



UNIVERSITÀ
DEGLI STUDI
FIRENZE

FLORE

Repository istituzionale dell'Università degli Studi di Firenze

Engineering Functional Safety Requirements for Automotive Systems: A Cyber-Physical-Social Approach

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

Engineering Functional Safety Requirements for Automotive Systems: A Cyber-Physical-Social Approach / Mohamad Gharib, Paolo Lollini, Andrea Ceccarelli, Andrea Bondavalli. - ELETTRONICO. - (2019), pp. 74-81. (IEEE INTERNATIONAL SYMPOSIUM ON HIGH ASSURANCE SYSTEMS ENGINEERING Hangzhou, China Jan 3 - 5, 2019) [10.1109/HASE.2019.00021].

Availability:

The webpage <https://hdl.handle.net/2158/1138925> of the repository was last updated on 2021-03-03T12:13:14Z

Publisher:

IEEE Computer Society

Published version:

DOI: 10.1109/HASE.2019.00021

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

Publisher copyright claim:

La data sopra indicata si riferisce all'ultimo aggiornamento della scheda del Repository FloRe - The above-mentioned date refers to the last update of the record in the Institutional Repository FloRe

(Article begins on next page)

Engineering Functional Safety Requirements for Automotive Systems: A Cyber-Physical-Social Approach

Mohamad Gharib, Paolo Lollini, Andrea Ceccarelli, Andrea Bondavalli
University of Florence - DiMaI, Viale Morgagni 65, Florence, Italy
{mohamad.gharib,paolo.lollini,andrea.ceccarelli,andrea.bondavalli}@unifi.it

Abstract—Several approaches have been developed to assist automotive system manufacturers in designing safer vehicles by complying with functional safety standards. However, most of these approaches either mainly focus on the technical aspects of automotive systems and ignore the social ones, or they are not equipped with an adequate automated support. To this end, we propose a model-based approach for modeling and analyzing the Functional Safety Requirements (FSR) for automotive systems, which is based on the ISO 26262 standard and considers both technical and social aspects of such systems. This approach proposes a UML profile for modeling the FSR starting from item definition until safety validation, and it proposes constraints expressed in OCL to be used for the verification of FSR models. We illustrate the utility of the approach using an example from the automotive domain.

Index Terms—Functional safety requirements, Automotive systems, ISO 26262, Cyber-Physical-Social systems, GORE

I. INTRODUCTION

The automotive industry is one of the largest industries in the world and it is responsible for producing millions of new vehicles every year to be used by humans on a daily basis. Therefore, ensuring the safety of these vehicles has always been a growing concern for their manufacturers. In particular, automotive systems are safety-critical systems that have to fulfill safety requirements in addition to their functional requirements [1], where safety requirements describe characteristics a system must have in order to be safe [1]. Moreover, the complexity of current automotive systems has increased significantly in terms of their implemented functionalities, which increase the complexity while dealing with their functional safety requirements.

To maintain acceptable levels of safety a functional safety standard named ISO 26262:2011 [2] has been developed. The ISO 26262 provides appropriate development processes, requirements and safety integrity levels specific for the automotive domain. However, this standard mainly covers Electrical and Electronic (E/E) systems of vehicles leaving the driver and its behavior outside the scope of the standard, i.e., it is assumed that drivers can perform the necessary actions to stay safe [3]. This is not always the case since drivers are a main reason for many accidents [4], i.e., vehicle safety is more than pure technical issue.

More specifically, an automotive system can be seen as a Cyber-Physical-Social System (CPSS), i.e., a combination of

cyber components (e.g., software, sensors), controlled components (e.g., vehicles, traffic lights) and social components (e.g., drivers). Therefore, the safety of such systems cannot be ensured without considering their three main components. In other words, ignoring the social components during the CPSS design leaves the system open to different kinds of vulnerabilities, since vulnerabilities of a CPSS are not only generated by technical (e.g., cyber and physical) issues, but they can be also generated due to social issues. In this context, a safe automotive system can be designed only if the driver behavior is also considered during the system design [4].

