Towards Qualitative and Quantitative Dependability Analyses for AR-Equipped Socio-Technical Systems

Soheila Sheikh Bahaei, Barbara Gallina
School of Innovation, Design and Engineering
Mälardalen University
Västerås, Sweden
soheila.sheikhbahaei, barbara.gallina@mdh.se

2021 5th International Conference on System Reliability and Safety (ICSRS) | 978-1-6654-0049-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSRS53853.2021.9660642

***Abstract*—Augmented Reality technologies are becoming essential components in various socio-technical systems. New kinds of risks, however, may emerge if the concertation between AR, other technical components and socio-components is not properly designed. To do that, it is necessary to extend techniques for risk assessment to capture such new risks. This may require the extension of modelling languages and analysis techniques. In the literature, modeling languages have been already extended by including specific language constructs for socio aspects in relation to the AR-impact. No satisfying contribution is available regarding analysis techniques. Hence, to contribute to filling the gap, in this paper, we propose an extension of previously existing analysis techniques. Specifically, we build on top of the synergy of qualitative and quantitative dependability analysis techniques and we extend it with the capability of benefiting from AR-related modelled aspects. In addition, we apply our proposed extension to an illustrative example. Finally, we provide discussion and sketch future work.**

***Index Terms*—Risk Analysis, Augmented Reality, Socio-technical Systems, AR-equipped Systems**

I. Introduction

Analyzing Augmented Reality (AR)-equipped socio-technical systems requires contemplating effect of various AR-related aspects on system behavior. Socio-technical systems are systems including socio entities such as human and organization and technical entities [1] such as augmented reality. Augmented reality technology is a technology that superimposes virtual content on the real environment of the user [2]. Augmented reality affects on human performance and it also affects on influencing factors on human performance. In order to automatize system analysis, model-based techniques [3] are used, which contain modeling system architecture and system behavior.

In [4], an extension to the Architecture Analysis and Design Language (AADL) [5] is proposed to enable modeling various types of faults and intertwining them into the system model to be used for analysis. It contains language extensions for modeling technical faults. In [6], a conceptual framework, called WAx (Work-As-x) is presented for the analysis of cyber-socio-technical systems. It contains concepts, and a language to develop information-driven model for understanding system functioning. It encompasses effect of digitalization on systems and organizations.

In [7], human entity is modeled by characterizing its behavior through human functions. In this study, effect of augmented reality on human functions are also considered. Human functions are provided based on state-of-the-art human failure taxonomies and they are extended by AR-extended human functions based on studies and experiments on AR. In [8], effect of augmented reality on influencing factors on human functions are considered and a taxonomy of faults leading to human failures based on state-of-the-art taxonomies are proposed. Then, this taxonomy is extended by considering AR-related factors causing human failures based on studies and experiments on augmented reality. SafeConcert metamodel [9], which is a metamodel for modeling socio-technical systems is extended in [10] to enable modeling of AR-equipped socio-technical systems. In [11], new concepts are proposed for modeling effect of digitalization, globalization and networked structure of organizations while performing risk assessment in AR-equipped socio-technical systems.

Currently, there is no analysis technique considering AR-related risks to be used for analyzing AR-equipped socio-technical systems' behavior. In this paper, we aim at proposing an extension for a synergy of qualitative and quantitative dependability analysis technique. The extension is based on AR-related modeling extensions and Concerto-FLA analysis plugin [1], which is Eclipse plugin for dependability analysis and risk assessment implemented in CHESS project [12]. We use Concerto-FLA analysis technique, because it includes constructs for including socio-technical systems' concepts. We also use AR-related modeling extensions to include AR-related concepts. Finally, we use a monitoring system case study to illustrate our contributions.

The rest of the paper is organized as follows. In section II, we provide essential background information. In Section III, we propose our extension on synergy of qualitative and

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

978-1-6654-0049-7/21/$31.00 ©2021 IEEE

quantitative dependability analysis represented as an extended process. In Section IV, we present the extension on a monitoring system case study. In Section V, we provide a discussion about the contribution of our proposed extension. In Section VI, we provide related work. Finally, in Section VII, we present some concluding remarks and describe the future work.

II. Background

This section provides essential background information onto which our work is based. First, the metamodel extensions for modeling AR-equipped socio-technical systems are recalled. Then, toolchain for automated dependability evaluation and a synergy of qualitative and quantitative dependability analysis techniques are recalled. Finally, analyzing socio-technical systems is explained.

A. Metamodel extensions for AR-equipped Socio-technical Systems

To capture AR-equipped socio-technical systems, constructs for modelling socio and technical (including AR-specific aspects) entities are needed. In [1], new constructs are proposed for modeling human and organization and their related aspects. In [10], AR-related concepts in addition to various socio concepts are considered and modeling elements related to these concepts categorized into two types are proposed. First category is human modeling elements for characterizing human functions (including AR-extended human functions) and human internal states (including AR-related human internal states). Second category is organization modeling elements characterizing external influencing factors on human performance (including AR-related factors).

For example, modeling elements of paying attention, deciding, executing and etc. are used for characterizing human functions. Modeling elements of human physical state, mental state, experience and etc. are used for characterizing human internal states. Modeling elements of environmental condition, time pressure, supervision and etc. are used for characterizing external influencing factors. Modeling element of surround detecting is an AR-extended modeling element, which characterize AR-extended human function. The reason is that using AR technology would help human to detect surrounding environment, thus augmenting the human to an extended human.

B. Toolchain for Automated Dependability Evaluation

A toolchain is introduced in [13], to perform the dependability analysis automatically. The toolchain contains five metamodels and four model-transformation algorithms. The relationship between five models (m1...m5) conforming to these five metamodels and four model-transformations (t1...t4) are shown in Fig. 1.

- **Metamodel 1:** This metamodel contains constructs to model various concepts of system architecture. The extended metamodel explained in Subsection II-A provides the constructs for preparing a model of system architecture at this level.

