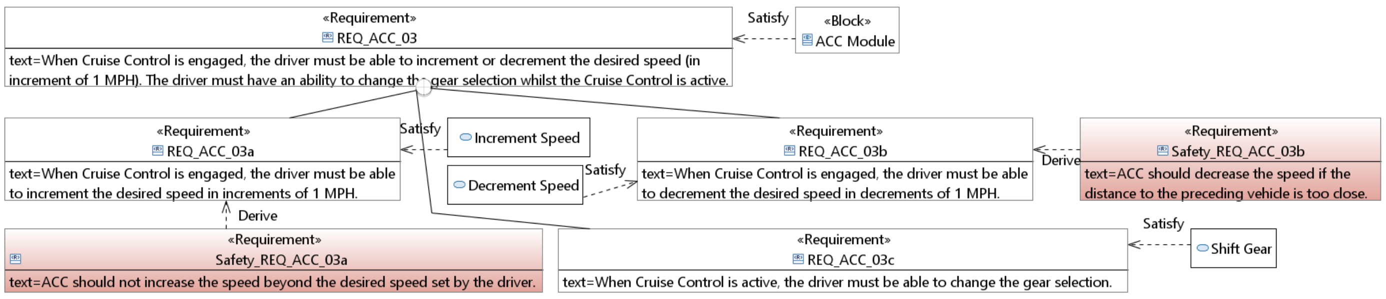

«Requirement» REQ_ACC_03
text=When Cruise Control is engaged, the driver must be able to increment or decrement the desired speed (in increment of 1 MPH). The driver must have an ability to change the gear selection whilst the Cruise Control is active.
Satisfy
«Block» ACC Module
«Requirement» REQ_ACC_03a
text=When Cruise Control is engaged, the driver must be able to increment the desired speed in increments of 1 MPH.
Satisfy
Increment Speed
Decrement Speed
Satisfy
«Requirement» REQ_ACC_03b
text=When Cruise Control is engaged, the driver must be able to decrement the desired speed in decrements of 1 MPH.
Derive
«Requirement» Safety_REQ_ACC_03b
text=ACC should decrease the speed if the distance to the preceding vehicle is too close.
Derive
«Requirement» Safety_REQ_ACC_03a
text=ACC should not increase the speed beyond the desired speed set by the driver.
«Requirement» REQ_ACC_03c
text=When Cruise Control is active, the driver must be able to change the gear selection.
Satisfy
Shift Gear

Figure 3. Excerpt of functional (shown in white) and safety (shown in red) requirements of the ACC system.