In [3], we have proposed an approach based on the ISO 26262 standard that considers the E/E systems along with driver's behavior, but it was not equipped with any kind of automated support. Thus, it is not possible to depend on it for manually dealing with a large number of hazards, Functional Safety Requirements (FSR), safety goals, etc. [5]. To tackle this problem, we propose a model-based approach that is based on [3] and extends it with the following components:

- An engineering methodology to assist designers while modeling and analyzing FSR for automotive systems.
- A UML profile for modeling FSR that adopts social and organizational concepts from Goal-Oriented Requirements Engineering (GORE) [6], [8] and integrates them with concepts adopted from the ISO 26262. This allows for capturing the cyber, physical and social aspects of automotive systems.
- An automated analysis support that allows for verifying the FSR models. More specifically, several properties of the design, represented as Object Constraint Language (OCL) [7], have been formulated to verify the correctness and consistency of FSR models.
- A tool¹ that allows the models of FSR to be generated and verified.

The rest of the paper is organized as follows: Section II presents the research baseline, and an illustrative example concerning a Maneuver Assistant System is described in Section III. In Section IV, we present and discuss our approach, and we discuss threats to its validity in Section V. Related work is presented in Section VI, and we conclude and discuss future work in Section VII.

¹The tool is available and downloadable at <https://goo.gl/g45S8t>

II. RESEARCH BASELINE

A. ISO 26262

The ISO 26262 [2] is a functional safety standard applicable to all road vehicles with a weight under 3500 kg, and it has been developed with a main objective to provide guidelines and best practices to increase the safety of vehicles. More specifically, the ISO 26262 focuses on hazards caused by malfunctions of E/E systems and their associated risks, where each associated risk is then assigned an Automotive Safety Integrity Level (ASIL). ASILs can be classified under Quality Management (QM), A, B, C, or D, where QM is assigned to hazards with a very low probability that might cause only slight injuries, and it does not require risk reduction efforts. ASIL A, B, C, or D require risk reduction efforts, where ASIL D requires the highest reduction efforts. Table I shows the main clauses of the ISO 26262 relevant to the different phases of product development.

B. Modeling Requirements via Goal Models

Goal-Oriented Requirements Engineering (GORE) has emerged as a main approach for Requirements Engineering (RE). In GORE, goal models can serve as abstract specifications of the system-to-be. Although several goal-based modeling languages have been introduced (e.g., KAOS [6]). Tropos [8] has been proven effective for modeling requirements in their social and organizational context. Tropos introduces primitives for modeling *actors* of the system (agentive entities), and *goals* that *actors* intend to achieve. A *task* represents an abstract way to do something, and its execution can be a means for satisfying a *goal*. When *goals/tasks* are at high abstraction levels, they can be refined through *and/or-decomposition* into finer sub-goals/sub-tasks. A *resource* represents a physical or an informational entity. Finally, a *dependency* allows *actors* to *depend* on one another for the fulfillment of *goals*, execution of *tasks*, and provision of *resources*.

III. ILLUSTRATIVE EXAMPLE: MANEUVER ASSISTANCE SYSTEM

Our example is a highly automated driving system, namely Maneuver Assistance System (MAS)², which is expected to increase the safety of the driver by detecting and preventing unintended tactical and operational maneuvers. Usually, a *tactical maneuver* is motivated by a recently modified desire of the driver (e.g., lane changes) and it is associated with a short-term timescale (tens of seconds), while an *operational maneuver* is generally a result of a driver's desire to remain safe (e.g., avoid collision) and it is associated with a very short timescale (tens of milliseconds). In particular, MAS collects information about the vehicle, vehicle surroundings, as well as driver behavior, and then analyzes such information to determine whether there is a need and/or desire for the maneuver. If there is a need and/or desire for such maneuver it is considered an *intended* one. Otherwise, it is considered as

an *unintended* one. Accordingly, MAS should allow *intended* maneuvers and prevents *unintended* ones.

IV. A MODEL-BASED APPROACH FOR ENGINEERING FUNCTIONAL SAFETY REQUIREMENTS

In this section, we present our approach. First, we introduce the methodological, followed by the UML profile that allows for modeling FSR, and then we discuss the automated reasoning support that can be used to verify FSR models. Finally, we describe our tool that allows FSR models to be generated and verified.