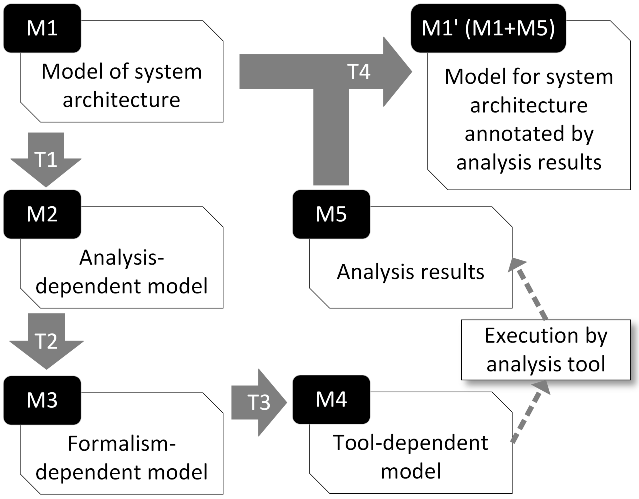

M1
Model of system architecture
T1
M2
Analysis-dependent model
T2
M3
Formalism-dependent model
T3
M4
Tool-dependent model
Execution by analysis tool
M5
Analysis results
T4
M1' (M1+M5)
Model for system architecture annotated by analysis results

Fig. 1. Relationship between models and transformations adapted from [13].

- **Metamodel 2:** This metamodel contains constructs for performing the intended analysis. The model prepared at this level is analysis-dependent. For example, in order to perform performance analysis, only information related to performance are extracted from the system architecture and other details are not considered.
- **Metamodel 3:** This metamodel contains constructs for implementing the analysis model in a specific formalism. The model prepared at this level is formalism-dependent. For example, Stochastic Petri Nets (SPNs) [14] or Fault Tree [15] can be considered as formalisms for the analysis.
- **Metamodel 4:** This metamodel contains constructs for preparing the code of the implementation by a specific tool. The model prepared at this level is tool-dependent. For example, a file including header, variable definitions, etc. that can be provided as input of a tool is a model at this level.
- **Metamodel 5:** This metamodel contains constructs for describing the results provided by the analysis tool. For example, a text file conforming to standard interchange formats such as XML can be considered as a model at this level.
- **Model-transformation 1:** This transformation extracts the information required for the intended analysis from the mass of information representing the system architecture. It is applied to m1 to produce m2.
- **Model-transformation 2:** This transformation implements the analysis algorithm using the intended formalism. It is applied to m2 to produce m3.
- **Model-transformation 3:** This transformation provides the implemented code to be used as the input of the analysis tool. It is applied to m3 to produce m4.
- **Model-transformation 4:** This transformation propagates the analysis results back into the system architecture. It uses m5 and m1 to produce a modified version of m1, which contains analysis results in addition to system

architecture.

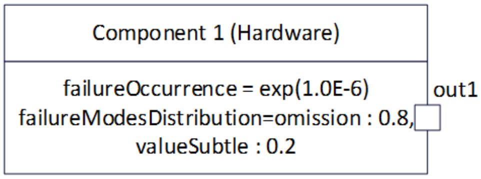
This toolchain presents how the dependability analysis can be implemented to perform the analysis automatically.

C. Synergy of Qualitative and Quantitative Dependability Analysis Techniques

A synergy of qualitative and quantitative dependability analysis techniques is proposed in [16]. It contains State-based analysis and Failure Logic Analysis (FLA). State-based analysis technique [17] is a quantitative technique, which is implemented based on the toolchain explained in Subsection II-B. FLA is a qualitative analysis based on qualitative behavior of components and their causes.

It is required to have information or assumptions about the system architecture to be used for modeling system architecture. Formalism used in state-based analysis is Stochastic Petri Nets (SPNs) [14] with general probability distributions. There are three types of behavior modeling used in these two analysis techniques, which are simple stochastic behavior, error model and Failure Logic Analysis (FLA) [16]. These three types of behavior modeling are described in the following paragraphs.

Simple stochastic behavior uses probability distribution for specifying the time to the occurrence of a failure and the time required to fix the component after failure occurrence, if available. Possible failure modes and their probabilities also can be provided. As it is shown in Fig. 2, exponential distribution with rate of 1.0e-6 per hour of operation is used for illustrating time to failure of this hardware component. Possible failure modes in case of failure in the output and their probabilities are shown in this example, which are omission (means output is not provided when expected) with probability of 80% and valueSubtle (means output is not in the expected range and it is not detected by user) with probability of 20%.



Component 1 (Hardware)
failureOccurrence = exp(1.0E-6)
failureModesDistribution=omission : 0.8,
valueSubtle : 0.2
out1

Fig. 2. Modeling a hardware component with stochastic behavior [16].

Error model is defined by using a set of finite state machines modeling internal faults, external faults and their probabilities. It also models transitions between states. Error models are used when there are detail information about the component's failure behavior [16]. For example in Fig. 3, a software is modeled by two error models modeling internal fault occurrence and effect of external faults. In the top part of the picture, probability of occurrence of internal fault is defined as exp(6.0E-6) and it would propagate to an undetected error state leading to output failure mode omission with weight 0.8 or it would propagate to an error state incorrect value with weight 0.2. In the bottom part of the picture, omission external fault is considered propagated to undetected error state leading to omission failure mode in the output.

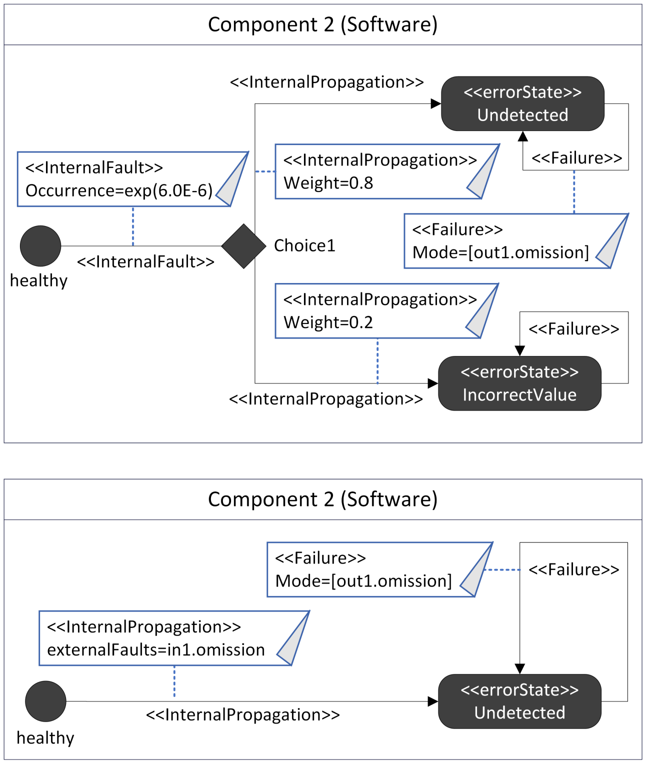
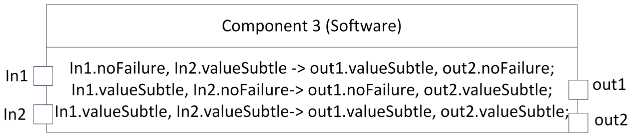


Fig. 3. Modeling a software component with error models [16].

