Journal of Systems Architecture 119 (2021) 102250

Contents lists available at ScienceDirect

Journal of Systems Architecture

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

A case study for risk assessment in AR-equipped socio-technical systems ☆

Soheila Sheikh Bahaei [a,*], Barbara Gallina [a,*], Marko Vidović [b]

[a] School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
[b] Xylon Electronics Company, Zagreb, Croatia

ARTICLE INFO

Keywords:
Socio-technical systems
Augmented reality
Risk assessment
ISO 26262
ISO/PAS 21448-SOTIF

ABSTRACT

Augmented Reality (AR) technologies are used as human–machine interface within various types of safety-critical systems. Several studies have shown that AR improves human performance. However, the introduction of AR might introduce risks due to new types of dependability threats. In order to avoid unreasonable risk, it is required to detect new types of dependability threats (faults, errors, failures). In our previous work, we have designed extensions for the SafeConcert metamodel (a metamodel for modeling socio-technical systems) to capture AR-related dependability threats (focusing on faults and failures). Despite the availability of various modeling techniques, there has been no detailed investigation of providing an integrated framework for risk assessment in AR-equipped socio-technical systems. Hence, in this paper, we provide an integrated framework based on our previously proposed extensions. In addition, in cooperation with our industrial partners, active in the automotive domain, we design and execute a case study. We aim at verifying the modeling and analysis capabilities of our framework and finding out if the proposed extensions are helpful in capturing system risks caused by new AR-related dependability threats. Our conducted qualitative analysis is based on the Concerto-FLA analysis technique, which is included in the CHESS toolset and targets socio-technical systems.

1. Introduction

Several studies have shown that Augmented Reality (AR) technology contributes to human performance [1]. The combination of the AR technology and humans constitute an AR-equipped socio-technical system. We focus on AR-equipped socio-technical systems because of the context of the ImmerSAFE project [2] and also due to the increased AR applications. AR technology superimposes virtual and computer generated information on the reality of the user [3]. The information can be visual, auditory, etc., for enhancing human capabilities [4]. An example of visual augmented reality is using navigational information superimposed on the windshield of a car for driver guidance.

In some cases the inclusion of the AR technology might undermine user reaction. For example, it can increase cognitive-processing load [4] and it would lead to new risks. Thus, exploiting AR in socio-technical systems demands risk assessment to make sure that it is not harmful for people and the environment. To assess risk of socio-technical systems equipped with augmented reality, it is required to identify new uncertainties, threats and their propagation.

According to ISO 26262 [5], the automotive standard for functional safety, risk assessment is a "method to identify and categorize hazardous events of items and to specify safety goals and ASILs (Automotive Safety Integrity Level) related to the prevention or mitigation of the associated hazards in order to avoid unreasonable risk". The focus in this standard is on risks emanated from malfunctions of electrical and/or electronic (E/E) system. In contrast, ISO/PAS 21448:2019, defined as safety of the intended functionality (SOTIF) [6], considers risks emanated from other types of hazardous behavior related to intended functionality or performance limitation of the system. The reason for hazardous behavior in many instances is a triggering event. For example, lack of attention while driving an automated vehicle (triggering event) would lead to incorrect decision (hazardous behavior) causing system risk. For analyzing SOTIF related hazards qualitative analysis is used and ASIL is not determined [6].

In our previous works [7,8], in order to identify AR-related dependability threats, we have proposed two taxonomies. Based on these taxonomies, we extended SafeConcert to investigate additional socio aspects and AR-related dependability threats in system architecture modeling and analysis [9,10]. So far, an integrated framework for risk assessment of AR-equipped socio-technical systems has not been

☆ This work is funded by EU H2020 MSC-ITN grant agreement No 764951. The author B. Gallina is also partially supported by Sweden's Knowledge Foundation via the SACSys (Safe and Secure Adaptive Collaborative Systems) project.

* Corresponding authors.
E-mail addresses: soheila.sheikhbahaei@mdh.se (S. Sheikh Bahaei), barbara.gallina@mdh.se (B. Gallina).

https://doi.org/10.1016/j.sysarc.2021.102250
Received 13 January 2021; Received in revised form 20 July 2021; Accepted 23 July 2021
Available online 31 July 2021

proposed. Current frameworks do not contain modeling and analysis constructs for modeling and analyzing several social aspects, AR-extended human functions and AR-related organizational factors. In addition, there has been little investigation about how effective current modeling and analysis techniques are for industrial systems containing new technologies and if it is possible to capture risk caused by augmented reality-related dependability threats.

In this paper, we build on our previous work and provide an integrated framework for risk assessment of AR-equipped socio-technical systems. In addition, we use an industrial case study for verifying the framework in capturing risks caused by AR-related dependability threats. More specifically, in this paper, we build on our previously proposed conceptual extensions on SafeConcert metamodel [12]. SafeConcert metamodel [12] is part of the modeling language included in the CHESS framework [13] for modeling socio-technical systems. Extended metamodel provides modeling and analysis capabilities, which can be used for assessing risk of AR-equipped socio-technical systems. Concerto-FLA [14] analysis technique is also used in our framework. Concerto-FLA is an analysis technique for socio-technical systems and it uses FPTC (Fault Propagation and Transformation Calculus) [15] syntax. In addition, we provide a case study based on SEooC (Safety element out of context) concept of ISO 26262 standard. SEooC concept refers to elements that are not developed in the context of a particular vehicle. Based on this concept, assumptions should be defined for the context in which a component is going to be used [16]. Finally, we provide threats to validity and limitations and benefits of the extensions. The results of our work can support modeling items and analyzing the behavior of AR-equipped socio-technical systems in compliance with ISO 26262 and SOTIF safety standards, which can be used by stakeholders, including designers and developers.

The rest of the paper is organized as follows. In Section 2, we provide essential background information. In Section 3, we provide an integrated framework for assessing risk of AR-equipped socio-technical systems. In Section 4, we design and conduct the case study to verify modeling and analysis capabilities of the proposed framework and we discuss about lessons learnt based on limitations and benefits of our research. In Section 5, we discuss about threats to validity in relation to our research. In Section 6, we provide a discussion about the contribution of our proposed extensions in determining ISO 26262 controllability and other applications. In Section 7, we extensively discuss related works. Finally, in Section 8, we present some concluding remarks and sketch future work.

2. Background

This section provides essential background information onto which our work is based. First, CHESS framework is introduced. Then, SafeConcert metamodel and AR-related modeling extensions are presented. FPTC syntax and Concerto-FLA analysis technique are also explained. Finally, ISO 26262, SOTIF, SEooC and SAE automation levels are presented.

2.1. CHESS framework

CHESS framework [13] provides a methodology, a language and a toolset for developing high-integrity systems.

The CHESS methodology, which is component-based and model-driven, is based on an incremental and iterative process. Based on this methodology, components are defined incrementally with functional and also extra-functional properties, such as dependability information [17]. Then, developers can use a set of analysis techniques and back propagate the results iteratively.

CHESS-ML (CHESS Modeling Language) [18] is based on UML and provides the modeling elements required for modeling high-integrity systems.

CHESS toolset includes a set of plugins for code generation and provides various analysis capabilities. For example, Concerto-FLA (Failure Logic Analysis) [14] is a plugin related to analysis. In Concerto-FLA, component-based model of the system is provided and dependability information is used for decorating components. Then, analysis results can be back propagated to the system model. In this paper, we use Concerto-FLA as the analysis technique.

2.2. SafeConcert and its extension of AR

SafeConcert [12] is a metamodel for modeling socio and technical entities in socio-technical systems. In this metamodel, which is part of the CHESS-ML modeling language [18], technical (i.e., software, hardware) or socio entities can be modeled as components/composite components in component-based systems representing socio-technical systems. SERA taxonomy [19] is used for modeling human and organization, which are the socio entities of the system. In this metamodel human sub-components are modeled based on twelve categories of human failures including failures in perception, decision, response, etc.

In [9], we extended the human modeling elements based on AREXTax, which is an AR-extended human function taxonomy [7]. This taxonomy is obtained by harmonizing about six state-of-the-art human failure taxonomies (Norman [20], Reason [21], Rasmussen [22], HFACS (Human Factor Analysis and Classification System) [23], SERA (Systematic Error and Risk Analysis) [19], Driving [24]) and then extending the taxonomy based on various studies and experiments on augmented reality. These extended modeling elements are divided to four categories, shown in Fig. 1. Three of these categories are human functions including human process unit, human SA (situational awareness) unit, and human actuator unit. The one other category is human fault unit, which is related to human internal influencing factors affecting on human functions. We explain these modeling elements in the next two paragraphs. In the first paragraph we explain modeling elements related to human functions and in the second paragraph we explain modeling elements related to human fault unit and also other fault categories. Extended modeling elements are shown with white color and AR-stemmed modeling elements are shown with dotted line border.

The extended modeling elements in human process unit, human SA unit, and human actuator unit enable modeling of AR-extended human functions. For example, detection failure, which represents a failure in *detecting* human function, is a human failure introduced by several human failure taxonomies such as Reason [21] and Rasmussen [22] taxonomies. Based on experiments and studies on augmented reality including [25] and [26], *detecting* function would be extended to *surround detecting* while using AR (surrounding information would be augmented on real world view of the user by AR). Thus, *surround detecting* can be considered as an extended sub-component of human component; in other words *surround detecting* is an extended modeling element proposed for analysis of AR-equipped socio-technical systems.

In [10], we extended organization and human modeling elements based on AREFTax, which is a fault taxonomy including AR-caused faults [8]. This taxonomy is obtained by harmonizing about five state-of-the-art fault taxonomies (Rasmussen [22], HFACS [23], SERA [19], Driving [24] and SPAR-H (Standardized Plant Analysis Risk Human Reliability Analysis) [27]) and then extending the taxonomy based on various studies and experiments on augmented reality.

In [11], we proposed more specifications for organization and human modeling elements by considering digitalization, globalization and networked structure of organizations. More specifically, we extended the human and organization modeling elements based on post normal accident theory [28] and global distance metric [29]. The extended modeling elements are helpful to prevent post normal accidents and to include global distance metric while assessing risk in new AR-equipped socio-technical systems. These extended modeling elements are shown in Fig. 2 and human fault unit of Fig. 1. Extended modeling elements

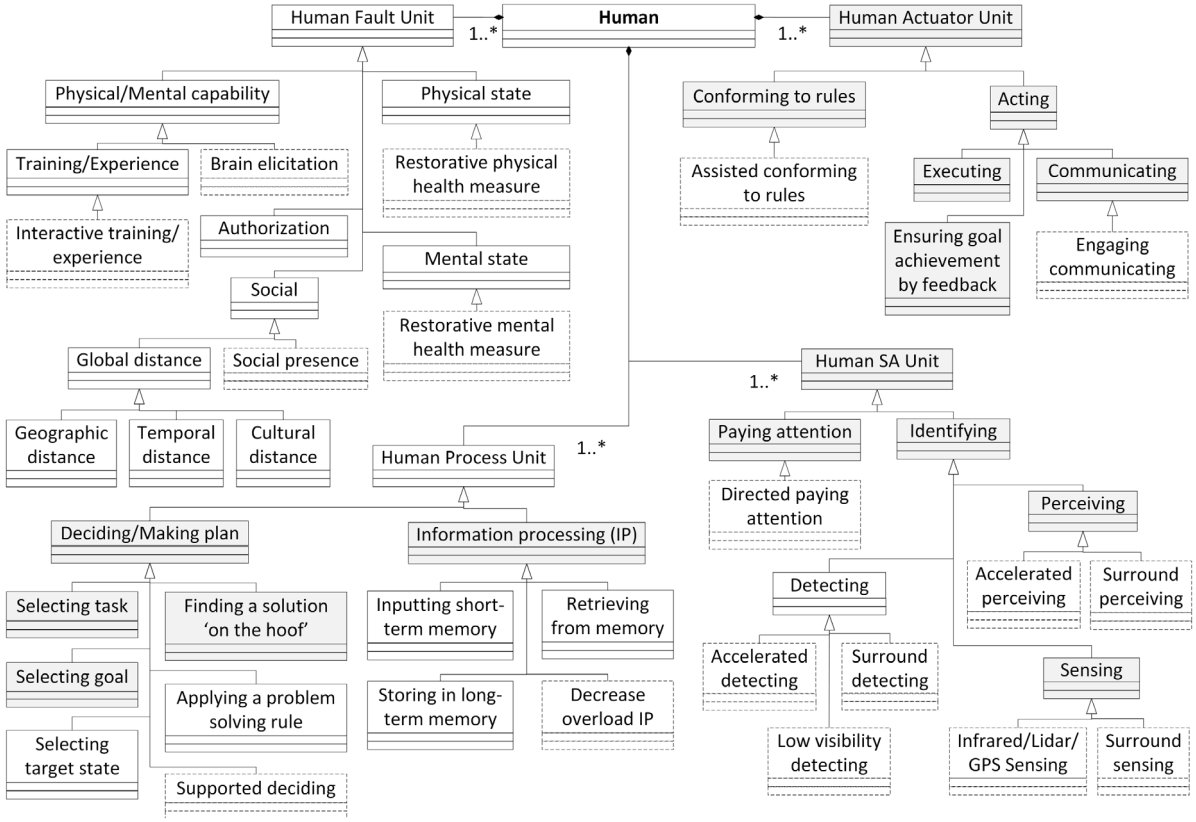


Fig. 1. Extended modeling elements for human components [11].

are shown with white color and AR-stemmed modeling elements are shown with dotted line border. These extended modeling elements enable modeling of various faults leading to human failures including AR-caused faults. Faults would be caused by human, environment, organization, etc. Human related faults are categorized as human fault unit of Fig. 1 and non-human faults are categorized as three categories of organizational factors including organization and regulation unit, environment unit and task unit. For example, failure in *physical state* of a human is a human internal fault leading to human failure. This is shown as human modeling element in human fault unit category shown in Fig. 1. Another example is *condition*, which is a non-human factor and it is categorized as extended modeling elements for organization components shown in Fig. 2. One example of the AR-extended modeling elements is *social presence* shown in Fig. 1. Based on studies on augmented reality [30], using AR would decrease social presence and failure in *social presence* can be considered as fault leading to human failure.

2.3. The FPTC syntax

FPTC syntax was proposed as part of FPTC analysis technique [15]. FPTC rules are set of logical expressions that relate output failure modes to combinations of input failure modes in each individual component [31].

Components' behavior can be classified as source (if component generates a failure), sink (if component is able to detect and correct input failure), propagational (if component propagates failures received in its input to its output) and transformational (if component transforms the type of failure received in its input to another type in its output).