A. Methodology

The process underlying our methodology is shown in Figure 1, and it has been developed based on the approach proposed in [3] and extends it with the modeling and analysis activities. Note that P. and C. represent the Parts and Clauses of the ISO 26262 standard respectively that have been used as a basis for defining some activities of the methodology³. The process is composed of two main phases, namely modeling and analysis: (1) **Modeling phase** aims to model the functional safety concept of an automotive system starting from item definition and modeling until the definition of safety validation, and it is composed of eight activities:

- 1.1 **Item definition and modeling** is the first activity of the process, in which we define and model the item along with the main functional requirements it aims to achieve.
- 1.2 **Hazard Analysis and Risk Assessment (HARA) modeling**, identifies and models possible hazards that can endanger the achievement of each functional requirement, which has been identified in the previous activity. Then for each hazard a risk assessment is performed to assign it with an ASIL based on its *severity*, *exposure* and *controllability* levels. After that, each hazard that is associated with ASIL level as ASIL A, ASIL B, ASIL C or ASIL D should be addressed by at least one Safety Goal (SG).
- 1.3 **Functional Safety Requirements (FSR) modeling**, derives at least one FSR from each SG that have been identified in the previous activity.
- 1.4 **Technical Safety Requirements (TSR) modeling**, defines at least one TSR from each FSR that have been identified in the FSR modeling activity.
- 1.5 **Defining Specification of Hardware Safety Requirements (HWSRs)**, defines a specification that can be used for the operationalization of each identified TSR that can be allocated to hardware.
- 1.6 **Defining Specification of Software Safety Requirements (SWSRs)**, defines a specification that can be used for the operationalization of each identified TSR that can be allocated to software.
- 1.7 **Defining Specification of SoCial Safety Requirements (SCSRs)**, defines a specification that can be

²For more information about the example, please refer to [3]

³A short description about these clauses can be found in Table I

Table I
MAIN CLAUSES OF THE ISO 26262 FOR THE DIFFERENT PHASES OF PRODUCT DEVELOPMENT

Clause	Description
C. 3-5	Item definition develops a description of the item with regard to its functionality, interfaces, known hazards, etc.
C. 3-6	Hazard Analysis and Risk Assessment (HARA) estimates the probability of exposure, controllability and severity of hazardous events with regard to the item. Then the ASILs of the hazardous events are determined based on these parameters, and assigned to corresponding safety goals.
C. 3-7	Functional safety concept is developed by deriving functional safety requirements from safety goals.
C. 4-6	Technical safety concept defines the technical implementation of the functional safety requirements, and verifies that the technical safety requirements comply with the functional safety requirements.
C. 5-6	Specification of Hardware Safety Requirements (HWSRs) provides specifications on how to elicit and manage the HWSRs.
C. 6-6	Specification of Software Safety Requirements (SWSRs) provides specifications on how to elicit and manage the SWSRs.
C. 4-9	Safety validation provides evidence that the safety goals are adequate, can be achieved at the vehicle level, and the safety concepts are appropriate for the functional safety of the item.

used for the operationalization of each identified TSR that can be allocated to social behavior.

- 1.8 **Defining safety validation**, defines acceptance criteria for the validation and verification of the identified HWSRs, SWSRs and SCSRs. This can provide evidence that the safety goals can be achieved at the vehicle level, and the FSRs are appropriate for the functional safety of the item.

(2) **Analysis phase** aims to verify the correctness and consistency of the FSR model depending on a set of properties of the design that we have defined and formulated as OCL constraints.

B. Modeling Phase

In what follows, we present our UML profile (shown in Figure 2) for modeling the FSR for automotive systems, and then we describe how it can be used to model the FSR of the MAS system.

The item in our approach can be a social entity or it may interact with a social entity. Therefore, we adopted the <<AgentiveElement>> and <<Actor>> concepts from Tropos to propose two stereotypes with the same names to capture the social aspects of the item. The <<Actor>> stereotype has a property to identify requirements it aims for. For capturing intentional entities related to the item, we follow Tropos and propose the <<IntentionalElement>> stereotype, which has been adopted mainly to capture the strategic goals/requirements of the item in their social and organizational context. This allows for capturing both the technical and social aspects of such goals/requirements. The <<IntentionalElement>> stereotype is specialized into three stereotypes: <<Requirement>>, <<SafetyGoal>> and <<OperationalElement>>.

The <<Requirement>> stereotype is further specialized into three different stereotypes: 1- <<FunctionalRequirement>> captures the functionalities an item aims to achieve, 2- <<FunctionalSafetyRequirement>> captures

the safety functionalities of the item without specifying how such functionalities can be implemented, and 3- <<TechnicalSafetyRequirement>> captures detailed technical requirements that can be defined from FSR, which can be operationalized. The <<SafetyGoal>> stereotype has been adopted to be consistent with the terminology offered by the ISO 26262 standard, and it is used to define a safety objective to be used for addressing a <<Hazard>>.

