The Journal of Systems & Software 189 (2022) 111310

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

A cyber–physical–social approach for engineering Functional Safety Requirements for automotive systems☆

Mohamad Gharib [a,b,*], Andrea Ceccarelli [b], Paolo Lollini [b], Andrea Bondavalli [b]

[a] University of Tartu, Estonia
[b] University of Florence, Italy

ARTICLE INFO

Article history:
Received 19 August 2021
Received in revised form 15 March 2022
Accepted 16 March 2022
Available online 24 March 2022

Keywords:
Functional safety requirements
Automotive
ISO 26262
ISO/PAS 21448
SOTIF
Cyber–Physical–Social systems

ABSTRACT

Several approaches have been developed to assist automotive system manufacturers in designing safer vehicles by facilitating compliance 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 provide inadequate analysis of such important aspects. To this end, we propose a model-based approach for modeling and analyzing the Functional Safety Requirements (FSR) for automotive systems, which considers both the technical and social aspects of such systems. This approach is based on both the ISO 26262 and ISO/PAS 21448 standards, and it proposes a detailed engineering methodology to assist designers while modeling and analyzing FSR. In particular, this approach proposes a UML profile for modeling the FSR of the automotive system starting from item definition until safety validation, and it offers constraints expressed in Object Constraint Language (OCL) to be used for the verification of FSR models. We demonstrated the applicability and usefulness of the approach relying on a realistic example from the automotive domain, and we also evaluated the usability and utility of the approach with potential end-users.

1. Introduction

The automotive industry is responsible for producing millions of new vehicles every year to be used by humans daily. That is why assuring the safety of these vehicles has been always a growing concern for their manufacturers (Gharib et al., 2019). More specifically, automotive systems are safety-critical systems that have to comply with strict safety requirements before they are used in their operational environment (Ridderhof et al., 2007; Kirovskii and Gorelov, 2019). Additionally, the complexity of current automotive systems has increased significantly in terms of its implemented functionalities as they depend on more sophisticated software, sensors, actuators, etc. (Wagner et al., 2010; Dardar et al., 2012), which may increase the risk of systematic failures (Zhang et al., 2010).

To guarantee that such complex automotive systems achieve an acceptable level of safety, a functional safety standard, named ISO 26262 (ISO, 2011), has been developed. The ISO 26262 provides appropriate development processes, requirements and safety integrity levels mainly for the Electrical and Electronic (E/E) systems of vehicles leaving the driver and its behavior outside the scope of the standard (Gharib et al., 2018). To tackle this and other limitations in the ISO 26262, the ISO/PAS 21448:2019 (Road vehicles - Safety Of The Intended Functionality (SOTIF)) (ISO - International Organization for Standardization, 2019) has been developed with a main objective of mitigating hazards resulting from functional insufficiency of the intended functionality or from reasonably foreseeable misuse by persons (Kirovskii and Gorelov, 2019). Although SOTIF was supposed to cover the human aspects, its analysis of possible hazardous triggering events covers only two categories, 1- algorithms and 2- sensors and actuators (Walker, 2019), i.e., it does not specifically focus on human aspects.

Drivers have been consistently considered as one of the main factors in a high proportion of accidents (Salmon et al., 2005; McCall and Trivedi, 2007; Hoess and Gmbh, 2009; Sathyanarayana et al., 2010). To this end, assuring the safety of automotive systems requires considering also the driver and his/her behavior (Gharib et al., 2019), i.e., it is clear that vehicle safety is more than a purely technical issue (Saffarian et al., 2012). In particular, an automotive system can be seen as a Cyber–Physical–Social system (CPSS), which is composed of cyber components (e.g., E/E systems), controlled components (e.g., vehicles, traffic lights) and social components (e.g., drivers) (Gharib et al., 2019). Therefore, the safety of such systems cannot be assured without considering

☆ Editor: Doo-Hwan Bae.

* Corresponding author at: University of Tartu, Estonia.
E-mail addresses: mohamad.gharib@unifi.it, mohamad.gharib@ut.ee (M. Gharib), andrea.ceccarelli@unifi.it (A. Ceccarelli), paolo.lollini@unifi.it (P. Lollini), andrea.bondavalli@unifi.it (A. Bondavalli).

https://doi.org/10.1016/j.jss.2022.111310
0164-1212/

their three main components. More specifically, ignoring the social components during the CPSS design leaves the system open to various 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's behavior is also considered during the system design (Regan et al., 2005; Sathyanarayana et al., 2010).

To address this problem, we proposed a model-based approach for modeling and analyzing the FSR for automotive systems (Gharib et al., 2019), which is based on the ISO 26262 standard but considers both the E/E systems as well as the drivers' behavior. In particular, we refine and improve our previous work (Gharib et al., 2019) by extending our approach to cover safety-related issues resulting from drivers' behavior that might influence the intended functionality of the item. This new extension is grounded on the ISO/PAS 21448:2019 standard. Further, the approach follows a well-defined methodology (Selic, 2007) for the development of Domain Specific Modeling Language (DSML) that relies on a UML profile, which is associated with OCL constraints to verify the correctness and consistency of models. In particular, the approach proposes a UML profile for modeling FSR, and a set of constraints expressed in OCL (OMG-OCL, 2014) to be used for the verification of FSR model. Moreover, we demonstrated the applicability and usefulness of the approach relying on a realistic example from the automotive domain. Moreover, we evaluated the usability and utility of the approach with potential end-users.

Our approach has been developed following a Design Science Research (DSR) approach (Hevner and Park, 2017), which identifies the problem that needs to be solved, motivates the development of the solution as a design artifact (e.g., an approach), then, evaluates the application of the developed solution through a relevant scenario (Bell et al., 2007; Wieringa, 2009).

More specifically, our research methodology is based on the steps suggested by Bell et al. (2007): 1- *Identification of the problem,* as discussed earlier there is a need for an approach specialized for modeling and analyzing the FSR for automotive systems that considers both the E/E systems as well as the drivers' behavior. 2- *Approach design,* we develop an approach to tackle the aforementioned need. 3- *Approach evaluation,* we evaluate the approach based on how well it supports solutions in the problem space. In particular, we demonstrate its applicability and usability for modeling FSR, and its effectiveness in capturing wrong/bad design practices concerning FSR models. 4- *Improve and re-evaluate the approach,* focuses on identifying limitations or areas of improvement and refining the approach. In this paper, we identify the limitations and threats to the validity of the approach and consider addressing them in future research.

The rest of the paper is organized as follows; Section 2 presents the research baseline, and an illustrative example concerning a Maneuver Assistant System is described in Section 3. In Section 4, we present and discuss our approach, and we evaluate it in Section 5. We discuss threats to the approach validity in Section 6. Related work is presented in Section 7, and we conclude and discuss future work in Section 8.

2. Research baseline

2.1. ISO 26262

ISO 26262:2011 (ISO, 2011) is a functional safety standard that has been developed to provide guidelines and best practices to increase the safety of E/E systems for all road vehicles with a weight under 3500 kg. ISO 26262 provides safety specifications that cover the whole life cycle (e.g., design, specification, implementation, integration, verification and validation) of critical "items", which can be defined as a single safety-critical automotive E/E system in the context of ISO 26262. The ISO 26262 focuses on the hazards of the E/E systems and their associated risks, where each associated risk is then assigned an Automotive Safety Integrity Level (ASIL). The ASILs can be classified under Quality Management (QM) that is assigned to hazard with very low probability, causing only slight injuries or can be avoided by the driver, and it does not require risk reduction effort. Then, ASIL A-D require risk reduction effort, where ASIL D requires the highest reduction. Usually, ASILs are used to specify the extent to which the safety activities of the item need to be addressed. Table 1 shows the main clauses of ISO 26262 relevant to the different phases of product development with a short description of each.

Table 1
Main clauses of 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, 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.

2.2. ISO/PAS 21448

The ISO/PAS 21448:2019 (Road vehicles - SOTIF) (ISO - International Organization for Standardization, 2019) is a standard that aims at guiding automotive system development to avoid potentially hazardous system behavior, in the absence of unreasonable risk due to hazards resulting from functional insufficiency of the intended functionality or from reasonably foreseeable misuse by persons. The ISO PAS 21448 consists of various clauses that run along with the V life-cycle model of the ISO 26262 (Kirovskii and Gorelov, 2019). Please note that the objective clauses of ISO 21448 (Clauses 5.1, 6.1, 7.1, 8.1, 9.1, 10.1, 11.1 and 12.1) are considered normative, and the rest of the content is considered informative (ISO - International Organization for Standardization, 2019). Table 2 shows the main clauses of ISO/PAS 21448 (SOTIF) that are relevant to this work with a short description of their objectives.

2.3. Human error and road transport

Humans by their very nature make mistakes; therefore, it is unreasonable to expect error-free human performance (Shappell and Wiegmann, 1997). Recent studies indicate that humans' errors are the main source of incidents occurring in the aviation domain (McFadden and Towell, 1999), rail domain (Lawton and Ward, 2005), medical domain (Helmreich, 2000) as well as

the automotive domain (Sathyanarayana et al., 2010). Moreover, humans' errors have contributed or led to several catastrophes incidents such as the Tenerife airport disaster (Weick, 1990), the Papa India/Staines (Stanton et al., 2010) air disasters, and the three-mile island accident and Chernobyl nuclear power disasters (Meshkati, 1991), etc. Accordingly, humans' errors have received considerable attention from academia, industry and also from the general public (Salmon et al., 2005).

To get a better understanding of humans' errors, various attempts have been made for clarifying this concept, yet no generally accepted definition has been reached (Salmon et al., 2005). Although several definitions have been introduced (e.g., Senders and Moray (1991), Salmon et al. (2005)), we adopt the one proposed by Reason (Reason, 1991) that defined human error as "a generic term to encompass all those occasions in which a planned sequence of mental or physical activities fails to achieve its intended outcome, and when these failures cannot be attributed to the intervention of some chance agency". Based on this definition, Salmon et al. (2005) concluded that *human error can, therefore, be generally defined as any mental or physical activity, or failure to perform the activity, that leads to either an undesired or unacceptable outcome*.

On the other hand, several researchers suggested linking human error to unsafe acts/operations (Rasmussen, 1982; Reason, 1991; Shappell and Wiegmann, 1997), and several taxonomies have been proposed for such unsafe acts/operations. The most prominent and commonly referred to within the literature is the taxonomy of unsafe acts proposed in Reason (1991) that is shown in Fig. 1. In this taxonomy, *Unsafe acts* are classified into *Unintended actions* and *Intended actions*. The first can be defined as actions that unwittingly deviate from planned intentions due to failures of execution or memory and can be classified into *Slips* or *Lapses* (Shappell and Wiegmann, 1997). *Slips* are characteristic of attention failures and may take the form of inadvertent activations, interference errors, omissions following interruptions, and order reversals, among others. For example, a driver meant to slow down but applied the full-break instead. In contrast, *Lapses* arise from memory failures and include errors such as omitted items in a checklist, place losing, or forgotten intentions. For instance, a driver forgot that the autonomous mode has been deactivated.

Intended actions includes *Mistakes* and *Violations*. *Mistakes* can be defined as intentional behavior that proceeds as an intended behavior that failed to achieve the desired outcome due to problem-solving or planning failures. For example, a driver deactivated the autonomous mode due to wrong assessment of a situation. *Violations* include any behavior that deviates from accepted procedures, standards and/or rules. For instance, a driver saw a stop sign but chooses not to comply with it. Further, Reason classified *Slips*, *Lapses* and *Mistakes* as the *Basic error types*, which we consider in this work.