FPTC syntax for modeling failure behavior at component and connector level is as follows:

behavior = expression+
expression = LHS '→' RHS
LHS = portname'.' bL | portname '.' bL (',' portname '.' bL) +
RHS = portname'.' bR | portname '.' bR (',' portname '.' bR) +
failure = 'early' | 'late' | 'commission' | 'omission' | 'valueSubtle' | 'valueCoarse'
bL = 'wildcard' | bR
bR = 'noFailure' | failure

Early and late failures refer to provided function at a wrong time (early or late). Commission failures refer to provided function at a time which is not expected and omission failures refer to not provided function at a time which is expected. Value failures refer to wrong value after computations, which would be valueSubtle (user cannot detect it) or valueCoarse (user can detect it).

Wildcard in an input port shows that the output behavior is the same regardless of the failure mode on this input port. NoFailure in an input port shows normal behavior.

Based on this syntax, "IP1.noFailure → OP1.omission" shows a source behavior and should be read as follows: if the component receives noFailure (normal behavior) on its input port IP1, it generates omission on its output port OP1.

2.4. Concerto-FLA analysis technique

Concerto-FLA [14], which extends FPTC [15], is a model-based analysis technique that provides the possibility for analyzing failure behavior of humans and organizations in addition to technical entities by using SERA [19] classification of socio-failures. As we recalled in Section 2.1, this technique is provided as a plugin within the CHESS toolset and allows users to define component-based architectural models composed of hardware, software, human and organization. This technique includes five main steps.

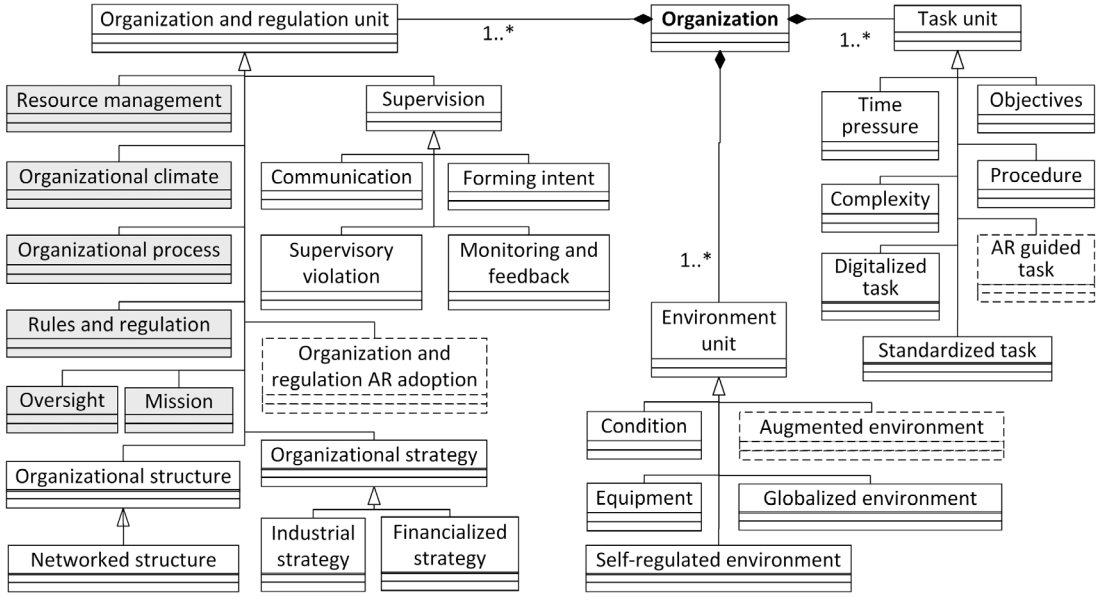


Fig. 2. Extended modeling elements for organization components [11].

1. Modeling architectural elements including software, hardware, human, organization, connectors, interfaces and etc.
2. Modeling failure behavior at component and connector level by using FPTC syntactical rules. Concerto-FLA has adopted the FPTC syntax for modeling failure behavior at component and connector level (explained in Section 2.3).
3. Modeling failure modes at system level by injection of inputs.
4. Performing qualitative analysis through automatic calculation of the failure propagations. This step is similar to FPTC technique that system architecture is considered as a token-passing network and set of possible failures that would be propagated along a connection is called tokenset (default value for each tokenset is noFailure, which means normal behavior). In order to obtain system behavior, maximal tokenset is calculated for each connection through a fixed-point calculation.
5. Interpreting the results at system level. Based on the interpretation it will be decided to do the re-design or not.

2.5. ISO 26262, SOTIF, SEooC and SAE

ISO 26262 [5] is the standard for functional safety. ISO 26262 provides the requirements and set of activities that should be performed during the lifecycle phases such as development, production, operation, service and decommissioning. ISO 26262 addresses functional safety and specifies risk assessment for risks due to malfunctioning behavior of the items. If the risk is because of intended functionality or performance limitation of a system, it is addressed in ISO/PAS 21448-SOTIF [6]. In ISO 26262, ASIL (Automotive Safety Integrity Level) is determined and used for applying the requirements to avoid unreasonable residual risk. ASIL specifies item's necessary safety requirements to achieve an acceptable residual risk. Residual risks are remaining risks after using safety measures. An ASIL value is one of four levels (A–D) and it is determined based on three factors: severity, exposure and controllability. The severity factor indicates class of severity in case of hazard occurrence and it is classified from 0 to 3 (shown by S0–S3). S3 shows the category with the highest severity and it is related to situations with life threatening injuries. The exposure factor indicates class of probability of exposure with respect to operational situations and it is classified from 0 to 4 (shown by E0–E4). E4 shows the category with the highest probability of exposure (with time in use more than 10%). The controllability factor indicates the class of driver controllability and it is classified from 0 to 3 (shown by C0–C3). C3 shows the category with the highest controllability (more than 99% of drivers can control). ASIL classification based on these three factors is shown in Fig. 3. QM (quality management) shows that no safety requirement is necessary. ASIL value A shows the lowest safety requirements and ASIL value D shows the highest safety requirements.

Safety element out of context (SEooC), introduced by ISO 26262, refers to an element that is not defined in the context of a special vehicle, but it can be used to make an item, which implements functions at vehicle level. SEooC is based on ISO 26262 safety process and information regarding system context such as interactions and dependencies on the elements in the environment should be assumed [33].

The SEooC development contains 4 main steps:

1. (a) Definition of the SEooC scope: assumptions related to the scope, functionalities and external interfaces of the SEooC should be defined.
   (b) Definition of the assumptions on safety requirements for the SEooC: assumptions related to item definition, safety goals of the item and functional safety requirements related to SEooC functionality, which are required for defining technical safety requirements of the SEooC should be defined.
2. Development of SEooC: based on the assumed functional safety requirements, technical safety requirements are derived and then SEooC is developed based on ISO 26262 standard.
3. Providing work products: work products are documents that show the fulfilled functional safety requirements and assumptions on the context of SEooC.
4. Integration of the SEooC into the item: safety goals and functional safety requirements defined in item development should match with assumed functional safety requirements for the SEooC. In case of a SEooC assumption mismatch, change management activity based on ISO 26262 standard should be conducted.

The process required for improving the intended functionality to ensure safety includes eight activities. Possible interactions between these activities and ISO 26262 activities and SEooC are shown in Fig. 4.

Safety process of the ISO 26262 standard starts with *concept phase* containing *item definition, hazard analysis and risk assessment* and *functional safety concept* [33]. An *item* implements a vehicle level function.

Severity		S1 Light injuries				S2 Sever injuries, not life threating				S3 Life threating injuries			
Exposure (time in use)		E1< 0.1%	E2< 1%	E3< 10%	E4> 10%	E1< 0.1%	E2< 1%	E3< 10%	E4> 10%	E1< 0.1%	E2< 1%	E3< 10%	E4> 10%
Controllability (likelihood controllable by avg.)	C1 ≥ 99%	QM	QM	QM	QM	QM	QM	QM	A	QM	QM	A	B
	C2 ≥ 90%	QM	QM	QM	A	QM	QM	A	B	QM	A	B	C
	C3 < 90%	QM	QM	A	B	QM	A	B	C	A	B	C	D

Fig. 3. ASIL classification [32].

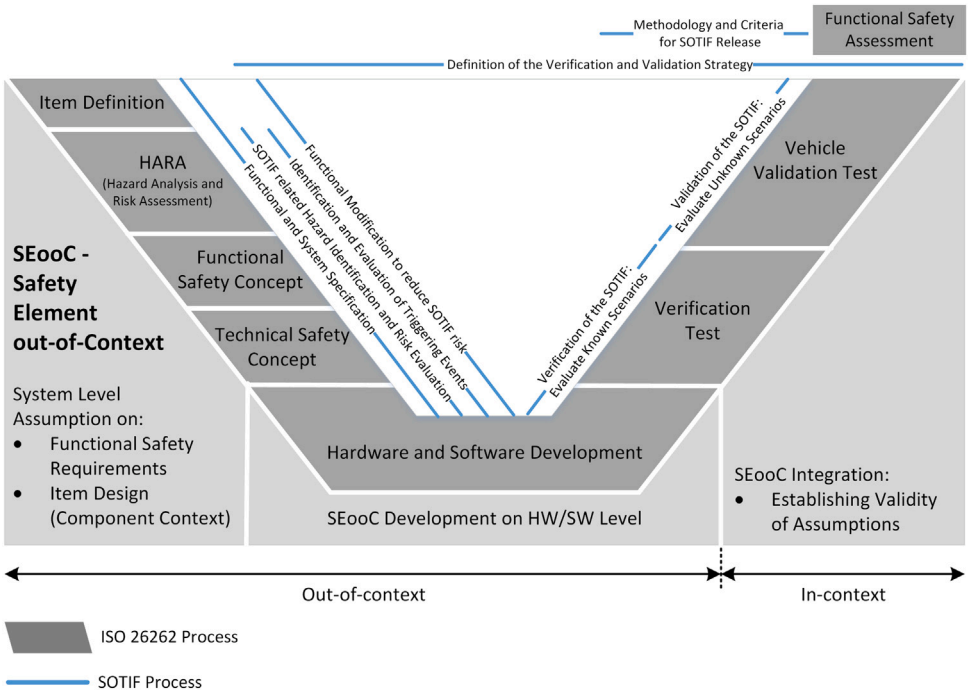


Fig. 4. Alignment of SOTIF activities to ISO 26262 activities and SEooC.
Source: Adapted from [33] and [6].

In *item definition* the main objective is defining items. Defining items requires defining the dependencies and interactions with environment. Then, related hazards are identified and functional safety requirements are obtained. In SEooC, assumptions related to system context are the main output of the *concept phase*. *Functional safety concept* includes providing functional safety requirements. Output provided by *Functional safety concept* is used by *technical safety concept*. *Technical safety concept* includes technical safety requirements and system design. Then, *hardware and software development* is done based on *technical safety concept*. *HW/SW development* is based on assumptions provided in concept phase. Next steps in the process are *verification test*, *validation test* and *functional safety assessment*. In SEooC, these steps require establishing validity of assumptions.

SOTIF process starts with *functional and system specification*, which includes functional description and considerations on system design and architecture. Then, potential hazardous events should be identified. If the harm is possible for the identified potentially hazardous events, then analysis of their triggering events should be conducted. Functional modification is the next activity for avoiding the hazards or for reducing the resulting risk. Next activities are verification and validation strategy specification and then in verification and validation activities arguments are provided to illustrate that the residual risk is below acceptable level by testing on various known and unknown scenarios. Finally, evaluation on residual risk should be performed based on the verification and validation results and specified criteria.

Based on the taxonomy and definitions related to driving automation systems for on-road motor vehicles performing part or the entire dynamic driving task (DDT) on a sustained basis, there are six levels of driving automation. SAE level 0 refers to no driving automation and SAE level 5 refers to full driving automation [34]. These levels with description and example are shown in Fig. 5. Assessing human factor in driver–vehicle interface is not only important on lower SAE levels, but also on higher levels because of the importance of safe transition between automated and non-automated vehicle operation [35]. In order to improve safety, various scenarios of driver/vehicle interaction should be considered.

3. An integrated framework for assessing risk of AR-equipped socio-technical systems

Our provided framework for assessing risk of AR-equipped socio-technical systems is based on our previously proposed modeling extensions and the Concerto-FLA analysis technique [14]. We name this framework FRAAR (Framework for Risk Assessment in AR-equipped socio-technical systems).

	Description	Example
SAE Level 0	The driver controls the vehicle completely at all times and system provides only warning.	Forward collision warning and blind spot monitoring
SAE Level 1	The driver controls the vehicle, but can choose an automation function under limited conditions.	Adaptive cruise control
SAE Level 2	The driver controls the vehicle, but can use combined function automation of at least two control functions under limited conditions.	Adaptive cruise control in combination with lane centering
SAE Level 3	The driver can transfer control of the vehicle to the system under limited conditions, but should be available for occasional transition.	Self-driving car that may signal driver to regain control with proper transition time
SAE Level 4	The system controls the vehicle under limited conditions and it is not required for the driver to be available.	Local driverless taxi
SAE Level 5	It is not required for the driver to be available and system controls the vehicle in all conditions.	Driverless vehicle

Fig. 5. SAE levels of driving automation.

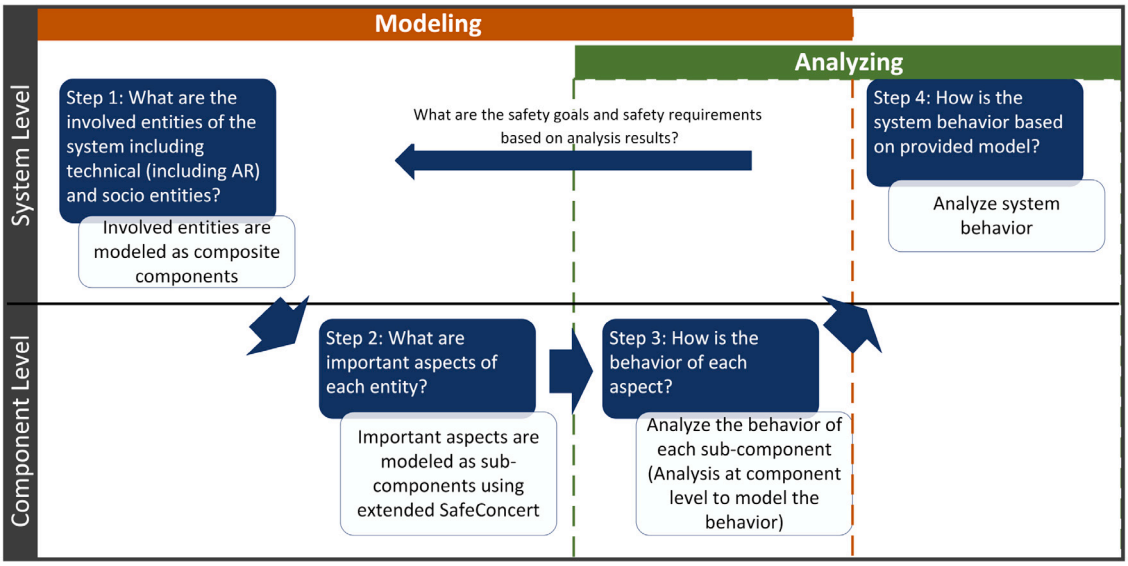


Fig. 6. Methodology of the provided framework for assessing risk of AR-equipped socio-technical systems.