OperationalElement (OE) stereotype has been developed based on the same idea of the task in Tropos, and it is further specialized into three stereotypes <<SHWSR>>, <<SSWSR>>, and <<SSCSR>> that define Specification of Hardware, Software and Social Safety Requirements respectively. Moreover, the OE stereotype has two properties. The first one identifies Verification and Validation (V&V) acceptance criteria that an OE should achieve to be considered satisfied. The second property identifies whether the OE has been satisfied. These two properties have been included to *define safety validation* for each OE, i.e., define acceptance criteria for the validation and verification of the identified HWSRs, SWSRs and SCSRs, and determine whether such criteria have been satisfied.

The <<Hazard>> stereotype has been developed based on the Hazard concept presented in the ISO 26262 standard, and it captures hazards that can endanger the achievement of a functional requirement. <<Hazard>> has several properties that can be used for the assessment of its related risk: 1- *SeverityLevel*, measures the potential harm of hazard, and it ranges from S0 to S3, where S0 means no injuries and S3 means life-threatening injuries; 2- *ExposureLevel*, measures the probability of exposure of the item being in a hazardous event situation, and it ranges from E0 to E4, where E4 is the highest exposure level; 3- *ControllabilityLevel*, measures the ability to avoid a specified *harm* through timely reactions, and it ranges from C0 to C3, where C0 means controllable in general and C3 means difficult to control or uncontrollable; 4- *ASILLevel* measures of necessary risk reduction, and its level range from QM, ASIL A, ASIL B, ASIL C, and ASIL D,

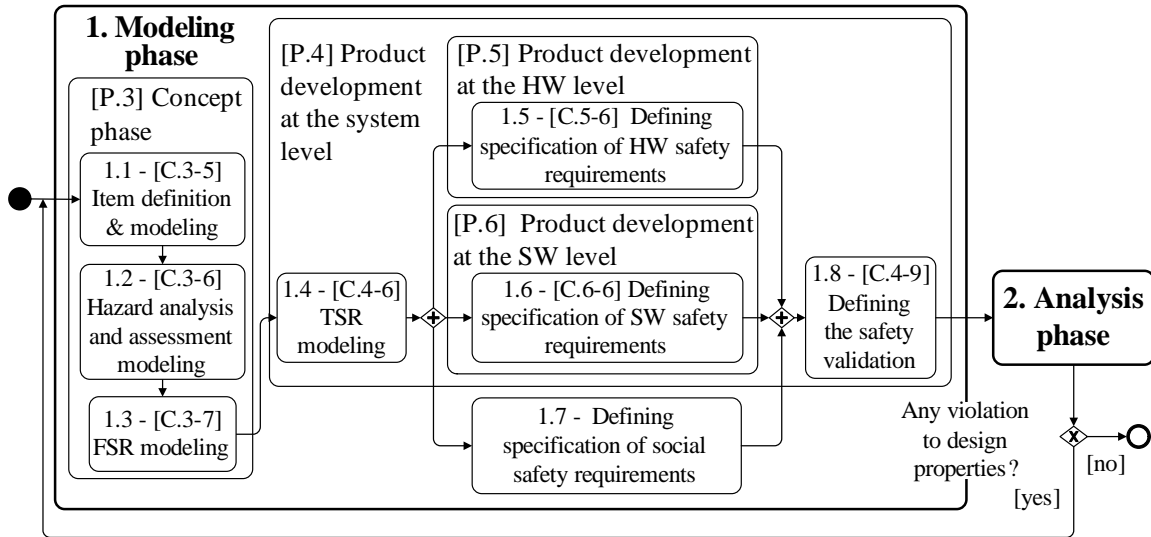


Figure 1. A process for modeling and analyzing the FSR for Automotive System compliant with the ISO 26262

where ASIL D is the highest. In the ISO 26262, the ASIL level is determined based on the levels of severity, probability of exposure and controllability in accordance with Table II. In the case of S0, E0, or C0, no ASIL is assigned and NA is used as a value of ASIL level.

Moreover, several stereotypes have been specialized from the <<Dependency>> Metaclass to capture the relations among the previously mentioned stereotypes. The <<endanger>> stereotype captures dependencies starting from the <<Hazard>> stereotype and points towards the <<FunctionalRequirement>> stereotype, and the <<address>> stereotype captures dependencies starting from the <<SafetyGoal>> stereotype and points towards the <<Hazard>> stereotype. The <<derivedFrom>> stereotype captures dependencies starting from <<FunctionalSafetyRequirement>> stereotype and points towards <<SafetyGoal>> stereotype. The <<definedFrom>> stereotype captures dependencies starting from the <<TechnicalSafetyRequirement>> stereotype and points towards the <<FunctionalSafetyRequirement>> stereotype. Finally, the <<operationalize>> stereotype captures dependencies starting from a <<operationalElement> stereotype and points towards <<TechnicalSafetyRequirement>> stereotype, and it has a type property ('SHWSR', 'SSWSR' or 'SSCSR') to guarantee a correct operationalization.