FLA behavior is defined by assigning possible failure modes in the input to possible failure modes in the output. In this type of behavior modeling probabilities are not considered. For example, in Fig. 4, a software is modeled by defining FLA behavior. In this example, there are two inputs (In1, In2) and two outputs (out1, out2) for the software component. NoFailure (normal behavior) at input In1 and valueSubtle at input In2 will lead to valueSubtle at out1 and noFailure at out2. Relationship of other possible failure modes at inputs and outputs are defined similarly.



Component 3 (Software)
In1
In2
In1.noFailure, In2.valueSubtle -> out1.valueSubtle, out2.noFailure;
In1.valueSubtle, In2.noFailure-> out1.noFailure, out2.valueSubtle;
In1.valueSubtle, In2.valueSubtle-> out1.valueSubtle, out2.valueSubtle;
out1
out2

Fig. 4. Modeling a software component with FLABehavior [16].

The following metrics can be measured by the quantitative analysis:

- Reliability: the probability that the system continuously remains in proper state from the time 0 up to time t.
- Availability:
  - Immediate: the probability that the system is in proper state at time t.
  - In a time interval: the fraction of time that system is in proper state in a given time interval.
- Probability of Failure on Demand (PFD): the probability

that the system fails to provide a requested service. It can be obtained by calculating 1 minus immediate reliability.

D. Analyzing Socio-technical systems

Concerto-FLA analysis technique [1] is an extension of FLA qualitative technique. The extension includes capabilities for analyzing socio aspects. It is implemented as plug-in within CHESS toolset [18] for developing high integrity socio-technical systems. Model of the system architecture is used for running the analysis and results are back propagated in order to support an iterative and incremental system development [13]. Formalism used in Concerto-FLA is based on fixed-point calculation used in FPTC technique [19]. In Concerto-FLA analysis technique [1], FPTC rules are used.

FPTC rules are expressions for illustrating components' behavior by relating input failure modes to output failure modes. Failure modes include early (provided function early), late (provided function late), commission (provided function at a time which is not expected), omission (not provided function at a time which is expected), valueSubtle (provided wrong value after computation that user can not detect it) and valueCoarse (provided wrong value after computation that user can detect it). Components' behavior can be classified as the following categories:

- Sink: when component detects failure in the input and corrects it in the output.
- Propagational: when component propagates the same failure mode or normal behavior in the input to the output.
- Transformational: when component transforms the failure mode in the input to another failure mode in the 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

NoFailure shows normal behavior. Wildcard on a specific input shows that the output is provided regardless of the failure mode or normal behavior of this specific input. For example, IP1.wildcard → OP1.noFailure is an example of a FPTC rule which shows that regardless of the failure mode or normal behavior on the input port with the name IP1 the output on the port OP1 will be provided with normal behavior. This shows the behavior of a component with sink behavior.

III. Proposed Analysis Process

In this section, we propose an extension based on AR-related modeling extensions and Concerto-FLA analysis technique [1]. We build on top of the synergy of qualitative and quantitative analysis in [16]. We aim at extending this synergy by incorporating socio-related and AR-related aspects explained in Subsection II-A. Our proposed analysis process is illustrated in Fig. 5.

The added value with respect to the synergy of quantitative and qualitative analysis is the possibility of analyzing various socio and AR-related aspects and their effects on system behavior. AR-related metamodel extensions are used in the system modeling by including AR-related modeling elements in the system model. In case of using qualitative analysis, Concerto-FLA analysis can be used for defining FPTC rules for AR-related components and automated analysis is used for obtaining the annotated model by analysis results. In case of quantitative analysis, error model or stochastic behavior are used for analyzing system behavior including AR-related effects.

Part A of Fig. 5 contains the activity that should be done for preparing the system model. This activity is defining components and sub-components. Then, we need to decide about analysis type. If we need to do qualitative analysis, we perform the next activities based on Concerto-FLA analysis technique (Part B), otherwise we perform based on State-based analysis technique (Part C).

Based on Part B of the figure, FPTC rules should be defined for all components. Then, Concerto-FLA analysis will be executed and model annotated by analysis results will be provided.

Based on Part C of the figure, failure behavior modeling type should be defined. If we want to use error model, then we need to create the error model of the desired component. If we want to use stochastic behavior, then we should define the related parameters. Next step is to execute the state-based analysis and to measure the evaluation result.

Result of the analysis can be used for hazard identification, defining safety goals and safety requirements.

We explain the activities of all the steps in the following sub-sections and in Table I, we compare these steps of our proposed extended process with the previous process in [16].

A. Define Components and Sub-components

Main entities incorporating in a system are considered as the main components. It is important to consider socio entities, which are human and organization. Defining sub-components are based on important aspects of each entity. In technical components, important aspects are defined based on technical description of the system. Human important aspects are defined based on human functions and human internal states. Organization important aspects are defined based on organizational important aspects. Human and organization modeling elements introduced in the extended metamodel explained in Subsection II-A are the modeling constructs that can be used for defining human and organization sub-components. For example, condition, environment and any other influencing factor on human performance can be considered as organizational important aspects. The extensions include AR-related aspects, which should be considered in defining sub-components.