Our previously proposed modeling extensions on SafeConcert was recalled in Section 2.2. Concerto-FLA analysis technique [14] is also recalled in Section 2.4. Essentially, the added value with respect to SafeConcert/Concerto-FLA is the availability of modeling and analysis capabilities for modeling and analyzing various socio aspects, AR-extended human functions and AR-related influencing factors on human functions.

We use V-model structure to illustrate methodology of the provided framework. Different steps of the methodology are shown in Fig. 6.

As it is shown in Fig. 6, there are four main steps.

In the first step, we need to answer to the question of what are the involved entities in the system. Since we model the system as a component-based system, defining involved entities determines the composite components. In an AR-equipped socio-technical system, involved entities include technical (including AR) and socio entities.

In the second step, we need to identify important aspects of each entity. These important aspects are used to determine sub-components of each composite component. In this step, our proposed taxonomies and extended modeling elements explained in Section 2.2 can be helpful to have a list of important aspects. Based on scenario and the selected case study, required sub-components can be selected. For example, paying attention can be considered as an important aspect of a human driving a car. Not paying attention would lead to failure in deciding, which is a hazardous behavior that would lead to system risk.

Third step is to model the behavior of each sub-component, which should be done based on analysis of each sub-component individually. FPTC syntax explained in Section 2.3, can be used for modeling the behavior of each sub-component.

Finally, last step is analyzing system behavior, which provides system behavior based on the provided model. We do this step based on Concerto-FLA analysis technique explained in Section 2.4.

Based on the analysis results there would be feedback for changing the system design in order to decrease risk. This feedback can be suggestions for safety requirements or functional modifications.

Proposed risk assessment activities support several ISO 26262 and SOTIF development process activities, shown in Table 1. Defining involved entities in step 1 and important aspects of each entity in step 2 supports *Item definition* activity of ISO 26262 standard and *functional and system specification* of SOTIF standard. In step 1 and 2 of our proposed activities, components and sub-components are defined,

Table 1
Risk assessment activities of our provided framework and supported ISO 26262 and SOTIF development process activities.

The proposed activity	ISO 26262 activity	SOTIF activity
Defining involved entities and important aspects of each entity (Step1 and 2)	Item definition	Functional and system specification
Defining important aspects of each entity, analyzing its behavior and system behavior (Step2, 3 and 4)	HARA	SOTIF related hazard identification and risk evaluation and Identification and evaluation of triggering events
Analyzing system behavior (Step 4)	Functional safety concept	Functional modification to reduce SOTIF risk
Analyzing system behavior (Step 4)	Technical safety concept	Functional modification to reduce SOTIF risk
Analyzing system behavior (Step 4)	Verification test	Verification of the SOTIF

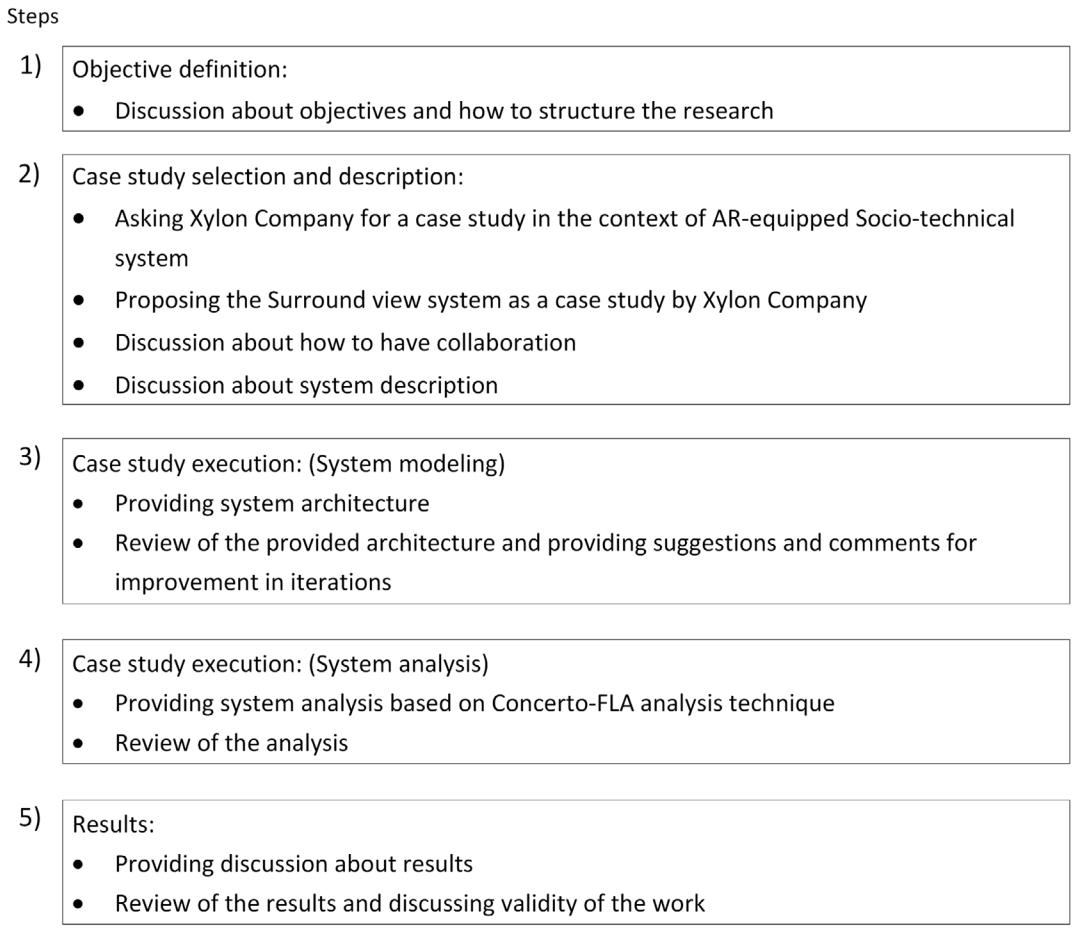


Fig. 7. Steps taken for the carried out research.

which can support provision of items and functional specification. System model including all components and sub-components support provision of system specification. Provided component-based model in step 1 and 2 of our proposed framework can be used as work products expected by the standards.

Modeling important aspects of each entity, analyzing their behavior and analyzing system behavior supports *hazard analysis and risk assessment (HARA)* of ISO 26262 standard and *SOTIF related hazard identification and risk evaluation* and also *identification and evaluation of triggering events* of SOTIF standard. In *hazard analysis and risk assessment* of ISO 26262, the aims are to identify the hazards and formulating safety goals. Step 2, 3 and 4 of our proposed activities support hazard identification by modeling failure propagation and by providing analysis results of different scenarios. These results support formulating safety goals to avoid unaccepted risks. In *SOTIF related hazard identification and risk evaluation*, the aims are identifying and evaluating SOTIF related hazards and their consequences. Modeling and analyzing activities in step 2, 3 and 4 provide the support for identification and evaluation of SOTIF related hazards and their consequences. For example, failing to pay attention leads to deciding incorrectly, which is a SOTIF related hazard and it leads to executing incorrectly. The modeling elements, used in step 2 and 3, provide the possibility to model and analyze paying attention, deciding and executing functions. Analysis in step 4 also provides the consequences at system level. Provided model in step two, three and analysis results in step four can be used as work products expected by the standards.

Analyzing system behavior in step 4 also supports defining functional and technical safety requirements, which are used in *functional*

Fig. 8. Sample images from 3D videos provided in surround view system.

and technical safety concept of ISO 26262 standard and it also supports *functional modification to reduce SOTIF risk* of SOTIF standard. In addition, analysis results are based on considering various scenarios, which support *verification test* in ISO 26262 and *verification of the SOTIF*. Required work products for verification test in ISO 26262 and SOTIF standards can be prepared based on analysis results in step four of our proposed framework.

4. Case study design and execution

In this section, we design a case study to present the modeling and analyzing capabilities of the proposed framework that can be used to qualitatively analyze the risks for AR-equipped socio-technical systems.

4.1. Objectives

Our objectives include presenting the modeling capabilities and analysis capabilities of our proposed framework containing AR-related extensions. In other words, we aim at estimating the effectiveness of the provided framework in predicting risk caused by new AR-related dependability threats. In order to do that, we use an industrial case study from automotive domain to evaluate the proposed extensions. Analysis results can be used for defining related safety requirements

4.2. Research methodology

We use case study research methodology based on [36]. The steps carried out for the presented research are presented in Fig. 7. In the first step, objectives and the structure of the research are discussed.

In the second step, we asked Xylon Company for a case study in the context of augmented reality socio-technical systems. Surround view system as a case study was suggested by this company and a meeting was organized to decide about the collaboration. We also discussed about system description.

In the third step, system architecture was provided based on information provided by the company and it was reviewed in several iterations for improvement.

In the fourth step, analysis of the case study was provided based on Concerto-FLA analysis technique and it was reviewed in iterations for improvement.

In the fifth step, a discussion about results and lessons learnt was provided. Then, the results are reviewed and a discussion about validity of the work is provided.

4.3. Case study selection and description

The case study is conducted in collaboration with Xylon, an electronic company providing intellectual property in the fields of embedded graphics, video, image processing and networking.

In this study, we select as case study subject a socio-technical system containing the following entities:

- Road transport organization (socio entity): representing the organization responsible for providing transport rules and regulations, proper road conditions and etc.
- Driver (socio entity): representing a human who is expected to drive a vehicle and park it safely by utilizing augmented reality technology used in the surround view system of the vehicle.
- Vehicle (technical entity): representing vehicle containing surround view system (a SEooC with the potential for using in vehicles with high levels of driving automation. However, currently it is used at driving automation level 0. It includes augmented reality technology to empower drivers).

Surround view systems are used to assist drivers to park more safely by providing a 3D video from the surrounding environment of the car. In Fig. 8, it is illustrated how the 3D video is shown to the driver. As it is shown in Fig. 8, driver can have a top view of the car while driving. This top view is obtained by compounding 4 views captured by 4 cameras mounted around the car and by changing point of view. It is like there is a flying camera visualizing vehicle's surrounding, which is called virtual flying camera feature. A picture of a virtual car is also augmented to the video to show the position of the car. Navigation information and parking lines also can be annotated to the video by visual AR technology. The current surround view system is a SEooC of driving automation level 0. However, Xylon plan to develop automated driving system features in higher levels for the future versions of the system.

Assumptions on the scope of the SEooC are:

- The system can be connected to the rest of the vehicle in order to obtain speed information. In case of drawing parking path lines, steering wheel angle and information from gearbox would also be obtained to determine reverse driving.

Assumptions on functional requirements of the SEooC are:

- The system is enabled either at low speed or it can be activated manually by the driver.
- The system is disabled either when moving above some speed threshold or it can be deactivated by driver.

Assumptions on the functional safety requirements allocated to the SEooC are:

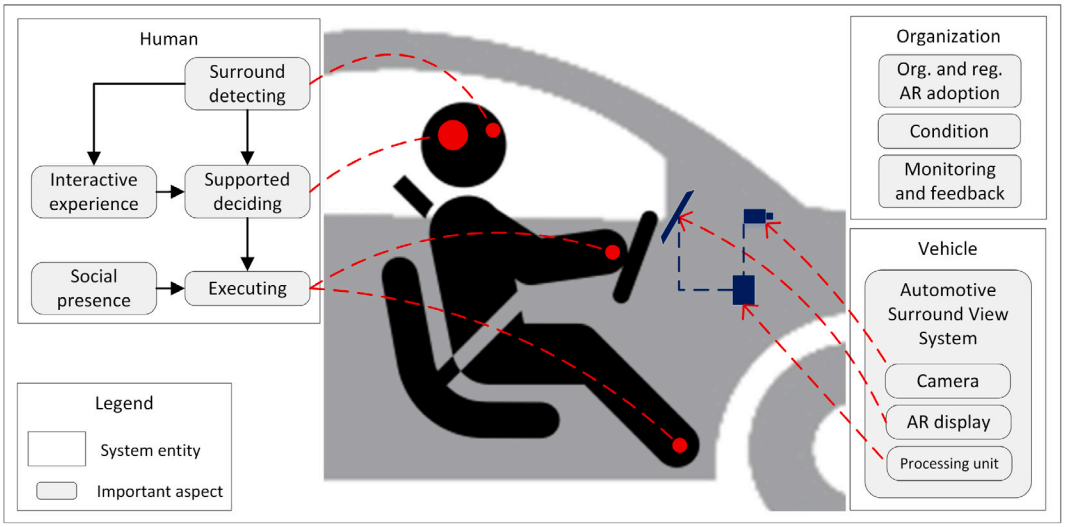


Fig. 9. Integration of the human, organization and vehicle effective aspects.

- The system does not activate the function at high vehicle speed automatically.
- The system does not deactivate the functionality at low speed automatically.

4.4. Case study execution: System modeling

This subsection reports on how we model the described system in Section 4.3 using our proposed framework.

Section 4.3 provides the required information for the first step of the risk assessment process defined in Fig. 6, which is identifying the entities for defining composite components. Based on the selected case study explained in Section 4.3, organization, driver and vehicle containing an automotive surround view system are three composite components of this system. In this subsection, we provide information for the second and third steps of risk assessment process.

Important aspects of each entity are modeled as sub-components of each composite component. For socio entities, the important aspects are selected from extended modeling elements explained in Section 2.2 and for vehicle, which is a technical entity the important aspects are based on system description.