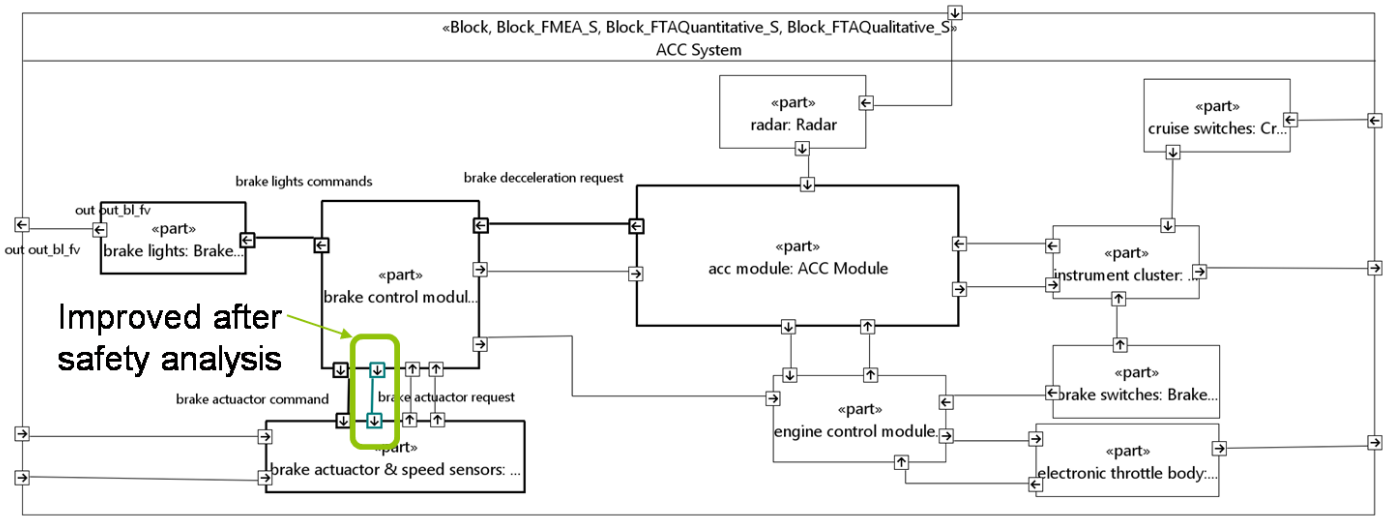

«Block, Block_FMEA_S, Block_FTAQuantitative_S, Block_FTAQualitative_S»
ACC System
«part» radar: Radar
«part» cruise switches: Cr...
brake lights commands
brake decceleration request
out put_bl_fv
out out_bl_fv
«part» brake lights: Brake...
«part» brake control modul...
«part» acc module: ACC Module
«part» instrument cluster: ...
Improved after safety analysis
brake actuactor command
brake actuactor request
«part» brake switches: Brake...
«part» engine control module...
«part» brake actuator & speed sensors: ...
«part» electronic throttle body:...

Figure 4. Top level architecture of the ACC system shown in Internal Block Diagram. The ACC architecture was modified (shown in green) to satisfy safety requirements

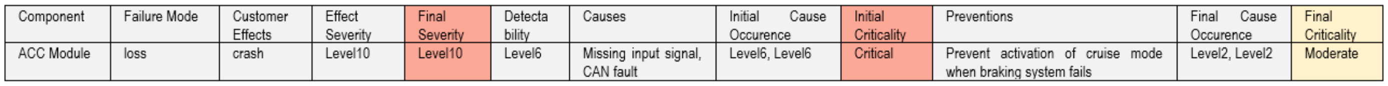
the throttle injection. The Cruise Switches component allows the driver to command the ACC functionalities and to set the selected speed and clearance. The Instrument Cluster is a panel in front of the driver that processes the Cruise Switches and sends them to the ACC and Engine Control modules. The Instrument Cluster also displays information regarding the ACC system state. The Brake Switches can deactivate the Cruise Control operation. The Brake Lights component allows illumination of the stop lamps during automatic braking from the ACC module request. The Brake Actuators & Speed Sensors component includes the sensors and devices, such as the brake pedal, the accelerator pedal, etc. The communication bus and the Controller Area Network (CAN) transmit all the signals between the components. We link the technical safety requirements to appropriate system components. The requirement REQ_ACC_03 given in Figure 3 is satisfied by the component ACC module (Figure 4). The sub-requirements REQ_ACC_03a, REQ_ACC_03b, REQ_ACC_03c, are satisfied by the system functions Increment speed, Decrement speed and Shift gear, respectively. Those system functions are realized by components that we reuse from the IP Core libraries, namely the Brake Lights, Brake Switches and Cruise Switches.

## *D. System Safety Analysis*

Hereby, we focus on performing the HARA and FMEA analyses with Sophia tool [26] to illustrate the proposed concept of IP Cores and safety artifacts reuse.