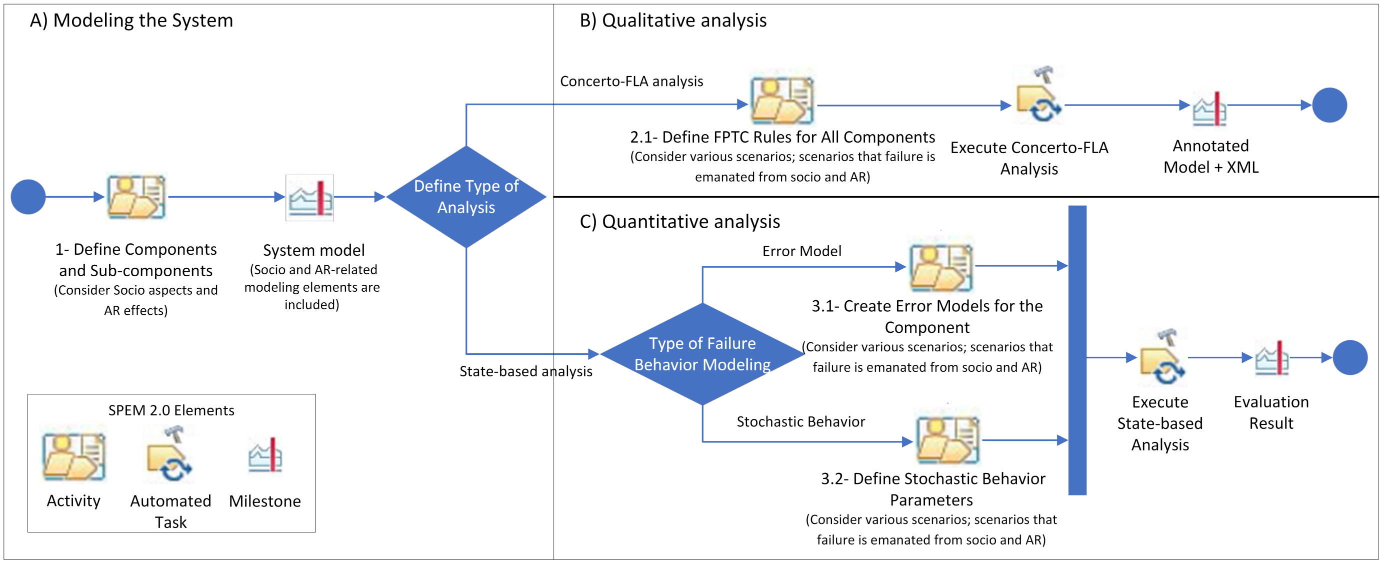

A) Modeling the System
B) Qualitative analysis
Concerto-FLA analysis
2.1- Define FPTC Rules for All Components
(Consider various scenarios; scenarios that failure is emanated from socio and AR)
Execute Concerto-FLA Analysis
Annotated Model + XML
1- Define Components and Sub-components
(Consider Socio aspects and AR effects)
System model
(Socio and AR-related modeling elements are included)
Define Type of Analysis
C) Quantitative analysis
Error Model
3.1- Create Error Models for the Component
(Consider various scenarios; scenarios that failure is emanated from socio and AR)
State-based analysis
Type of Failure Behavior Modeling
Stochastic Behavior
3.2- Define Stochastic Behavior Parameters
(Consider various scenarios; scenarios that failure is emanated from socio and AR)
Execute State-based Analysis
Evaluation Result
SPEM 2.0 Elements
Activity
Automated Task
Milestone

Fig. 5. The proposed extended analysis process.

B. *Define FPTC rules for All Components*

This activity should be done based on the syntax explained in Subsection II-D. In order to define FPTC rules, each component/sub-component should be analyzed individually. We should define the possible failure modes at each of their inputs and outputs for various scenarios. Then, FPTC rules can be used for relating the failure modes at inputs to the failure modes at outputs. For example, a camera would not receive the input (raw image) because of the obstacle in front of it. Input failure mode in this example is omission as explained in Subsection II-D. Based on technical analysis of the camera, we would model it as propagational (explained in Subsection II-D). It means that the failure mode in input propagates to the output port and it does not provide the output.

C. *Create Error Models for the Component*

This activity should be done based on the syntax explained in Subsection II-C. In order to define error models, the intended component/sub-component should be analyzed individually. State machine for each component including internal and external faults and their probabilities should be defined for various scenarios.

D. *Define Stochastic Behavior Parameters*

This activity should be done based on the syntax explained in Subsection II-C. In order to define stochastic behavior parameters, the intended component/sub-component should be analyzed individually. Possible failure modes and their probabilities should be defined for various scenarios.

IV. Case study

In this section, we design a case study with the objective of presenting the analysis capabilities provided by the proposed process. First step is to model the system, as shown in part A of the process. Then, Concerto-FLA analysis can be used for qualitative analysis (Part B) and state-based analysis can be used for quantitative analysis (Part C). We consider an industrial monitoring system introduced in [20]. We use this system as a case study for analyzing AR-equipped socio-technical system.

The industrial monitoring system uses a sensor for receiving raw data. Raw data is processed in server and it is organized to be represented to the user for making decisions. AR can be used for providing graphical or textual instructions for solving a problem, configuring an equipment or maintenance activities. In this example, we consider using AR for providing visual alarm in case of problem in a special equipment under control.

A. *Modeling the System*

This system includes technical and socio entities. Technical entity is the monitoring system and socio entities are the user and organization. We model each of these entities based on their description and based on their important aspects.

The technical components of this system are defined based on description of monitoring system as follows:

- **Sensor**: it is a hardware component. It can be various sensors, for example a camera receiving raw data of a specific equipment, which is considered for monitoring.
- **Server**: it is a hardware component. It is a computer that contains processing unit for processing the data.
- **Processing unit**: it is a software component. It processes the received data from sensor and organizes it in a format to be used by the user.
- **AR application interface**: it is a hardware component. It is the interface between the user and the server. It is an screen containing AR technology notations.

The user can be characterized based on its important aspects, which are human functions and human internal states.

TABLE I
COMPARISON OF OUR PROPOSED EXTENDED PROCESS WITH THE PREVIOUS PROCESS IN [16]

Steps	**In the previous process in** [16]	**In our proposed extended process**
Define components and sub-components	Technical components/sub-components are defined.	Technical + socio + AR-related components/sub-components are defined.
Define FPTC rules for all components	Scenarios including failures emanated from technical components/sub-components are considered.	Scenarios including failures emanated from technical + socio + AR-related components/sub-components are considered.
Create error models for the component	Scenarios including failures emanated from technical components/sub-components are considered.	Scenarios including failures emanated from technical + socio + AR-related components/sub-components are considered.
Define stochastic behavior parameters	Scenarios including failures emanated from technical components/sub-components are considered.	Scenarios including failures emanated from technical + socio + AR-related components/sub-components are considered.

We use four following modeling elements of the extended metamodel explained in Subsection II-A.

- **Directed paying attention**: it refers to an AR-extended human function. It models the function paying attention when it is directed to a specific position by using AR technology. For example, in this case study, if there is something strange related to the equipment which is under monitoring, then AR technology can be used for displaying a red circle around the strange area. Thus, the user attention will be directed to the position to make a decision to prevent any probable risk.
- **Training**: it refers to training received by the human.
- **Deciding**: it refers to human deciding function.
- **Executing**: it refers to human executing function.

The organization can be modeled based on important organizational aspects. We use the following modeling elements of the extended metamodel explained in Subsection II-A.