- Important aspects of road transport organization (selected from Fig. 2):
  - *Organization and regulation AR adoption*: it refers to upgrading rules and regulations of road transport organization based on AR technology.
  - *Condition*: it refers to road condition.
  - *Monitoring and feedback*: it refers to the monitoring task and feedback provided by organization.
- Important aspects of driver (selected from Fig. 1):
  - *Surround detecting*: it is an AR-extended function, because driver can detect surround environment through AR technology.
  - *Supported deciding*: it is an AR-extended function, because driver can decide with the support of AR technology.
  - *Executing*: it is human executing function.
  - *Interactive experience*: it is an AR-caused factor, because AR provides interactive ways for enhancing user experience.
  - *Social presence*: it is an AR-caused factor, because AR may decrease social presence and lead to human failure.
- Important aspects of vehicle containing surround view system (selected based on system description received from Xylon Company):
  - A set of speed sensors: each sensor is a hardware for providing speed of the vehicle based on its movement.
  - A set of cameras: each camera is a hardware for providing raw data for a video receiver. Usually there are four cameras that can be attached to four sides of the car.
  - Switch: switch is a hardware for receiving on/off command from driver. It is also possible to send on/off command automatically based on driving requirement.
  - Peripheral controller: peripheral controller includes hardware and driver for receiving user inputs such as on/off command and speed and for sending them to user application implementation.
  - A set of video receivers: each video receiver includes a hardware and a driver. Its hardware is used for transforming raw data to AXI-stream based on the command from its driver implementation.
  - Video storing unit: video storing unit includes a hardware and a driver. Its hardware is used for receiving AXI-stream and storing it to the memory by means of DDR memory controller based on the command received form its driver.
  - DDR controller: DDR controller is a hardware for accessing DDR memory, which stores video in DDR memory and provides general memory access to all system IPs.
  - Video processing IP: Video processing IP includes hardware and driver for reading prepared data structures and video from memory, for processing video accordingly and finally for storing the processed video to memory through DDR controller. The prepared data is stored to memory by video processing IP driver based on the data structures received from memory.
  - Display controller: Display controller includes hardware and driver for reading memory where processed video is stored and for converting it in the format appropriate for driving displays.
  - Processing unit: processing unit includes hardware and software, which its software contains all the software and drivers of all other IPs. The software also contains user application implementation and video processing engine implementation. User application implementation receives inputs from peripheral unit and controls operation of all

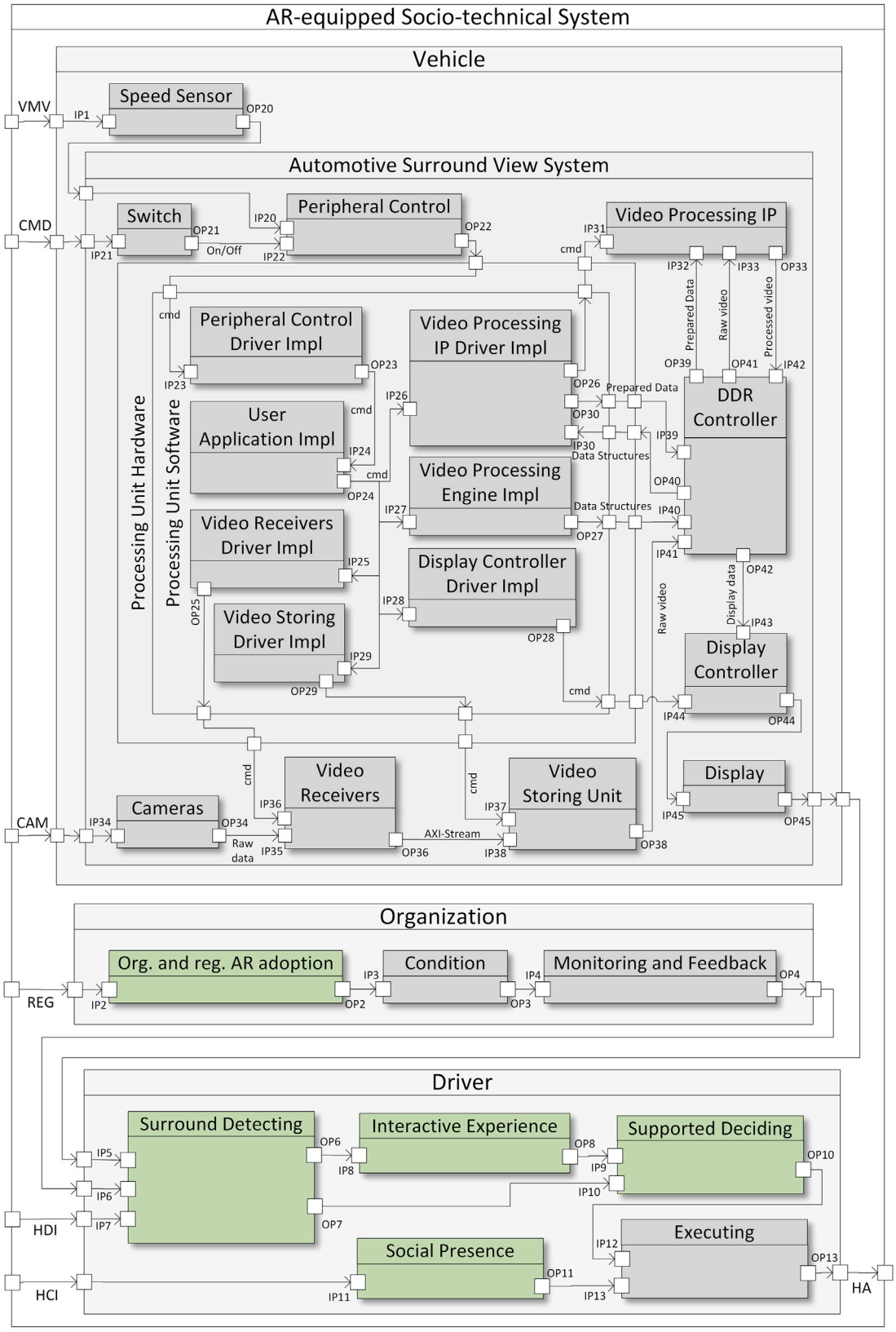


Fig. 10. AR-equipped socio-technical system modeling.

IPs by means of their software drivers. Video processing engine implementation prepares data structures to be stored in DDR memory through DDR controller.

Fig. 9 provides an overview of the integration of the human, organization and some of vehicle important aspects. In Fig. 10, we show how this AR-equipped socio-technical system is modeled using SafeConcert AR-extended modeling elements. Driver is composed of five sub-components. Driver has four inputs and two of its inputs are from system inputs with the names human detection input (HDI) and human communication input (HCI). Two other inputs are from organization and surround view system. We consider *interactive experience* and *social presence* as two sub-components of human component, which are influencing factors on human functions. *Interactive experience* affects on *supported deciding* and is affected by *surround detecting*. *Social presence* affects on human *executing*. Driver output, which is output of the system is human action shown by HA.

Organization and regulation AR adoption, *condition* and *monitoring and feedback* are three sub-components of organization composite component. Organization component receives input from system, which represents influences from regulation authorities on the organization (REG).

Vehicle is also modeled with three inputs including user command shown by CMD, vehicle movement shown by VMV and camera input shown by CAM. Green color is used to show the extended modeling elements used in this system.

4.5. Case study execution: System analysis

This subsection reports on the analysis of the system using AR-related extensions, which refers to the last step of the risk assessment process defined in Fig. 6. We follow the five steps of Concerto-FLA analysis technique explained in Section 2.4 for system analysis.

1. First step is provided in Fig. 10. We explained how the system is modeled in Section 4.4.
2. Second step is shown by providing FPTC rules, which are used for linking possible failure modes in the input of each component to the possible failure modes in the output. "IP.variable → OP.variable" shows propagational behavior of the component, which means that any failure mode in its input is propagated to its output. FPTC rules of modeled sub-components are shown in Figs. 11–14. There is one box for each component. The left part of the box shows the name of the component. The right part of the box shows possible failure modes in the input (up left), possible failure modes in the output (up right) and FPTC rules (bottom). Based on dependability-related terminology in literature such as [37,38] and [39], we consider omission, commission, etc. as failure modes. However, these are named failures in FPTC terminology.
   In this paragraph, we explain how the possible failure modes at input and output are identified/defined in Figs. 11–14. For example, the camera takes in input a raw image. Based on the definitions of failure modes in Section 2.3, omission and valueSubtle are the possible failure modes for the case of camera. The reason for having omission as a possible failure mode at input is the possibility of an occlusion in front of the camera, which prevents receiving raw image as input. The reason for having valueSubtle as possible failure mode at input is the possibility of intervene, which leads to receiving input not in the expected range. For example, when image is blurred because of foggy weather. Possible failure modes at output can be obtained by considering the possible input failure modes in the FPTC rules. Defining the FPTC rules are explained in the next paragraph.
   In this paragraph, we explain how the FPTC rules are defined in Figs. 11–14. FPTC rules show how the component behaves. For example, the camera would not produce any failure, but if the input image is not in the expected range, then the output would not be in the expected range either. Moreover, if the input is not provided when expected, then the output would not be provided when expected. Thus, the camera propagates possible input failure modes to the output and it does not behave as source, sink and transformational (explained in Section 2.3).
   In scenarios, we may change some components' failure behavior to source based on assumptions related to that scenario. For example, if we assume that an AR-related component is producing failure, then we need to change its failure behavior to source and update its FPTC rules.
3. Third step is to consider failure modes in inputs of the system to calculate failure propagation. In this case study, we inject noFailure to four inputs of the system, because we aim at analyzing system for scenarios that failure is originating from our modeled system and more specifically from our AR-related part of the system.
4. Fourth step is calculating the failure propagations. We consider three scenarios and show the analysis results in Figs. 15–17.
5. Last step is back propagation of results. Interpretation of the back-propagated results can be used to make decision about design change or defining safety barrier, if it is required.

4.5.1. Scenario 1:

- **Description of the scenario:** In this scenario, we assume that failure in the system is emanated from the technical part of the system. We assume video processing IP produces processed video incorrectly. For example, we assume that the expected visual mark for parking lot striping is assigned on an incorrect position (value failure mode). As a consequence, the driver cannot detect the surround environment correctly and decides and executes incorrectly (value failure mode).
- **Modeling failure behavior:** We show the failure propagation with underlined FPTC rules, which are the rules that are activated, shown in Fig. 15. In this scenario, video processing IP behaves as source and while its inputs are noFailure, it produces valueSubtle failure mode in its output. This activated rule is shown on its sub-component. DDR controller, display controller and display sub-components behave as propagational and propagate valueSubtle from inputs to outputs.
- **Analysis of system behavior:** ValueSubtle failure mode in IP5 means that displayed information on the display is not correct. ValueSubtle propagates to *surround detecting*, *interactive experience* and *supported deciding* and it transforms to valueCoarse in *executing*. The reason for this transformation is that if there is value failure mode in *executing* function, it can be detected by user, which means valueSubtle transforms to valueCoarse. We show the failure propagation by blue color of the underlined FPTC rules.
- **Interpreting the results:** Based on back propagation of the results, we can explain how the rules have been triggered. ValueCoarse on OP13 is because of valueSubtle on IP12. ValueSubtle on IP12 is because of valueSubtle on OP10 and we continue this back propagation to reach a component originating the failure, which is component with inputs IP31, IP32 and IP33. This component is video processing IP.
  The analysis results can be helpful in hazard identification and categorization. Since the reason for system failure is a technical component, functional safety is addressed by ISO 26262.
  In this case, unintended displayed information is the identified hazard and the reason is failure in video processing IP. System failure in this scenario would lead to light accident and light injuries. The reason is that the speed is not usually high while parking the car. Based on the explanation in Section 2.5 and Fig. 3, severity in this case is S1. Class of exposure is E4, because probability of exposure is more than 10%. It means that it is more than 10 percent probable that a driver be exposed to parking

Name of the component	Possible failure modes at input	Possible failure modes at output
	FPTC rules	
Camera	IP34: omission, valueSubtle	OP34: omission, valueSubtle
	IP34.variable → OP34.variable;	
Speed Sensor	IP1: omission, valueSubtle	OP20: omission, valueSubtle
	IP1.variable → OP20.variable;	
Switch	IP21: late, omission, commission	OP21: late, commission, omission
	IP21.variable → OP21.variable;	
Peripheral Control	IP20: late, omission, valueSubtle IP22: late, omission, commission, valueSubtle	OP22: late, omission, commission, valueSubtle
	IP20.noFailure, IP22.noFailure → OP22.noFailure; IP20.variable, IP22.noFailure → OP22.variable; IP20.noFailure, IP22.variable → OP22.variable; IP20.variable, IP22.variable → OP22.variable; IP20.wildcard, IP22.omission → OP22.omission; IP20.omission, IP22.wildcard → OP22.omission; IP20.late, IP22.commission → OP22.commission; IP20.late, IP22.valueSubtle → OP22.valueSubtle; IP20.valueSubtle, IP22.late → OP22.valueSubtle; IP20.valueSubtle, IP22.commission → OP22.valueSubtle;	
Peripheral Control Driver Imp	IP23: late, omission, commission, valueSubtle	OP23: late, omission, commission, valueSubtle
	IP23.variable → OP23.variable;	
User Application Imp	IP24: late, omission, commission, valueSubtle	OP24: late, omission, commission, valueSubtle
	IP24.variable → OP24.variable;	
Video Receiver Driver Imp	IP25: late, omission, valueSubtle, commission	OP25: late, omission, commission, valueSubtle
	IP25.variable → OP25.variable;	
Video Processing Engine Imp	IP27: late, omission, valueSubtle	OP27: late, omission, valueSubtle
	IP27.variable → OP27.variable;	
Display Controller Driver Imp	IP28: late, omission, valueSubtle, commission	OP28: late, omission, commission, valueSubtle
	IP28.variable → OP28.variable;	
Video Storing Driver Imp	IP29: late, omission, valueSubtle, commission	OP29: late, omission, commission, valueSubtle
	IP29.variable → OP29.variable;	
Video Processing IP Driver Imp	IP26: late, omission, commission IP30: late, omission, valueSubtle	OP26: late, omission, commission, valueSubtle OP30: late, omission, valueSubtle
	IP26.noFailure, IP30.noFailure → OP26.noFailure, OP30.noFailure; IP26.variable, IP30.variable → OP26.variable, OP30.variable; IP30.valueSubtle, IP26.late → OP30.valueSubtle, OP26.late; IP30.wildcard, IP26.omission → OP26.omission, OP30.omission; IP30.omission, IP26.wildcard → OP30.valueSubtle, OP26.valueSubtle; IP30.late, IP26.commission → OP30.commission, OP26.valueSubtle; IP30.valueSubtle, IP26.commission → OP30.commission, OP26.valueSubtle;	

Fig. 11. Modeling failure behavior of components.

