A Case Study for Risk Assessment in AR-equipped Socio-technical Systems*

Soheila Sheikh Bahaei[1], Barbara Gallina[1], and Marko Vidović[2]

[1] School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
{soheila.sheikhbahaei, barbara.gallina}@mdh.se
[2] Xylon Electronics Company, Zagreb, Croatia
Marko.Vidovic@logicbricks.com

Abstract. Augmented Reality (AR) technologies are used as human-machine interface within various types of safety-critical systems. In order to avoid unreasonable risk, it is required to anticipate new types of dependability threats (faults, errors, failures), which could be introduced within the systems by these technologies. In our previous work, we have designed an extension for CHESS framework to capture AR-related dependability threats (focusing on faults and failures) and we have extended its metamodel, which provides qualitative modeling and analysis capabilities that can be used for AR-equipped socio-technical systems. In this paper, we conduct a case study from automotive domain to present modeling and analysis capabilities of our proposed extensions. We conduct qualitative modeling and analysis based on Concerto-FLA analysis technique, which is an analysis technique for socio-technical systems to find out if the proposed extensions would be helpful in capturing new system failures caused by AR-related dependability threats.

Keywords: Risk Assessment · Augmented Reality · Socio-tecnical Systems.

1 Introduction

Augmented Reality (AR) technology is used for superimposing virtual and computer generated information on the reality of the user [7]. This information would be visual, auditory, etc., for enhancing human capabilities [30]. An example of visual augmented reality is using navigational information superimposed on the windshield of a car for driver guidance.

Utilizing augmented reality technology in socio-technical systems demands analysis to make sure that it is not harmful for people and the environment, while interacting with humans. Thus, it is required to identify the threats and their propagation via modeling the system and analyzing its behavior in order to enable risk analysis of systems containing augmented reality.

* This work is funded by EU H2020 MSC-ITN grant agreement No 764951.

According to ISO 26262 [9] standard, which is related to automotive domain, 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". In order to identify AR-related hazardous events or dependability threats, which are risk sources, we have proposed two taxonomies in our previous works. Based on these taxonomies extensions are provided to investigate AR-related dependability threats in architecture modeling and analysis. So far, however, there has been little investigation about how effective are current modeling and analysis techniques for industrial systems containing new technologies and if it is possible to capture new risks caused by augmented reality.

In this paper, we use an industrial case study for evaluating our proposed conceptual extensions on CHESS framework for capturing AR-related dependability threats in AR-equipped socio-technical systems. Conceptual extensions are mostly associated with SafeConcert metamodel [12], which is part of the modeling language included in the CHESS framework for modelling socio-technical systems. Extended metamodel provides modeling and analysis capabilities. In order to show the analysis capabilities of the proposed extensions, we use Concerto-FLA [5], which is an analysis technique for socio-technical systems. Concerto-FLA uses Fault Propagation and Transformation Calculus (FPTC) [31] syntax to provide the means for analysis in system level. We present the case study based on SEooC (Safety element out of context) concept of ISO 26262 standard, which 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 [18]. Finally, we provide a discussion related to threats to validity and limitations and benefits of the extensions.

The rest of the paper is organized as follows. In Section 2 we provide essential background information. In Section 3, we design and conduct the case study to evaluate modeling and analysis capabilities of the proposed extensions. In Section 4, we discuss about threats to validity and limitations of our research. Finally, in Section 5, 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 modelling technique and AR-related modeling extensions are presented. Concerto-FLA analysis technique is also explained. Finally, SEooC concept of ISO 26262 is presented.

2.1 CHESS Framework

CHESS framework [10] provides a methodology and toolset for developing high-integrity systems. 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 [2]. Then, developers can use the analysis and back propagate the results iteratively. CHESS methodology contains CHESS-ML (CHESS Modeling Language) [1] based on UML and a set of plugins for code generation and providing various analysis capabilities. Plugins related to analysis are Failure Logic Analysis (FLA) and State-Based Analysis (SBA). For executing 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 contrast, SBA allows quantitative analysis using quantitative dependability information such as probability. In this paper, our focus is on failure logic analysis and we consider Concerto-FLA as the analysis technique used in this toolset. Concerto-FLA is based on Fault Propagation and Transformation Calculus (FPTC) [31] syntax. We have also proposed extensions in CHESS-ML through extending SafeConcert, which is part of this modeling language. Details about our extensions and Concerto-FlA technique are provided in the next sections.

2.2 SafeConcert and its Extension of AR

SafeConcert [12] is a metamodel for modeling socio and technical entities in socio-technical systems. This metamodel is part of CHESS-ML modeling language [1], which is a UML-based modeling language. In SafeConcert metamodel, software, hardware or socio entities can be modelled as components in component-based systems representing socio-technical systems. SERA taxonomy [8] is used for modeling human and organization, which are the socio entities of the system. In this metamodel human sub-components are modelled based on twelve categories of human failures including failures in perception, decision, response, etc.

In [24], we extended human modeling elements based on AREXTax, which is an AR-extended human function taxonomy [22]. These extended modeling elements are shown in Fig. 1. Human functions are divided to three categories including human process unit, human SA unit, and human actuator unit. Human fault unit are related to human internal influencing factors on human function. This part will be explained in next paragraph. Extended modeling elements are shown with white color and AR-stemmed modeling elements are shown with dotted line border. These extended modeling elements enable modeling of AR-extended human functions. For example, detection failure is a human failure introduced by several human failure taxonomies such as Reason [17] and Rasmussen[16] taxonomies, which is failure in detecting human function. Based on experiments and studies on augmented reality including [4] and [20], 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, which is an extended modeling element proposed for analysis of AR-equipped socio-technical systems.

In [23], we extended organization modeling elements based on AREFTax, which is a fault taxonomy including AR-caused faults [25]. These extended mod-

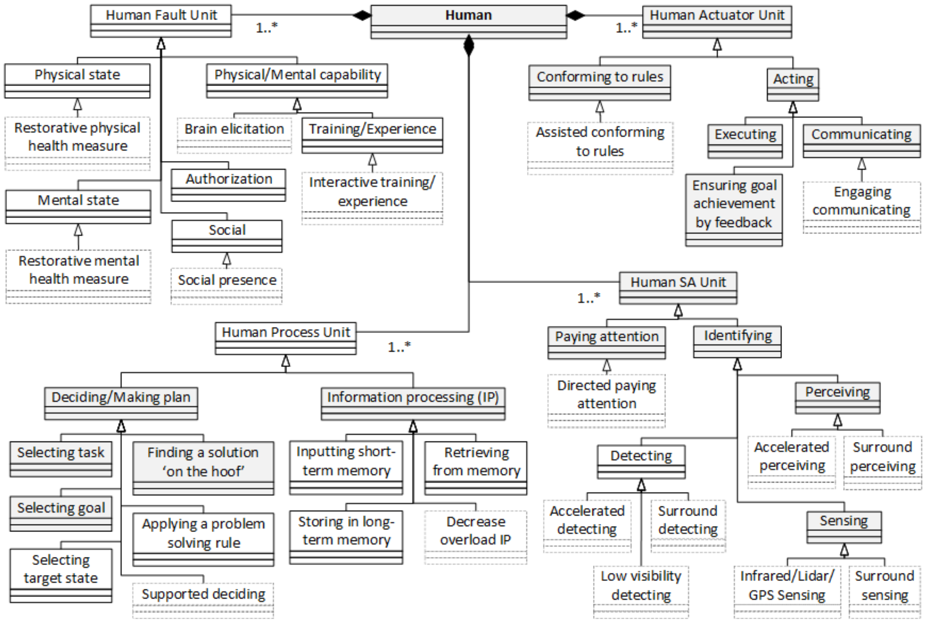


Fig. 1: Extended modeling elements for human components [23].

eling elements are shown in Fig. 2 and human fault unit of Fig. 1. Extended modeling elements are shown with white color and AR-stemmed modeling elements are shown with dotted line border. These extended modeling elements enable modeling of AR-caused faults leading to human failures. Faults would be caused by human, environment, organization, etc. Human related faults are categorized in 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 fault leading to human failure, so it is categorized in organization taxonomy 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 [11], using AR would decrease social presence and failure in social presence can be considered as fault leading to human failure.

