IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 2, 2020, accepted August 28, 2020, date of publication September 3, 2020, date of current version September 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021460

Analyzing Safety of Collaborative Cyber-Physical Systems Considering Variability

NAZAKAT ALI, (Graduate Student Member, IEEE), MANZOOR HUSSAIN, AND JANG-EUI HONG

Laboratory for Software Engineering, Department of Computer Science, Chungbuk National University, Cheongju 28644, South Korea

Corresponding author: Jang-Eui Hong (jehong@chungbuk.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (MSIT), Korea Government under Grant 2020R1A2C1007571 and Grant 2017M3C4A7066479.

ABSTRACT Cyber-Physical System (CPS) is co-engineered interacting networks of physical and computational components that operate on different spatial and temporal scales. The safety goal of a single CPS is usually achieved by applying hazard analysis techniques and by following the standard processes defined in ISO 26262 and IEC 61508. However, the safety property may not be satisfied when multiple CPSs collaborate due to complexity, uncertainty, and variability. Therefore, a technique that would provide a hazardous-free collaboration for multiple CPSs is required to preserve sustainability. In this paper, we analyze the hazards arising due to variabilities in collaborative CPSs. We extend the hazard analysis techniques (FTA, FMEA, and ETA) to explore hazards with variability and developed a fault traceability graph from our extended techniques to trace the faults considered by multiple hazard analyses in collaborative CPSs with variability. To justify our proposed approach, a case study on the human rescue robot system was conducted to analyze hazards emerging as a result of variabilities. Finally, a tool (CPS Tracer) was developed to model the FTA, ETA, and FMEA with variability (v_FTA, v_FMEA, and v_ETA). It also and generates the fault traceability graph (v_FTG) that represents fault propagation route.

INDEX TERMS Variability, cyber-physical system, SOTIF, safety, hazard analysis techniques.

I. INTRODUCTION

Cyber-Physical Systems (CPSs) are highly connected and massively networked systems composed of cyber (computation and networking) and physical (sensors and actuators) components that interact with each other in order to achieve the goals of, enhance reliability, efficiency, robustness, and sustainability when dealing with a specific task [1], [2]. CPSs cover autonomous and adaptive operations due to their properties of robustness and heterogeneity.

Safety in a single CPS can be ensured by applying standards such as ISO 26262 and IEC 61501. These standards define functional safety and safety integrity level to confirm the degree of safety-related to CPS's fail-safe and to describe a risk-based methodology for determining the safety integrity level for CPS. Despite this, CPS safety remains a thorny challenge as mentioned in [3], [4]. One of the major problems is to ensure the safety of a collaborative CPS: a system where multiple CPSs collaborate to complete a specific mission. Safety for a collaborative CPS may not be satisfied due to complexity, uncertainty, and variability [5]. Therefore, designing a collaborative CPS is one of the challenging tasks due to variable operating environment and a diverse set of heterogeneous computing and communicating devices. The collaborative CPSs may not be working in a controlled environment and have to operate in a robust way to cope with uncertainty. The uncertainty can be originated either from the unintended behavior of a failure-free system due to its performance limitations, lack of robustness with respect to environmental influences that might disturb sensors, or due to insufficient situational awareness.

The associate editor coordinating the review of this manuscript and approving it for publication was Lorenzo Ciani.

A. MOTIVATION

The uncertainty emerging from CPSs collaboration critically calls for safety of collaborative system. The collaboration makes system safety to an inevitable challenge because CPSs operate in a physical environment. Therefore, safety mechanisms and fault mitigation methodologies must be designed into the system at a design time.

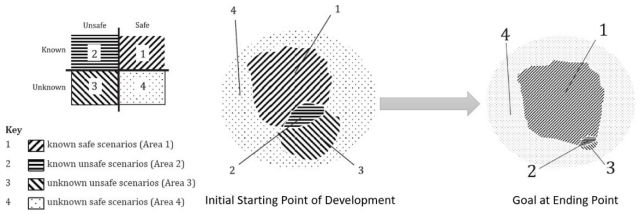


FIGURE 1. Scenario classification and application goals of SOTIF [4].