Video Processing IP	IP31: late, omission, valueSubtle IP32: late, omission, valueSubtle IP33: late, omission, valueSubtle	OP33: late, omission, valueSubtle
	IP31.noFailure, IP32.noFailure, IP33.noFailure → OP33.noFailure; IP31.omission, IP32.wildcard, IP33.wildcard → OP33.omission; IP31.wildcard, IP32.omission, IP33.wildcard → OP33.omission; IP31.wildcard, IP32.wildcard, IP33.omission → OP33.omission; IP31.late, IP32.noFailure, IP33.noFailure → OP33.late; IP31.noFailure, IP32.late, IP33.noFailure → OP33.late; IP31.noFailure, IP32.noFailure, IP33.late → OP33.late; IP31.value, IP32.noFailure, IP33.noFailure → OP33.valueSubtle; IP31.noFailure, IP32.value, IP33.noFailure → OP33.valueSubtle; IP31.noFailure, IP32.noFailure, IP33.valueSubtle → OP33.valueSubtle; IP31.late, IP32.valueSubtle, IP33.noFailure → OP33.valueSubtle; IP31.valueSubtle, IP32.late, IP33.noFailure → OP33.valueSubtle; IP31.noFailure, IP32.late, IP33.valueSubtle → OP33.valueSubtle; IP31.noFailure, IP32.valueSubtle, IP33.late → OP33.valueSubtle; IP31.valueSubtle, IP32.noFailure, IP33.late → OP33.valueSubtle; IP31.late, IP32.noFailure, IP33.valueSubtle → OP33.valueSubtle; IP31.late, IP32.late, IP33.late → OP33.late; IP31.valueSubtle, IP32.valueSubtle, IP33.valueSubtle → OP33.valueSubtle; IP31.late, IP32.late, IP33.valueSubtle → OP33.valueSubtle; IP31.valueSubtle, IP32.late, IP33.late → OP33.valueSubtle; IP31.late, IP32.valueSubtle, IP33.late → OP33.valueSubtle; IP31.valueSubtle, IP32.late, IP33.valueSubtle → OP33.valueSubtle; IP31.valueSubtle, IP32.valueSubtle, IP33.late → OP33.valueSubtle; IP31.late, IP32.valueSubtle, IP33.valueSubtle → OP33.valueSubtle;	
Video Receiver	IP35: late, omission, valueSubtle IP36: late, omission, commission, valueSubtle	OP36: late, omission, valueSubtle, commission
	IP35.noFailure, IP36.noFailure → OP36.noFailure; IP35.variable, IP36.noFailure → OP36.variable; IP35.noFailure, IP36.variable → OP36.variable; IP35.variable, IP36.variable → OP36.variable; IP35.wildcard, IP36.omission → OP36.omission; IP35.omission, IP36.wildcard → OP36.omission; IP35.late, IP36.commission → OP36.commission; IP35.late, IP36.valueSubtle → OP36.valueSubtle; IP35.valueSubtle, IP36.late → OP36.valueSubtle; IP35.valueSubtle, IP36.commission → OP36.valueSubtle;	
Video Storing Unit	IP37: late, omission, commission, valueSubtle IP38: late, omission, valueSubtle	OP38: late, omission, valueSubtle, commission
	IP38.noFailure, IP37.noFailure → OP38.noFailure; IP38.variable, IP37.noFailure → OP38.variable; IP38.noFailure, IP37.variable → OP38.variable; IP38.variable, IP37.variable → OP38.variable; IP38.wildcard, IP37.omission → OP38.omission; IP38.omission, IP37.wildcard → OP38.omission; IP38.late, IP37.commission → OP38.commission; IP38.late, IP37.valueSubtle → OP38.valueSubtle; IP38.valueSubtle, IP37.late → OP38.valueSubtle; IP38.valueSubtle, IP37.commission → OP38.valueSubtle;	
DDR Controller	IP39: Late, omission, valueSubtle IP40: late, omission, valueSubtle IP41: late, omission, valueSubtle IP42: late, omission, valueSubtle	OP39: late, omission, valueSubtle OP40: late, omission, valueSubtle OP41: late, omission, valueSubtle OP42: late, omission, valueSubtle
	IP39.variable, IP40.wildcard, IP41.wildcard, IP42.wildcard → OP39.variable; IP39.wildcard, IP40.variable, IP41.wildcard, IP42.wildcard → OP40.variable; IP39.wildcard, IP40.wildcard, IP41.variable, IP42.wildcard → OP41.variable; IP39.wildcard, IP40.wildcard, IP41.wildcard, IP42.variable → OP42.variable;	

Fig. 12. Modeling failure behavior of components (Cont.).

Display Controller	IP43: late, omission, valueSubtle IP44: late, omission, commission, valueSubtle	OP44: late, omission, valueSubtle
	IP43.noFailure, IP44.noFailure → OP44.noFailure; IP43.variable, IP44.noFailure → OP44.variable; IP43.noFailure, IP44.variable → OP44.variable; IP43.variable, IP44.variable → OP44.variable; IP43.wildcard, IP44.omission → OP44.omission; IP43.omission, IP44.wildcard → OP44.omission; IP43.late, IP44.commission → OP44.commission; IP43.late, IP44.valueSubtle → OP44.valueSubtle; IP43.valueSubtle, IP44.late → OP44.valueSubtle; IP43.valueSubtle, IP44.commission → OP44.valueSubtle;	
Display	IP45: late, omission, commission, valueSubtle	OP45: late, omission, commission, valueSubtle
	IP45.variable → OP45.variable;	
Org. and Reg. AR Adoption	IP2: late, omission, valueSubtle, valueCoarse	OP2: late, omission, valueSubtle, valueCoarse
	IP2.variable → OP2.variable;	
Condition	IP3: late, omission, valueSubtle, valueCoarse	OP3: late, omission, valueSubtle, valueCoarse
	IP3.variable → OP3.variable;	
Monitoring and Feedback	IP4: late, omission, valueSubtle, valueCoarse	OP4: late, omission, valueSubtle, valueCoarse
	IP4.variable → OP4.variable;	
Surround Detecting	IP5: late, omission, valueSubtle IP6: late, omission, valueSubtle IP7: omission, valueSubtle, late	OP6: late, omission, valueSubtle OP7: late, omission, valueSubtle
	IP5.noFailure, IP6.noFailure, IP7.noFailure → OP6.noFailure, OP7.noFailure; IP5.omission, IP6.wildcard, IP7.wildcard → OP6.omission, OP7.omission; IP5.wildcard, IP6.omission, IP7.wildcard → OP6.omission, OP7.omission; IP5.wildcard, IP6.wildcard, IP7.omission → OP6.omission, OP7.omission; IP5.late, IP6.noFailure, IP7.noFailure → OP6.late, OP7.late; IP5.noFailure, IP6.late, IP7.noFailure → OP6.late, OP7.late; IP5.noFailure, IP6.noFailure, IP7.late → OP6.late, OP7.late; IP5.valueSubtle, IP6.noFailure, IP7.noFailure → OP6.valueSubtle, OP7.valueSubtle; IP5.noFailure, IP6.valueSubtle, IP7.noFailure → OP6.valueSubtle, OP7.valueSubtle; IP5.noFailure, IP6.noFailure, IP7.valueSubtle → OP6.valueSubtle, OP7.valueSubtle; IP5.late, IP6.valueSubtle, IP7.noFailure → OP6.valueSubtle, OP7.valueSubtle; IP5.valueSubtle, IP6.late, IP7.noFailure → OP6.valueSubtle, OP7.valueSubtle; IP5.noFailure, IP6.late, IP7.valueSubtle → OP6.valueSubtle, OP7.valueSubtle; IP5.noFailure, IP6.valueSubtle, IP7.late → OP6.valueSubtle, OP7.valueSubtle; IP5.valueSubtle, IP6.noFailure, IP7.late → OP6.valueSubtle, OP7.valueSubtle; IP5.late, IP6.noFailure, IP7.valueSubtle → OP6.valueSubtle, OP7.valueSubtle; IP5.late, IP6.late, IP7.late → OP6.late, OP7.late; IP5.valueSubtle, IP6.valueSubtle, IP7.valueSubtle → OP6.valueSubtle, OP7.valueSubtle; IP5.late, IP6.late, IP7.valueSubtle → OP6.valueSubtle, OP7.valueSubtle; IP5.valueSubtle, IP6.late, IP7.late → OP6.valueSubtle, OP7.valueSubtle; IP5.late, IP6.valueSubtle, IP7.late → OP6.valueSubtle, OP7.valueSubtle; IP5.valueSubtle, IP6.late, IP7.valueSubtle → OP6.valueSubtle, OP7.valueSubtle; IP5.valueSubtle, IP6.valueSubtle, IP7.late → OP6.valueSubtle, OP7.valueSubtle; IP5.late, IP6.valueSubtle, IP7.valueSubtle → OP6.valueSubtle, OP7.valueSubtle;	
Interactive Experience	IP8: late, omission, valueSubtle	OP8: late, omission, valueSubtle
	IP8.variable → OP8.variable;	

Fig. 13. Modeling failure behavior of components (Cont.).

Supported Deciding	IP9: late, omission, valueSubtle IP10: late, omission, valueSubtle	OP10: late, omission, valueSubtle
	IP9.noFailure, IP10.noFailure → OP10.noFailure; IP9.variable, IP10.noFailure → OP10.variable; IP9.noFailure, IP10.variable → OP10.variable; IP9.variable, IP10.variable → OP10.variable; IP9.wildcard, IP10.omission → OP10.omission; IP9.omission, IP10.wildcard → OP10.omission; IP9.late, IP10.valueSubtle → OP10.valueSubtle; IP9.valueSubtle, IP10.late → OP10.valueSubtle;	
Social Presence	IP11: late, omission, valueSubtle	OP11: late, omission, valueSubtle
	IP11.variable → OP11.variable;	
Executing	IP12: late, omission, valueSubtle IP13: late, omission, valueSubtle	OP13: late, omission, valueCoarse
	IP12.noFailure, IP13.noFailure → OP13.noFailure; IP12.late, IP13.noFailure → OP13.late; IP12.noFailure, IP13.late → OP13.late; IP12.late, IP13.late → OP13.late; IP12.valueSubtle, IP13.noFailure → OP13.valueCoarse; IP12.noFailure, IP13.valueSubtle → OP13.valueCoarse; IP12.valueSubtle, IP13.valueSubtlev → OP13.valueCoarse; IP12.wildcard, IP13.omission → OP13.omission; IP12.omission, IP13.wildcard → OP13.omission; IP12.late, IP13.valueSubtle → OP13.valueCoarse; IP12.valueSubtle, IP13.late → OP13.valueCoarse;	

Fig. 14. Modeling failure behavior of components (Cont.).

situation while driving a car. Finally, class of controllability is C1 or normally controllable. It means that more than 90% of drivers can control this situation. Therefore, ASIL level for this case is A, based on Fig. 3. If we aim at overcoming risks with ASIL level A, then we should define safety goal, functional and technical safety requirements in order to overcome this risk. For example, for this scenario safety goal, functional safety requirement and technical safety requirement can be defined as follows to prevent failure in processing unit IP:

- **Safety goal**: The driver shall be notified, if there is failure in processing.
- **Safety requirement**:
  - **Functional safety requirement**: A monitoring component should be used to check the processing actively.
  - **Technical safety requirement**: Monitoring function should check the processing output every 10 ms.

After interpreting the results and providing safety requirements, system design would be updated. Then, failure behavior can also be updated and failure propagation analysis can be repeated for another iteration.

4.5.2. Scenario 2:

- **Description of the scenario:** In this scenario, we assume that the technical part of the system works without failure, but driver does not have interactive experience. For example, it is the first time driver is working with systems containing AR and he/she cannot understand the meaning of AR notations. Therefore, driver would decide and execute incorrectly.
- **Modeling failure behavior:** We show the failure propagation with underlined FPTC rules, which are the rules that are activated, shown in Fig. 16. Surround view sub-components behave as propagational and propagate noFailure from inputs to outputs. *Interactive experience* behaves as source and while its input is noFailure, it has omission failure mode in its output. This activated rule is shown on this component.
- **Analysis of system behavior:** Omission failure mode in *interactive experience* transforms to valueSubtle in *supported deciding*, because lack of interactive experience causes wrong decision and in *executing*, it transforms to valueCoarse. Similar to the first scenario, the reason for this transformation is that if there is value failure mode in *executing* function, it can be detected by user, which means valueSubtle transforms to valueCoarse.
- **Interpreting the results:** Based on back propagation of the results, we can explain how the rules have been triggered. ValueCoarse on OP13 is because of valueSubtle on IP2. ValueSubtle on IP12 is because of valueSubtle on OP10 and we continue to IP8, which is related to *interactive experience* component.
  In this scenario, we considered failure in AR-related part of the system and since it refers to limitation in intended functionality (SOTIF related hazards), we do not determine ASIL level. If the expected severity and controllability of the scenario is higher than S0 and C0 respectively, we need to consider SOTIF safety process [40]. As we explained in the previous scenario, severity and controllability are higher than S0 and C0. Lack of interactive experience leads to system failure and incorrect deciding is the identified hazard. Safety goal and safety requirement can be defined as follows. Since the failure is not emanated from technical part of the system, we do not need to specify technical safety requirement:
  - **Safety goal**: Interactive experience shall be provided for the driver.
  - **Safety requirement**: The Company should provide a training video for all drivers at the first time of using the system.