2.4. 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 such as KAOS (Dardenne et al., 1993), GBRAM (Anton and Potts, 1998), *i** (Yu, 1995), etc., Tropos (Bresciani et al., 2004), an agent-oriented software engineering (AOSE) methodology that is developed based on *i**, has been proven efficient for modeling requirements in their social and organizational context.

In Tropos, the system is modeled in terms of its main actors (agentive entities) along with their objectives, entitlements, and social dependencies for objectives and entitlements. In particular, the language introduces primitives for modeling *actors* in terms of *agents* and the *roles* they are playing. *Goals* are intentional entities and they are used to represent a strategic interest that *actors* intend to achieve. A *task* represents an abstract way to do something, and the execution of a *task* can be a mean 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. Refining a root-goal/root-task into sub-goals/sub-tasks through *and-decomposition* implies that all sub-goals/sub-tasks need to be achieved in order to achieve the parent goal/task, and refining them through *or-decomposition* implies achieving any of them achieve the root-goal/root-task. A *resource* represents a physical or an informational entity. Finally, a *dependency* allows *actors* to *depend* on one another for fulfillment of *goals*, execution of *tasks*, and provision of *resources*. Fig. 2 shows a partial goal model of Maneuver Assistant System represented with Tropos main concepts.

Table 2
Main clauses of ISO/PAS 21448 (SOTIF) for analyzing the Safety Of The Intended Functionality.

Clause	Objective
C. 5–1	**Functional and system specification:** 1- Compile and create evidence containing the information sufficient to initiate the SOTIF related activities; 2- Update the evidence as necessary after each iteration of the SOTIF related activities.
C. 6–1	**Identification and Evaluation of hazards caused by the intended functionality:** the potential hazards related to the SOTIF shall be systematically identified and evaluated.
C. 7–1	**Identification and Evaluation of triggering events:** triggering events that can trigger potentially hazardous behavior shall be identified, and they shall be evaluated for their acceptability with respect to the SOTIF.
C. 8–1	**Functional modifications to reduce SOTIF related risks:** The development activities of the functional modifications to reduce the SOTIF related risks shall achieve the following objectives: 1-identification and allocation of measures to avoid, reduce, or mitigate the SOTIF related risks; 2-estimation of the effect of the SOTIF related measures on the intended function; and 3- improvement of the information required by Clause 5 (Functional and system specification).
C. 9–1	**Definition of the Verification and Validation strategy:** A verification and validation strategy shall be defined such that: it supports the rationale for the SOTIF; the necessary evidence (e.g. analysis results, test reports) is generated; and the procedures to generate the evidence are developed.

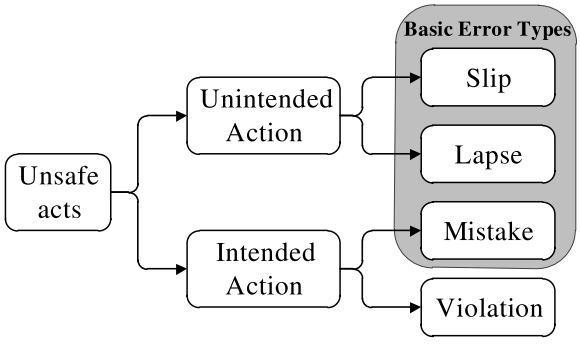

Fig. 1. Unsafe acts taxonomy (Reason, 1991).

3. Illustrative example: Maneuver Assistance System

Our example concerns the Maneuver Assistance System (MAS),[1] which is a highly automated driving system. MAS is

[1] For more information about the example, please refer to Gharib et al. (2018).

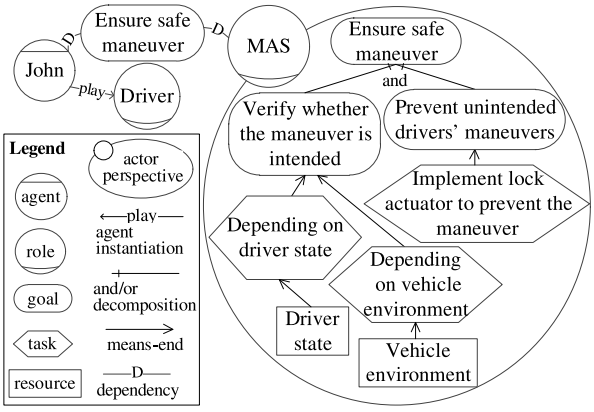

Fig. 2. A partial goal model of the MAS represented in Tropos.

expected to increase the safety of the driver by detecting and preventing driver's unintended maneuvers. According to Tawari et al. (2014), there are three types of maneuvers: 1- *strategically-planned maneuver* that is motivated by the destination goal of the driver, and it is associated with long time scale (minutes or even hours); 2- *tactical-planned maneuver* that is motivated by a recently modified desire of the driver (e.g., lane changes, turns), and it is associated with a short-term timescale (tens of seconds); and 3- *operational maneuver* that is, usually, a result of a driver's desire to remain safe (e.g., avoid collision), and it is associated with a very short time scale (hundreds of milliseconds). MAS focuses on the last two types of maneuvers since the first one does not involve safety-critical situations.

MAS collects information about the vehicle, vehicle surroundings, as well as the driver's behavior to predict her/his intentions. Then, MAS analyzes such information to determine the driver's intentions and 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, i.e., it is a result of a driver's *error* (e.g., *a slip*, a *lapse* or a *mistake*). Accordingly, MAS allows *intended* maneuvers and prevents *unintended* ones.

Fig. 3 illustrates three different types of maneuvers: (1) The driver may have a *desire* to perform a maneuver in order to overtake a leading vehicle; (2) The driver *needs* to perform a maneuver to get back to her original lane, especially, because there is a vehicle in the opposing direction; and (3) The driver does not have any *desire* nor *need* to perform a maneuver, especially, because performing such a maneuver will lead to a head-on collision. The first two maneuvers are motivated by a *desire* or a *need*, thus, they should be considered as *intended* ones and allowed by the MAS. The third maneuver is not motivated by any *desire* or any *need*, therefore, it should be considered as an *unintended* one and prevented by the MAS.

4. A model-based approach for engineering Functional Safety Requirements

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

4.1. Methodology

The process underlying our methodology is shown in Fig. 4, and it adopts our approach proposed in Gharib et al. (2019). The process has been developed based on the ISO 26262 standard and it considers the E/E systems as well as the drivers' behavior. Additionally, it has several activities that have been developed based on the ISO PAS 21448 to enrich the analysis of the intended functionality. Note that C. represents the Clauses of the ISO 26262 and ISO PAS 21448 standards that have been used as a basis for the definition of some activities of the methodology.[2] The process is composed of two main phases, namely modeling and analysis:

(1) **Modeling phase** aims at modeling the functional safety concept of an automotive system starting from item definition and modeling until the definition of safety validation. This phase is composed of 12 activities, four of them (S1, S2, S3 and S4) are dedicated for the analysis of the intended functionality of the item that might be influenced by the behavior of the driver, and these four activities may start after activity *1.1*.

1.1- Item definition and modeling is the first activity of the process, in which the item needs to be defined and modeled along with the main functional requirements it aims to achieve. Moreover, if the intended functionality of the item might be influenced by the driver' behavior/activities, S1. activity starts, otherwise, there is no need for the SOTIF-based analysis (e.g., S1, S2, S3 and S4).

S1. Functional and system specification: aims at collecting sufficient information concerning the drivers' intended behavior/actions while interacting with the item. Such information should cover also the drivers' unintended behavior/actions that may result due to his errors (e.g., Slips, Lapses and Mistakes), which might influence the intended functionality of the item.

S2. Identification and Evaluation of hazards caused by the intended functionality: aims at identifying and evaluating hazards that might be caused by the behavior/actions of the drivers and influence the intended functionality of the item. Unlike ISO 26262, SOTIF standard does not consider (neither implicitly nor explicitly) the acceptable level of risk nor the effort required to address it (Kirovskii and Gorelov, 2019) that is why we integrate such activities into ISO 26262 -based activity (*1.2- HARA modeling*), i.e., activity *1.2* is extended to also consider hazards related to the intended functionality of the item that are identified in **S2.** The outcome of this activity is evaluated and if the *Risk of Harm is Acceptable*, we end the analysis without considering such Hazard as its related harm is insignificant, i.e., we do not modify the item definition. Otherwise, we proceed to activity **S3.**

S3. Identification and Evaluation of triggering events: aims at identifying Triggering Events (TEs) that can trigger the identified hazards. Then, evaluating their acceptability, i.e., a TE is acceptable if it has low exposure probability, high controllability, or risks associated with them bear low severity. If all TEs are acceptable, we end the analysis without modifying the item definition. Otherwise, we proceed to activity **S4.**

[2] A short description of these clauses can be found in Tables 1 and 2 respectively.

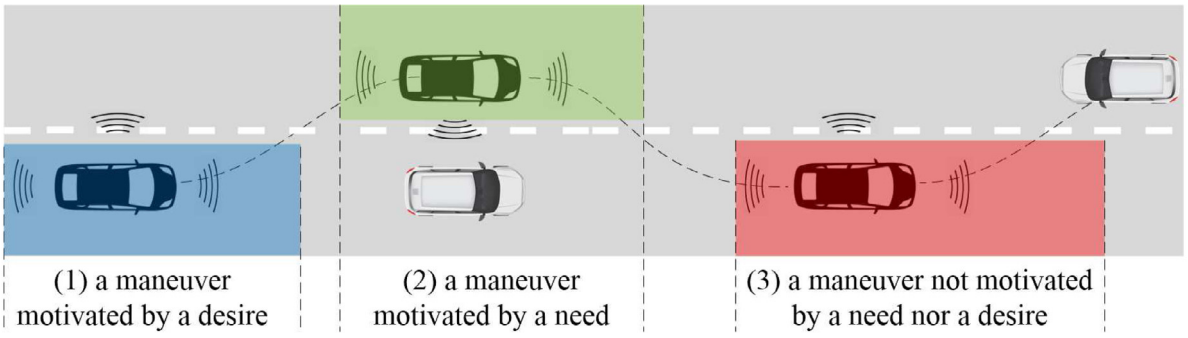

Fig. 3. A diagram representing three different types of maneuvers.