Figure 3 shows a partial model of the MAS system using our UML profile. In which, we can identify the item that has been represented as an Actor along with two Functional Requirements (Freq_01 and Freq_02) it aims to achieve. Both of these Functional Requirements are represented along with a short description about each of them. Following our methodology, after modeling the item and its functional requirements, we identify and model possible hazards that

Table II
DETERMINING ASIL LEVEL BASED ON SEVERITY, PROBABILITY AND CONTROLLABILITY

Severity level	Probability level	Controllability level		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

can *endanger* the achievement of each of these functional requirements. Hazards H_01 and H_02 *endanger* Freq_01 and Freq_02 respectively. For each identified hazard we perform a risk assessment to assign the appropriate ASIL level. Both of H_01 and H_02 have been associated with ASIL level C, therefore they should be *addressed* by safety goals, e.g., H_01 is *addressed* by safety goal SG_01.

During the third activity of the methodology, we derive at least one FSR from each SG. Two FSRs (FSR_01 and FSR_02) have been *derived from* SG_01. In the fourth activity, we define at least one TSR from each FSR. Four TSRs (TSR_01, TSR_02, FSR_03 and FSR_04) have been *defined from* FSR_02. Fulfilling the complete set of TSRs is considered sufficient to ensure that the item is compliant with its functional safety concept. Therefore, TSRs should be detailed enough to be allocated to different hardware, software and/or social components, which is performed in activities 5, 6 and 7 respectively. For instance, TSR_02 is *operationalized* into SHWSR_01, SSWSR_01 and SSCSR_01 that defines specification for hardware, software, and social safety require-