  After applying the requirements the behavior of this component would change from source to other types and analysis can be repeated.
  It is not possible to detect risk originated from failure in interactive experience, without using the proposed representation means, because using these representation means or modeling elements provide the possibility to analyze their failure propagation

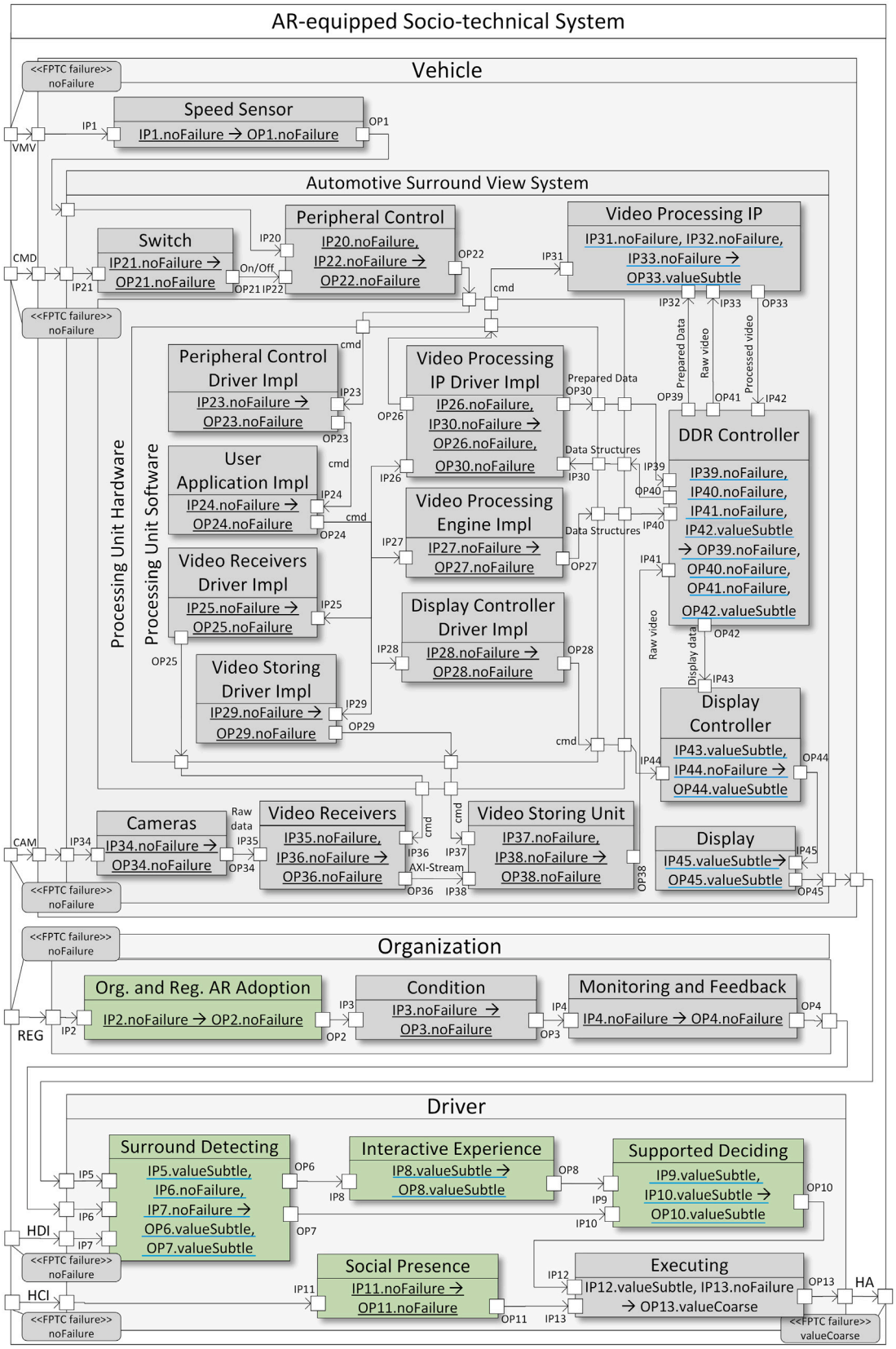


Fig. 15. Analyzing AR-equipped socio-technical system (Scenario1).

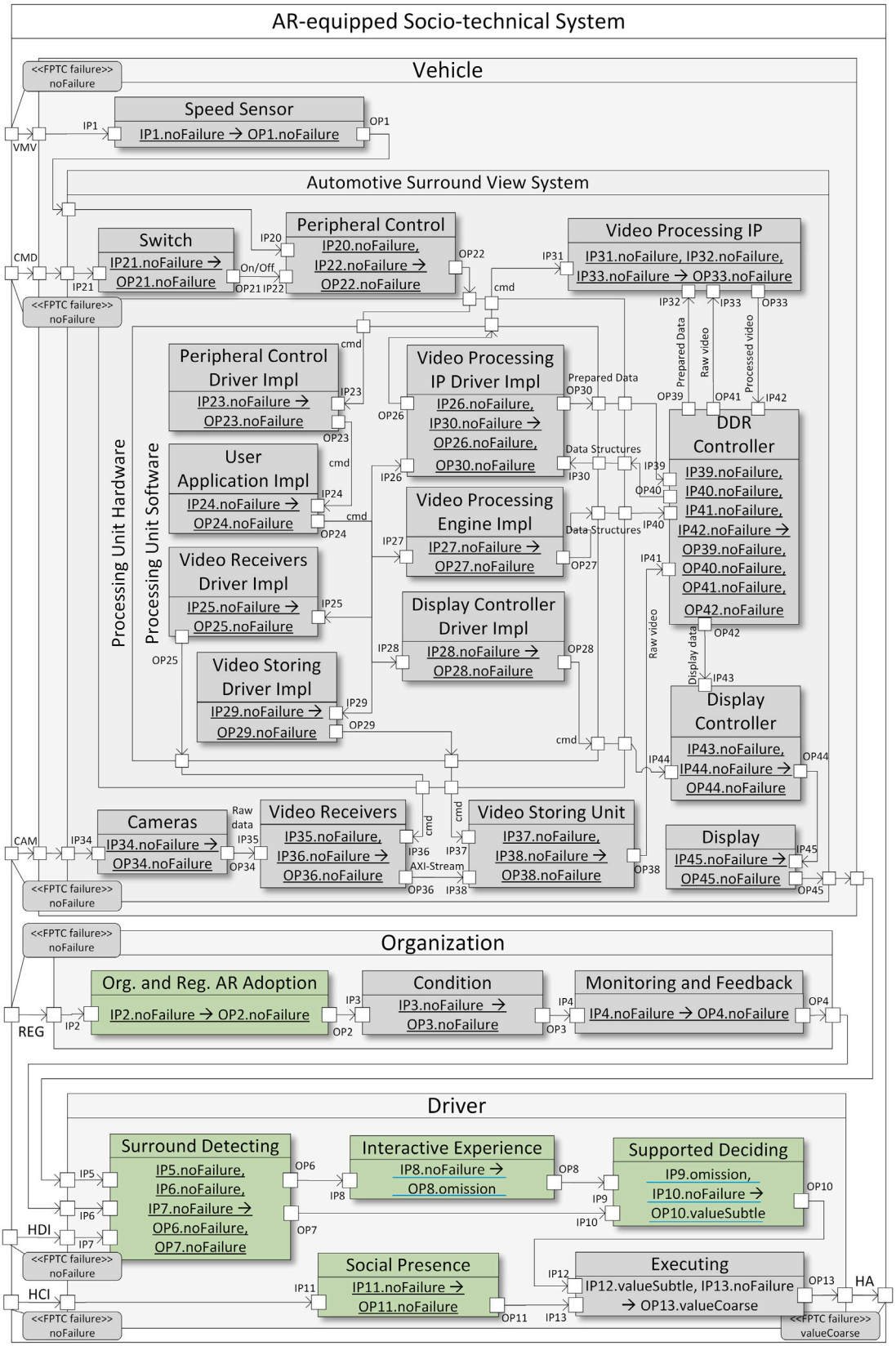


Fig. 16. Analyzing AR-equipped socio-technical system (Scenario2).

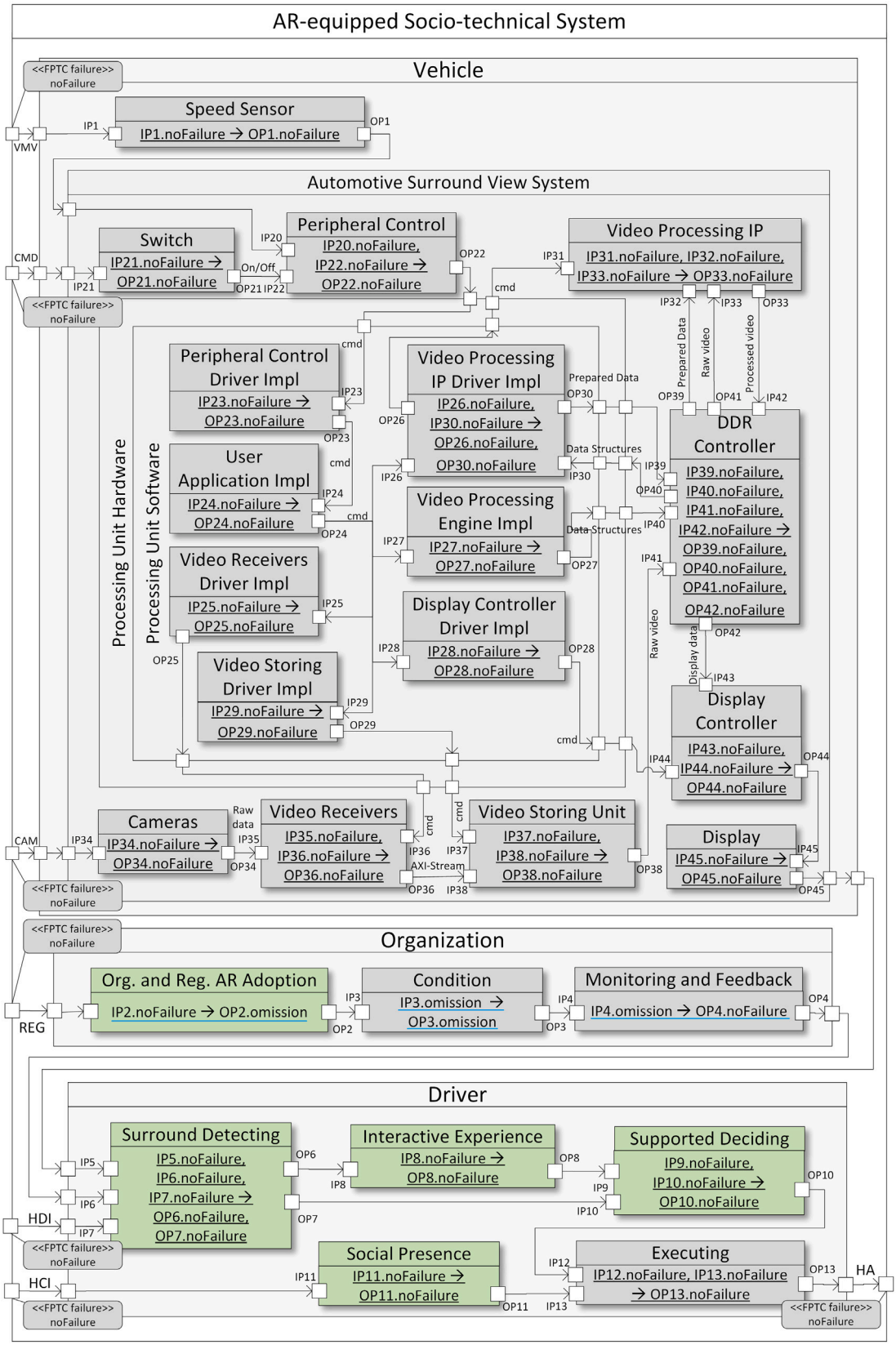


Fig. 17. Analyzing AR-equipped socio-technical system (Scenario3).

and provides the possibility to analyze effect of these failures on system behavior. Then based on analysis results decision about design change or fault mitigation mechanisms would be taken.

4.5.3. Scenario 3:

- **Description of the scenario:** In this scenario, we assume that road transport organization has not updated rules and regulations based on AR technology, which is a limitation in intended functionality. For example, parking lot striping is not updated to be used by AR applications and it affects on road condition, but *monitoring and feedback* component detect this problem and provide a feedback to driver. This feedback would be a visual text alarm showing that there is a problem in AR information. Therefore, driver will not depend on shown result and try to decide and execute correctly.
- **Modeling failure behavior:** We show the failure propagation with underlined FPTC rules, which are the rules that are activated, shown in Fig. 17. Similar to the previous scenario, surround view sub-components behave as propagational and propagate noFailure from inputs to output. *Organization and regulation AR adoption* behaves as source and while its input is noFailure, it has omission failure mode in its output. This activated rule is shown on this component. *Monitoring and feedback* component behaves as sink and while its input is omission, it has noFailure in its output.
- **Analysis of system behavior:** Omission failure mode propagates from *organization and regulation AR adoption* to *condition* and *monitoring and feedback*. In *monitoring and feedback* it will transform to noFailure. Then, noFailure is propagated from *surround detecting* to *interactive experience*, *supported deciding* and *executing*.
- **Interpreting the results:** In this scenario, system output is provided without failure. Thus, there is no hazard and no safety requirement is required.

4.6. Compliance with ISO 26262 and SOTIF

Based on the explanation in Section 2.5, the first step of ISO 26262 safety process is *item definition* and the first step of SOTIF safety process is *functional and system specification*. In Fig. 10, we defined the components which are used for modeling items, their interactions and dependencies. We also specified system and functions through entities specification.

The second step of ISO 26262 is *hazard analysis and risk assessment* and second step of SOTIF is *SOTIF related hazard identification and risk evaluation*. Based on the interpreted results of each scenario, hazards are identified (if there are) and categorized based on ASIL level, if they are emanated from technical failures, otherwise they are evaluated qualitatively. If the hazard is emanated from a technical fault, functional safety is addressed by ISO 26262, otherwise it is addressed by SOTIF.

The third step of SOTIF is *identification and evaluation of triggering events*. Sub-components in Fig. 10 are the identified potential triggering events and failure behavior of each of these sub-components in Figs. 11–14 are evaluation of the triggering events.

The third and fourth steps of ISO 26262 are *functional and technical safety concept* and fourth step of SOTIF is *functional modification to reduce SOTIF risk*. The aim in the two steps of ISO 26262 is to define functional and technical safety requirements. Defining functional and technical safety requirements should be based on the analysis results as explained in the first scenario. Functional modification is also provided based on the analysis results as explained in the second scenario.

Finally, *verification test* of ISO 26262 and SOTIF includes considering several scenarios and verifying system functioning. This step is supported by the provided analysis results.