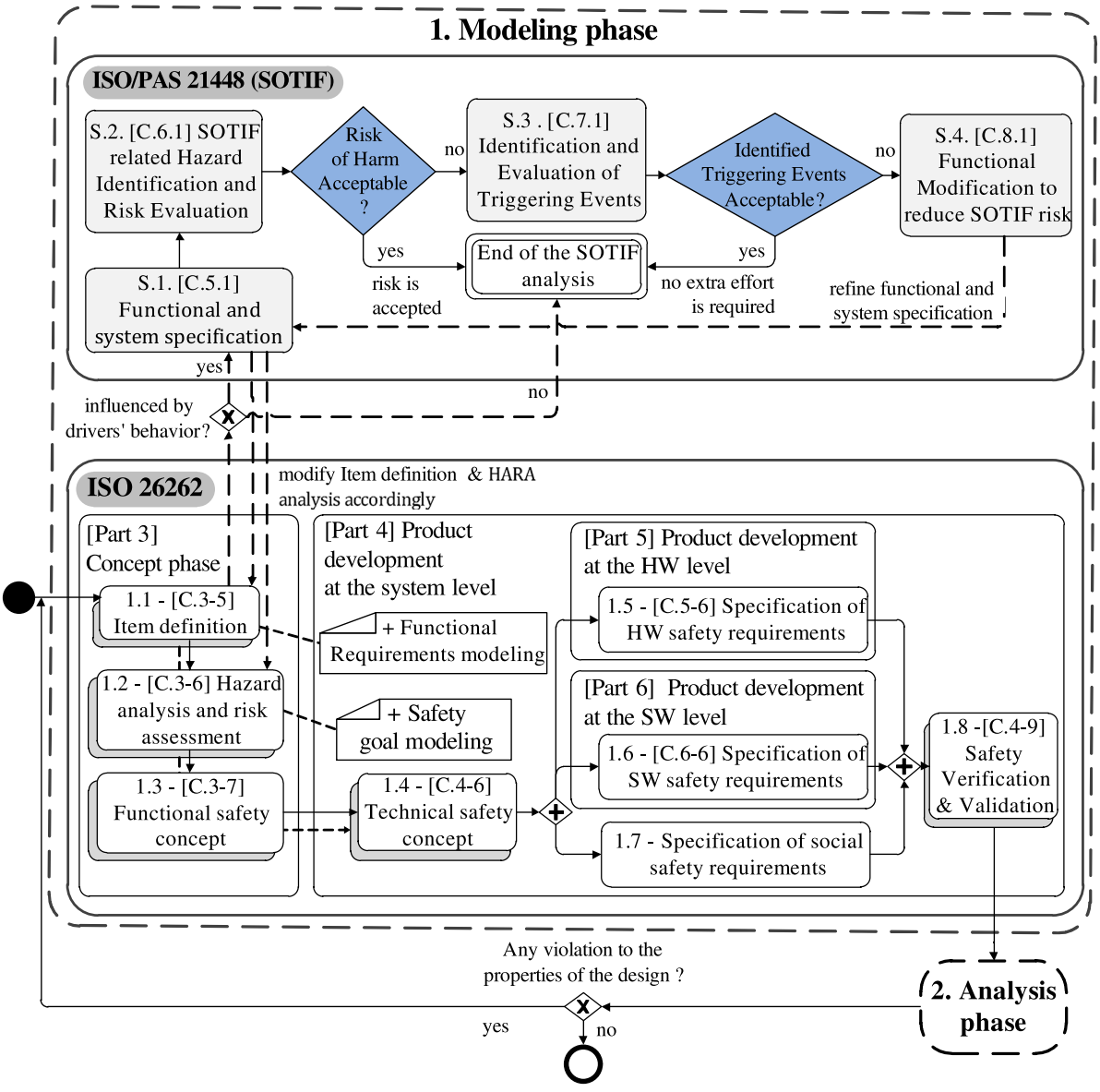

Fig. 4. A process for modeling and analyzing the FSR for Automotive System based on the ISO 26262 & ISO PAS 21448.

S4. Functional modifications to reduce SOTIF related risks: aims at 1- identifying and allocating measures to avoid, reduce, or mitigate the identified risks; 2- estimate the effect of the related measures on the intended function; and 3- improving/refining the functional and system specification taking into consideration identified measures and their effect on the

intended function. The outcome of this activity is used to refine the functional and system specification (S1.), which in turn updates the Item definition (activity 1.1) and HARA analysis (activity 1.2) accordingly based on the analysis performed in S1, S2, S3 and S4 activities.

1.2- HARA modeling, this activity has two sub-activities: **(i) Hazard identification** that identifies and models all possible hazards that can endanger the achievement of each functional requirement of the item, and **(ii) Risk assessment** that perform a risk assessment for each identified hazard 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 A, B, C or D should be addressed by at least one Safety Goal (SG). Note that this activity covers also hazards related to the intended functionality of the item that have been identified in **S2.**

1.3- FSR modeling, derives at least one FSR from each SG that have been identified in the previous activity. According to the ISO 26262, FSRs are used for defining the safety functionalities without specifying how such functionalities can be implemented.

1.4- Technical Safety Requirements (TSR) modeling, defines at least one TSR from each FSR that have been identified in the FSR modeling activity. TSRs should be defined in a way that provides more detailed and specific information about their corresponding FSRs, which in turn facilitates the allocation of these TSRs to the various hardware, software, and social safety requirements (HWSRs, SWSRs, and SCSRs). Note that defining TSRs is not an easy task and it may require domain experts.

1.5- Defining Specifications of Hardware Safety Requirements (HWSRs), defines specifications that can be used for the operationalization of each identified TSR that can be allocated to physical hardware. According to the ISO 26262, HWSRs shall include information about safety mechanisms relevant to (a) control internal failures of the hardware of the element; (b) control or tolerate failures external to the element; (c) comply with the safety requirements of other elements; and (d) detect and signal internal or external failures.

1.6- Defining Specifications of Software Safety Requirements (SWSRs), defines specifications that can be used for the operationalization of each identified TSR that can be allocated to software. According to ISO 26262, SWSRs shall be derived from TSRs considering the required safety-related functionalities and properties of the software. Then, SWSRs can be used to define software design specifications.

1.7- Defining Specifications of SoCial Safety Requirements (SCSRs), defines specifications that can be used for the operationalization of each identified TSR that can be allocated to social behavior. Unlike the previous two activities, this activity is not based on any of the ISO 26262 clauses. However, it follows the same pattern of activities 5 and 6, i.e., SCSRs also use TSRs to define clear design specifications concerning the driver's behavior and its interactions and dependencies with other components of the item.

1.8- Defining safety Verification and Validation (V&V), 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, we have defined and formulated as OCL. This phase will be discussed in detail in Section 4.3.

4.2. Modeling phase

In what follows, we present our UML profile (shown in Fig. 5) that can be used for modeling the FSR for automotive systems.

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>>` from Tropos to propose two stereotypes with the same names to capture the social aspects of the item. `<<Actor>>` has a property to identify the requirements it aims for. For capturing intentional entities related to the item, we follow Tropos and propose the `<<IntentionalElement>>` stereotype, which is specialized into three stereotypes namely, `<<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 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 notion of the task concept in Tropos, and it is further specialized into three stereotypes `<<SHWSR>>`, `<<SSWSR>>`, and `<<SSCSR>>` that define the specification of hardware, software and social safety requirements respectively. The `OE` stereotype has two properties, the first one identifies the V&V acceptance criteria that an `OE` should achieve to be considered satisfied, and 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 has been satisfied or not.

The `<<Hazard>>` stereotype has been developed based on the Hazard concept presented in the ISO 26262 standard, and it captures any hazard that can endanger the achievement of a functional requirement. `<<Hazard>>` has several properties that can be used for the assessment of its risk: 1- *Severity Level,* measures the potential harm of hazard that can range from S0 to S3, where S0 means no injuries and S3 means life-threatening injuries. 2- *Exposure Level,* measures the probability of the item being in an operational situation that is described in the hazardous event, and it can range from E0 to E4, where E0 means the lowest occurrence probability and E4 mean high probability. 3- *Controllability Level,* 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* represents the necessary risk reduction, and its level range from QM, ASIL A, ASIL B, ASIL C, and ASIL D, where ASIL D is the highest. In the case of S0, E0, or C0, no ASIL is assigned and NA is used as a value of ASIL level. The ASIL level is determined based on the levels of severity, probability and controllability in accordance with Table 3.

Moreover, several stereotypes have been specialized from the `<<Dependency>>` metaclass to capture the different relations among the previously mentioned stereotypes. `<<endanger>>` stereotype captures dependencies starting from `<<Hazard>>` and pointing towards `<<FunctionalRequirement>>`,

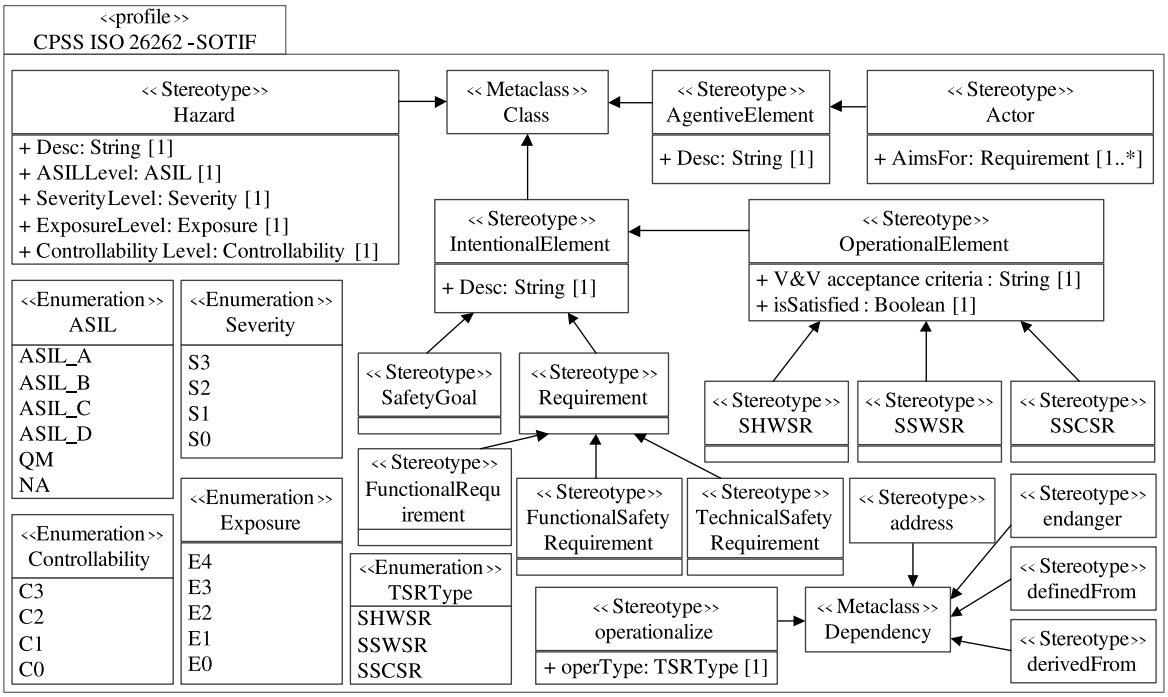

Fig. 5. UML Profile for modeling functional safety concept.

and <<address>> stereotype captures dependencies starting from <<SafetyGoal>> and pointing towards <<Hazard>>. <<derivedFrom>> stereotype captures dependencies starting from <<FunctionalSafetyRequirement>> and pointing towards <<SafetyGoal>>. <<definedFrom>> stereotype captures dependencies starting from <<TechnicalSafetyRequirement>> and pointing towards <<FunctionalSafetyRequirement>>.

Finally, <<operationalize>> stereotype captures dependencies starting from a <<operationalElement> and pointing towards <<TechnicalSafetyRequirement>>, and it has a type property that can be 'SHWSR' or 'SSWSR' or 'SSCSR' to facilitate the allocation of <<TechnicalSafetyRequirement>> to the correct <<operationalElement>> (e.g., <<SHWSR>>, <<SSWSR>>, <<SSCSR>>).