- **Condition**: it refers to the condition of the organization where the monitoring task is performed.
- **Organization and regulation AR adoption**: it refers to an AR-extended aspect. It models the adoption process needed in the organization to be able to use AR.
- **AR guided task**: it refers to the task that AR is used for guiding the human to do that. For example, a task should be defined in an organization that in case of special AR alarm the user should react.

Based on the described entities and their important aspects, we provided the model shown in Fig. 6. Sensor receives raw data (shown by RD in Fig. 6) and provides the output for processing unit. Data is processed in processing unit and its output is shown in AR application interface to the user. Organization and regulation AR adoption is influenced by regulation authorities (REG) and it affects on AR guided task defined by organization. AR guided task is also influenced by condition of the organization, which is influenced by condition out of the organization (shown by CON). Output of monitoring system which is a visual description on a screen influences on human directed paying attention and output of the organization influences on training. Finally, human deciding function is influenced by directed paying attention and training. Human executing function is influenced by human deciding function. Output of the system, which is output of the human component is human function (HF).

B. Qualitative Analysis

As it is shown on part B of the Fig. 5, in order to provide the qualitative analysis, we need to define FPTC rules for all components. These rules should be defined based on individual analysis of components and based on the assumptions of various scenarios. For example, we provide the FPTC rules for a specific scenario and we provide the system behavior based on failure propagation.

- **Definition of scenario**: We assume that the equipment under monitoring is in a situation that it can harm a person. The information is received by the sensor and it is processed by the processing unit and a visual alarm is displayed on the AR display. However, we assume that there is a failure in organization and regulation AR adoption. For example, organization should update regulations in order to include AR related considerations and trainings. Since there is no rule defined in the organization, the required training is not provided for the user. The user's attention is directed to the alarm, but the user does not take the correct decision and does not provide the required execution function to prevent the harm.
- **Modeling of the failure behavior**: In this scenario the organization and regulation AR adoption is behaving as a source (source behavior is explained in Subsection II-D). The input of this component receives noFailure, but in the output it provides valueSubtle. The reason is that organization has not updated rules and regulations to adopt AR (valueSubtle) and the user does not receive the required AR-related training (omission). Since the user does not receive the required AR-related training, the deciding component provides valueSubtle failure mode in its output. Thus, the user does not provide the required execution (omission). Monitoring system components are behaving as propagational and propagate noFailure from input to output.
- **Analyzing the system behavior**: Analysis annotations are shown in Fig. 7. ValueSubtle in OP4 means that the AR adoption in organization and regulation is not

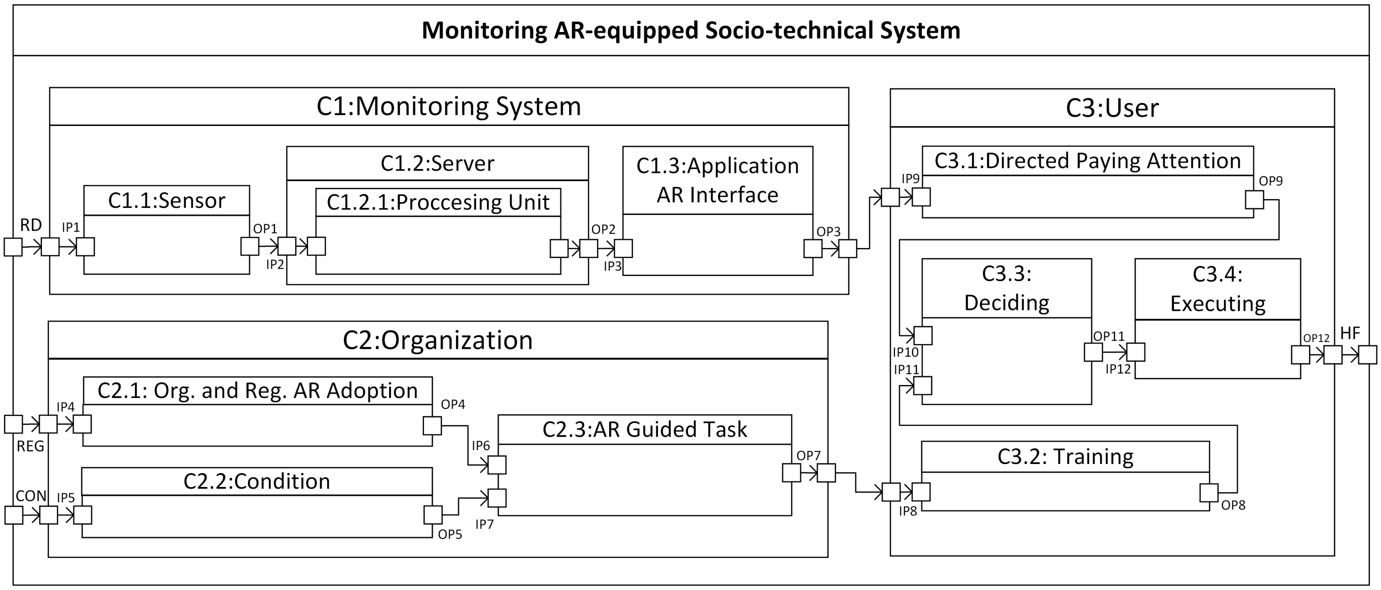


Fig. 6. Modeling the system.

performed correctly. ValueSubtle failure mode transforms to omission in AR guided task and it propagates in training. Then, in deciding it transforms to valueSubtle and in executing transforms to omission. The failure propagation is shown by blue color.

- **Interpreting the results**: Based on the back propagation of the results, we can explain how the rules have been triggered. Omission in HF is because of valueSubtle in OP11. ValueSubtle in OP11 is because of omission in IP11 and we continue until IP4, which is input port of organization and regulation AR adoption. Thus, this component caused the failure in the system output. The identified hazard is as follows:
  - **Hazard**: Lack of required AR training

  The reason for this hazard is failure in organization and regulation AR adoption. System failure in this scenario may lead to fatal injuries for people around the intended equipment. Thus, safety goal should be defined to overcome this risk. For example, for this scenario, safety goal can be defined as follows:
  - **Safety goal**: The organization should update rules and regulations based on AR and should provide the required AR training.

  By using the qualitative analysis and by considering various possible scenarios, various safety goals can be defined. Based on safety goals, system design can be updated and analysis of system behavior can be performed for more iterations to reach the accepted level of safety.