In order to cope with uncertainties emerging due to unintended behavior, a new standard namely SOTIF (Safety of the Intended Functionality) also known as ISO/PAS21448 [6] was introduced in January 2019. The goal of SOTIF is to maintain or maximize area 1(known-safe) while minimizing area 2 (known-unsafe) and area 3 (unknown-unsafe) to improve the safety of a system as shown in Fig. 1. The focus on area 2 (known-unsafe) with technical measures can minimize risks to an acceptable level and can shift hazardous scenarios to area 1(known-safe) by improving the system functionalities. The research focus on area 3 (unknown-unsafe) can minimize potential risks as much as possible with an acceptable level of effort. Efforts may include improving the performance of algorithms, improving the perception of algorithms, or analyzing the safety of the system by considering potential variabilities of the environment.

Our work falls under these efforts to maximize area 1(known-safe) by minimizing area 2 (known-unsafe) and 3(unknown-unsafe). We extended hazard analysis techniques such as Event Tree Analysis (ETA), Fault Tree Analysis (FTA), and Failure Mode and Effect Analysis (FMEA) to analyze the safety of the system with variabilities. This can identify associated risks with different variabilities of the environment and try to mitigate the associated risks by recommending safety guards. The content relationship among multiple hazard analysis techniques (ETA, FTA, and FMEA) is defined in order to know fault propagation in the collaborative CPSs, and also to discover the source of faults in the system.

Runtime behavioral models for collaborative adaptive systems is discussed in [7]. The authors proposed a framework to develop safe and secure adaptive collaborative systems with runtime safety guarantees. The aim of this framework was to develop runtime behavioral models for collaborative adaptive distributed systems, hazard analysis techniques for continuous safety and cyber security assurance, with real-time safety guarantees for the assumptions made in the model. To enable this, it is required to design behavioral models, and techniques to analyze and check safety and cyber-security at both design time and runtime. The analysis of these models is proposed to be executed in a cloud-based platform capable of providing real-time safety guarantees. Particularly, safety requirements are needed to be identified and analyzed in adaptive collaborative systems. The authors considered a platoon-based collaborative driving example to validate their study. However, the uncertainty due to variabilities was not considered in conducting a platoon driving case study. Rerouting of the platoon in case of an accident, and cyber-attacks to the communication of the platooning system were considered in platoon driving case study.

Developing safety cases requires gathering safety evidence during the development of a safety-critical system to ensure the identified failures are addressed and unwanted interactions between systems and environment are considered. In [8] Medawar *et al*. have proposed an approach that provides cooperative safety in CPS platooning within the safeCOP project. The safeCOP project was developed to analyze safety-critical systems that need to provide safety assurance - the clear safety arguments assuring that CPS system is acceptably safe. The authors argue that safety in cooperative CPSs is a challenging task because all the cooperative systems have to operate in a safe mode to achieve a single task. The continuous safety assurance was handled through the runtime manager of truck platooning use case. In order to fully utilize the runtime manager, the safety contracts based on the safety analysis of the local system and overall cooperative system need to be specified. Such safety contracts are first checked during design time to ensure their validity. The context under which safety contracts are checked for their validity changes on cooperating CPSs such as truck platoons. Therefore, safety contracts need a continuous check whether the safety contracts are violated and provide a healing strategy to behave in case of violation.

However, the focus of above study is about cooperative work rather than a collaboration. The collaborative CPSs also pose other safety concerns such as interaction uncertainties, and dynamic environmental variabilities [4] that need to be analyzed in detail.

B. CONTRIBUTIONS

In summary, we make the following major contributions in this paper.

- First, we analyze the collaborative behavior of CPSs and extract variability factors that potentially trigger uncertainties at runtime.
- After identification of variability factors, hazard analysis is required to mitigate hazards related to the identified variability. Therefore, we extended hazard analysis techniques such as FTA, FMEA, and ETA for variability.
- In collaborative CPSs, hazard traceability and propagation are critical to know the source of hazards and their effects in the system. We developed a fault traceability graph with variability (v_FTG) to trace the faults among multiple hazard analyses in collaborative CPSs.
- A tool (CPS Tracer) is developed that supports modeling FTA, ETA, and FMEA with variability (v_FTA, v_ETA, and v_FMEA), and generates FTG with variability also known as v_FTG.