2.3 Modeling Failure Behavior based on FPTC Syntax

Fault Propagation and Transformation Calculus (FPTC) [31] syntax is proposed in FPTC dependability analysis technique. This syntax is used by several methods such as Concerto-FLA, because it provides the possibility for calculating the behavior at system level based on behavior of individual components. FPTC rules are set of logical expressions that relate output failure modes to combinations of input failure modes in each individual component [26].

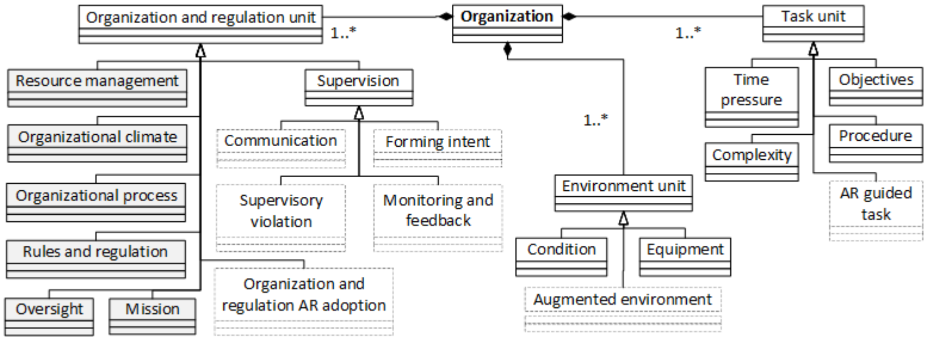


Fig. 2: Extended modeling elements for organization components [23].

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 can not 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 [5] 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 [8] classification of socio-failures. As we explained in Sub-section 2.1, this approach 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 method includes five main steps.

1. Modeling architectural elements including software, hardware, human, organization, connectors, interfaces and etc.
2. Using FPTC syntactical rules to model failure behavior at component and connector level. Concerto-FLA has adopted the FPTC syntax for modeling failure behavior at component and connector level (explained in Subsection 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 SEooC in ISO 26262

ISO 26262 standards [9] provide the requirements and set of activities that should be performed during the lifecycle phases such as development, production, operation, service and decommissioning. Integrity level or ASIL (Automotive Safety Integrity Levels) are 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.

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 [27].

SEooC system 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 in this step.
   (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 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 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.

Based on the taxonomy and definitions related to driving automation systems for on-road motor vehicles performing part or all of the 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 [29]. 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 [3]. In order to improve safety, various scenarios of driver/vehicle interaction should be considered.

Safety process of the ISO 26262 standard, shown in Fig. 3 , starts with *concept phase* containing *item definition*, *hazard analysis and risk assessment* and *functional safety concept* [27]. An *item* implements a vehicle level function. In *item* definition the main objective is defining items, which 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* sent to the *product development phase. Product development phase* contains *system level* and *HW/SW level*. Functional safety concept is used to provide technical safety requirements and to design system in *product development phase* at system level. Then, hardware and software development and testing is done based on system design. HW/SW safety requirements are based on assumptions provided in concept phase. Next step in the process is integration and testing of HW/SW elements and then in system level integration of elements that compose an item, safety validation and functional safety assessment are done, which requires establishing validity of assumptions. Finally, the last step is production and operation.

3 Case Study Design

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

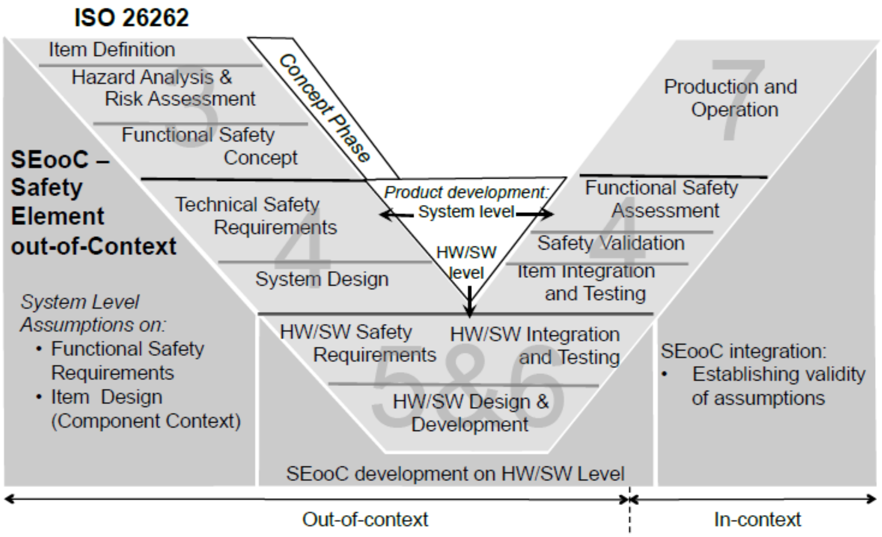

ISO 26262
Item Definition
Hazard Analysis & Risk Assessment
Functional Safety Concept
Concept Phase
SEooC – Safety Element out-of-Context
Technical Safety Requirements
System Design
Product development:
System level
HW/SW level
System Level Assumptions on:
• Functional Safety Requirements
• Item Design (Component Context)
HW/SW Safety Requirements
HW/SW Integration and Testing
HW/SW Design & Development
SEooC development on HW/SW Level
Production and Operation
Functional Safety Assessment
Safety Validation
Item Integration and Testing
SEooC integration:
• Establishing validity of assumptions
3
4
5&6
4
7
Out-of-context
In-context

Fig. 3: Projection of the ISO 26262 lifecycle activities to SEooC development and integration process [27].

Projection of the risk assessment activities to the ISO 26262 development process is shown in Fig. 4. There are four main steps. The first step is to define composite components of the system. In order to find composite components, we need to answer to the question of what are the involved entities. Second step is to determine sub-components of each composite component. In order to determine sub-components, we need to identify different effective aspects of each entity. In this step, our proposed taxonomies and extended modeling elements can be helpful to provide a list of effective aspects and based on scenario and the selected case study, required sub-components can be selected. Third step is to model the behavior of each sub-component, which should be done based on analysis of each sub-component individually. In order to model each sub-component behavior, effect of related aspect to the sub-component's behavior should be identified. Finally, last step is analyzing system behavior, which provides effect of various aspects on the system.

3.1 Objectives

Our objectives include presenting the modeling capabilities and analysis capabilities of our proposed AR-related extensions in order to estimate how effective they are in predicting new kinds of risks caused by AR-related factors. In order to do that, we use an industrial case study from automotive domain to evaluate the proposed extensions.