C. Quantitative Analysis

Based on part C of the Fig. 5, in order to provide the quantitative analysis, we should model the failure behavior using error models or stochastic parameters. Similar to qualitative analysis these models should be defined based on individual analysis of components and based on the assumptions of various scenarios. For example, we provide stochastic behavior modeling for a specific scenario and we provide the analysis result.

- **Definition of scenario**: Similarly, we assume that the equipment under monitoring is in a situation that it can harm a person. The information should be received by the sensor and it should be processed by the processing unit. Then, a visual alarm should be illustrated through AR display and the user should decide based on illustrated alarm and based on received training from organization to execute a needed task preventing the risk.
- **Modeling of the failure behavior**: In this scenario, for each component we consider possible failure modes and their probabilities as it is shown in figure Fig. 8. Probabilities can be defined based on previous accident reports or based on expert opinion. For example, in this scenario, organization has not updated rules and regulations based on AR technology. Thus, failure probability in the Org. and Reg. AR adoption component is high (0.9).
- **Analyzing the system behavior**: In order to perform the analysis, we can consider the hazard related to this scenario and calculate the intended measure or failure mode probability in system output. We consider the same hazard as the one we considered in qualitative analysis, which is lack of required AR training. In this case, we want to calculate the probability of omission failure mode in system output. The result for this assumed scenario is shown in Fig. 8. Calculation is an automatic task, which can be performed by running the analysis in the toolset. For example, failure in output of executing function would be of type omission or valueSubtle. The probability of omission failure mode is calculated based on the probability of executing function providing an omission

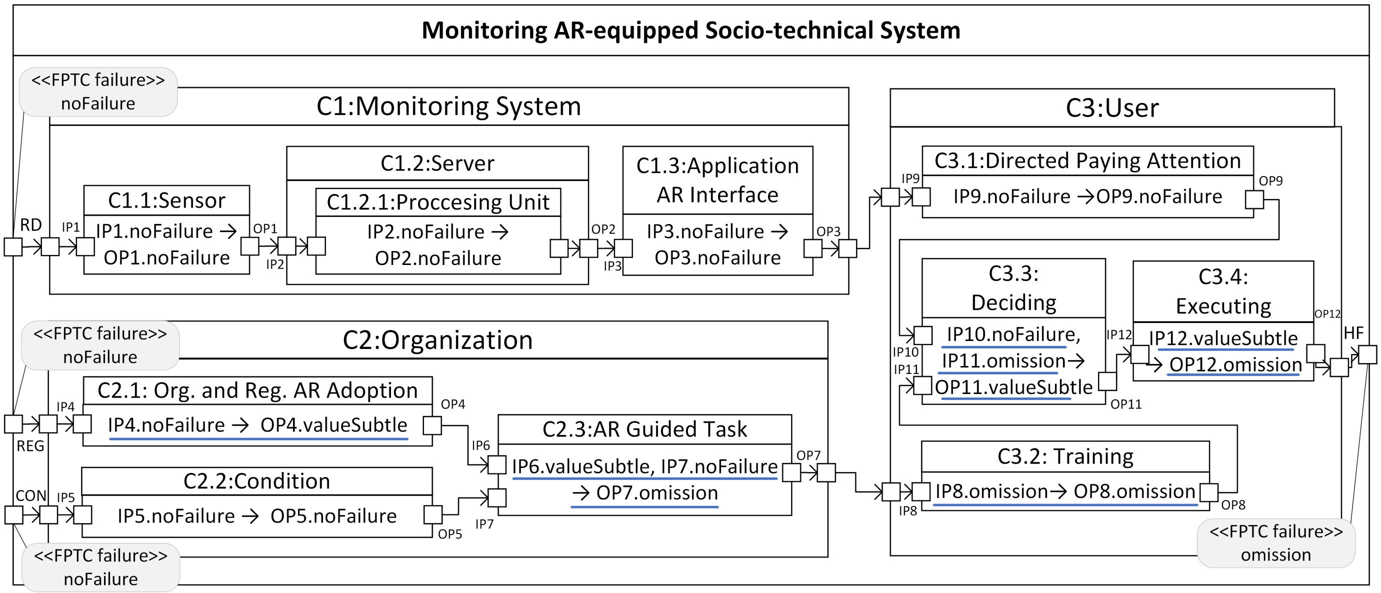

Monitoring AR-equipped Socio-technical System
<<FPTC failure>> noFailure
C1:Monitoring System
C1.1:Sensor
IP1.noFailure → OP1.noFailure
C1.2:Server
C1.2.1:Proccesing Unit
IP2.noFailure → OP2.noFailure
C1.3:Application AR Interface
IP3.noFailure → OP3.noFailure
RD
IP1
OP1
IP2
OP2
IP3
OP3
C3:User
C3.1:Directed Paying Attention
IP9.noFailure →OP9.noFailure
IP9
OP9
C3.3: Deciding
IP10.noFailure, IP11.omission→ OP11.valueSubtle
C3.4: Executing
IP12.valueSubtle → OP12.omission
IP10
IP11
OP11
IP12
OP12
HF
C3.2: Training
IP8.omission→ OP8.omission
IP8
OP8
<<FPTC failure>> omission
<<FPTC failure>> noFailure
C2:Organization
C2.1: Org. and Reg. AR Adoption
IP4.noFailure → OP4.valueSubtle
IP4
OP4
REG
C2.2:Condition
IP5.noFailure → OP5.noFailure
CON
IP5
OP5
C2.3:AR Guided Task
IP6.valueSubtle, IP7.noFailure → OP7.omission
IP6
IP7
OP7
<<FPTC failure>> noFailure

Fig. 7. Qualitative analysis of the system.

failure mode while its input can be different failure modes with different probabilities and all the possible conditions should be considered in the calculation. In this example, probability of failure occurrence in system output (human function) is 0.9, which shows that the reliability of the system from time 0 up to time 1000 hours is around $1 - 0.900 = 0.100$. The probability for omission failure mode will be $0.9 * 0.9875 = 0.88875$.

- **Interpreting the results**: Based on the back propagation of the results, we can explain how the hazard would happen and how much is the probability. For example, in this scenario the probability of omission failure mode in output is 0.88875 and the reason is high probability of failure in organization and regulation AR adoption.
  Similar to the previous scenario, safety goal can be defined in order to decrease the probability and prevent the risk. The probability can be helpful to decide if a special failure mode in the system output should be overcome or it can be ignored due to low probability of its occurrence.
  By using the quantitative analysis and by considering various possible scenarios, various safety goals can be defined. Based on safety goals, safety requirements can be defined and system design can be updated. Then, analysis of system behavior can be performed for more iterations to reach the accepted level of safety.