Table 3
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

4.3. Analysis phase

After completing the modeling phase, we have a model that represents the FSR of the item/automotive system. However, we cannot rely only on the model to perform the required analysis to verify the correctness and consistency of the FSR model. More specifically, software engineers will require automated analysis support to verify the correctness of the model with respect to already defined properties of interest/design. Such analysis can be very useful when dealing with large and complex models that cannot be verified/checked manually. In this context, we selected OCL since it is a highly expressive language and can substantially enrich modeling languages like UML by formulating precise definitions of model properties (Kuhlmann et al., 2011). Moreover, OCL can be used to specify constraints while defining the UML profile, allowing a more expressive profile that assures a more complete specification of a system (Ali et al., 2011; Richters and Gogolla, 2002; Kuhlmann et al., 2011).

Therefore, we have defined a set of properties of the design (shown in Table 4) expressed in OCL, which specify logical constraints that the designers should consider during the automotive system design. In particular, these constraints restrict the existence of some of the relations among the elements of the model, forcing the existence of other relations, as well as evaluating the value of some attributes. Additionally, they can be used to evaluate the criteria for the validation and verification of the model. If all of these properties hold, the model is considered correct and consistent. However, if any of them has been violated (e.g., missing an element or a relation, mismatching relation, invalid value, etc.) the designer will be notified of such violation, which enables him/her to perform the required modifications to address it.

We briefly discuss these properties as follows: **Pro1**, **Pro2**, **Pro3**, **Pro4** & **Pro5** are used to restrict the use of some relationships, i.e., verify that these relationships are used as they supposed to in the model. For example, **Pro1** is used to restrict the use of *endanger* relationship to point only from a *Hazard* class toward a *Functional Requirement* class, i.e., a *Hazard endangers* a *Functional Requirement*. In case, the *endanger* relationship has been used to point from any class other than the *Hazard* class and/or to point toward any class other than a *Functional Requirement*, the designer will be notified about such violation. **Pro6** is

Table 4
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>> pointing towards a class with a stereotype <<FunctionalRequirement>>.
Pro8.	Each class with a stereotype <<Hazard>> that have ASIL level of ASIL_A-D should has 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>> pointing towards a class with a stereotype <<Hazard>> and at least one supplier dependency with a stereotype <<derivedFrom>> from a class with a stereotype <<FunctionalSafetyRequirement>>.
Pro10.	Each class with a stereotype <<FunctionalSafetyRequirement>> should have at least one dependency with a stereotype <<derivedFrom>> pointing 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>> pointing 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 3.
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.

used to verify that the type of the *operationalize* relationship is compliant with the type of the *operational Element* class that is used for the operationalization.

Pro7 is used to verify that the model does not include any *Hazard* without specifying the *Functional Requirement* that is *endangered* by such *Hazard*, and **Pro8** is used to verify that the model does not include any *Hazard*, which has ASIL A-D that is not *addressed* by at least one *Safety Goal*. **Pro9** is used to verify that the model does not include any *Safety Goal* that is not used for addressing at least one *Hazard*, and/or not derived from a *FSR*. **Pro10** is used to verify that the model does not include any *FSR* without specifying the *Safety Goal* that such *FSR* is derived from, and the *TSR* that is defined based on this *FSR*.

Pro11 is used to verify that the model does not include any *TSR* without specifying the *FSR* that such *TSR* is defined from, and the *Operational Element* that is used for the operationalization of such *TSR*, and **Pro12** is used to verify that the model does not include any *Hazard* that its ASIL is not determined based on Table 3. Finally, **Pro13** is used to verify that the model does not include any *Operational Element* that is not satisfied.

In what follows, we present three listings that show how such properties are expressed in OCL:

Listing 1. shows an OCL concerning Pro 2 that constraints the client (source) of any dependency with the stereotype <<address>> to a class with a stereotype <<SafetyGoal>>, and the supplier (destination) of such dependency to a class with a stereotype <<Hazard>>. This guarantees that dependencies with a stereotype <<address>> can only points from a class with a stereotype <<SafetyGoal>> towards a class with a stereotype <<Hazard>>.

Listing 1: OCL constrain for verifying Pro2.

```
{OCL} -- context = address
self.base_Dependency.client->any(true).g
etAppliedStereotypes().name->includes
('SafetyGoal') and self.base_Dependency.s
upplier->any(true).getAppliedStereotyp
es().name->includes('Hazard')
```

Listing 2. shows an OCL concerning Pro 6, which verifies that the type of an <<operationalize>> dependency matches the type of the <<operationalElement>> class that is used for its operationalization. This guarantees that any class with a stereotype <<operationalElement>> is operationalized relying on the correct hardware (<<SHWSR>>), software (<<SSWSR>>) or social safety requirements specification (<<SSCSR>>).

Listing 2: OCL constrain for verifying Pro6.

```
{OCL} -- context = operationalize
self.operationalizationType = TSRType::
SHWSR implies self.base_Dependency.client
->any(true).getAppliedStereotypes().na
me->includes ('SHWSR') and self.operat
ionalizationType = TSRType:: SSWSR im
plies self.base_Dependency.client->any(tr
ue).getAppliedStereotypes().name->incl
udes ('SSWSR') and self.operationalizat
ionType = TSRType:: SSCSR implies self.ba
se_Dependency.client->any(true).getApp
liedStereotypes().name->includes('SSC
SR')
```

Listing 3. shows an OCL concerning Pro 8, which constraints classes with a stereotype <<Hazard>> that is associated with ASIL level of ASIL_A-D to have at least one (more than zero) supplier (incoming) dependency with the stereotype <<address>>. This guarantees that any class with a stereotype <<Hazard>>, which is associated with ASIL level of ASIL_A-D is addressed by at least one class with a stereotype <<SafetyGoal>>.

Listing 3: OCL constrain for verifying Pro8.

```
{OCL} -- context = Hazard
self.ASILLevel = ASIL::ASIL_A or self.AS
ILLevel = ASIL::ASIL_B or self.ASILLev
```

```
el= ASIL::ASIL_C or self.ASILLevel = AS
IL::ASIL_D implies self.base_Class.suppl
ierDependency->any(true).getAppliedSte
reotypes().name->includes('address')->
size()> 0
```

4.4. Tool: Prototype implementation

We have developed a tool,[3] depending on Eclipse-Papyrus[4] 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 4. 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.

5. Evaluation

Design artifacts are evaluated based on how well they support solutions (e.g., validity, usability, and utility) for the problems, which they have been developed to solve (Hevner and Park, 2017), i.e., the artifact achieves the purpose for which it was designed (Peffers et al., 2008; Venable et al., 2012). In what follows, we discuss how we demonstrated the **validity** of the approach. Then, we present the evaluation of its **usability and utility**.

5.1. Demonstrating the validity of the approach

We demonstrate the validity of our approach depending on a simulation method (experiment) (Hevner and Park, 2017; Venable et al., 2012) by developing a prototype implementation (e.g., a tool) of our approach and demonstrate its applicability and effectiveness for modeling and analyzing the FSR of a realistic example (e.g., MAS). Following de França and Travassos (2016), we focused on identifying the features of the system that are sufficient to serve the specific objectives of the study, i.e., the main actors of MAS, their interaction and FSR. Next, we used the reasoning support techniques to verify the correctness and consistency of the model.

Applicability and effectiveness: in what follows, we demonstrate the applicability of our approach by using it to model and analyze the FSR of the MAS system.

Following our methodology, we start by defining and modeling the Item as well as its Functional Requirements (FRs). In particular, MAS is an ADAS that rely on sensors to collect information about the driver's behavior to predict her/his intentions. MAS also depends on a LIDAR and Radar to collect information about surrounding vehicles, which helps in analyzing whether there is a need for such maneuvers. Moreover, MAS includes a software system that enables for analyzing all collected information in a timely manner to decide whether a driver's maneuver is intended or unintended. Finally, MAS depends on the lock actuator to prevent a driver's unintended maneuvers. To this end, MAS aims for two main functional requirements (FRs):

FR_01: preventing unintended drivers' tactical and operational maneuvers when the vehicle is moving faster than 50 km/h,

FR_02: allowing intended drivers' tactical and operational maneuvers when the vehicle is moving faster than 50 km/h

Since the intended functionality of the item relies heavily on the driver's intended/unintended behavior, i.e., might be influenced by a driver's error (e.g., *a slip*, a *lapse* or a *mistake*), we proceed to activity **S1**.

S1 aims at further investigating how the drivers' intended/unintended behavior/actions may influence the intended functionality of the item. To this end, MAS needs to differentiate between drivers' intended and unintended maneuvers. More specifically, MAS shall collect enough information concerning the drivers' behavior/actions that enable it to correctly categorize unintended drivers' maneuvers as unintended ones, and also correctly categorize intended drivers' maneuvers as intended ones.

S2 identify and evaluate hazards related to the unintended behavior/actions of the drivers that might influence the intended behavior of the item. Two hazards related to the behavior/actions of the drivers have been identified:

H_01: categorizing an unintended driver's maneuver due to driver's error as an intended one when the vehicle is moving faster than 50 km/h.

H_02: categorizing an intended driver's maneuver as an unintended one when the vehicle is moving faster than 50 km/h.

If it is shown that a potentially hazardous event does not lead to harm, then no improvement is required and the intended functionality can be considered free from unreasonable risk (ISO - International Organization for Standardization, 2019). However, miss-categorizing an intended or unintended maneuver at such speed may lead to life-threatening injuries. Therefore, an analysis of the possible hazardous triggering events should be conducted (ISO - International Organization for Standardization, 2019), i.e., we proceed to **S3**.

S3. identifies and evaluates triggering events. `H_01` can be triggered due to a driver's error (e.g., a slip, a lapse or a mistake) that leads to an unintended maneuver and misinterpreting such errors as an intended action. Such a triggering event is not acceptable since it has high exposure probability, may have low controllability, and its risks are associated with high severity. `H_02` can be triggered due to misinterpreting the driver's behavior/actions, which heavily rely on the type, amount and quality of information collected about the driver's behavior/actions. This triggering event is also not acceptable since it may have high exposure probability, especially, when relying on inappropriate information to analyze the driver's intentions; it may have low controllability, and its risks are associated with high severity. Therefore, we proceed to **S4**.

In **S4**, we need to identify measures to avoid, reduce, or mitigate the identified risks, where such measures may include modifications to the functional and system specification. In particular, based on the analysis performed in **S1-3**, **S4** suggests the following modifications: (i) the item definition is extended as follows: "MAS is an ADAS that rely on sensors to collect information about the driver: 1- head pose and motion that can be used to predict driver's maneuvers since head motion may precede a maneuver; 2- hands and foot location and motions that can be used to predict some driver's maneuvers". This will reduce the probability of misinterpreting the driver's behavior/actions, which contributes to correctly categorizing intended and unintended driver's maneuver. (ii) "the software decision should be sufficiently guaranteed to be correct". This will assure that the software will fulfill the safety requirements related to the intended functionality of the item. Moreover, we update the two main FRs of MAS to cover the result of the intended functionality of the item, as follows:

[3] The tool is available at https://bit.ly/31SyF9H.

[4] https://www.eclipse.org/papyrus/.

FR_01: categorizing unintended drivers' tactical and operational maneuvers as unintended ones, and preventing such maneuvers when the vehicle is moving faster than 50 km/h;

FR_02: categorizing intended drivers' tactical and operational maneuvers as intended ones, and allowing such maneuvers when the vehicle is moving faster than 50 km/h.