4.7. Lessons learnt

In this section, we present the lessons learnt while conducting the case study. As it is shown in the second and third scenarios in Section 4.5, in an AR-equipped socio-technical system, there are system failures which are not caused by technical entities of the system and new AR-related dependability threats are the reason for these system failures. These new AR-caused dependability threats are related to intended functionality of socio entities of the system. Our proposed framework provides the required means to take into account these new AR-related dependability threats. We can consider these extensions from two perspectives:

- **Augmented reality concepts coverage**: from a coverage point of view, as shown in Section 4.4, modeling capabilities obtained by our proposed framework, allow architects and safety managers to model augmented reality effects on socio-technical systems by using modeling elements related to AR-extended human functions as well as modeling elements related to AR-caused faults leading to human failures. For example, in the second scenario, failure in *interactive experience* is considered as an AR-related dependability threat and its modeling element provides representation mean for taking into account AR effect as an AR-caused fault leading to human failures. In the third scenario, failure in updating rules and regulations based on AR technology is considered as AR-related dependability threat and its modeling element provides representation mean for taking into account AR effects. It is also shown in Section 4.5 that analysis capabilities allow architects and safety managers to have at disposal means to reveal effect of AR-related dependability threats on system behavior. It is done by analyzing failure propagation that might be effective in emerging risks within an AR-equipped socio-technical system.
- **Expressiveness**: Expressiveness refers to the power of a modeling language to express or describe all things required for a given purpose [41]. Set of symbols or possible statements that can be described by modeling languages can be used for measuring expressiveness. Statement means “a syntactic expression and its meaning”. As it is explained in Section 2.2, the extensions on human modeling elements used to extend the modeling language is based on an AR-extended human function taxonomy (AREXTax [7]). This taxonomy is obtained by extracting human functions from about six state-of-the-art human failure taxonomies (Norman [20], Reason [21], Rasmussen [22], HFACS [23], SERA [19], Driving [24]). This taxonomy is also extended based on various studies and experiments on augmented reality. In addition, the extension for extending organization modeling elements is based on a fault taxonomy (AREFTax [8]) containing AR-caused faults leading to human failures. This taxonomy is gained by harmonizing about five state-of-the-art fault taxonomies (Rasmussen [22], HFACS [23], SERA [19], Driving [24] and SPAR-H [27]). The taxonomy is also extended based on various studies and experiments on augmented reality. According to the basis of the extensions and as it is also shown in Section 4.4, the extensions increase power of modeling language to express new AR-caused risks.

We used Concerto-FLA analysis technique as the basis of the analysis in order to disclose the advantages of the proposed AR-related extensions included in our proposed framework at analysis level. Concerto-FLA uses FPTC syntax for the modeling failure behavior of each component or sub-component, which includes defining FPTC rules for a component/sub-component in isolation. It is possible to define FPTC rules for the AR-extended modeling elements characterizing their behavior. In addition, as known, modeling the failure behavior can be challenging, because the number of FPTC rules grows exponentially with the increase of the cardinality of the input ports. It is important

to consider possible failure modes for each input in a component/sub-component and skip the others. It is not conspicuous in small and academic examples, but it is really challenging when we use an industrial case study.

5. Threats to validity

In this section, we discuss threats to validity in relation to our research based on best practices available in the scientific literature [36]. Validity of a study denotes to what extent the results can be trusted.

External validity refers to possibility of generalizing the findings. We provided a case study with three scenarios from automotive domain, but the proposed framework is not limited to specific scenarios and specific domain and the baseline for the included extensions, which are AREXTax and AREFTax taxonomies are attained from taxonomies in various domains. Thus, there is the possibility of generalizing the findings for automotive domain in general and also for other domains.

Construct validity refers to the quality of choices and measurements. In our case, we used SafeConcert, which is an accepted work, as the basis of our work. Proposed extensions are also based on state-of-the-art taxonomies (Norman [20], Reason [21], Rasmussen [22], HFACS [23], SERA [19], Driving [24] and SPAR-H [27] taxonomies) and studies and experiments for the new technologies. The modeling and analysis process is done based on standardized process to increase the repeatability of the work. However, it cannot be guaranteed that different people have same answer using our proposed framework, because it depends on the analyzer skills and ability for modeling and analysis.

In this paper we used a realistic and sufficiently complex case at a level that can be found in industry to verify our proposed framework including AR-related extensions. Although we were not allowed to access confidential information related to their customers, we have been able to model system architecture and failure behavior of system components using SafeConcert metamodel, its AR-extensions and FPTC rules.

In this case study, we illustrated the modeling and analysis capabilities of our proposed framework including AR-related extensions through three different scenarios with different assumptions about the AR-related components' failure behavior. We have not shown that the modeling elements are complete for modeling all possible scenarios. Instead, we have focused on the provided elements to check if they are able to capture new system failure behaviors.

The benefit of using our proposed extensions for a particular case depends on the ability to choose the best elements and the ability to establish failure behavior of the component related to that element. Still, this case provides evidence for the applicability and usefulness of our proposed framework. Further investigations are required to provide more beneficial results on limitations of modeling and analysis applications.

6. Discussion

Statistical information is used for determining exposure, severity and controllability of ASIL value of systems with SAE-levels 0–2. It would be possible to use the same statistical information for determining exposure and severity in AR-equipped systems with higher levels of automation, but controllability is a factor, which is affected by augmented reality used in higher levels of automation. Thus, it is required to model system and include effect of augmented reality on the model to be able to involve AR effect in specifying controllability factor of ISO 26262. For providing automated driving safety, Responsibility Sensitive Safety (RSS) standard [42] can be helpful. This standard provides formalization for safe decisions by self-driving cars in cases where machine learning mechanisms are used [43].

Surround view system can be mounted on vehicles with higher levels of automation (for example level 1–3) alongside more advanced systems for providing driver assistance functionalities. In these cases, driver is not supervising the car and controllability factor should be defined by modeling system as an AR-equipped socio-technical system. In [32], a controllability classification is proposed based on human takeover time and analysis of human driver models. The value of human action times, based on studies in literature, are used for predicting mean takeover times. Since classification of controllability according to ISO 26262 requires description of percentiles, normal distribution is assumed for each action time. Normal distribution can be obtained by mean value and its standard deviation. Based on the reaction times and distributions, it is possible to calculate controllability of the situation. The proposed modeling extensions included in our proposed framework provide the possibility to model effect of augmented reality on human and effect of augmented reality on influencing factors on human functions. Thus, mean takeover time and as a consequence controllability can be updated while using augmented reality by using the proposed extensions on humans and influencing factors modeling.

The generated model using our proposed framework and analysis results can be used to provide safety case for AR-equipped industrial products. Safety case contains arguments based on evidences to demonstrate that the system is acceptably safe to work on a given environment. However, it is required to provide also some documentation explaining the results and showing how the safety requirements are achieved. Goal Structuring Notation (GSN) [44] can be used for SOTIF argumentation [45].

Extended human modeling elements can be used for modeling integration of human aspects with interactive systems in system testing. For example, MIODMIT architecture [46] is a generic architecture for interactive systems. As it is discussed in [47], human aspects should be considered and integrated while testing. Using extended modeling elements for modeling different aspects of human as a user of an interactive system would be of value for the system testing.

7. Related work

A comparative study about architecture-based risk analysis techniques is provided in [48]. Specifically, in this work, authors compare: the modeling capabilities, process and tool support of various techniques. Traditional methods such as Fault Tree Analysis (FTA) [49] and Failure Modes and Effects Analysis (FMEA) [50] are manual analysis techniques. In comparison, there are also model-driven techniques, which provide the analysis results (semi-)automatically based on the system architecture and annotated failure behavior information. Model-driven techniques such as Failure Propagation and Transformation Notation (FPTN) [51], FPTC, Hierarchically Performed Hazard Origin and Propagation Studies (HIP HOPS) [52] and techniques using Architecture Analysis and Description Language (AADL) and its technical error annex [53] are considered in this study. All these techniques consider risks emanated from technical parts. Human and organization are not considered as part of the system that would introduce risk.

A framework for construction safety management and visualization system (SMVS) is proposed in [54]. This framework includes a safety management process, which includes planning, education and inspection phases. A prototype system is also developed and tested. The results shown that this framework improves risk identification and communication between managers and workers in construction sites. Augmented reality is used for improving the safety management process. In comparison to our work, in this paper the proposed framework is specific to construction domain. AR is also used for safety management process improvement, but it is not considered as part of the system, which is going to be evaluated. Thus, risks emanated from AR and AR-related factors are not included in the process.

In risk analysis techniques for socio-technical systems, failures emanated from human and organizational factors are also considered in addition to technical failures. Human failure taxonomies provide the possible human failures while working in a socio-technical system.

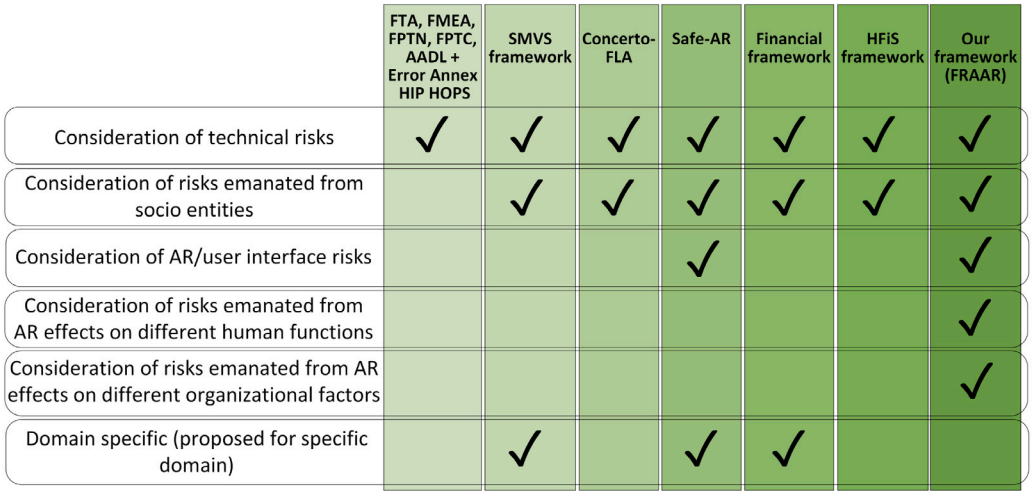


Fig. 18. Comparative analysis summary.

There are also taxonomies on organizational factors that provide the factors influencing human performance. In [14], Concerto-FLA analysis technique is proposed based on SERA taxonomy including human failures and organizational factors. Human reliability quantification techniques can be used for quantifying human error probability and providing quantitative risk assessment. Expert judgment and analysis of accident reports can be used for determining likelihood. However, error likelihood estimation usually has low accuracy. We also do not aim at using quantitative assessment, because based on SOTIF standard, SOTIF related hazards require qualitative analysis.

A risk analysis technique for systems containing augmented reality, named Safe-AR, is proposed in [55]. Safe-AR integrates failures of AR/user interface at three levels: perception, comprehension and decision-making. Likely risks and their severity are based on reports available in literature. The proposed technique is shown on an AR left-turn assist app, which is an example from automotive domain. Human functions and failure modes in this study are limited to the provided example and a generalization is required to be used for other domains and more complicated case studies.

A framework for risk management in financial services is provided in [56]. The paper focuses on risk management from a human centered perspective. In comparison to our work, this paper is specific to financial domain and it does not provide a general framework. The proposed framework does not include modeling and analysis constructs to be used for risk assessment. The required activities in different steps for assessing risk are not defined specifically.

Human Functions in Safety (HFiS) framework is proposed in [57]. This framework focuses on the role of human in system safety in socio-technical systems. Organizational factors are also considered in this framework. The output from applying the framework is a description of safety related activities through human functions, organizational goals and contextual factors. It is developed for railway context, but there are guidance for generic application of HFiS. In comparison to our work, in this paper there is no consideration on effects of new technologies such as augmented reality on human functions and organizational factors.

In comparison to the above-mentioned works, our framework provides more general risk assessment technique with the integration of risk emanated from human, organization and technology (augmented reality). In addition, effects of augmented reality on human functions and organizational factors are considered in our framework. We highlight the features provided by our framework and pre-existing related work in Fig. 18.

8. Conclusion and future work

In this paper, we provided a framework for assessing risk of AR-equipped socio-technical systems. This framework provides the possibility to detect faults and failures leading to system risk and provides the possibility to model and analyze system behavior. In addition, we conducted a case study to illustrate how our proposed framework can be used for predicting risk caused by new AR-related dependability threats. The predicted risk can then be used as a basis for developing e.g., the safety concept in compliance with ISO 26262 and SOTIF related work products.

The framework includes extensions for modeling and analyzing AR effects on human functioning and AR effects on faults leading to human failures. We showed the analysis results by providing three scenarios. In two of the scenarios, the failure was emanated from the AR-related faults. We provided failure propagation manually and we showed that in some scenarios there would be no failure in technical entities of the system, but risk would be identified caused by non-technical AR-related faults. By implementing our proposed conceptual extensions for CHESS toolset, failure propagation calculation can be provided automatically to be used for AR-equipped socio-technical systems.

Our proposed framework supports ISO 26262 and SOTIF development process activities and can be used for providing expected work products by these safety standards. In addition, we discussed that the modeling capabilities within our proposed framework is helpful for determining ISO 26262 controllability. ISO 26262 controllability requires to be updated in order to be used for AR-equipped socio-technical systems, especially in higher levels of driving automation.

Further research is required to show the potential benefits of the proposed framework. Specifically, we intend to conduct case studies where there are scenarios with higher safety criticality. In addition, having two or more teams composed of three or four experienced analysts would help to have more advanced scenarios including more complicated propagation of failures. In future, we also plan to evaluate a safety-critical socio-technical system within the rail industry, the passing of a stop signal (signal passed at danger; SPAD) [58], to verify if the results are transferable to the rail domain.

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.

References

[1] R.T. Azuma, A survey of augmented reality, Presence: Teleoperators Virtual Environ. 6 (4) (1997) 355–385.

[2] ImmerSAFE - Immersive Visual Technologies for Safety-Critical Applications, 2021, URL https://immersafe-itn.eu/.

[3] B.F. Goldiez, N. Saptoka, P. Aedunuthula, Human Performance Assessments when using Augmented Reality for Navigation, Tech. Rep., University of Central Florida Orlando Inst for Simulation and Training, 2006.

[4] D. Van Krevelen, R. Poelman, A survey of augmented reality technologies, applications and limitations, Int. J. Virtual Real. (IJVR) 9 (2) (2010) 1–20, http://dx.doi.org/10.20870/IJVR.2010.9.2.2767.

[5] International Organization for Standardization (ISO), ISO 26262: Road Vehicles — Functional Safety, 2018.

[6] International Organization for Standardization (ISO), ISO/PAS 21448: Road Vehicles — Safety of the Intended Functionality (SOTIF), 2019.

[7] S. Sheikh Bahaei, B. Gallina, Augmented reality-extended humans: towards a taxonomy of failures – focus on visual technologies, in: European Safety and Reliability Conference (ESREL), Research Publishing, Singapore, 2019, http://dx.doi.org/10.3850/978-981-11-2724-3_0922-cd.