The remaining part of this paper is structured as follows: Section II presents a literature review in our research field. In section III, we present our proposed approach,

and the evaluation of the proposed approach is presented in section IV. Finally, section V concludes this paper with some research directions.

II. RELATED WORK

The need to organize collaboration of various heterogeneous systems leads to the emergence of self-organizing, self-adaptive, and self-healing systems, with a higher level of collaboration among systems. This ultimately leads to the concept of collaborative CPSs which combines the concept of autonomy and collaboration aspects [9]. Brings *et al.* [10] have proposed an automated approach for reasoning about the relations between goals and configurations of collaborative CPSs. The proposed approach depends on model transformation generative view on configuration and goal models which enables system developers to see how goals and configurations impact each other. The proposed approach presents an explicit specification of the participant system goals that arise in the interplay with the connected systems and individual systems aim at accomplishing in collaboration, in a devoted goal model. The authors also presented a traceability concept that connects the goal model and configuration model in order to see the nature of a configuration while accomplishing certain goals.

Leite *et al.* [11] have presented a taxonomy which facilitates research on safety challenges and solutions for collaborative CPSs. The authors extended the taxonomy of Avizienis *et al.* [12] and redefined the meaning of the term function, behavior, functional specification, and service with respect to system collaboration. The authors argued that uncertainty is the biggest challenge that arises due to collaboration among multiple CPSs. Furthermore, the authors presented safety contracts and dynamic risk management as a solution concept to cope with uncertainty.

Törsleff *et al.* [13] proposed an approach for modeling the context of collaborative CPSs. The authors generated an ontology that systems use at runtime to communicate with each other and perform specific context-related reasoning. The proposed ontology was made considering two important properties of CPSs. First, the participant CPS of collaborative CPS may leave the collaboration at run time or a new CPS may join the collaboration at runtime. This property has been referred to as openness of the context. Second, the property of the context of CPS may change at runtime. This property is referred to as the dynamicity of the context. Both properties are one of the foundations of collaborative CPSs where the operating environment can change frequently, and CPS has to adapt its behavior to achieve its goal. The proposed approach considers above two properties of collaborative CPSs to design a safe system. The proposed approach is divided into three steps: 1) modeling the interaction between CPSs, 2) building the ontologies that are used by participant CPSs at runtime, and 3) modeling dynamic context of CPSs. Ensuring the safety of a CPS is a challenging task. However, ensuring safety for collaborative CPSs is the super challenging due to heterogeneity, complexity, interoperability, and variability.

Khalid *et al.* [14] proposed an approach for collaboration between humans and CPSs in the production industry. The authors identified the requirements of human-robot collaboration in a production environment. In collaborative robotic CPS, the human worker is considered to be an integrated part of the system. The human worker interacts with other CPS components in a fully interconnected way. In this study, the human worker cooperates with CPS in order to solve a separated portion of the problem. Lastly, the summing-up of each activity that is performed by each participating system can solve the problem as a whole. However, collaboration is a coordinated and synchronous activity that solves a commonly shared problem.

Daun *et al.* [15] report on findings gained from a case study conducted together with participants from the avionics industry (innovative collision avoidance system) to know the unsafe behavior that emerges from the collaboration of various instances of the same system type of a collaborative CPS. The authors conducted a case study where a collision-avoidance system (CAS) participates in a collaborative system network with various CAS to prevent potential aircraft collision. The CAS instance flight data is used to determining potential collision hazards. As a result of investigation, the authors identified several research challenges including automatic validation of instance configuration at run-time, design-time support for validating and verifying potential unsafe behavior, and automatic generation of multiple instance configurations for collaborative CPSs.

Ou *et al.* [16] have proposed a *SafeTrace* framework that has the ability to manage traceability among safety requirements, design, and safety analysis in the medical device plug-and-play system. The authors investigated the links between hazards analysis artifacts, requirements artifacts, and design artifacts to see whether a change of requirements or design may cause a safety violation in the system. Specifically, the proposed framework defined trace links between design documents and basic events in FTA and between requirements and top events of each tree. Once, the relationships have been identified, an impact-analysis algorithm was used to identify the effects on the safety analysis of the system that are caused due to a change in requirements and design artifacts. The authors used a case study on an airway laser-surgery system to validate their *SafeTrace* framework. Kim *et al.* [17] developed NuDE 2.0 tool for verification and safety analysis environment for safety critical systems. However, this tool does not consider collaboration among multiple CPS.