Finally, relying on the analysis performed in **S1, S2, S3 & S4** activities, **S1** updates the definition of the item and its related functional requirements (Activity 1.1). Moreover, its outcome is used to enrich the HARA analysis (Activity 1.2) by covering the Hazards identified in **S2 & S3**.

Fig. 6 shows a partial model of the MAS system using our UML profile.[5] In this, we can identify the updated definition of the item that is represented as an `Actor` along with its main two FRs (`FR_01` and `FR_02`) it aims to achieve.

In the **1.2 HARA modeling** activity, we have identified two Hazards[6] `H_01` and `H_02` that *endanger* `FR_01` and `FR_02` respectively.

H_01: categorizing an unintended driver's maneuver due to driver's error (e.g., a slip, a lapse or a mistake) as an intended one when the vehicle is moving faster than 50 km/h, which allows an unintended maneuver to be performed.

H_02: categorizing an intended driver's maneuver as an unintended one when the vehicle is moving faster than 50 km/h, which prevents an intended maneuver to be performed.

Next, we perform a risk assessment for the hazard to assign the appropriate ASIL level. The occurrence of `H_01` is of the highest severity level (`S3`) because allowing an unintended maneuver to be performed may lead to life-threatening injuries or even death. The exposure level is of a medium probability (`E3`) since MAS may not identify some unintended maneuver due to messing or wrong sensor information. Moreover, the highest controllability level `C3` is chosen as the driver might not be aware of such maneuver to perform any corrective action to avoid potential harm/ damage. Based on the severity (`S3`), exposure (`E3`) and controllability (`C3`) of `H_01`, ASIL C is determined for this hazard.

Similarly, the occurrence of `H_02` prevents a driver from performing an intended maneuver, which may lead to life-threatening injuries or even death. Therefore, the highest severity level (`S3`) is chosen. The exposure level `E3` is chosen because several reasons could result in categorizing an intended maneuver as an unintended one (e.g., wrong informational about the head pose and motion, hands/foot location, etc.). Finally, the highest controllability level `C3` is chosen as the driver will not have the required time to perform any corrective action to avoid potential harm. Hence, ASIL C is determined for this hazard.

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`. Following our approach, at least one Safety Goal (`SG`) should be assigned to each hazard rated as ASIL A, B, C or D. Therefore, we assign two SGs (`SG_01` and `SG_02`) to hazard `H_01`, and two SGs (`SG_03` and `SG_04`) to hazard `H_02`:

SG_01: a driver's unintended maneuver should be sufficiently guaranteed to be categorized as an unintended one, when the vehicle is moving faster than 50 km/h.

SG_02: a driver's unintended maneuver shall be prevented, when the vehicle is moving faster than 50 km/h.

SG_03: a driver's intended maneuver should be sufficiently guaranteed to be categorized as an intended one, when the vehicle is moving faster than 50 km/h.

SG_04: a driver's intended maneuver shall be allowed, when the vehicle is moving faster than 50 km/h.

During the **1.3. FSR modeling** activity, at least one FSR should be derived from each SG that has been identified in the previous activity. Accordingly, we derived at least one FSR from each SGs. In particular, we derive the following `FSRs` from `SG_01`:

FSR_01.1: MAS shall be activated when the vehicle is moving faster than 50 km/h.

FSR_01.2: MAS shall be able to collect sufficient information that allows determining whether there is a need for a maneuver.

FSR_01.3: MAS shall be able to collect sufficient information that allows determining whether the driver has a desire or a motivation to make a maneuver.

FSR_01.4: MAS shall be able to identify driver's unintended maneuvers within an appropriate time.

FSR_01.5: MAS shall sufficiently guarantee that driver's unintended maneuvers are categorized as an unintended one.

We derive the following `FSR` from `SG_02`[7]:

FSR_02.1: MAS shall prevent unintended maneuvers, when the vehicle is moving faster than 50 km/h.

From `SG_03`, we derive the following `FSRs`:

FSR_03.1: MAS shall be able to identify driver's intended maneuvers within an appropriate time.

FSR_03.2: MAS shall sufficiently guarantee that driver's intended maneuvers are categorized as an intended one.

From `SG_04`, we derive the following `FSR`:

FSR_04.1: MAS shall not prevent intended maneuvers.

In the **1.4. TSR modeling** activity, we define at least one TSR from each FSR. Due to space limitations, the defined list of TSR is not complete. We define `TSR_01.1.1`, `TSR_01.2.1` and `TSR_01.3.1` based on `FSR_01.1`, `FSR_01.2` and `FSR_01.3`, respectively:

TSR_01.1.1: MAS shall depend on reliable sensor(s) to identify the speed of the vehicle and activate/deactivate MAS when the vehicle is moving faster/slower than 50 km/h.

[5] The complete model is included within the Profile, which is available at https://bit.ly/31SyF9H.

[6] The identified hazards are not complete nor exclusive due to space limitation, and we focused only on the Hazards identified in Activity **S2**.

[7] `FSR_01.1`, `FSR_01.2` and `FSR_01.3` can also be derived from `SG_02`, `SG_03` or `SG_04`, but since ISO 26262 requires to keep the list of `FSRs` atomic, they are not derived.

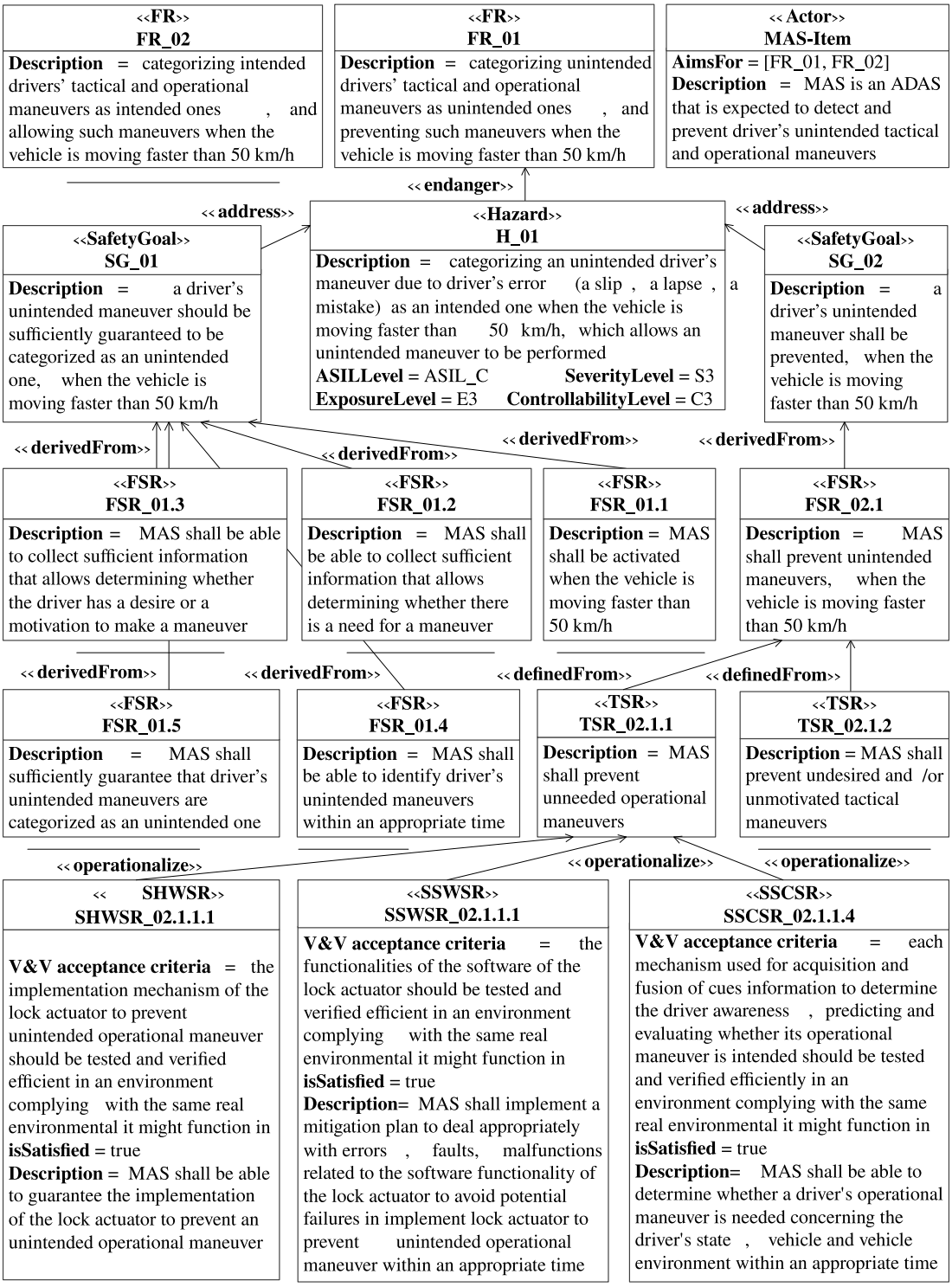

Fig. 6. A partial diagram of applying the UML Profile for modeling the FSR of MAS.

TSR_01.2.1: MAS shall depend on reliable means (e.g., sensors, LIDAR, Radar) to provide sufficient information that allows to predict whether there is a need for an operational maneuver.

TSR_01.3.1: MAS shall depend on reliable means (e.g., head pose and motion, hands and foot location and motions) to provide sufficient information that allows to predict whether there is a desire and/or a motivation for a tactical maneuver.

From `FSR_01.4.`, we define `TSR_01.4.1` and `TSR_01.4.2`, and based on `FSR_01.5.`, we define `TSR_01.5.1`:

TSR_01.4.1: MAS shall depend on reliable technique(s) that allows identifying unneeded operational maneuvers within an appropriate time.

TSR_01.4.2: MAS shall depend on reliable technique(s) that allows identifying undesired and/or unmotivated tactical maneuvers within an appropriate time.

TSR_01.5.1: MAS shall guarantee that the percentage of incorrectly identified unintended maneuvers shall not exceed a predefined safety threshold.

From FSR_02.1., we define both of TSR_02.1.1 and TSR_02.1.2:

TSR_02.1.1: MAS shall prevent unneeded operational maneuvers.

TSR_02.1.2: MAS shall prevent undesired and/or unmotivated tactical maneuvers.

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 the various hardware, software or social specification. In what follows, we define the hardware, software or social specifications concerning only the TSR_02.1.1 in activities **1.5**, **1.6** and **1.7** respectively.

In the **1.5. HWSRs** activity, we define specifications that can be used for the operationalization of each identified TSR that can be allocated to hardware. Based on TSR_02.1.1, we have defined the following SHWSRs:

SHWSR_02.1.1.1: MAS shall be able to guarantee the implementation of the lock actuator to prevent an unintended operational maneuver

SHWSR_02.1.1.2: the lock actuator shall not allow any unintended signal on their outputs.

SHWSR_02.1.1.3: the lock actuator shall be able to deal with any disturbances/noise on their inputs.

SHWSR_02.1.1.4: any unusual behavior of the lock actuator that may result due to error, fault or failure shall be identified by diagnosing its inputs/outputs signals

In the **1.6. SWSRs** activity, we define specifications that can be used for the operationalization of each identified TSR that can be allocated to software. Based on TSR_02.1.1, we have defined the following SWSRs:

SSWSR_02.1.1.1: MAS shall implement a mitigation plan to deal appropriately with errors, faults, malfunctions related to the software functionality of the lock actuator to avoid potential failures in implement lock actuator to prevent unintended operational maneuver within an appropriate time.