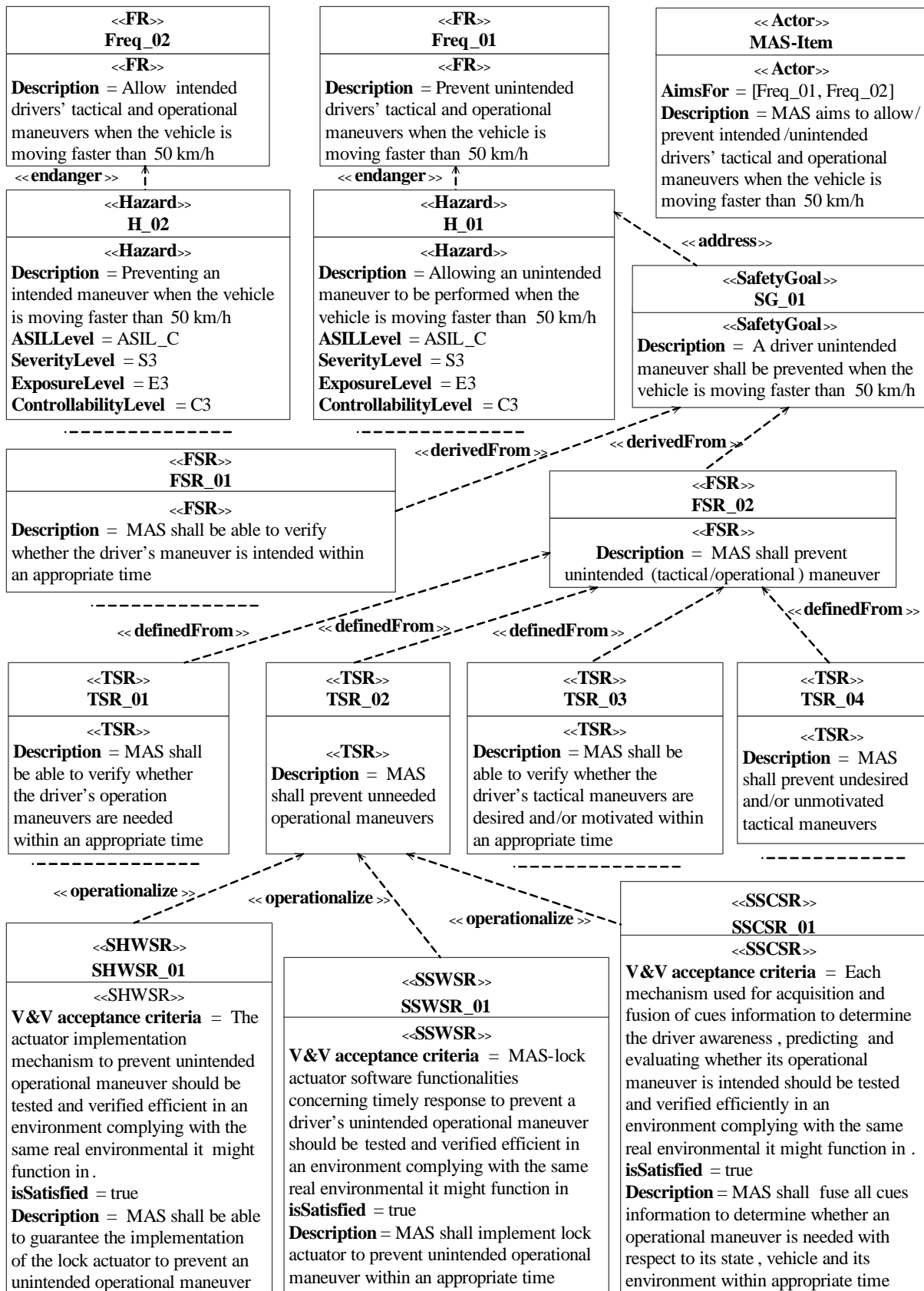


Figure 3. Applying the UML Profile for modeling functional safety concept of MAS

Table III
PROPERTIES OF THE DESIGN.

Pro1.	Dependencies with the stereotype <<endanger>> can only have a class with a stereotype <<Hazard>> as a source of the dependency and a class with a stereotype <<FunctionalRequirement>> as a destination.
Pro2.	Dependencies with the stereotype <<address>> can only have a class with a stereotype <<SafetyGoal>> as a source of the dependency and a class with a stereotype <<Hazard>> as a destination.
Pro3.	Dependencies with the stereotype <<derivedFrom>> can only have a class with a stereotype <<FunctionalSafetyRequirement>> as a source of the dependency and a class with stereotype <<SafetyGoal>> as a destination.
Pro4.	Dependencies with the stereotype <<definedFrom>> can only have a class with a stereotype <<TechnicalSafetyRequirement>> as a source of the dependency and a class with a stereotype <<FunctionalSafetyRequirement>> as a destination.
Pro5.	Dependencies with the stereotype <<operationalize>> can only have a class with a stereotype <<SHWSR>>, <<SSWSR>> or <<SSCSR>> as a source of the dependency and a class with a stereotype <<TechnicalSafetyRequirement>> as a destination.
Pro6.	The type of dependencies with the stereotype <<operationalize>> (e.g., <<SHWSR>>, <<SSWSR>>, <<SSCSR>>) should match the type of a class with the stereotype <<operationalElement>> that is used for the operationalization.
Pro7.	Each class with a stereotype <<Hazard>> should have at least one dependency with a stereotype <<endanger>> points towards a class with a stereotype <<FunctionalRequirement>>.
Pro8.	Each class with a stereotype <<Hazard>> that have ASIL level of ASIL_A, ASIL_B, ASIL_C or ASIL_D should have at least one supplier dependency with a stereotype <<address>> from a class with a stereotype <<SafetyGoal>>.
Pro9.	Each class with a stereotype <<SafetyGoal>> should have at least one dependency with a stereotype <<address>> points towards a class with a stereotype <<Hazard>> and at least one supplier dependency with a stereotype <<derivedFrom>> a class with a stereotype <<FunctionalSafetyRequirement>>.
Pro10.	Each class with a stereotype <<FunctionalSafetyRequirement>> should have at least one dependency with a stereotype <<derivedFrom>> points towards a class with a stereotype <<SafetyGoal>> and at least one supplier dependency with a stereotype <<definedFrom>> from a class with a stereotype <<TechnicalSafetyRequirement>>.
Pro11.	Each class with a stereotype <<TechnicalSafetyRequirement>> should have at least one dependency with a stereotype <<definedFrom>> points towards a class with a stereotype <<FunctionalSafetyRequirement>> and at least one supplier dependency with a stereotype <<operationalize>> from a class with a stereotype <<operationalElement>>.
Pro12.	The ASIL level of each class with a stereotype <<Hazard>> should be determined based on the levels of severity, probability and controllability of the <<Hazard>> in accordance with the Table II.
Pro13.	All classes with stereotype <<operationalElement>> (e.g., <<SHWSR>>, <<SSWSR>>, <<SSCSR>>) that are used for the operationalization of classes with stereotypes <<TechnicalSafetyRequirement>> should be satisfied.