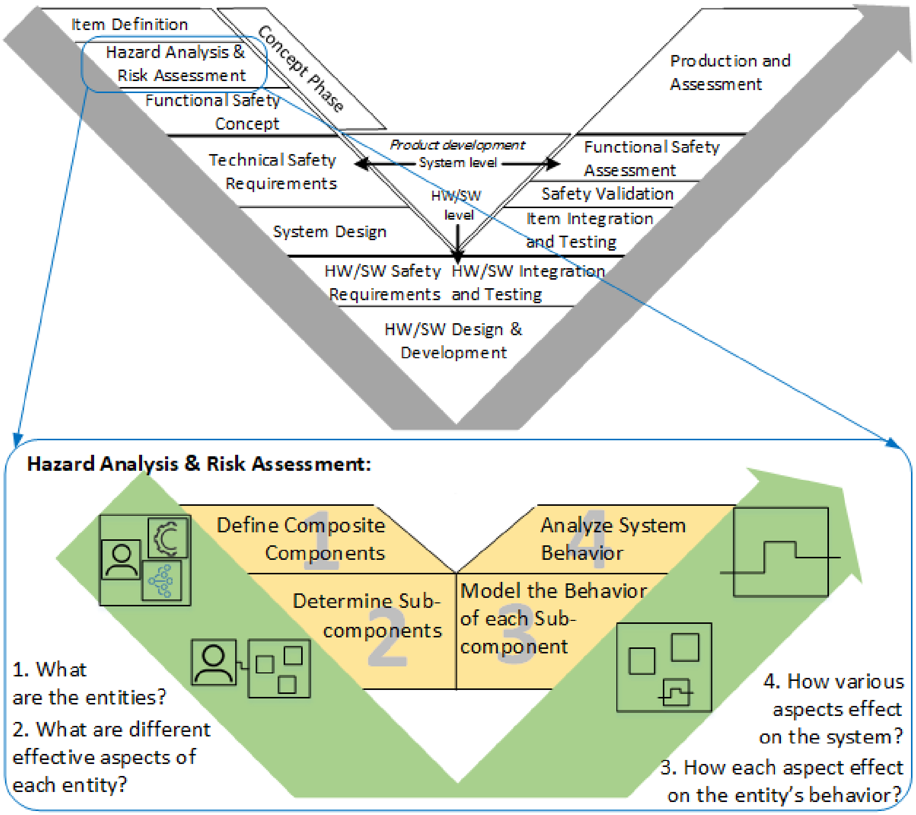

Item Definition
Hazard Analysis & Risk Assessment
Functional Safety Concept
Concept Phase
Technical Safety Requirements
System Design
HW/SW Safety Requirements
HW/SW Design & Development
Product development
System level
HW/SW level
HW/SW Integration and Testing
Item Integration and Testing
Safety Validation
Functional Safety Assessment
Production and Assessment
Hazard Analysis & Risk Assessment:
Define Composite Components
1
Determine Sub-components
2
Model the Behavior of each Sub-component
3
Analyze System Behavior
4
1. What are the entities?
2. What are different effective aspects of each entity?
4. How various aspects effect on the system?
3. How each aspect effect on the entity's behavior?

Fig. 4: Projection of the risk assessment activities to ISO 26262 development process.

3.2 Research Methodology

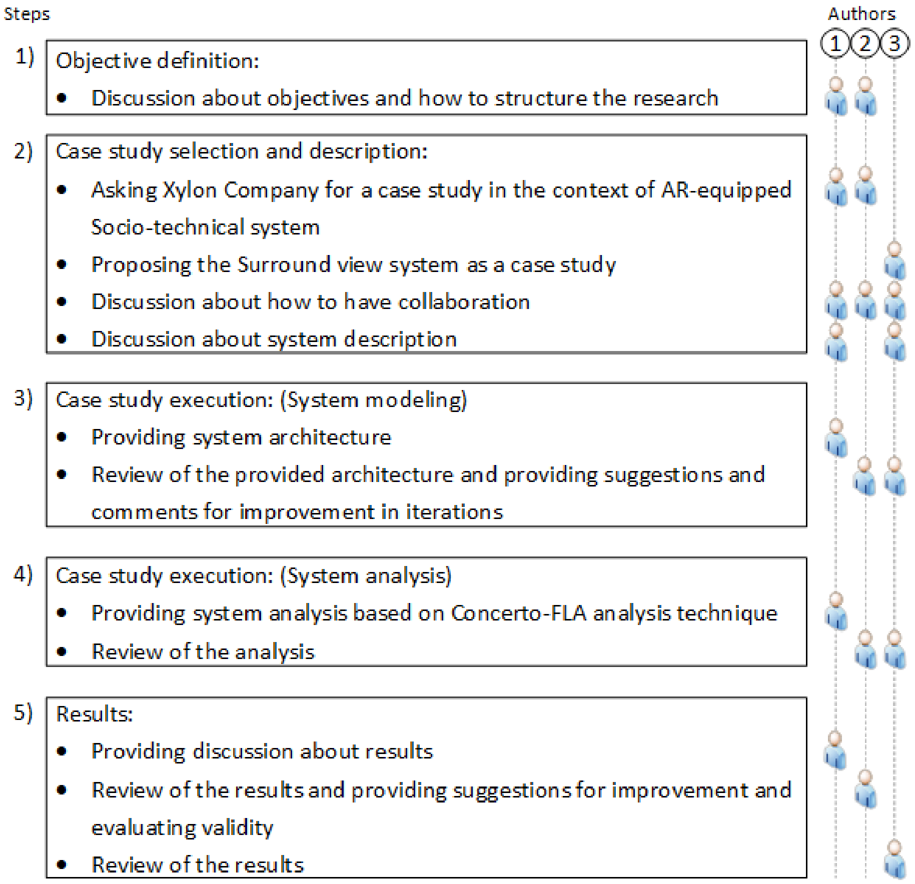
The steps carried out for the presented research is presented in Fig 5. In the first step, the first and second authors discussed about objectives and the structure of the research.

In the second step, the first and second authors asked from Xylon Company for a case study in the context of augmented reality socio-technical systems and third author suggested surround view system as a case study and a meeting was organized between three authors to decide about the collaboration. First and third authors also discussed about system description.

In the third step, system architecture was provided by the first author and it was reviewed by third author in some iterations for improvement. Second author also reviewed the architecture and provided comments and suggestions for improvement.

In the fourth step, analysis of the case study was provided by the first author based on Concerto-FLA analysis technique and it was reviewed by the second and third authors.

In the fifth step, the first author provided discussion about results and second and third authors reviewed the results. Second author also provided suggestions for improvement and for discussing about validity of the work.



1) Objective definition:
- Discussion about objectives and how to structure the research

2) Case study selection and description:
- Asking Xylon Company for a case study in the context of AR-equipped Socio-technical system
- Proposing the Surround view system as a case study
- Discussion about how to have collaboration
- Discussion about system description

3) Case study execution: (System modeling)
- Providing system architecture
- Review of the provided architecture and providing suggestions and comments for improvement in iterations

4) Case study execution: (System analysis)
- Providing system analysis based on Concerto-FLA analysis technique
- Review of the analysis

5) Results:
- Providing discussion about results
- Review of the results and providing suggestions for improvement and evaluating validity
- Review of the results

Fig. 5: Steps taken for the carried out research.

3.3 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)

- Driver (socio entity)
- Surround view system (a SEooC that includes augmented reality technology used to empower drivers).

Road transport organization and driver are two socio-entities of this system that we aim to use our extended modeling elements for modeling different aspects of their behaviors.

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. 6, it is illustrated how the 3D video is shown to the driver. As it is shown in Fig. 6, 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 not included in driving automation systems, because it does not perform part or all of the DDT on a sustained basis. However, Xylon plan to develop automated driving system features for the future versions of the system.

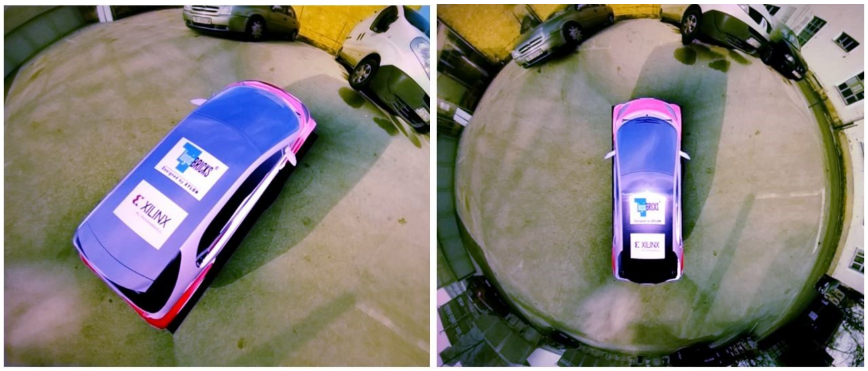


Fig. 6: Sample images from 3D videos provided in surround view system.

Surround view system as a SEooC includes:

- 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 software 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 software for reading prepared data structures and video from memory and 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 software for reading memory via DDR memory controller 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 IPs by means of their software drivers. Video processing engine implementation prepares data structures to be stored in DDR memory through DDR controller.

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:

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

3.4 Case Study Execution: System Modelling

This sub-section reports how we model the described system in Sub-section 3.3 using our proposed extensions.