SSWSR_02.1.1.2: MAS shall be able to detect if any of the lock actuator functionalities is not responding in an appropriate time.

SSWSR_02.1.1.3: MAS shall assign a special code for each error, faults, malfunctions, etc., related to the lock actuator functionalities, which enables to easily identify them and differentiate them from one another.

In the **1.7. SCSRs** activity, we define specifications that can be used for the operationalization of each identified TSR that can be allocated to social aspects. Based on TSR2.1.1, we have defined the following SCSRs:

SSCSR_02.1.1.1: MAS shall be able to collect all possible information concerning the driver state at any point in time.

SSCSR_02.1.1.2: MAS shall be able to evaluate the correctness of the collect information concerning the driver state.

SSCSR_02.1.1.3: MAS shall be able to fuse all available information to determine the driver's awareness state (e.g., attention, inattention) within an appropriate time.

SSCSR_02.1.1.4: MAS shall be able to determine whether a driver's operational maneuver is needed concerning the driver's state, vehicle and vehicle environment within an appropriate time.

In the final modeling activity (**1.8.**), we define acceptance criteria for the V&V of the identified SHWSRs, SSWSRs and SSCSRs, which helps in assuring that the safety goals can be achieved at the vehicle level, and the FSRs are appropriate for the functional safety of the item. In what follows, we define the V&V acceptance criteria for SHWSR_02.1.1.1, SSWSR_02.1.1.1 and SSCSR_02.1.1.1:

SHWSR_02.1.1.1: V&V acceptance criteria: the implementation mechanism of the lock actuator to prevent unintended operational maneuver should be tested and verified efficient in an environment complying with the same real environmental it might function in.

SSWSR_02.1.1.1: V&V acceptance criteria: the functionalities of the software of the lock actuator should be tested and verified efficient in an environment complying with the same real environmental it might function in.

SSCSR_02.1.1.4: V&V acceptance criteria: each mechanism used for acquisition and fusion of cues information to determine the driver awareness, predicting and evaluating whether its operational maneuver is intended should be tested and verified efficiently in an environment complying with the same real environmental it might function in.

After modeling the MAS in terms of its main CPSs, activities, etc., we proceed to the analysis phase. In which, we depend on the automated analysis to verify the correctness and consistency of the model by detecting and addressing any violation to the properties of the design. In particular, this phase aims at assuring that all the properties of the stereotypes that are used for modeling the main constructs of MAS as well as the relationships among them have been defined correctly. Additionally, this phase verifies the existence of mandatory dependencies/relationships among some constructs. Finally, it verifies that all the V&V acceptance criteria for all defined SHWSR, SWSR and SCSR have been met. The model is verified correct and consistent with respect to our properties of the design, after solving all detected violations by correcting the model accordingly.

5.2. Evaluating the usability and utility of the approach with potential end-users

Following Venable et al. (2012), an *ex post naturalistic evaluation* method has been chosen since the approach has been evaluated after its implementation (*ex post*) with potential end-users. In particular, we conducted an initial experiment to explore how well the approach can support its potential end-users. Note that the evaluation was conducted considering a simplified form of the process that does not include the SOTIF analysis part as such analysis requires experts not users with just basic knowledge.

A. Method. 16 masters students were encouraged to attend the experiment. Most of the participants have basic knowledge of modeling and analysis techniques (15 out 16 have less than 1 year experience), have less than one year experience with UML, and they have basic knowledge of the application domain. The evaluation was structured into two parts:

Briefing, in which the students were invited to a seminar (lasts around 30 min) about the approach and its main components. In particular, we start by explaining the purpose of the experiment, then, the UML profile has been presented, and they were provided with a printed copy of the modeling language, analysis manual, a realistic example concerning the functional safety concept of an advanced automotive system (e.g., MAS), and the methodology to be followed during the modeling and analysis of the MAS example. Finally, we explained the tasks they are supposed to perform.

Evaluation, in which each student was asked to use the UML profile to model and analyze the MAS example.[8] The MAS example has been designed carefully to cover all main aspects of an advanced automotive system. More specifically, the participants were asked to depend on the methodology, and use the modeling language to model the MAS example, and then use the analysis support to analyze the model until it is verified correct. Moreover, participants were asked to complete a paper-based short survey[9] after finalizing their models. The main purpose of this survey was to evaluate how the approach can support end-users for modeling and analyzing FSR.

B. Result. First, we present and discuss the result of the survey. Then, we discuss the four aspects we considered to evaluate how well the approach can support its potential end-users:

Result of the survey: the survey contained only 6 questions (shown in Table 5), which were dedicated to collect feedback concerning the usability aspect of the approach. In particular, *Q1* aims at evaluating the difficulties in importing and setting up the Profile, and *Q2* evaluates the usability of the Methodology. *Q3* and *Q4* are used to evaluate the difficulties in understanding and using the profile for modeling. Finally, *Q5* and *Q6* are used to evaluate the difficulties in understanding and using the automated analysis. Each of these questions is graded on 5 points Likert scale ranging from 1 (Strongly Disagree) to 5 (Strongly Agree).

The result of the survey is presented in Table 5, where the result of *Q1* shows that 82% of the participants strongly agree that they were able to import and run the profile without any help, and the result of *Q2* shows that 41%/47% of the participants strongly agree/agree that they were able to understand and follow the methodology. Concerning the usability of the Profile for modeling, the result of *Q3* demonstrates that 30%/64% of the participants strongly agree/agree that the Profile is easy to be understood and used. The result of *Q4* shows that 30%/64% of the participants strongly agree/agree that they were able to model the example scenario without any help. Concerning the usability of the analysis support, the result of *Q5* demonstrates that 24%/59% of the participants strongly agree/agree that they were able to understand and address the violations to the properties of the design easily. Finally, the result of *Q6* shows that 41%/47% of the participants strongly agree/agree that they were able to use the automated analysis without any help.

Reviewing the results of the survey, it is easy to note that the great majority of the participants did not face any problem concerning the usability of the methodology as well as the modeling and analysis components of the Profile.

Result of the evaluation: we mainly focused on four aspects to evaluate how well the approach can support its potential end-users:

- *Ability to use the methodology,* evaluates how well the participant was able to follow the methodology.
- *Ability to use the modeling component,* evaluates how well the participant was able to use the modeling component.
- *Ability to use the analysis support,* evaluates how well the participant was able to use the automated analysis support.
- *Modeling and analysis time,* measures the time a participant spent to produce the FSR model, which evaluates how well the participant was able to efficiently use the modeling concepts, as well as his efficiency in using the automated analysis support.

The result is shown in Table 6, and they have been collected and determined by the examiner. A qualitative measure was adopted to evaluate participant's ability to use the *methodology*, *modeling component*, and *analysis support*: **H**igh means that the participant was capable of using the artifact without any support, **M**edium means that the participant needed a minimum to medium support, and **L**ow means that the participant was not capable of using the artifact without support.

Concerning the *ability to use the methodology*, 88% (15/17) of the participants were able to follow the methodology without any support, and 12% (2/17) of the participants were able to follow the methodology with a minimum to medium support. The results were the same concerning the *ability to use the modeling component,* where 88% (15/17) of the participants were able to perform the modeling activities without any support, and 12% (2/17) were able to perform the modeling activities with a minimum to medium support. The results concerning the *ability to use the analysis support* were lower, where 41% (7/17) of the participants were able to perform the analysis activities without any support, 47% (8/17) were able to perform the analysis activities with a minimum to medium support, and 12% (2/17) were not able to perform the analysis activities without the examiner support.

Modeling and analysis time was measured in minutes and also evaluated on a qualitative measure: **E**fficient means that the participant has finished all its assigned tasks in less than 90 min, which means that the participant has no or only very few modeling errors and/or violations to the properties of the design that can be tackled easily. **A**dequate means that the participant has finished all its assigned tasks in a time range from 90 to 120 min. This means that the participant has several modeling errors and/or violations to the properties of the design, which needed more time to be tackled. **P**oor means that the participant has finished all its assigned tasks in time over 120 and the maximum allowed time is 140 min. This means that the participant has some serious modeling errors and/or a considerable number of violations to the properties of the design.

The average *modeling and analysis time* of the participants was 101 min, the best performer required 78 min, and the worst took 140 min. Moreover, 29% (5/17) of the participants were considered **E**fficient, 53% (9/17) of the participants were considered **A**dequate, and 18% (3/17) of the participants were considered **P**oor.

The results of the evaluation performed by the examiner do not vary much from the results of the survey. More specifically, the great majority of the participants (all but 2) were able to model and analyze the provided example with no or minimum to medium support and within acceptable time (e.g., **E**fficient or **A**dequate time). Generally speaking, the result of the evaluation is encouraging since most participants have basic knowledge of modeling and analysis. Despite this, they were able to perform well. It is worth mentioning that we are, currently, applying the approach in an industrial context, trying to get more accurate feedback concerning its applicability.

[8] The example along with all documentation used in the experiment can be found at http://tiny.cc/zwfhrz.

[9] The survey template can be found at http://tiny.cc/u88grz.

Table 5
The result of the participants survey.

		S. Disagree	Disagree	Uncertain	Agree	S. Agree
Q1.	I was able to import and run the profile without any help	0%	0%	6%	12%	82%
Q2.	The Methodology is easy to be understood and followed	0%	0%	6%	53%	41%
Q3.	The Profile is easy to be understood and used	0%	0%	6%	64%	30%
Q4.	I was able to model the example scenario without any help	0%	6%	12%	41%	41%
Q5.	The violations to the properties of the design are easy to be understood and addressed	0%	0%	17%	59%	24%
Q6.	I was able to use the automated analysis without any help	0%	0%	12%	47%	41%

Table 6
The result of the participants experiment.

	L	M	H
Methodology ability	0%	12%	88%
Modeling ability	0%	12%	88%
Analysis ability	12%	47%	41%
	P	A	E
Modeling and analysis time	18%	53%	29%

6. Threats to validity

Following Wohlin et al. (2012), we classify threats to validity under the following types:

1- Internal validity concerns the factors that have not been considered in the study, and they could have influenced the investigated factors (Trochim and Donnelly, 2006; Runeson and Höst, 2009). We have identified the following internal threats: *(i) Other factors might influence the system:* our analysis has focused on the three main aspects (e.g., Cyber, Physical and Social) 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. *(ii) Testing threat in the end-users experiment,* occurs when performing a pre-experiment may affect the participants' performance. That is why we did not perform any pre-test. *(iii) Experimenter bias in the end-users experiment,* to reduce the probability of such threat, the experimenter role during the evaluation was limited only to assist the participants when they ask, and such assistance was considered as a part of the evaluation.

2- 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. The evaluation was mainly done with students, not with safety engineers or automotive domain experts, which threatens the validity of the research. However, involving safety engineers or automotive domain experts as well as applying our approach to other automotive systems is on our list for future work. *(iii) The demonstration evaluation does not capture/represent the corresponding real-world problem.* To mitigate this threat, the simulation model/illustrative example has been carefully designed based on several papers that describe the ADAS (e.g., Hayashi et al. (2012), Tawari et al. (2014), Tran and Trivedi (2009)), driver behavior, and intent inference (e.g., Doshi and Trivedi (2011), Schubert et al. (2010), Bevly et al. (2016), Kuge et al. (2010), Doshi and Trivedi (2009), Hegeman et al. (2020)), driver distraction, errors and mistakes (e.g., Lee et al. (2008), Devlin et al. (2011), Rasmussen (1982), Reason (1991), Shappell and Wiegmann (1997)) to get a better understanding of main components of the system and especially drivers, their capabilities and intents while building the model.