**HARA.** We perform the HARA on the system design to determine new hazards and effects that may arise as we reuse some libraries elements in a new context. The HARA is based on the usage scenarios and the main functionalities of the ACC system. It takes into account requirement and architecture models defined in the Safety Requirements and Process Management viewpoints. Some HARA artifacts are defined in a specific library as model elements reusable from one viewpoint to another, e.g., the set of operating conditions are derived from the vehicle states and the malfunctions are specified for all functions that satisfied the functional requirements of the system. The hazards, nature of injuries are also coming from a predefined list corresponding to the injury category described in the ISO26262 standard. During the analysis, we reused the HARA results for the Brake Lights, Brake Switches and Cruise Switches that improved analysis time.

Hereafter, we analyze the following operational situation: the ACC system being active when the vehicle is driving on highway at medium speed, following a preceding vehicle. We



| Component | Failure Mode | Customer Effects | Effect Severity | Final Severity | Detecta bility | Causes | Initial Cause Occurence | Initial Criticality | Preventions | Final Cause Occurence | Final Criticality |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ACC Module | loss | crash | Level10 | Level10 | Level6 | Missing input signal, CAN fault | Level6, Level6 | Critical | Prevent activation of cruise mode when braking system fails | Level2, Level2 | Moderate |

Figure 5. Excerpt of the FMEA results for the ACC system. Columns 5, 9 and 12 are calculated automatically according rules given ISO26262

analyze the ACC function Increment speed (Figure 4) used to maintain the desired distance with the preceding vehicle. Some malfunctions associated with this function are the ACC system increases vehicle speed when it is too close to preceding vehicle and the ACC system increases vehicle speed beyond desired speed set by driver. The generic hazard Unintended acceleration is associated with these malfunctions. The resulting hazardous event, as defined by ISO26262, is a combination of the hazard and the operational situation, i.e., the ACC module requests an unintended acceleration when preceding vehicles are too close. In our example, the hazardous event is evaluated at the Automotive Safety Integrity Level (ASIL) C, with Exposure=E4, Controllability=C2, Severity=S3. Finally, we determine the safety goals for the hazardous events to prevent an unacceptable risk level from those events or reduce their impact. The safety goals refine/extend the ACC requirements defined in the Safety Requirement Engineering viewpoint (Figure 3). For our example of hazardous event, we define two safety goals, Safety_REQ_ACC_03a: ACC should not increase the speed beyond the desired speed set by the driver, and Safety_REQ_ACC_03b: ACC should decrease the speed if the distance to the preceding vehicle is too close. The newly analyzed ACC components and safety artifacts are stored into the IP Core and safety artifacts libraries (as shown in Figure 1).

**FMEA.** The FMEA complements the HARA by determining the corrective actions to be implemented to meet previously defined safety objectives. This analysis uses as inputs the usage scenarios, the ACC system architecture, as well as the results of the HARA model elements (libraries of accidents, malfunctions, hazardous events, accidents, etc.) and their properties (severity, ASIL, etc.). The analysis helps determine the effects and the criticality of single basic causes of failure modes at the component level until the system level.

With the help of the safe IP Core libraries, the tool traces the FMEA artifacts to the hazardous events and accidents previously identified in the HARA. Figure 5 shows the FMEA table generated for the ACC module component. The malfunctions found during the HARA are stored into the safety artifacts library and then reused for failure mode identification. As an example, we specify the failure mode Loss of the ACC module, its causes (missing input signal, CAN fault) and effects (loss). This failure mode can lead to different effects until the crash of the vehicle at customer level referring to the accident identified during the HARA. We found that the ACC module changes its criticality level from critical to moderate after application of recommended and well-trusted preventive actions (already existing in the libraries). The list of safety requirements is derived from the specified preventive actions: for our example, it is prevent activation of cruise module when braking system fails. These new safety requirements are traced by the tool to the safety goals elicited during the HARA. As during the HARA, the analyzed components with the FMEA results are stored to safe IP Core library (as shown in Figure 1). We reused existing results of the analysis for the Brake Lights, Brake Switches and Cruise Switches that reduce efforts to perform the FMEA.