Sub-section 3.3 provides the required information for the first step of the risk assessment process defined in Fig. 4, which is identifying the entities for defining composite components. Based on the selected case study explained in Sub-section 3.3, automotive surround view system, organization and driver are three composite components of this system. In this sub-section we provide information for the second and third steps of risk assessment process. In order to find effective aspects of each entity and determine sub-components of each composite component, AREXTax and AREFTax explained in Sub-section 2.2, can be used. Based on the case study, surround detecting, supported deciding and executing selected from AREXTax are three effective aspects in this entity. Surround detecting and supported deciding, which are two AR-extended human functions effecting on executing human function, are used from AREXTax and we consider them as sub-components of human component. Surround detecting is an AR-extended function, because driver can detect surround environment through AR technology. Interactive experience and social presence are also two effective aspects that effect on human functions selected from AREFTax. Fig. 7 provides an overview of the integration of the human functions and influencing factors with SEooC.

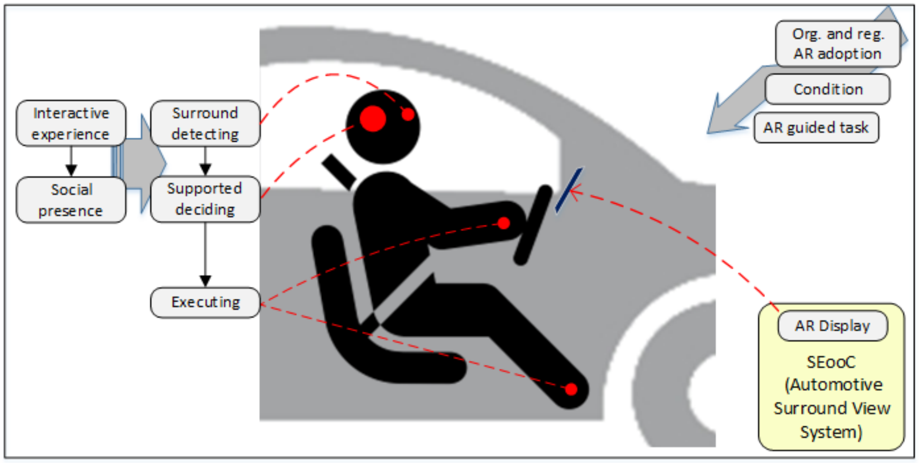


Fig. 7: Integration of the human functions and influencing factors with SEooC.

Effective aspects of organization entity in this case study by considering AREFTax, include organization and regulation AR adoption, condition and AR guided task. Organization and regulation AR adoption refers to upgrading of rules and regulations of road transport organization based on AR technology. Condition refers to road condition. AR guided task refers to the task, which AR is used for guiding driver to do that. For example, if AR is used to guide driver to park the car more safely, parking safely is the AR-guided task.

Effective aspects or sub-components of automotive surround view system as a SEooC can be determined based on the provided description in Sub-section 3.3.

In Fig. 8, we show how this AR-equipped socio-technical system is modeled. Driver is composed of five sub-components. Driver has four inputs and one of its inputs is from system input with the name human detection input (HDI). Two other inputs are from organization and surround view system and the last input is human communication input (HCI). We consider also interactive experience and social presence as two sub-components of human component, which are influencing factors on human functions. Interactive experience effects on supported deciding and is effected by surround detecting. Social presence receives input from system with the name human communication input (HCI) and effects on human executing. Driver output, which is output of the system is human action shown by HA.

Organization and regulation AR adoption, condition and AR guided task are three sub-components of organization composite component. Organization component receives input from system, which represents influences from regulation authorities on the organization (REG). Human and organization and their relation with surround view system are modelled in Fig. 8. Gray color is used to show the extended modelling elements used in this system.

Automotive surround view system is also modelled based on description provided in Sub-section 3.2. Three inputs of this system are user command shown by CMD, speed shown by SPD and camera input shown by CAM.

3.5 Case Study Execution: System Analysis

This sub-section reports about the analysis of the system using our proposed extensions, which refers to the last step of the risk assessment process defined in Fig. 4. We use the five steps of Concerto-FLA analysis technique explained in sub-section 2.4 for system analysis.

1. First step is provided in Fig. 8. We explained how the system is modeled in Sub-section 3.4.
2. Second step is shown by providing FPTC rules, which is used for linking possible failure inputs of each component to failure outputs. (These rules for modeled sub-components are shown in Table.1-4)
3. Third step is to consider possible failures in inputs of the system to evaluate failure propagation. In this example, 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.
4. Fourth step is calculating the failure propagations. We consider three scenarios and show the analysis results in Fig. 9 - 11.
5. Last step is back propagation of results (Shown in Fig. 12). Interpretation of the back-propagated results can be used to make decision about design change or defining safety barrier, if it is required.

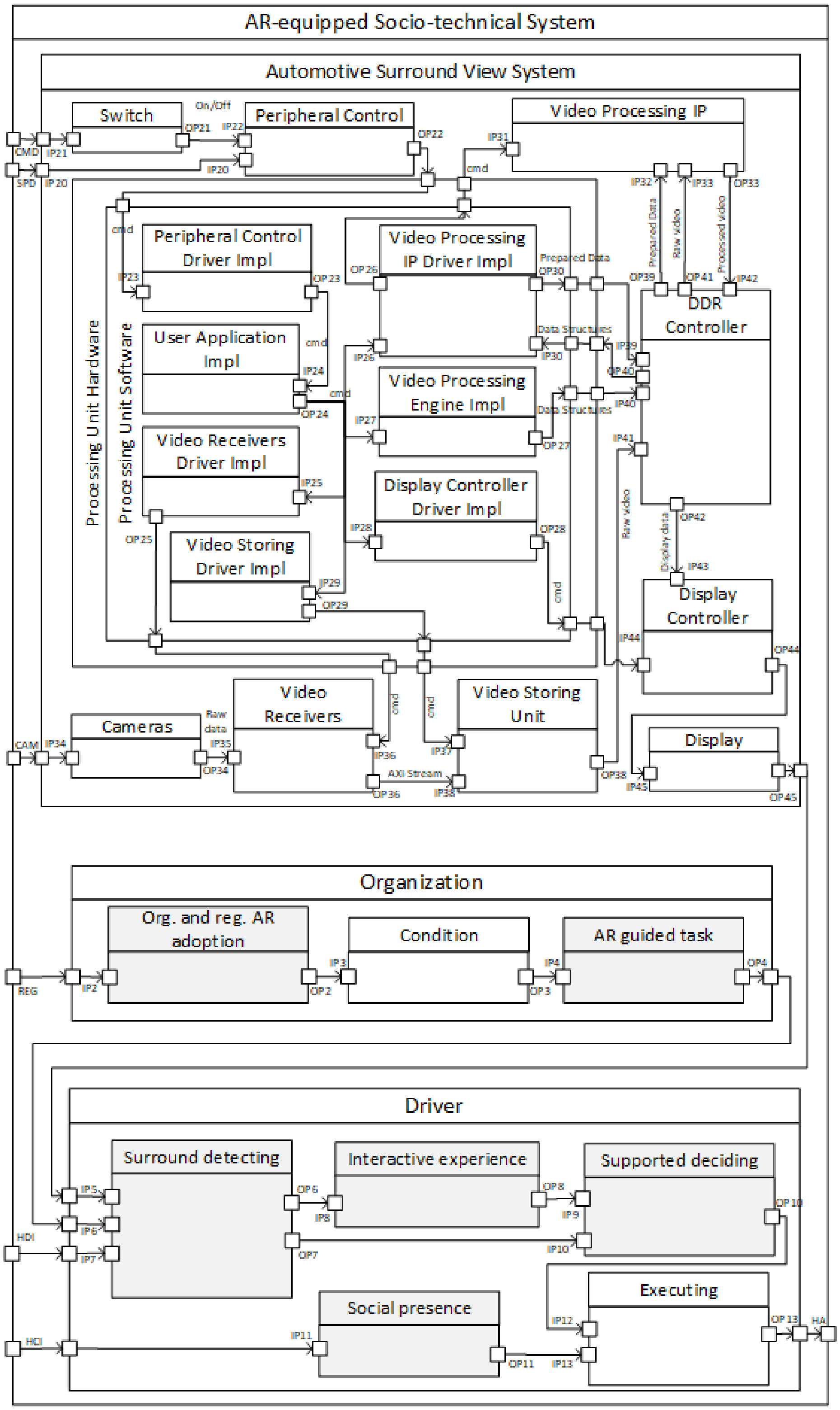


Fig. 8: AR-equipped socio-technical System Modelling.