3- Conclusion validity concerns the extent to which the conclusions we obtained are reasonable. We have identified two threats, *(i) Fishing for a specific result:* the process we followed starting from item definition until safety validation is based on well-adopted standards (ISO 26262 and ISO/PAS 21448), which reduces the possibility of this threat. Moreover, the importance of considering driver behavior has been reported by many other researchers/experts in the automotive domain. *(ii) Reliability of measures in the potential end-users experiment,* to mitigate this threat, we used clear well-defined and adopted measures to evaluate the participants' performance (e.g., time, qualitative measures).

7. Related work

Several approaches, methods and processes for dealing with functional safety requirements for automotive systems have been proposed in the literature. For instance, one of the early works concerning safety requirements for vehicle-based systems is the work of Jesty et al. (2000), in which they propose guidelines for the safety analysis, including, hazard identification and analysis, identification of safety integrity levels, etc. They also use Failure Mode and Effect Analysis (FMEA) (Stamatis, 2003), Fault Tree Analysis (FTA), and HAZard and OPerability study (HAZOP) (BIS, 2001) for analyzing the system. Giese et al. (2004) propose an approach to support the compositional hazard analysis of UML models. Their 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.

Papadopoulos and Grante (2005) present a process that considers safety concerns along with the cost. In particular, their process combines techniques for safety and reliability analysis of system models and optimization techniques to maximize profit within pragmatic development cost constraints. Zhang et al. (2010) introduce a comprehensive hazard analysis method based on functional models. They argued that their process overcomes the narrower scope of available individual hazard analysis techniques while obtaining the benefits of all of them. Li and Zhang (2011) present a software hazard analysis method for automotive control systems that use the traditional software development

process as a basis and extend it to incorporate safety procedures as a fundamental part of the whole software development.

Basir et al. (2010) propose an approach for system safety requirements in the automotive domain, which adopts the Goal Structuring Notation (GSN) (Kelly and Weaver, 2004) to construct safety cases. In their approach, the defined safety cases reflect the results of the system analysis and provide a high-level argument that traces the requirements on the model via the inferred model structure to the code. Palin et al. (2011) provide guidelines for researchers to create safety cases compliant with the ISO 26262 standard. They propose extensions to GSN, patterns, and several reusable safety arguments covering all parts of ISO 26262 for creating safety cases. Moreover, a method to define functional safety requirements depending on GSN notation has been presented in Beckers et al. (2014). Beckers et al. (2017) present a model-based method for hazard analysis and risk assessment for automotive systems in the context of ISO26262, which offers a UML profile and several constraints expressed in OCL to validate the system model. Sinha (2011) presents a system architecture for a brake-by-wire for vehicles system, where safety and reliability analysis is performed as per the ISO 26262 standard for the functional safety of E/E systems.

The work of Habli et al. (2010) focuses on addressing one component of the overall safety case, namely the assurance of the functional safety concept. They examine how model-driven development and assessment can provide a basis for the systematic generation of functional safety requirements. Baumgart (2012) proposes a method to deal with the reuse of components and functional safety compliance at the same time. Their proposed method considers the entire safety lifecycle of functional safety with special emphasis on hazard analysis and risk assessment. Mehrpouyan et al. (2012) introduce a model-based hazard analysis methodology for early identification of potential safety issues caused by unexpected environmental factors and subsystem interactions. This methodology maps hazard and vulnerability models to specific components in the system and analyzes the hazard propagation paths for risk control and protection strategies.

Born et al. (2010) report on their experience with the application of ISO 26262 standard in a pilot project at a German car manufacturer, and recommend a transition from a document-centric approach to a model-based approach. Gosavi et al. (2018) provide an overview of several methods and techniques that have been developed to ensure the functional safety of smart E/E components, which can be incorporated into autonomous vehicles. The authors also concluded based on this review that the current ISO 26262 standard is not adequate for autonomous vehicles as they are more safety-critical and complex. In Conrad et al. (2010), the authors report on their experiences with one of the first ISO 26262 tool qualifications of commercially available production code generation and verification tools.

On the other hand, Walker (2019) investigated how the consideration of human factors (e.g., human use errors) can be enhanced in the SOTIF specification. The author tries to identify the potential use errors relying on Perception, Cognition, Action (PCA) task Analysis, which allows to answer whether the human is unable to perceive a task, unable to interpret/process it, and unable to act. Kirovskii and Gorelov (2019) present a Generalized Safety Life-cycle approach that weaves the SOTIF analysis into the analysis of Functional Safety Requirements performed relying on ISO 26262. Feth et al. (2018) proposed a safety engineering approach specialized for highly automated driving, which incorporates components that have FSRs that cannot be addressed relying on established safety methods and standardization. Their approach adds a layer on top of the safety aspects tackled by ISO PAS 21448 and ISO 26262 to address the untackled requirements.

Although most of the previously mentioned approaches, methods and processes propose solutions to improve the functional safety of automotive systems, none of them proposes to consider both technical and social aspects of the item to be developed, i.e., they do not consider the driver and his/her behavior as an integral part of the item.

8. Conclusions and future work

In the automotive domain, as well as in other main domains of our life, humans' behavior and especially their errors may influence the intended behavior of many systems they interact with. Understanding such behavior and errors and tackling them will have great economic and societal benefits. In this paper, we presented a model-based approach for modeling and analyzing the FSR for automotive systems in their social and organizational context, i.e., it considers both the technical and social aspects of such systems, which gives the driver a voice by considering him and his behavior while dealing with FSR.

This approach is based on both the ISO 26262 and ISO/PAS 21448 standards, and it proposes a detailed engineering methodology to assist designers while modeling and analyzing FSR. Moreover, it proposes a UML profile for modeling the FSR of the automotive system, and it also proposes various constraints expressed in OCL for the verification of the FSR model. Additionally, it offers a tool to facilitate the modeling and analysis of FSR. We demonstrated the applicability and usefulness of the approach relying on a realistic example from the automotive domain, and we also evaluated the usability and utility of the approach with potential end-users.

For future work, we will provide an SysML profile based on our UML profile, and we are planning to enrich the meta-model of our modeling language by integrating several social concepts from GORE modeling languages. For instance, we aim at considering the dependency concept, which allows for capturing the dependency relations among the different actors (items) of the system or even the relations among sub-components of an item. We also plan to adopt the and/or-decomposition concepts to refine the analysis of SGs, FSRs and TSRs by considering various alternatives for their achievements.

We will integrate the social trust concept in the modeling language that enables for analyzing the expectations of actors in one another concerning their dependencies. The OCL constraints proposed by the approach will be also extended to cover the modeling language extension and to perform more expressive analysis. As mentioned earlier, we are, currently, applying the approach in an industrial context, trying to get more accurate feedback concerning its applicability. Moreover, we intend to contact safety engineers, automotive domain experts, and peer researchers to collect their feedback concerning the approach, and we aim to better validate our approach by applying it to various case studies.

CRediT authorship contribution statement

Mohamad Gharib: Conceptualization, Methodology, Experiments, Writing – original draft. **Andrea Ceccarelli:** Writing – review & editing. **Paolo Lollini:** Provided expertise and feedback. **Andrea Bondavalli:** Provided expertise and feedback.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No. 823788 - ADVANCE, and by the European Social Fund via IT Academy programme.

References

Ali, S., Iqbal, M.Z., Arcuri, A., Briand, L., 2011. A search-based OCL constraint solver for model-based test data generation. In: Proceedings - International Conference on Quality Software. pp. 41–50.

Anton, A.I., Potts, C., 1998. The use of goals to surface requirements for evolving systems. In: Proceedings of the 20th International Conference on Software Engineering. IEEE, pp. 157–166.

Basir, N., Denney, E., Fischer, B., 2010. 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, pp. 68–81.

Baumgart, S., 2012. 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.

Beckers, K., Cote, I., Frese, T., Hatebur, D., Heisel, M., 2014. 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, pp. 65–80.

Beckers, K., Holling, D., Côté, I., Hatebur, D., 2017. A structured hazard analysis and risk assessment method for automotive systems - A descriptive study. In: Reliability Engineering & System Safety, vol. 158. IEEE, pp. 185–195.

Bell, D., De Cesare, S., Iacovelli, N., Lycett, M., Merico, A., 2007. A framework for deriving semantic web services. Inf. Syst. Front. 9 (1), 69–84.

Bevly, D., Cao, X., Gordon, M., Ozbilgin, G., Kari, D., Nelson, B., Woodruff, J., Barth, M., Murray, C., Kurt, A., Redmill, K., Ozguner, U., 2016. Lane change and merge maneuvers for connected and automated vehicles: A survey. IEEE Trans. Intell. Veh. 1 (1), 105–120.

BIS, 2001. BS IEC 61882: Hazard and Operability (HAZOP) Studies. Tech. Rep.

Born, M., Favaro, J., Kath, O., 2010. Application of ISO DIS 26262 in practice. In: Proceedings of the 1st Workshop on Critical Automotive Applications Robustness & Safety - CARS '10. ACM, p. 3.

Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J., 2004. Tropos: An agent-oriented software development methodology. Auton. Agents Multi-Agent Syst. 8 (3), 203–236.

Conrad, M., Munier, P., Rauch, F., 2010. Qualifying software tools according to ISO 26262: Tool certification/qualification approaches in standards and guidelines. In: Mbees. Citeseer, pp. 117–128.

Dardar, R., Gallina, B., Johnsen, A., Lundqvist, K., Nyberg, M., 2012. Industrial experiences of building a safety case in compliance with ISO 26262. In: Proceedings - 23rd IEEE International Symposium on Software Reliability Engineering Workshops. ISSREW 2012. IEEE, pp. 349–354.

Dardenne, A., Lamsweerde, A.van., Fickas, S., 1993. Goal-directed requirements acquisition. Sci. Comput. Program. 20 (1–2), 3–50.

de França, B.B.N., Travassos, G.H., 2016. Simulation based studies in software engineering: A matter of validity. CLEI Electron. J. 18 (1), 4:1–4:18.

Devlin, A., Candappa, N., Corben, B., Logan, D., 2011. Designing safer roads to accommodate driver error. In: Psychopharmacology, vol. 10. pp. 193–212.

Doshi, A., Trivedi, M., 2009. Investigating the relationships between gaze patterns, dynamic vehicle surround analysis, and driver intentions. In: Proceedings of Intelligent Vehicles Symposium. IEEE, pp. 887–892.

Doshi, A., Trivedi, M.M., 2011. Tactical driver behavior prediction and intent inference: A review. In: Proceedings of Conference on Intelligent Transportation Systems, ITSC. IEEE, pp. 1892–1897.

Feth, P., Adler, R., Fukuda, T., Ishigooka, T., Otsuka, S., Schneider, D., Uecker, D., Yoshimura, K., 2018. Multi-aspect safety engineering for highly automated driving: Looking beyond functional safety and established standards and methodologies. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11093 LNCS, pp. 59–72.