Figure 4 shows how the ACC architecture was modified to satisfy the safety requirements derived during the ACC development process. The improved scenario is that the ACC module should send a brake actuator request to a new added Actuator Controller in order to duplicate the brake actuator command from both the Brake Control and the Actuator Controller.

## VI. Conclusion and Discussion

The ever growing complexity of modern automotive systems presents certain challenges in meeting time to market constraints. To address this issue, we suggest a methodology that helps in formalizing, synchronizing and semi-automating the system development and the safety analysis activities recommended in ISO26262. The methodology includes the ability to reuse libraries of already pre-modeled and pre-analyzed IP Cores as basic elements for building more complex automotive systems. In addition, we can reuse safety artifacts (e.g., hazards, failure modes, etc.) for analysis of other systems. It makes the design process more flexible and reduces the design time.

We implement the proposed methodology in Sophia tool. Sophia is a modeling tool offering a graphical development environment for system design and safety analyses. It relies on SysML and UML languages, so that the models and libraries can be imported and reused in different modeling environments. Beside, as both the development and safety activities are conducted in the same environment, we avoid interoperability and traceability issues that undermine the reuse capabilities of certain tools. Although we focus on the system development process to demonstrate the methodology, the proposed methodology is applicable to later development phases, in particular, to software and hardware development and testing activities. For example, we may refer to FIDES to refill our IP Core libraries, as it is a well-known and standardized database for reliability prediction of hardware components.

We apply the model-based safety analysis and IP Core reuse approach to an ACC system. During the case study, we model the ACC system architecture and conduct safety analyses according to the proposed methodology. As a result, we identify critical components of the ACC system and propose architectural changes to reduce the system overall criticality level. In the case study, we reuse pre-analyzed IP Cores (in particular, Brake Lights, Brake Switches and Cruise Switches), as well as various HARA and FMEA safety artifacts (e.g., hazards, malfunctions, failure modes, safety mechanisms, etc.) across the development process. This possibility helps us reduce the design and analysis effort.

The case study shows the important efforts for the deployment of the methodology the first time: it takes time to model the system and to fill in the libraries. However, this effort may be rapidly amortized during next iterations or in future projects by saving time and cost on the analyses thanks to the reusability inherent to the model-based and IP Core reuse paradigms.

From our case study, we also notice the lack of certain domain specific expertise about which safety artifacts can be reused in specific context. Another difficulty comes from maintenance of the libraries while IP Cores or safety artifacts must be accompanied with justifications about their application context. As future work, we might consider exploration of solutions based on Safety Element out of Context (SEooC) concept from ISO26262. We would also develop dedicated libraries per domain to be able to address other standards (e.g., aerospace, robotic, and medical).

## REFERENCES

[1] ISO 26262: Road Vehicles : Functional Safety. International Organization for Standardization, 2018.

[2] D. Brugali, “Model-driven software engineering in robotics: Models are designed to use the relevant things, thereby reducing the complexity and cost in the field of robotics,” Robotics & Automation Magazine, IEEE, vol. 22, no. 3, 09 2015, pp. 155–166.

[3] D. D. Gajski et al., “Essential issues for ip reuse,” in Proceedings 2000. Design Automation Conference. (IEEE Cat. No.00CH37106), 02 2000, pp. 37 – 42.

[4] I. Sljivo, B. Gallina, J. Carlson, H. Hansson, and S. Puri, Tool-Supported Safety-Relevant Component Reuse: From Specification to Argumentation. Cham: Springer International Publishing, 01 2018, pp. 19–33.

[5] N. Yakymets, S. Dhouib, H. Jaber, and A. Lanusse, “Model-driven safety assessment of robotic systems,” in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 1137–1142.

[6] N. Yakymets, M. Perin, and A. Lanusse, “Model-driven multi-level safety analysis of critical systems,” in 2015 Annual IEEE Systems Conference (SysCon) Proceedings, 2015, pp. 570–577.