Table 1: Modeling failure behavior of components

Name of the component	Possible input failures	Possible output failures	FPTC rules
Switch	IP21: late, omission, commission, value	OP21: late, commission, omission	IP21.variable → OP21.variable;
Camera	IP34: omission, value	OP34: Omission, value	IP34.variable → OP34.variable;

Video Receiver	IP35: Late, omission, value-Subtle IP36: late, omission, commission, valueSubtle	OP36: late, omission, value-Subtle, 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	IP38 : Late, omission, value-Subtle IP37: late, omission, commission, valueSubtle	OP38 : late, omission, value-Subtle, 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;

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;

Scenario 1:

In this scenario, we assume that the road transport organization has not updated rules and regulations based on AR technology. So this component will produce an omission failure. We show the failure propagation with underlined FPTC rules, which are the rules that are activated, shown in Fig. 9. In this 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 in its output. This activated rule is shown on this component. Omission failure propagates through condition and AR guided task and in surround detecting it transforms to valueSubtle. The reason for this transformation is that omission failure in IP6 means that AR guided task is not defined by organization. This means that surround detecting would be done incorrectly because its input is not provided and this leads to valueSubtle failure in its output. ValueSubtle propagates to interactive experience and supported deciding and transforms to valueCoarse in executing. The reason for this transformation is that if there is value failure in executing function it can be detected by user, which means valueSubtle transforms to valueCoarse.

Based on back propagation of the results, shown in Fig. 12, we can explain how the rules have been triggered. ValueCoarse on OP13 is because of valueSubtle on IP12 and noFailure on OP11. 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 input IP2 that is organization and regulation AR adoption. In this case a solution would be an instruction for organization and regulation to update their rules and regulations based on AR technology. Then, the failure behavior will be updated and failure propagation analysis can be repeated to see the results.

It is not possible to detect risks originated from failure in updating rules and regulations based on AR technology, without using the proposed representation means, because using these representation means or modeling elements provide the possibility to analyze their failure propagation 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.

Scenario 2:

In this scenario, we assume that driver doesn't have interactive experience. So this component will produce a valueSubtle failure. We show the failure propagation with underlined FPTC rules, which are the rules that are activated, shown in Fig. 10. Similar to the first scenario, surround view sub-components behave

Table 2: Modeling failure behavior of components (Cont.)

Name of the component	Possible input failures	Possible output failures	FPTC rules
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 Storing Driver Imp	IP29: Late, omission, valueSubtle, commission	OP29: Late, omission, commission, valueSubtle	IP29.variable → OP29.variable;
Video Processing Engine Imp	IP27: Late, omission, valueSubtle	OP27: Late, omission, valueSubtle	IP27.variable → OP27.variable;
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;
Display Controller Driver Imp	IP28: Late, omission, valueSubtle, commission	OP28: Late, omission, commission, valueSubtle	IP28.variable → OP28.variable;
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;

Table 3: Modeling failure behavior of components(Cont.)

Name of the component	Possible input failures	Possible output failures	FPTC rules
Condition	IP3: Late, omission, value-Subtle, value-Coarse	OP3: Late, omission, value-Subtle, value-Coarsee	IP3.variable → OP3.variable;
AR guided task	IP4: Late, omission, value-Subtle, value-Coarse	OP4: Late, omission, value-Subtle, value-Coarse	IP4.variable → OP4.variable;
Video Processing IP driver Imp	IP30: Late, omission, valueSubtle IP26: Late, omission, commission	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;
Social presence	IP11: Late, omission, valueSubtle	OP11: Late, omission, valueSubtle	IP11.noFailure → OP11.noFailure; IP11.late → OP11.late; IP11.valueSubtle → OP11.valueSubtle; IP11.omission → OP11.omission;
Interactive experience	IP8: Late, omission, valueSubtle,	OP8: Late, omission, valueSubtle	IP8.noFailure → OP8.noFailure; IP8.late → OP8.late; IP8.valueSubtle → OP8.valueSubtle; IP8.omission → OP8.omission;
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;

Table 4: Modeling failure behavior of components(Cont.)

Name of the component	Possible input failures	Possible output failures	FPTC rules
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;
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.value, OP7.value; 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;

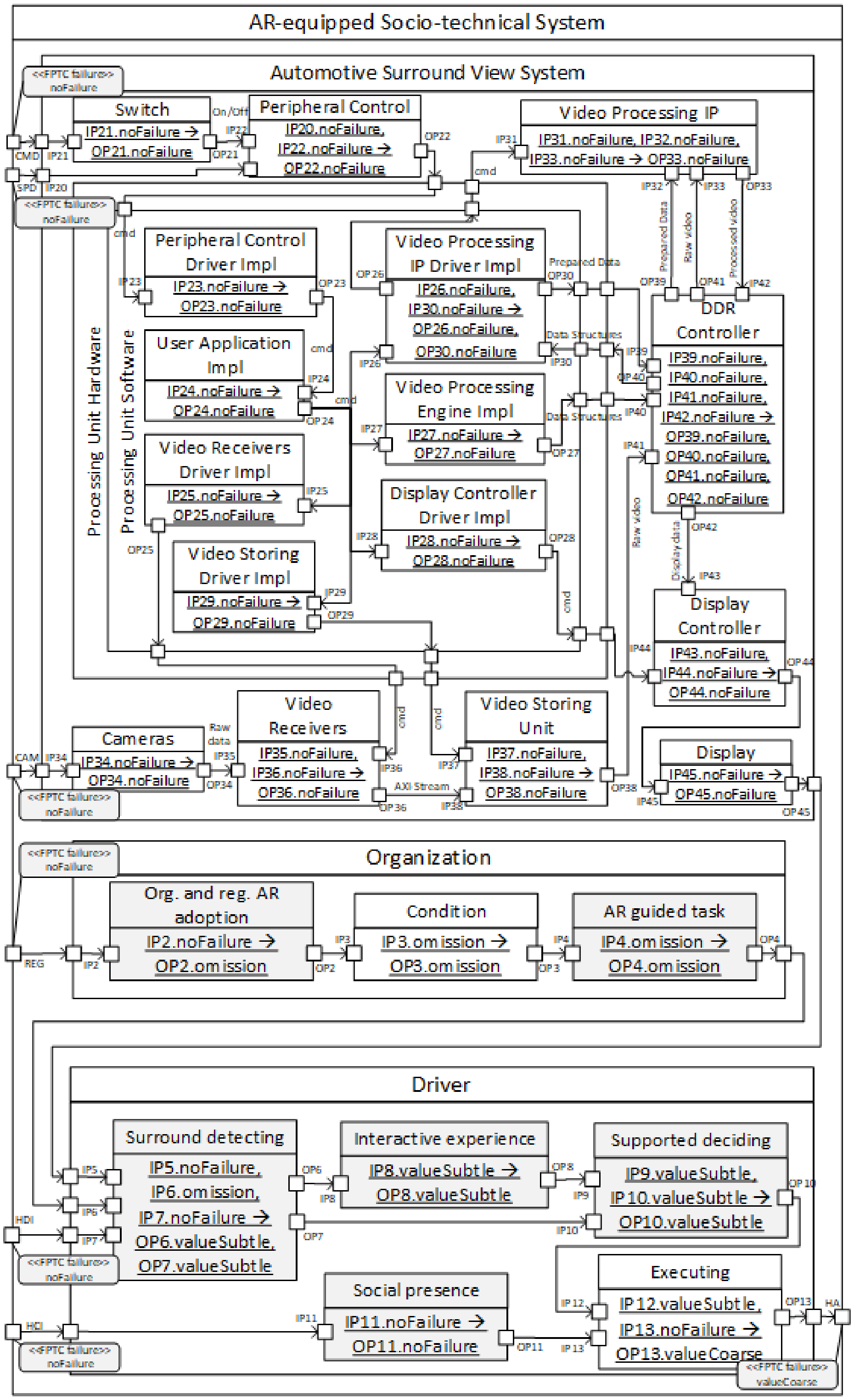


Fig. 9: Analyzing AR-equipped socio-technical system (Scenario1).

as propagational and propagate noFailure from inputs to output. Interactive experience behaves as source and while its input is noFailure, it has valueSubtle failure in its output. This activated rule is shown on this component. ValueSubtle failure propagates through supported deciding and in executing it transforms to valueCoarse. Similar to the first scenario, the reason for this transformation is that if there is value failure in executing function it can be detected by user, which means valueSubtle transforms to valueCoarse.