V. Discussion

As it is shown in the case study, there are human functions extended by augmented reality (directed paying attention) and there are AR-related organizational factors (organization and regulation AR adoption and AR guided task). Using modeling elements characterizing AR-extended human functions and AR-related organizational factors in modeling and analyzing the system provides the possibility to include their effects on system behavior while performing the analysis. In the extended metamodel used in our process, there are various modeling elements, which can be helpful in order to incorporate new features of organizations and their effects on human. Globalization, digitalization and networking structure of organizations are also considered. There are various factors leading to post normal accidents discussed in [21] and these factors are included in the extensions as it is explained in [11]. For example, industrial strategy is an organizational modeling element, which can be used to incorporate effect of industrial strategy on system behavior. A failure in industrial strategy can influence on human performance and can lead to system failure. Thus, it is important to model this factor in system modeling and it is crucial to consider its effects on analysis while we analyze system behavior.

Similar to the modeling and analysis capabilities for components, modeling and analysis constructs can be used for modeling and analysis of connections between components. It is an important feature, because based on accident reports there are a lot of situations that failure in the system is not caused by failure in components, but it is caused by failure in connections between components. It is important that we consider various scenarios including ones which system failure is emanated from failures in connections between components.

Analysis results can be used for preparing safety case and arguments to show that a system is acceptably safe. It is required to have several analysis iterations and brainstorm the possible scenarios and possible failures for all components, subcomponents and connections.

VI. Related Work

In [22], a framework is proposed for integrated socio-technical enterprise modelling. In this framework, social as-