[7] W. Pananurak, S. Thanok, and M. Parnichkun, “Adaptive cruise control for an intelligent vehicle,” in 2008 IEEE International Conference on Robotics and Biomimetics, 2009, pp. 1794–1799.

[8] The Unified Modeling Language Specification Version 2.5, 2015. Object Management Group, retrieved: September, 2020. [Online]. Available: https://www.omg.org/spec/UML/2.5/

[9] System Modeling Language Specification Version 1.5, 2017. Object Management Group, retrieved: September, 2020. [Online]. Available: https://www.omg.org/spec/SysML/

[10] P. Cuenot, C. Ainhauser, N. Adler, S. Otten, and F. Meurville, “Applying model based techniques for early safety evaluation of an automotive architecture in compliance with the ISO 26262 standard,” in Embedded Real Time Software and Systems (ERTS2014), Toulouse, France, 02 2014.

[11] G. Biggs, T. Juknevicius, A. Armonas, and K. Post, “Integrating safety and reliability analysis into mbse: overview of the new proposed OMG standard,” INCOSE International Symposium, vol. 28, no. 1, 07 2018, pp. 1322–1336.

[12] P. Feth et al., “Multi-aspect safety engineering for highly automated driving,” in Computer Safety, Reliability, and Security, B. Gallina, A. Skavhaug, and F. Bitsch, Eds. Cham: Springer International Publishing, 2018, pp. 59–72.

[13] Altarica. Alatarica Association, retrieved: September, 2020. [Online]. Available: https://altarica.labri.fr/

[14] G. Biggs, T. Juknevicius, A. Armonas, and K. Post, “The xsap safety analysis platform,” in Tools and Algorithms for the Construction and Analysis of Systems, M. Chechik and J.-F. Raskin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 04 2015.

[15] RiskWatch. Risk Watch International, retrieved: September, 2020. [Online]. Available: http://www.riskwatch.com/

[16] Pilar. EAR, retrieved: September, 2020. [Online]. Available: www.pilar-tools.com/en/tools/pilar/

[17] Hip-Hops. Hull University, retrieved: September, 2020. [Online]. Available: http://hip-hops.eu/

[18] Visual Figaro. Electricite De France, retrieved: September, 2020. [Online]. Available: https://sourceforge.net/projects/visualfigaro/

[19] Isograph Reliability Workbench, retrieved: September, 2020. [Online]. Available: https://www.isograph.com/software/reliability-workbench/

[20] B. Gallina, Z. Haider, and A. Carlsson, “Towards generating ECSS-compliant fault tree analysis results via ConcertoFLA,” IOP Conference Series: Materials Science and Engineering, vol. 351, 05 2018, p. 012001.

[21] Ansys, Medini Analyzer. Ansys, retrieved: September, 2020. [Online]. Available: https://www.ansys.com/products/systems/ansys-medini-analyze

[22] OPENOCSS project. The OPENCOSS Consortium, retrieved: September, 2020. [Online]. Available: http://www.opencossproject.eu

[23] AMASS project. The AMASS Consortium, retrieved: September, 2020. [Online]. Available: https://www.amassecsel.eu

[24] IEC 60812: Analysis techniques for system reliability - Procedures for FMEA. International Electrotechnical Commission, 1985.

[25] NASA, “Fault tree handbook with aerospace applications,” 2002.

[26] M. Adedjouma and N. Yakymets, “A framework for model-based dependability analysis of cyber-physical systems,” in 2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE), 2019, pp. 82–89.

[27] M. Mori et al., “Systems-of-systems modeling using a comprehensive viewpoint-based SysML profile,” Journal of Software: Evolution and Process, vol. 30, no. 3, 2018, p. e1878, e1878 JSME-16-0093.R2.

[28] NuSMV. NuSMV Project, retrieved: September, 2020. [Online]. Available: http://nusmv.fbk.eu/

[29] FIDES. The Fides Consortium, retrieved: September, 2020. [Online]. Available: https://www.fides-reliability.org/