Based on back propagation of the results, shown in Fig. 12, we can explain how the rules have been triggered. ValueCoarse on OP13 is because of valueSubtle on IP2 and noFailure on OP11. ValueSubtle on IP12 is because of valueSubtle on OP10 and we continue to IP8, which is related to interactive experience component. In this case a solution would be to suggest that the company provide a training video for all drivers at the first time of using the system. This would change the behavior type of this component from source to other types and analysis can be repeated.

It is not possible to detect risks 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 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.

Scenario 3:

In this scenario, we assume that AR guided task is not defined well. So this component will produce a valueSubtle failure. We show the failure propagation with underlined FPTC rules, which are the rules that are activated, shown in Fig. 11. Similar to the previous scenarios, surround view sub-components behave as propagational and propagate noFailure from inputs to output. AR guided task behaves as source and while its input is noFailure, it has valueSubtle failure in its output. This activated rule is shown on this component. ValueSubtle failure propagates through surround detecting, interactive experience and supported deciding and in executing it transforms to valueCoarse.

Based on back propagation of the results, shown in Fig. 12, we can explain how the rules have been triggered. ValueCoarse on OP13 is originated from component with input IP4, which is AR guided task component. In this case a solution would be to decrease the complexity of the task which AR is used for its guidance. For example dividing the task to sub-tasks decreases the complexity, which requires changes on AR design. After accomplishing the changes, modeling failure behavior should be provided to be used again in analysis.

It is not possible to detect risks originated from failure in AR guided task, without using the proposed representation means, because using these representation means or modeling elements provide the possibility to analyze their failure propagation 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.

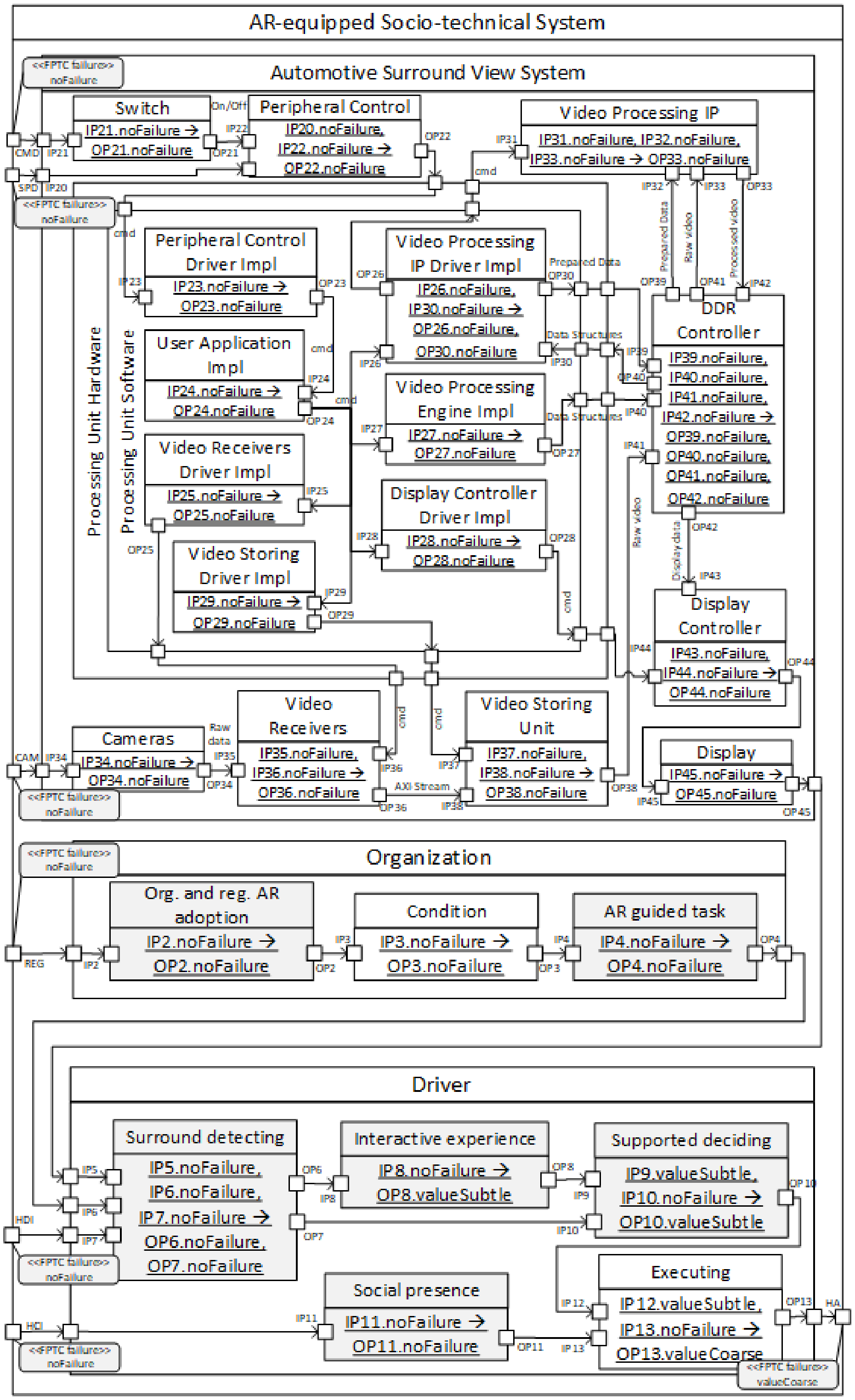


Fig. 10: Analyzing AR-equipped socio-technical system (Scenario2).