D. Tool Support

We have developed a tool⁴ depending on Eclipse-Papyrus⁵, which allows designers to use the various stereotypes offered by our UML profile for modeling the FSR for automotive systems. In addition, it allows the designer to verify the FSR model depending on the properties of the design (OCL constraints) presented in Table III. In case any of these properties has been violated, the designer will be notified by the exact name of the violation, which enables him/her to address it.

V. THREATS TO VALIDITY

Following Wohlin et al. [9], we classify threats to validity under four types:

1- Construct validity concerns the extent to which a study measures what it claims to be measuring. We have identified one threat, *Poor conceptualization*: occurs when few factors are considered to analyze the subject of the study. To mitigate this threat, our example has been very carefully chosen to cover all the three main aspects (e.g., cyber, physical and social) that might influence the functional safety concept of an automotive system.

2- Internal validity concerns the factors that have not been considered in the study, and they could have influenced

the investigated factors. Our analysis has focused on the three main aspects that we consider essential to guarantee the functional safety concept. However, other factors might be involved as well, which we were not able to identify. Further analysis is required to verify whether the aspects we considered are enough, or identifying other unrevealed aspects.

3- External validity concerns the extent to which the results of the study can be generalized. We have identified two threats, (i) *Completeness of the design properties*: we have identified these properties based on an extensive analysis of available reports and studies concerning FSR. However, we are planning to evaluate their completeness with domain experts. (ii) *Extensive evaluation*: the approach has been applied to only one example, but it covers the main aspects of many complex automotive systems. Moreover, applying our approach to other automotive systems is on our list for future work.

4- Conclusion validity concerns the extent to which the conclusions about relations between the treatment and the outcome of an experiment is correct. We have identified one threat, (i) *Fishing for a specific result*: the process we followed starting from item definition until safety validation is based on well-adopted standard (ISO 26262), which reduces the possibility of this threat. Moreover, the importance of considering the driver behavior has been reported by many other researchers/experts in the automotive domain.

⁴The tool is available at <https://goo.gl/g45S8t>

⁵<https://www.eclipse.org/papyrus/>

VI. RELATED WORK

Several approaches for dealing with functional safety requirements have been proposed in the literature. For instance, Giese et al. [10] propose an approach enables for systematically identifying which hazards/failures are most critical, which components require a more detailed safety analysis, and which restrictions to the failure propagation should be considered. Zhang et al. [11] introduce a comprehensive hazard analysis method based on functional models. Moreover, Li and Zhang [12] present a hazard analysis method for automotive control systems that incorporate safety procedures in the traditional development process. Basir et al. [13] propose an approach that adopts the Goal Structuring Notation (GSN) [14] to construct safety cases to trace requirements to the code. The work of Habli et al. [15] examines how model-driven development and assessment can provide a basis for the systematic generation of functional safety requirements.

Baumgart [16] proposes a method that considers the entire safety lifecycle of functional safety with special emphasis on hazard analysis and risk assessment. Palin et al. [17] provide extensions to GSN with patterns and a number of safety arguments to assist researchers in creating safety cases compliant with the ISO 26262. Moreover, a method to define functional safety requirements depending on GSN notation has been presented in [5], yet GSN does not provide constructs specialized for modeling the social aspects of a system. Finally, Beckers et al. [18] present a model-based method for hazard analysis and risk assessment for automotive systems in the context of ISO 26262, which offers a UML profile and several constraints expressed in OCL to validate the model.

Although most of these approaches propose solutions to improve functional safety, they mainly focus on the technical aspects of the system and ignore the social ones. Moreover, many of them are not equipped with an adequate automated support, which makes them inappropriate for dealing with a large number of hazards, FSR, safety goals, etc.

VII. CONCLUSIONS AND FUTURE WORK

We presented a model-based approach that has been developed based on the ISO 26262 standard and considers both technical and social aspects of such systems. Our approach allows for modeling and analyzing FSR in their social and organizational context, which gives the driver a voice by considering him and his behavior while dealing with FSR.