Gharib, M., Lollini, P., Ceccarelli, A., Bondavalli, A., 2018. Dealing with functional safety requirements for automotive systems: A cyber–physical-social approach. In: International Conference on Critical Information Infrastructures Security CRITIS. vol. 10707 LNCS, Springer International Publishing, Lucca, pp. 194–206.

Gharib, M., Lollini, P., Ceccarelli, A., Bondavalli, A., 2019. Engineering functional safety requirements for automotive systems: A cyber–physical-social approach. In: Proceedings of IEEE International Symposium on High Assurance Systems Engineering 2019. pp. 74–81.

Giese, H., Tichy, M., Schilling, D., 2004. Compositional hazard analysis of UML component and deployment models. In: Computer Safety, Reliability, and Security, vol. 3219. Springer, pp. 166–179.

Gosavi, M.A., Rhoades, B.B., Conrad, J.M., 2018. Application of functional safety in autonomous vehicles using ISO 26262 standard: A survey. In: Conference Proceedings - IEEE SoutheastCon., vol. 2018-April. pp. 1–6.

Habli, I., Ibarra, I., Rivett, R.S., Kelly, T., 2010. Model-Based Assurance for Justifying Automotive Functional Safety. Tech. Rep. June 2016, SAE Technical Paper.

Hayashi, R., Isogai, J., Raksincharoensak, P., Nagai, M., 2012. Autonomous collision avoidance system by combined control of steering and braking using geometrically optimised vehicular trajectory. Veh. Syst. Dyn. 50 (Suppl. 1), 151–168.

Hegeman, G., Brookhuis, K., Hoogendoorn, S., 2020. Opportunities of advanced driver assistance systems towards overtaking. Eur. J. Transp. Infrastruct. Res. 5 (4), 281–296.

Helmreich, R.L., 2000. On error management: Lessons from aviation. Br. Med. J. 320 (7237), 781–785.

Hevner, March, Park, Ram, 2017. Design science in information systems research. MIS Q. 28 (1), 75.

Hoess, A., Gmbh, C.A., 2009. Highly automated vehicles for intelligent transport. In: ITS World Congress, NY, USA. No. July 2015.

ISO, 2011. 26262: Road Vehicles-Functional Safety. International Standard ISO/FDIS 26262.

ISO - International Organization for Standardization, 2019. PAS 21448-Road Vehicles-Safety of the Intended Functionality. Tech. Rep.

Jesty, P.H., Hobley, K.M., Evans, R., Kendall, I., 2000. Safety analysis of vehicle-based systems. In: Proceedings of the 8th Safety-Critical Systems Symposium. Citeseer, pp. 90–110.

Kelly, T., Weaver, R., 2004. The goal structuring notation - A safety argument notation. In: Elements. Citeseer.

Kirovskii, O.M., Gorelov, V.A., 2019. Driver assistance systems: Analysis, tests and the safety case. ISO 26262 and ISO PAS 21448.

Kuge, N., Yamamura, T., Shimoyama, O., Liu, A., 2010. A Driver Behavior Recognition Method Based on a Driver Model Framework. SAE Technical Paper Series 1.

Kuhlmann, M., Hamann, L., Gogolla, M., 2011. Extensive validation of OCL models by integrating SAT solving into USE. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 6705 LNCS, pp. 290–306.

Lawton, R., Ward, N.J., 2005. A systems analysis of the Ladbroke Grove rail crash. Accid. Anal. Prev. 37 (2), 235–244.

Lee, J.D., Young, K.L., Regan, M.A., 2008. Defining driver distraction. Driv. Distraction: Theory Effects Mitig. 13 (4), 31–40.

Li, W., Zhang, H., 2011. A software hazard analysis method for automotive control system. In: Proceedings of International Conference on Computer Science and Automation Engineering, CSAE 2011, vol. 3. IEEE, pp. 744–748.

McCall, J.C., Trivedi, M.M., 2007. Driver behavior and situation aware brake assistance for intelligent vehicles. Proc. IEEE 95 (2), 374.

McFadden, K.L., Towell, E.R., 1999. Aviation human factors: A framework for the new millennium. J. Air Transp. Manag. 5 (4), 177–184.

Mehrpouyan, H., Bunus, P., Kurtoglu, T., 2012. Model-based hazard analysis of undesirable environmental and components interaction. In: IEEE Aerospace Conference Proceedings. IEEE, pp. 1–8.

Meshkati, N., 1991. Human factors in large-scale technological systems' accidents: Three Mile Island, Bhopal, Chernobyl. Organ. Environ. 5 (2), 133–154.

OMG-OCL, 2014. Object Constraint Language. Tech. Rep. May, URL http://www.omg.org/spec/OCL/2.4/.

Palin, R., Ward, D., Habli, I., Rivett, R., 2011. ISO 26262 safety cases: compliance and assurance. In: 6th IET International Conference on System Safety 2011. No. November 2014. IET, pp. 12–18.

Papadopoulos, Y., Grante, C., 2005. Evolving car designs using model-based automated safety analysis and optimisation techniques. J. Syst. Softw. 76 (1), 77–89.

Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S., 2008. A design science research methodology for information systems research. J. Manage. Inf. Syst. 24 (3), 45–77.

Rasmussen, J., 1982. Human errors. A taxonomy for describing human malfunction in industrial installations. J. Occup. Accid. 4 (2–4), 311–333.

Reason, J., 1991. Human Error, vol. 29. Cambridge University Press.

Regan, M.A., Young, K., Triggs, T.J., Horberry, T., E. Mitsopoulos, N., Tomasevic, I.J., 2005. Intelligent vehicle safety research at the monash university accident research centre. In: Australia Asian Transport Research Forum, 27th, 2004, Adelaide, South Australia, vol. 27. ATRF, pp. 1–11.

Richters, M., Gogolla, M., 2002. OCL: Syntax, Semantics, and Tools. pp. 42–68.

Ridderhof, W., Gross, H.-G., Doerr, H., 2007. Establishing evidence for safety cases in automotive systems–A case study. In: International Conference on Computer Safety, Reliability, and Security. Springer, pp. 1–13.

Runeson, P., Höst, M., 2009. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. 14 (2), 131–164.

Saffarian, M., de Winter, J.C.F., Happee, R., 2012. Automated driving: Human-factors issues and design solutions. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 56. Sage Publications Sage CA, Los Angeles, CA, pp. 2296–2300.
Salmon, P.M., Regan, M.A., Johnston, I., 2005. Human Error and Road Transport: Phase One – A Framework for an Error Tolerant Road Transport System. No. 256.
Sathyanarayana, A., Boyraz, P., Purohit, Z., Lubag, R., Hansen, J.H.L., 2010. Driver adaptive and context aware active safety systems using CAN-bus signals. In: Intelligent Vehicles Symposium. IV. 2010. IEEE, pp. 1236–1241.
Schubert, R., Schulze, K., Wanielik, G., 2010. Situation assessment for automatic lane-change maneuvers. IEEE Trans. Intell. Transp. Syst. 11 (3), 607–616.
Selic, B., 2007. A systematic approach to domain-specific language design using UML. In: 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, ISORC 2007. IEEE, pp. 2–9.
Senders, J., Moray, N., 1991. Human Error: Cause, Prediction, and Reduction. Lawrence Erlbaum Associates, Inc, New Jersey.
Shappell, S.A., Wiegmann, D.A., 1997. A human error approach to accident investigation: The taxonomy of unsafe operations. Int. J. Aviat. Psychol. 7 (4), 269–291.
Sinha, P., 2011. Architectural design and reliability analysis of a fail-operational brake-by-wire system from ISO 26262 perspectives. Reliab. Eng. Syst. Saf. 96 (10), 1349–1359.
Stamatis, D.H., 2003. Failure Mode and Effect Analysis: FMEA from Theory to Execution. ASQ Quality Press.
Stanton, N.A., Harris, D., Salmon, P.M., Demagalski, J., Marshall, A., Waldmann, T., Dekker, S., Young, M.S., 2010. Predicting Design-Induced Error in the Cockpit. Tech. Rep. 1.
Tawari, A., Sivaraman, S., Trivedi, M.M., Shannon, T., Tippelhofer, M., 2014. Looking-in and looking-out vision for urban intelligent assistance: Estimation of driver attentive state and dynamic surround for safe merging and braking. In: Proceedings of Intelligent Vehicles Symposium. IEEE, pp. 115–120.
Tran, C., Trivedi, M.M., 2009. Driver assistance for keeping hands on the wheel and eyes on the road. In: ICVES 2009 - International Conference on Vehicular Electronics and Safety. IEEE, pp. 97–101.
Trochim, W., Donnelly, J.P., 2006. The Research Methods Knowledge Base. Cengage Learning.
Venable, J., Pries-Heje, J., Baskerville, R., 2012. A comprehensive framework for evaluation in design science research. In: International Conference on Design Science Research in Information Systems. Springer, pp. 423–438.
Wagner, S., Schatz, B., Puchner, S., Kock, P., 2010. A case study on safety cases in the automotive domain: Modules, patterns, and models. In: 21st International Symposium on Software Reliability Engineering. ISSRE, 2010. IEEE, pp. 269–278.
Walker, A., 2019. Sotif the human factor. In: Communications in Computer and Information Science, vol. 1060. Springer Verlag, pp. 575–584.
Weick, K.E., 1990. The vulnerable system: An analysis of the tenerife air disaster. J. Manag. 16 (3), 571–593.
Wieringa, R., 2009. Design science as nested problem solving. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, DESRIST '09. pp. 1–12.
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2012. Experimentation in Software Engineering. pringer Science & Business Media.
Yu, E.S.-k., 1995. Modelling Strategic Relationships for Process (Ph.D. thesis). University of Toronto.
Zhang, H., Li, W., Chen, W., 2010. Model-based hazard analysis method on automotive programmable electronic system. In: Proceedings of the 3rd International Conference on Biomedical Engineering and Informatics, BMEI 2010, vol. 7. IEEE, pp. 2658–2661.

Dr. Mohamad Gharib is Lecturer of Software Engineering at the Institute of Computer Science, University of Tartu. He was a postdoctoral researcher at the University of Florence. Previously he was a postdoctoral researcher at the Department of Information Engineering and Computer Science, University of Trento, Italy, where he obtained his Ph.D. degree in April 2015. His current research interests focus mainly on designing Cyber–Physical System of Systems and Socio-Technical Systems.

Dr. Andrea Ceccarelli received the Bachelor, and Master degree in computer science, and the Ph.D. in Informatics and Automation Engineering at the University of Firenze, Italy, respectively in 2006, 2008 and 2012. He is currently a Research Associate at the same department. He published in International conferences and journals and regularly serves in the Program Committee of International Conferences and Workshops.

Dr. Paolo Lollini is an Assistant Professor at the Faculty of Science at the University of Florence. He has been continuously participating in European funded projects since 2002 up to present. He was a member of the program committee of important conferences in the area and he has coauthored papers that appeared in proceedings of international conferences, journals, and books. His current research interests include the modeling and evaluation of performability and resiliency attributes of large-scale critical infrastructures and systems-of-systems.

Prof. Andrea Bondavalli received the M.S. degree in computer science from the University of Pisa, in 1986. He has been a Researcher with the Italian CNR, and is currently a Professor at the University of Florence. He is a Member of the Editorial Board of the IJCCBS journal and the chair of the Steering Committee of the IEEE SRDS. He served as Program Chair and as General Chair of the most important conferences in Dependable Computing and the Conference Coordinator of IEEE DSN 2009.