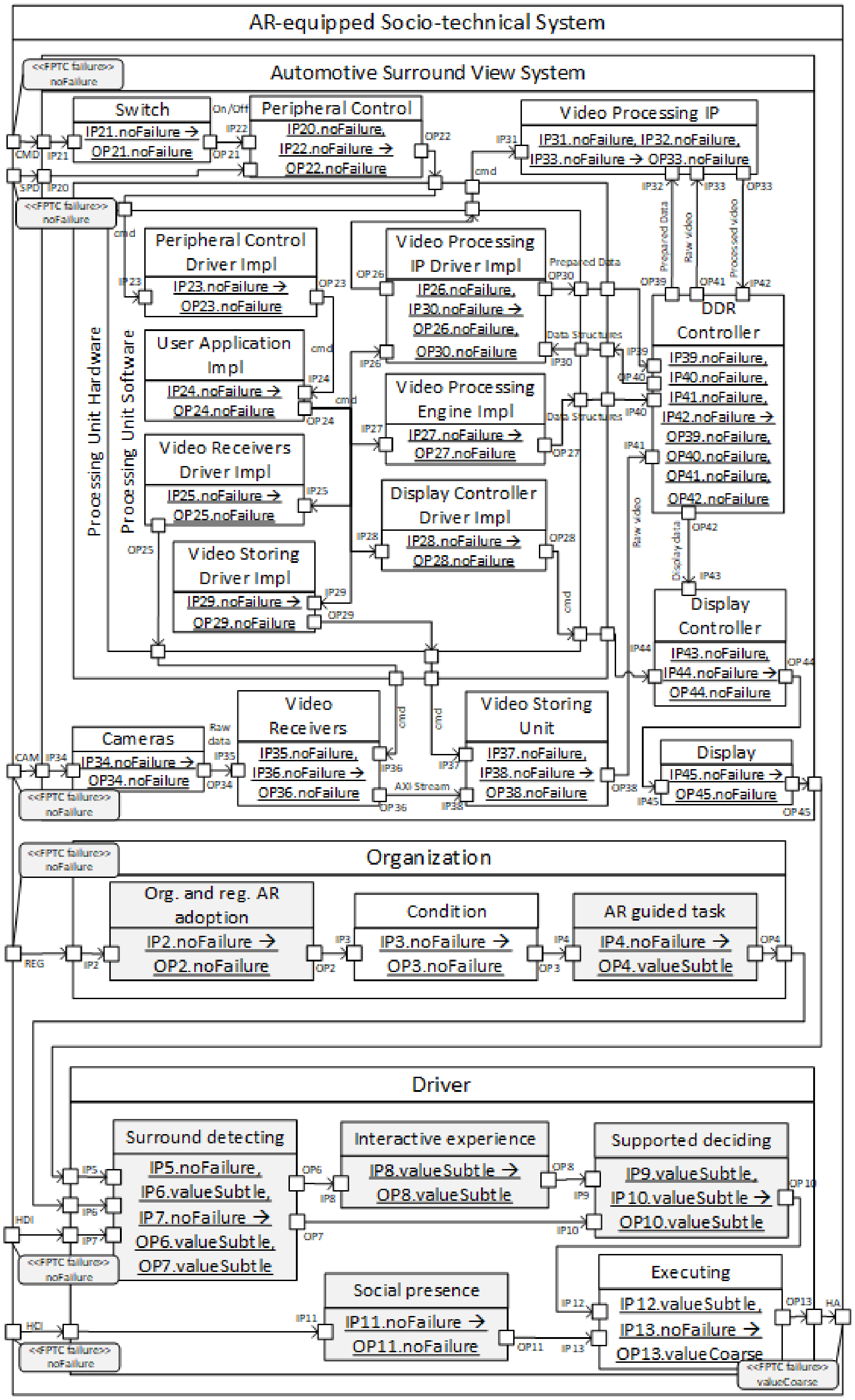
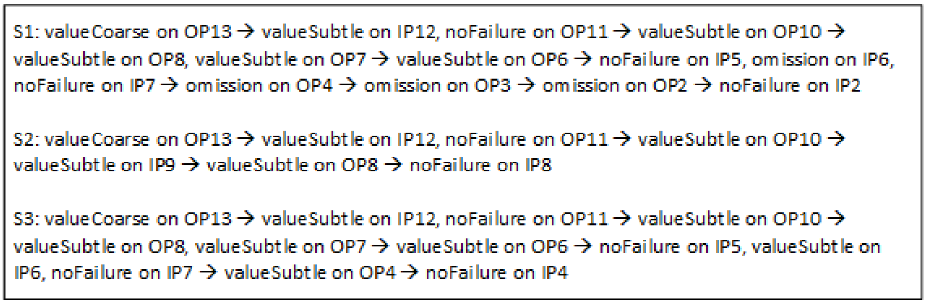


Fig. 11: Analyzing AR-equipped socio-technical system (Scenario3).



S1: valueCoarse on OP13 → valueSubtle on IP12, noFailure on OP11 → valueSubtle on OP10 → valueSubtle on OP8, valueSubtle on OP7 → valueSubtle on OP6 → noFailure on IP5, omission on IP6, noFailure on IP7 → omission on OP4 → omission on OP3 → omission on OP2 → noFailure on IP2

S2: valueCoarse on OP13 → valueSubtle on IP12, noFailure on OP11 → valueSubtle on OP10 → valueSubtle on IP9 → valueSubtle on OP8 → noFailure on IP8

S3: valueCoarse on OP13 → valueSubtle on IP12, noFailure on OP11 → valueSubtle on OP10 → valueSubtle on OP8, valueSubtle on OP7 → valueSubtle on OP6 → noFailure on IP5, valueSubtle on IP6, noFailure on IP7 → valueSubtle on OP4 → noFailure on IP4

Fig. 12: Back propagation of the results.

3.6 Lessons Learnt

In this section, we present the lessons learned that we have derived by manually applying Concerto-FLA analysis considering our proposed extensions for CHESS framework for an AR-equipped socio-technical system. The lessons are as follows:

- **Augmented reality concepts coverage**: from a coverage point of view, modeling and analysis capabilities obtained by our proposed extensions, allow architects and safety managers to model augmented reality effects on socio-technical systems that might be effective in emerging risks within an AR-equipped socio-technical system. As it is shown in this case study, by using modeling elements related to AR-extended human functions as well as modeling elements related to AR-caused faults leading to human failures and by analyzing their failure propagation, architects and safety managers have at disposal means to reveal effect of AR-related dependability threats on system behavior. For example, in the first scenario failure in updating rules and regulations based on AR technology 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 second scenario, failure in interactive experience and in the third scenario failure in AR guided task are considered as an AR-related dependability threats and their modeling elements provide representation means for taking into account AR effects as AR-caused faults leading to human failures.
- **Expressiveness**: Expressiveness refers to the power of a modelling language to express or describe all things required for a given purpose [15]. Set of symbols or possible statements that can be described by modelling languages can be used for measuring expressiveness. Statement means “a syntactic expression and its meaning”. The proposed extension on human modeling elements used to extend the modeling language is based on an AR-extended human function taxonomy (AREXTax [22]), which is gained by harmonizing about 6 state-of-the-art human failure taxonomies (Norman [14], Reason [17], Rasmussen [16], HFACS (Human Factor Analysis and Classification System) [21], SERA (Systematic Error and Risk Analysis) [8], Driving [28]) and then

> extending the taxonomy based on various studies and experiments on augmented reality. In addition, the proposed extension for extending organization modeling elements is based on a fault taxonomy (AREFTax [25]) containing AR-caused faults leading to human failures, which is gained by harmonizing about 5 state-of-the-art fault taxonomies (Rasmussen [16], HFACS (Human Factor Analysis and Classification System) [21], SERA (Systematic Error and Risk Analysis) [8], Driving [28] and SPAR-H (Standardized Plant Analysis Risk Human Reliability Analysis)[6]) and then extending the taxonomy based on various studies and experiments on augmented reality. Thus, we believe that these extensions increase power of modeling language to express new AR-caused risks as it is also shown in the case study provided in this paper.

One issue that is not covered by modeling extensions and analysis technique is that the result of the analysis is dependent on the elements used for modeling and how their failure behaviors are modeled. It is dependent on the skills of analyst to be able to choose most suitable modeling elements and to be able to model their failure behavior correctly. So it causes new challenges when the result is different while the techniques are used by different people.

4 Discussion

We have used Concerto-FLA analysis technique as the basis of the analysis in order to disclose the advantages of our proposed AR-related extensions for CHESS framework at analysis level. Concerto-FLA uses FPTC syntax for 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 proposed AR-extended modeling elements characterizing different aspects of a component. It is important to consider possible failure modes for each input in a component/sub-component and skipping the others, because the number of FPTC rules grows exponentially with increase of the input ports. It is not conspicuous in small and academic examples, but it is really challenging if we use an industrial case study. There are also some occasions that one failure mode in input would lead to different failure modes in output. This can not be modeled using FPTC rules, because the assumption in this technique is that behavior for each component is deterministic. In industrial case studies, there would be situations with a component with non-deterministic behavior. In order to overcome this challenge, we considered the most probable situation and we modeled the component based on that situation. However, if it is required to model more complicated situations, then it is required to have more research on the extensions for techniques based on FPTC to overcome this limitation.

The generated model using our proposed AR-extended modeling elements and analysis results based on the extensions can be used as arguments based on evidences in order to provide safety case for AR-equipped industrial products 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 how the safety requirements are achieved.

5 Threats to Validity

In this section, we discuss threats of validity in relation to our research based on literature [19]. Validity of a study denotes to what extent the results can be trusted.