For the future work, we aim to enrich our modeling language by integrating other social concepts from GORE such as the dependency concept that allows capturing dependencies among the different actors (items) of a system. Moreover, we plan to adopt and/or-decomposition concepts to refine SGs, FSRs, TSRs and provide alternatives for their achievements. Additionally, we intend to contact peer researchers to collect their feedback concerning the approach, and we aim to better validate our approach by applying it to several real case studies. Finally, we are planning to perform a set of experiments with industrial experts to evaluate our approach, i.e., how well it can support its users for modeling and analyzing FSR.

ACKNOWLEDGMENT

This work has been partially supported by the “Ente Cassa Di Risparmio di Firenze”, Bando per progetti 2016, and by the REGIONE TOSCANA POR FESR 2014-2020 SISTER “Signaling & Sensing Technologies in Railway application”.

REFERENCES

- [1] W. Ridderhof, H.-G. Gross, and H. Doerr, “Establishing evidence for safety cases in automotive systems—A case study,” in *Proceedings of International Conference on Computer Safety, Reliability, and Security*. Springer, 2007, pp. 1–13.
- [2] ISO, “26262:2011 Road vehicles-Functional safety,” *International Standard ISO/FDIS*, vol. 26262, 2011.
- [3] M. Gharib, P. Lollini, A. Ceccarelli, and A. Bondavalli, “Dealing with Functional Safety Requirements for Automotive Systems: A Cyber-Physical-Social Approach,” in *The 12th International Conference on Critical Information Infrastructures Security (CRITIS)*. Springer International Publishing, 2017.
- [4] A. Sathyanarayana, P. Boyraz, Z. Purohit, R. Lubag, and J. H. L. Hansen, “Driver adaptive and context aware active safety systems using CAN-bus signals,” in *Intelligent Vehicles Symposium (IV)*. IEEE, 2010, pp. 1236–1241.
- [5] K. Beckers, I. Cote, T. Frese, D. Hatebur, and M. Heisel, “Systematic derivation of functional safety requirements for automotive systems,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8666 LNCS. Springer, 2014, pp. 65–80.
- [6] A. Dardenne, A. van Lamsweerde, and S. Fickas, “Goal-directed requirements acquisition,” *Science of Computer Programming*, vol. 20, no. 1-2, pp. 3–50, 1993.
- [7] OMG-OCL, “Object Constraint Language,” Tech. Rep. May, 2014. [Online]. Available: <http://www.omg.org/spec/OCL/2.4/>
- [8] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, “Tropos: An agent-oriented software development methodology,” *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, 2004.
- [9] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [10] H. Giese, M. Tichy, and D. Schilling, “Compositional Hazard Analysis of UML Component and Deployment Models,” in *Computer Safety, Reliability, and Security*, vol. 3219. Springer, 2004, pp. 166–179.
- [11] H. Zhang, W. Li, and W. Chen, “Model-based hazard analysis method on automotive programmable electronic system,” in *Proceedings of the 3rd International Conference on Biomedical Engineering and Informatics, BMEI*, vol. 7. IEEE, 2010, pp. 2658–2661.
- [12] W. Li and H. Zhang, “A software hazard analysis method for automotive control system,” in *Proceedings of the International Conference on Computer Science and Automation Engineering, CSAE*, vol. 3. IEEE, 2011, pp. 744–748.
- [13] N. Basir, E. Denney, and B. Fischer, “Deriving safety cases for hierarchical structure in model-based development,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6351 LNCS. Springer, 2010, pp. 68–81.
- [14] T. Kelly and R. Weaver, “The Goal Structuring Notation - A Safety Argument Notation,” in *Elements*. Citeseer, 2004.
- [15] I. Habli, I. Ibarra, R. S. Rivett, and T. Kelly, “Model-Based Assurance for Justifying Automotive Functional Safety,” SAE Technical Paper, Tech. Rep. June 2016, 2010.
- [16] S. Baumgart, “Investigations on Hazard Analysis Techniques for Safety Critical Product Lines,” in *Proceedings of the Workshop on Interesting Results in Computer Science and Engineering (IRCSE)*. ACM, 2012.
- [17] R. Palin, D. Ward, I. Habli, and R. Rivett, “ISO 26262 safety cases: compliance and assurance,” in *the 6th IET International Conference on System Safety 2011*, no. November 2014. IET, 2011, pp. 12–18.
- [18] K. Beckers, D. Holling, I. Côté, and D. Hatebur, “A structured hazard analysis and risk assessment method for automotive systems - A descriptive study,” in *Reliability Engineering & System Safety*, vol. 158. IEEE, 2017, pp. 185–195.