[8] S. Sheikh Bahaei, B. Gallina, K. Laumann, M. Rasmussen Skogstad, Effect of augmented reality on faults leading to human failures in socio-technical systems, in: International Conference on System Reliability and Safety, ICSRS, IEEE, 2019, http://dx.doi.org/10.1109/ICSRS48664.2019.8987586.

[9] S. Sheikh Bahaei, B. Gallina, Towards assessing risk of reality augmented safety-critical socio-technical systems, in: Published As Proceedings Annex on the International Symposium on Model-Based Safety and Assessment (IMBSA) Website, 2019, URL http://easyconferences.eu/imbsa2019/proceedings-annex/.

[10] S. Sheikh Bahaei, B. Gallina, Extending safeconcert for modelling augmented reality-equipped socio-technical systems, in: International Conference on System Reliability and Safety, ICSRS, IEEE, 2019, http://dx.doi.org/10.1109/ICSRS48664.2019.8987702.

[11] S. Sheikh Bahaei, B. Gallina, A metamodel extension to capture post normal accidents in ar-equipped socio-technical systems, in: European Safety and Reliability Conference, ESREL, Research Publishing, Singapore, 2021.

[12] L. Montecchi, B. Gallina, SafeConcert: A metamodel for a concerted safety modeling of socio-technical systems, in: International Symposium on Model-Based Safety and Assessment, IMBSA, Springer, 2017, pp. 129–144, http://dx.doi.org/10.1007/978-3-319-64119-5_9.

[13] S. Mazzini, J.M. Favaro, S. Puri, L. Baracchi, CHESS: an open source methodology and toolset for the development of critical systems, in: Join Proceedings of EduSymp and OSS4MDE, 2016, pp. 59–66.

[14] B. Gallina, E. Sefer, A. Refsdal, Towards safety risk assessment of socio-technical systems via failure logic analysis, in: International Symposium on Software Reliability Engineering Workshops, ISSRE, IEEE, 2014, pp. 287–292, http://dx.doi.org/10.1109/ISSREW.2014.49.

[15] M. Wallace, Modular architectural representation and analysis of fault propagation and transformation, in: Proceedings of the Second International Workshop on Formal Foundations of Embedded Software and Component-Based Software Architectures, FESCA, Electron. Notes Theor. Comput. Sci. 141 (3) (2005) 53–71, http://dx.doi.org/10.1016/j.entcs.2005.02.051.

[16] A. Ruiz, A. Melzi, T. Kelly, Systematic application of ISO 26262 on a SEooC: support by applying a systematic reuse approach, in: Design, Automation & Test in Europe Conference & Exhibition, DATE, IEEE, 2015, pp. 393–396, http://dx.doi.org/10.7873/DATE.2015.0177.

[17] L.P. Bressan, A.L. de Oliveira, L. Montecchi, B. Gallina, A systematic process for applying the CHESS methodology in the creation of certifiable evidence, in: 14th European Dependable Computing Conference, EDCC, IEEE, 2018, pp. 49–56, http://dx.doi.org/10.1109/EDCC.2018.00019.

[18] CONCERTO D2.7 – Analysis and Back-Propagation of Properties for Multicore Systems – Final Version, 2016, URL http://www.concerto-project.org/results.

[19] K.C. Hendy, A Tool for Human Factors Accident Investigation, Classification and Risk Management, Tech. Rep., Defence Research And Development Toronto (Canada), 2003.

[20] D.A. Norman, Errors in Human Performance, Tech. Rep., California Univ San Diego LA JOLLA Center For Human Information Processing, 1980.

[21] J. Reason, The Human Contribution: Unsafe Acts, Accidents and Heroic Recoveries, CRC Press, 2017, http://dx.doi.org/10.1016/j.ssci.2009.09.006.

[22] J. Rasmussen, Human errors a taxonomy for describing human malfunction in industrial installations, J. Occup. Accid. 4 (2–4) (1982) 311–333, http://dx.doi.org/10.1016/0376-6349(82)90041-4.

[23] S.A. Shappell, D.A. Wiegmann, The Human Factors Analysis and Classification System–HFACS, Tech. Rep., Civil Aeromedical Institute, 2000, http://dx.doi.org/10.1177/1062860613491623.

[24] N.A. Stanton, P.M. Salmon, Human error taxonomies applied to driving: A generic driver error taxonomy and its implications for intelligent transport systems, Saf. Sci. 47 (2) (2009) 227–237, http://dx.doi.org/10.1016/j.ssci.2008.03.006.

[25] W.-T. Fu, J. Gasper, S.-W. Kim, Effects of an in-car augmented reality system on improving safety of younger and older drivers, in: International Symposium on Mixed and Augmented Reality, ISMAR, IEEE, 2013, pp. 59–66, http://dx.doi.org/10.1109/ISMAR.2013.6671764.

[26] M.C. Schall Jr., M.L. Rusch, J.D. Lee, J.D. Dawson, G. Thomas, N. Aksan, M. Rizzo, Augmented reality cues and elderly driver hazard perception, Hum. Factors 55 (3) (2013) 643–658, http://dx.doi.org/10.1177/0018720812462029.

[27] D. Gertman, H. Blackman, J. Marble, J. Byers, C. Smith, et al., The SPAR-H Human Reliability Analysis Method, vol. 230, US Nuclear Regulatory Commission, 2005.

[28] J.-C. Le Coze, Post Normal Accident: Revisiting Perrow's Classic, CRC Press, 2020, http://dx.doi.org/10.1201/9781003039693.

[29] J. Noll, S. Beecham, Measuring global distance: a survey of distance factors and interventions, in: International Conference on Software Process Improvement and Capability Determination, Springer, 2016, pp. 227–240, http://dx.doi.org/10.1007/978-3-319-38980-6_17.

[30] M.R. Miller, H. Jun, F. Herrera, J.Y. Villa, G. Welch, J.N. Bailenson, Social interaction in augmented reality, PLoS One 14 (5) (2019) e0216290, http://dx.doi.org/10.1371/journal.pone.0216290.

[31] I. Šljivo, B. Gallina, J. Carlson, H. Hansson, S. Puri, A method to generate reusable safety case argument-fragments from compositional safety analysis, J. Syst. Softw. 131 (2017) 570–590, http://dx.doi.org/10.1016/j.jss.2016.07.034.

[32] T. Hecht, M. Lienkamp, C. Wang, et al., Development of a human driver model during highly automated driving for the ASIL controllability classification, in: Tagung Fahrerassistenz, vol. 8, 2017.

[33] I. Sljivo, B. Gallina, J. Carlson, H. Hansson, et al., Using safety contracts to guide the integration of reusable safety elements within iso 26262, in: 21st Pacific Rim International Symposium on Dependable Computing, PRDC, IEEE, 2015, pp. 129–138, http://dx.doi.org/10.1109/PRDC.2015.12.

[34] Taxonomy and Definitions for Terms Related To Driving Automation Systems for on-Road Motor Vehicles, 2021, https://www.sae.org/st{and}ards/content/j3016_202104.

[35] G. Dimitrakopoulos, L. Uden, I. Varlamis, The Future of Intelligent Transport Systems, Elsevier, 2020, http://dx.doi.org/10.1016/C2018-0-02715-2.

[36] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empir. Softw. Eng. 14 (2) (2009) 131, http://dx.doi.org/10.1007/s10664-008-9102-8.

[37] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Trans. Dependable Secure Comput. 1 (1) (2004) 11–33.

[38] F. Ye, T. Kelly, Component failure mitigation according to failure type, in: Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004, IEEE, 2004, pp. 258–264.

[39] D.J. Pumfrey, The Principled Design of Computer System Safety Analyses (Ph.D. thesis), University of York, 1999.

[40] C. Becker, J.C. Brewer, L. Yount, Safety of the Intended Functionality of Lane-Centering and Lane-Changing Maneuvers of a Generic Level 3 Highway Chauffeur System, Tech. Rep., United States. National Highway Traffic Safety Administration, 2020.

[41] S. Patig, Measuring expressiveness in conceptual modeling, in: International Conference on Advanced Information Systems Engineering, CAiSE, Springer, 2004, pp. 127–141, http://dx.doi.org/10.1007/978-3-540-25975-6_11.

[42] S. Shalev-Shwartz, S. Shammah, A. Shashua, On a Formal Model of Safe and Scalable Self-Driving Cars, 2017, https://arxiv.org/pdf/1708.06374.pdf.

[43] Y. Zhang, G. Lintern, L. Gao, Z. Zhang, A Study on Functional Safety, Sotif and Rss from the Perspective of Human-Automation Interaction, Tech. Rep., SAE Technical Paper, 2021, http://dx.doi.org/10.4271/2021-01-0858.

[44] Goal structuring notation community standard, 2018, URL https://scsc.uk/SCSC-141B.

[45] International Organization for Standardization (ISO), ISO/DIS 21448: Road Vehicles — Safety of the Intended Functionality (SOTIF), 2021.

[46] G. Cockton, A. Woolrych, Understanding inspection methods: lessons from an assessment of heuristic evaluation, in: People and Computers XV—Interaction Without Frontiers, Springer, 2001, pp. 171–191, http://dx.doi.org/10.1007/978-1-4471-0353-0_11.

[47] A. Canny, E. Bouzekri, C. Martinie, P. Palanque, Rationalizing the need of architecture-driven testing of interactive systems, in: International Conference on Human-Centred Software Engineering, HCSE, Springer, 2018, pp. 164–186, http://dx.doi.org/10.1007/978-3-030-05909-5_10.

[48] L. Grunske, J. Han, A comparative study into architecture-based safety evaluation methodologies using ADDL's error annex and failure propagation models, in: High Assurance Systems Engineering Symposium, HASE, IEEE, 2008, pp. 283–292, http://dx.doi.org/10.1109/HASE.2008.32.

[49] D.F. Haasl, N.H. Roberts, W.E. Vesely, F.F. Goldberg, Fault Tree Handbook, Tech. Rep., Nuclear Regulatory Commission, 1981, http://dx.doi.org/10.1002/9780470612484.

[50] D.H. Stamatis, Failure Mode and Effect Analysis: FMEA from Theory To Execution, Quality Press, 2003.

[51] P. Fenelon, J.A. McDermid, New directions in software safety: causal modelling as an aid to integration, Workshop on Safety Case Construction, York, Citeseer, 1994.

[52] Y. Papadopoulos, J. McDermid, Safety-Directed System Monitoring using Safety Cases (Ph.D. thesis), Citeseer, 2000.

[53] P. Feiler, A. Rugina, Dependability Modeling with the Architecture Analysis & Design Language (AADL), Tech. Rep., Carnegie-Mellon Univ Pittsburgh PA Software Engineering INST, 2007.

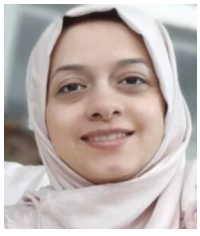
[54] C.-S. Park, H.-J. Kim, A framework for construction safety management and visualization system, Autom. Constr. 33 (2013) 95–103, http://dx.doi.org/10.1016/j.autcon.2012.09.012.

[55] R.R. Lutz, Safe-AR: Reducing risk while augmenting reality, in: 29th International Symposium on Software Reliability Engineering, ISSRE, IEEE, 2018, pp. 70–75, http://dx.doi.org/10.1109/ISSRE.2018.00018.

[56] J. Organ, L. Stapleton, A socio-technical systems framework for risk management in financial services: Some empirical evidence from a case study of the irish banking crisis, IFAC-PapersOnLine 52 (25) (2019) 148–153, http://dx.doi.org/10.1016/j.ifacol.2019.12.463.

[57] B. Ryan, D. Golightly, L. Pickup, S. Reinartz, S. Atkinson, N. Dadashi, Human functions in safety-developing a framework of goals, human functions and safety relevant activities for railway socio-technical systems, Saf. Sci. 140 (2021) 105279, http://dx.doi.org/10.1016/j.ssci.2021.105279.

[58] A. Naweed, J. Trigg, S. Cloete, P. Allan, T. Bentley, Throwing good money after spad? Exploring the cost of signal passed at danger (SPAD) incidents to australasian rail organisations, Saf. Sci. 109 (2018) 157–164, http://dx.doi.org/10.1016/j.ssci.2018.05.018.



Soheila Sheikh Bahaei is a Ph.D. student at Mälardalen University, Västerås, Sweden. She got Bachelor degree in Information Technology Engineering (Isfahan University of Technology (IUT), Iran) and she worked as a student teaching assistant on computer programming courses. She received Master's degree in Artificial Intelligence with a focus on Contextual Image Processing (Kharazmi University of Tehran, Iran). After graduation, she worked as a software developer in a company in Iran and she cooperated in developing a program with the aim of increasing children's creativity.

Her Ph.D. research focuses on dependability analysis of AR-equipped socio-technical systems as part of ImmerSAFE Project. More specifically, she works on extending modeling and analysis techniques of risk assessment to be used for AR-equipped socio-technical systems. She has defended her licentiate on September 2020 with the title "A Framework for Risk Assessment in AR-equipped Socio-technical Systems". Her main supervisor is Barbara Gallina.

Barbara Gallina is Associate Professor of Dependable Software Engineering at Mälardalen University, Västerås, Sweden, where she leads the Certifiable Evidences and Justification Engineering group. She holds a M.Sc. in Computer Science (Politecnico di Milano, Italy, 2003), a II-level Master in ICT (Politecnico di Milano/Cefriel, Italy, 2004), and a Ph.D. in Computer Science (University of Luxembourg, Luxembourg, 2010).

Her research focuses on developing languages, techniques, metrics, and processes for engineering evidence(s) and justifications for the purpose of certification/self-assessment of complex dependable (computer-based) systems. Currently, she is and has been involved as organizer, co-organizer, co-chair, and program committee member of relevant conferences (such as ISSRE, SafeComp, EDCC, AdaEurope, RSS-Rail, and PRDC) and workshops (such as WoSoCER, ISSA, and DeCPS) in dependability engineering. Dr. Gallina is the author of over 100 articles in the area of dependable software engineering and certification.

Marko Vidović is Advanced Driver Assistance Systems (ADAS) director in Xylon d.o.o. where he leads development teams on various ADAS related projects. He holds Master degree in Industrial Electronics from the University of Electrical Engineering and Computing in Zagreb. As an active developer and project leader Marko has successfully completed many complex embedded electronics projects, custom SoC chip architectures and various IP cores implementations in the FPGA and programmable SoC chips. Today he is focused on image processing algorithms, camera extrinsic and intrinsic parameters calibration, FPGA implementation and acceleration of image processing and various advanced algorithms for use in automotive applications. Marko holds joint patent on method for calibration of plurality of cameras for vehicle surround view systems.