External validity refers to possibility of generalization of the findings. We provided a case study with three scenarios from automotive domain, but the proposed extensions are not limited to specific scenarios and specific domain and the baseline for the 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 and the proposed extensions are also based on state-of-the-art taxonomies (Norman [14], Reason [17], Rasmussen [16], HFACS (Human Factor Analysis and Classification System) [21], SERA (Systematic Error and Risk Analysis) [8], Driving [28] and SPAR-H (Standardized Plant Analysis Risk Human Reliability Analysis) [6] taxonomies) in addition to 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 can not be guaranteed that different people will have same answer using our proposed extensions, because it depends on the analyzer skills and ability for modeling and analysis.

Our main focus in this paper is to validate our proposed AR-related extensions for CHESS toolset on a realistic and sufficiently complex case at a level that can be found in industry. 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, our proposed extensions and FPTC rules. The downside is that it was not possible to check correctness and completeness of the FPTC rules.

In this case study we examined the modeling and analysis capabilities of our proposed 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 modelling all possible scenarios. Instead, we have focused on the provided elements to check if they are able to capture new types of system failure behaviors.

The implications of the results of the case study can not be advantageous for all different scenarios. 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 extensions. Further investigations are required to provide more beneficial results on limitations of modelling and analysis applications.

6 Conclusion and Future Work

In this paper, we conducted a case study with the purpose of presenting the modeling capabilities and analysis capabilities of our proposed AR-related extensions for CHESS framework in order to estimate how effective they are in predicting new kinds of risks caused by AR-related factors. The extensions are for modelling and analyzing AR effects on human functioning and faults leading to human failures. We used an industrial case study to figure out if the extensions are effective in predicting new system failures caused by augmented reality. We showed how the analysis can be done manually, by implementing our proposed extensions for CHESS toolset, failure propagation calculation can be provided automatically to be used for AR-equipped socio-technical systems.

Further research is required to show the potential benefits of the proposed extensions. For example, using case studies with higher safety criticality in order to have scenarios with higher risks. 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 aim at implementing the conceptual extension of SafeConcert within CHESSML and we aim at extending analysis technique based on the proposed extensions to provide analysis results for AR-equipped socio-technical systems automatically. We can also consider a safety-critical socio-technical system within the rail industry, the passing of a stop signal (signal passed at danger; SPAD) [13], to verity if the results are transferrable within the rail domain.

References

1. CONCERTO D2.7 – analysis and back-propagation of properties for multicore systems – final version, http://www.concerto-project.org/results
2. Bressan, L.P., de Oliveira, A.L., Montecchi, L., Gallina, B.: A systematic process for applying the chess methodology in the creation of certifiable evidence. In: 2018 14th European Dependable Computing Conference (EDCC). pp. 49–56. IEEE (2018)
3. Dimitrakopoulos, G., Uden, L., Varlamis, I.: The Future of Intelligent Transport Systems. Elsevier (2020)
4. Fu, W.T., Gasper, J., Kim, S.W.: Effects of an in-car augmented reality system on improving safety of younger and older drivers. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). pp. 59–66. IEEE (2013)
5. Gallina, B., Sefer, E., Refsdal, A.: Towards safety risk assessment of socio-technical systems via failure logic analysis. In: 2014 IEEE International Symposium on Software Reliability Engineering Workshops. pp. 287–292. IEEE (2014)
6. Gertman, D., Blackman, H., Marble, J., Byers, J., Smith, C., et al.: The SPAR-H human reliability analysis method. US Nuclear Regulatory Commission **230** (2005)
7. Goldiez, B.F., Saptoka, N., Aedunuthula, P.: Human performance assessments when using augmented reality for navigation. Tech. rep., University of Central Florida Orlando Inst for Simulation and Training (2006)

8. Hendy, K.C.: A tool for human factors accident investigation, classification and risk management. Tech. rep., Defence Research And Development Toronto (Canada) (2003)
9. International Organization for Standardization (ISO). : ISO 26262: Road vehicles — Functional safety. (2018)
10. Mazzini, S., Favaro, J.M., Puri, S., Baracchi, L.: Chess: an open source methodology and toolset for the development of critical systems. In: EduSymp/OSS4MDE@ MoDELS. pp. 59–66 (2016)
11. Miller, M.R., Jun, H., Herrera, F., Villa, J.Y., Welch, G., Bailenson, J.N.: Social interaction in augmented reality. PloS one **14**(5), e0216290 (2019)
12. Montecchi, L., Gallina, B.: SafeConcert: A metamodel for a concerted safety modeling of socio-technical systems. In: International Symposium on Model-Based Safety and Assessment. pp. 129–144. Springer (2017)
13. Naweed, A., Trigg, J., Cloete, S., Allan, P., Bentley, T.: Throwing good money after spad? exploring the cost of signal passed at danger (spad) incidents to australasian rail organisations. Safety science **109**, 157–164 (2018)
14. Norman, D.A.: Errors in human performance. Tech. rep., California Univ San Diego LA JOLLA Center For Human Information Processing (1980)
15. Patig, S.: Measuring expressiveness in conceptual modeling. In: International Conference on Advanced Information Systems Engineering. pp. 127–141. Springer (2004)
16. Rasmussen, J.: Human errors. a taxonomy for describing human malfunction in industrial installations. Journal of occupational accidents **4**(2-4), 311–333 (1982)
17. Reason, J.: The human contribution: unsafe acts, accidents and heroic recoveries. CRC Press (2017)
18. Ruiz, A., Melzi, A., Kelly, T.: Systematic application of ISO 26262 on a SEooC: support by applying a systematic reuse approach. In: 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 393–396. IEEE (2015)
19. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering **14**(2), 131 (2009)
20. Schall Jr, M.C., Rusch, M.L., Lee, J.D., Dawson, J.D., Thomas, G., Aksan, N., Rizzo, M.: Augmented reality cues and elderly driver hazard perception. Human factors **55**(3), 643–658 (2013)
21. Shappell, S.A., Wiegmann, D.A.: The human factors analysis and classification system–HFACS. Tech. rep., Civil Aeromedical Institute (2000)
22. Sheikh Bahaei, S., Gallina, B.: Augmented reality-extended humans: towards a taxonomy of failures – focus on visual technologies. In: European Safety and Reliability Conference (ESREL). Research Publishing, Singapore (2019)
23. Sheikh Bahaei, S., Gallina, B.: Extending safeconcert for modelling augmented reality-equipped socio-technical systems. In: International Conference on System Reliability and Safety (ICSRS). IEEE (2019)
24. Sheikh Bahaei, S., Gallina, B.: Towards assessing risk of safety-critical socio-technical systems while augmenting reality. Published as proceedings annex on the International Symposium on Model-Based Safety and Assessment (IMBSA) website (2019), http://easyconferences.eu/imbsa2019/proceedings-annex/
25. Sheikh Bahaei, S., Gallina, B., Laumann, K., Rasmussen Skogstad, M.: 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)
26. Šljivo, I., Gallina, B., Carlson, J., Hansson, H., Puri, S.: A method to generate reusable safety case argument-fragments from compositional safety analysis. Journal of Systems and Software **131**, 570–590 (2017)

27. Sljivo, I., Gallina, B., Carlson, J., Hansson, H., et al.: Using safety contracts to guide the integration of reusable safety elements within iso 26262. In: 2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing (PRDC). pp. 129–138. IEEE (2015)
28. Stanton, N.A., Salmon, P.M.: Human error taxonomies applied to driving: A generic driver error taxonomy and its implications for intelligent transport systems. Safety Science **47**(2), 227–237 (2009)
29. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles: https://www.sae.org/standards/content/j3016_201806/ (2018), https://www.sae.org/standards/content/j3016_201806/
30. Van Krevelen, D., Poelman, R.: A survey of augmented reality technologies , applications and limitations. The International Journal of Virtual Reality **9**(2), 1–20 (2010)
31. Wallace, M.: Modular architectural representation and analysis of fault propagation and transformation. Electronic Notes in Theoretical Computer Science **141**(3), 53–71 (2005)