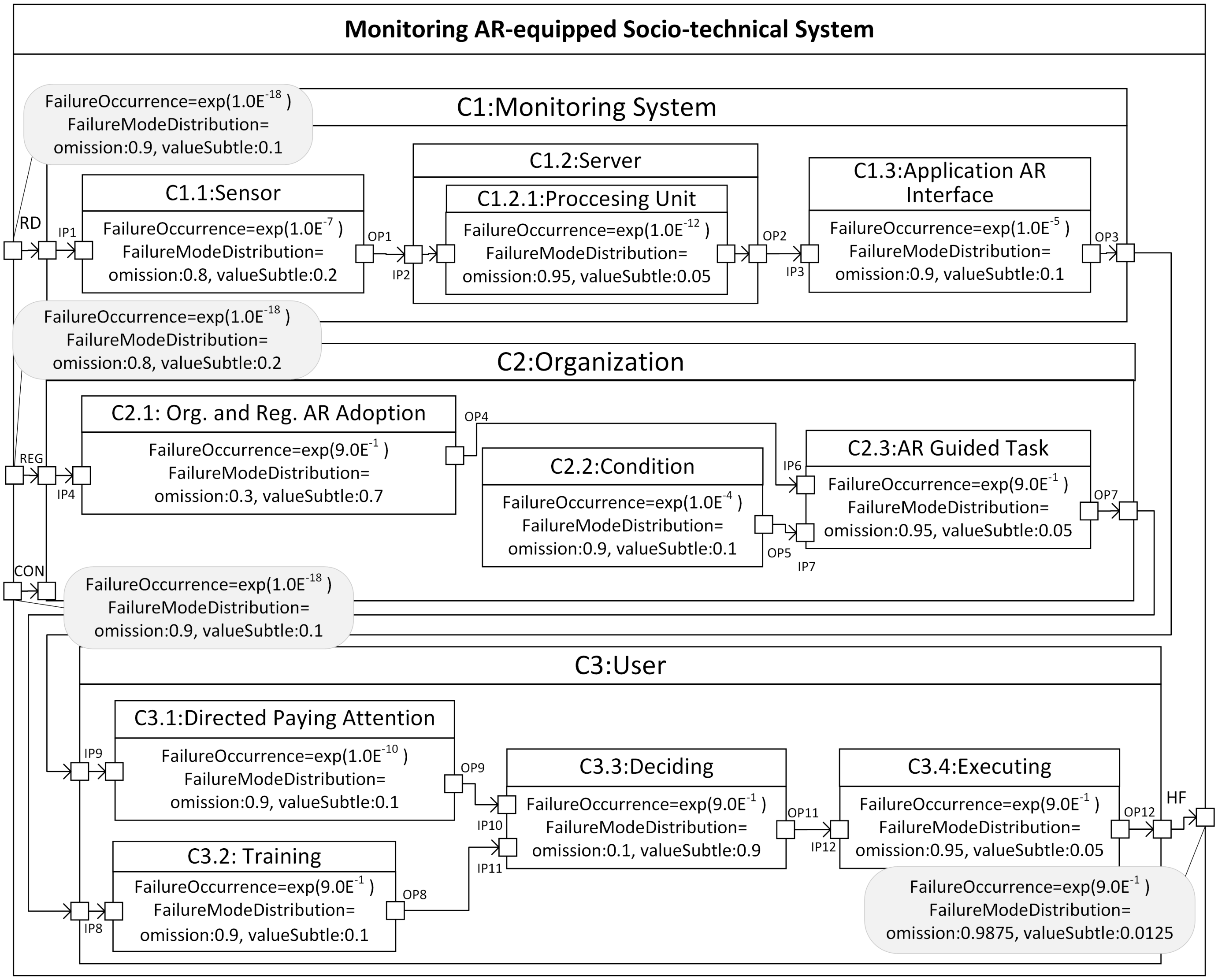

Monitoring AR-equipped Socio-technical System
C1:Monitoring System
FailureOccurrence=exp(1.0E-18)
FailureModeDistribution=
omission:0.9, valueSubtle:0.1
RD
IP1
C1.1:Sensor
FailureOccurrence=exp(1.0E-7)
FailureModeDistribution=
omission:0.8, valueSubtle:0.2
OP1
IP2
C1.2:Server
C1.2.1:Proccesing Unit
FailureOccurrence=exp(1.0E-12)
FailureModeDistribution=
omission:0.95, valueSubtle:0.05
OP2
IP3
C1.3:Application AR Interface
FailureOccurrence=exp(1.0E-5)
FailureModeDistribution=
omission:0.9, valueSubtle:0.1
OP3
FailureOccurrence=exp(1.0E-18)
FailureModeDistribution=
omission:0.8, valueSubtle:0.2
C2:Organization
C2.1: Org. and Reg. AR Adoption
FailureOccurrence=exp(9.0E-1)
FailureModeDistribution=
omission:0.3, valueSubtle:0.7
REG
IP4
OP4
C2.2:Condition
FailureOccurrence=exp(1.0E-4)
FailureModeDistribution=
omission:0.9, valueSubtle:0.1
OP5
IP6
IP7
C2.3:AR Guided Task
FailureOccurrence=exp(9.0E-1)
FailureModeDistribution=
omission:0.95, valueSubtle:0.05
OP7
CON
FailureOccurrence=exp(1.0E-18)
FailureModeDistribution=
omission:0.9, valueSubtle:0.1
C3:User
C3.1:Directed Paying Attention
FailureOccurrence=exp(1.0E-10)
FailureModeDistribution=
omission:0.9, valueSubtle:0.1
IP9
OP9
C3.2: Training
FailureOccurrence=exp(9.0E-1)
FailureModeDistribution=
omission:0.9, valueSubtle:0.1
IP8
OP8
C3.3:Deciding
FailureOccurrence=exp(9.0E-1)
FailureModeDistribution=
omission:0.1, valueSubtle:0.9
IP10
IP11
OP11
IP12
C3.4:Executing
FailureOccurrence=exp(9.0E-1)
FailureModeDistribution=
omission:0.95, valueSubtle:0.05
OP12
HF
FailureOccurrence=exp(9.0E-1)
FailureModeDistribution=
omission:0.9875, valueSubtle:0.0125

Fig. 8. Quantitative analysis of the system.

pects in addition to technical aspects, internal and external aspects are considered. Eight constructs such as goal, structure, task, etc. are mapped to enterprise models such as goal model, organizational model and process model respectively. The framework is illustrated on a case study from healthcare industry.

In [6], a framework is proposed for conducting the analysis of cyber-socio-technical systems. Concepts and a language are developed to characterize varieties of entities from a knowledge management perspective. Effects of modern challenges of digitalization on organizations and systems in various domains are included in the proposed framework.

Similarity of these studies with our work is consideration of social aspects and their interactions between various entities. The difference is that we also consider augmented reality effects on different socio aspects and its effects on system behavior in general. We incorporate augmented reality effects in the modeling and analysis process.

In [23], a literature review on various studies of risk management on socio-technical systems with the existence of digital transformation is proposed. Various studies are identified and they are categorized based on the steps they have considered for risk management and if human, organization and technology are considered in these steps. The results show that the researches are increasing on human and organization in addition to technology. However, in the risk controlling step, approaches considering all dimensions of socio-technical systems, are required. In our study, we considered all dimensions of socio-technical systems in risk identification, calculation of failure propagation and system behavior analysis. In addition, we considered effect of augmented reality on various parts of socio-technical systems.

VII. Conclusion

In this paper, we proposed an extension on the synergy of qualitative and quantitative dependability analysis techniques by incorporating AR and socio aspects. We presented this extension by an extended process. In the proposed extended process, we used extended metamodels for capturing AR-related aspects and considered their effects on system behavior. By implementing the proposed process in the CHESS toolset, it is possible to automatically calculate the failure propagation and failure mode probabilities for AR-equipped socio-technical systems. We illustrated the proposed process for analysis on an industrial monitoring system.

Further research is required to show the potential of the proposed process in more complex case studies within different domains. In addition, we plan to evaluate our proposed process by preparing a questionnaire and collecting expert opinions.

References

[1] 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.

[2] D. Van Krevelen, R. Poelman, A Survey of Augmented Reality Technologies , Applications and Limitations, The International Journal of Virtual Reality (IJVR) 9 (2) (2010) 1–20.

[3] D. Shcmidt, Guest editor's introduction: Model-driven engineering, IEEE Computer 2 (39) (2006) 25–31.

[4] D. Stewart, J. J. Liu, D. Cofer, M. Heimdahl, M. W. Whalen, M. Peterson, AADL-Based safety analysis using formal methods applied to aircraft digital systems, Reliability Engineering & System Safety 213 (2021) 107649.

[5] P. H. Feiler, D. P. Gluch, Model-based engineering with AADL: an introduction to the SAE architecture analysis & design language, Addison-Wesley, 2012.

[6] R. Patriarca, A. Falegnami, F. Costantino, G. Di Gravio, A. De Nicola, M. L. Villani, WAx: An integrated conceptual framework for the analysis of cyber-socio-technical systems, Safety science 136 (2021) 105142.

[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.

[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.

[9] 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.

[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.

[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] CHESS, ARTEMIS-JU-100022 – Composition with guarantees for high-integrity embedded software components assembly, http://www.chess-project.org (2010).

[13] L. Montecchi, P. Lollini, A. Bondavalli, A reusable modular toolchain for automated dependability evaluation, in: Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2013, pp. 298–303.

[14] G. Balbo, Introduction to Stochastic Petri Nets, in: School organized by the European Educational Forum, Springer, 2000, pp. 84–155.

[15] D. F. Haasl, N. H. Roberts, W. E. Vesely, F. F. Goldberg, Fault tree handbook, Tech. rep., Nuclear Regulatory Commission (1981).

[16] 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: 2018 14th European Dependable Computing Conference (EDCC), IEEE, 2018, pp. 49–56.

[17] CHESS-SBA, CHESS State-Based Analysis, https://ic.unicamp.br/ leonardo/tools.html (2021).

[18] 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.

[19] M. Wallace, Modular architectural representation and analysis of fault propagation and transformation, Electronic Notes in Theoretical Computer Science 141 (3) (2005) 53–71, proceedings of the Second International Workshop on Formal Foundations of Embedded Software and Component-based Software Architectures (FESCA).

[20] D. Pavlov, I. Sosnovsky, V. Dimitrov, V. Melentyev, D. Korzun, Case study of using virtual and augmented reality in industrial system monitoring, in: 2020 26th Conference of Open Innovations Association (FRUCT), IEEE, 2020, pp. 367–375.

[21] J.-C. Le Coze, Post Normal Accident: Revisiting Perrow's Classic, CRC Press, 2020.

[22] A. Fayoumi, R. Williams, An integrated socio-technical enterprise modelling: A scenario of healthcare system analysis and design, Journal of Industrial Information Integration 23 (2021) 100221.

[23] J. S. Menzefricke, I. Wiederkehr, C. Koldewey, R. Dumitrescu, Socio-technical risk management in the age of digital transformation-identification and analysis of existing approaches, Procedia CIRP 100 (2021) 708–713.