Human safety in a human-robot collaborative production system is important. Therefore, at runtime, the behavior of a robot should be adaptable depending on the human actions as discussed in [18]. The author in [18] developed a CPS architecture for human-robot collaboration and investigated its capabilities to ensure human safety in a production environment. The architecture considers a shared fenceless working space where industrial robots, humans, or other moving objects, such as auto-guided vehicles, may operate. However,

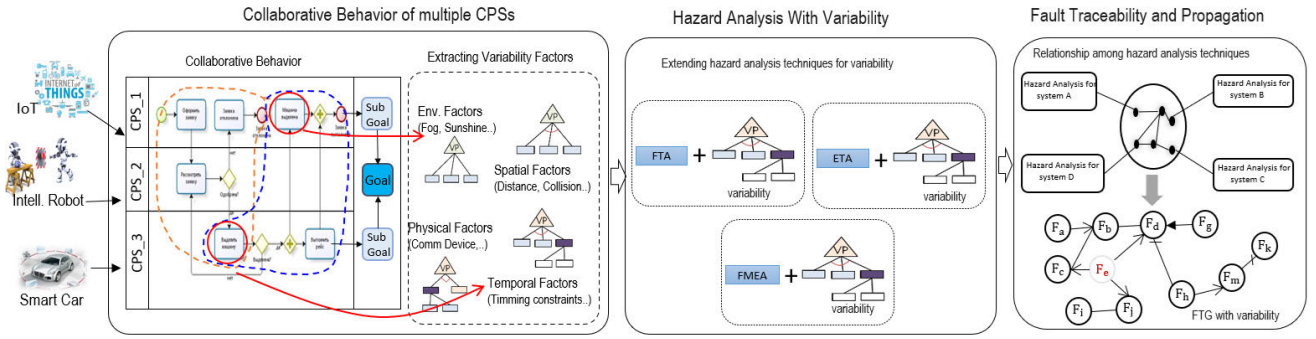


FIGURE 2. Proposed approach.

the main aim of this study was to find the response time needed to detect the human and activate the safety signal. Short detection time may lead to activating rapid collision avoidance strategies. Conversely, a high response time of a detection system to detect humans and activating a safety signal could be hazardous for an operator.

The proposed architecture was implemented in a particular use case in a laboratory machine shop with an industrial robot. The authors argued that proposed architecture, with its low cost and detection time performance, has enough capability to be used for human-robot collaboration. However, it has limitations such as the use of a Kinect camera sensor which has a limited field of view, can raise concerns on the validity of results. Furthermore, communication safety, uncertainty, and variability were not considered in the implementation.

III. PROPOSED APPROACH

When multiple CPSs collaborate with each other to accomplish a common goal, the safety property may not be ensured due to complexity, variability, and heterogeneity. In CPSs, hazard analysis allows safety engineers to rectify insufficiencies, identify failures, and provide information on essential safety guards [19]. Therefore, we propose an approach in order to analyze hazards considering variability as shown in Fig. 2. First, we identify the potential variability factors by considering the collaborative behavior of multiple CPSs and extended the hazard analysis techniques (FTA, FMEA, and ETA) to analyze the hazards related to the variabilities. Then, we have defined relationships among hazard analysis techniques (FTA, FMEA, and ETA) to visualize the fault relationships and to identify critical faults in collaborative CPSs. For visualization, a Fault Traceability Graph (FTG) was developed in order to represent the relationship between faults and/ or safety guards [20]. We extended FTG also known as v_FTG to reflect variabilities. We explain our proposed approach as follows:

A. COLLABORATIVE BEHAVIOUR OF MULTIPLE CPSs

CPSs are autonomous and adaptable technical systems. In a collaborative environment, every CPS contributes its part to accomplish a particular task, e.g. multiple electrical vehicles in a virtual power plant collaborate to cope with power

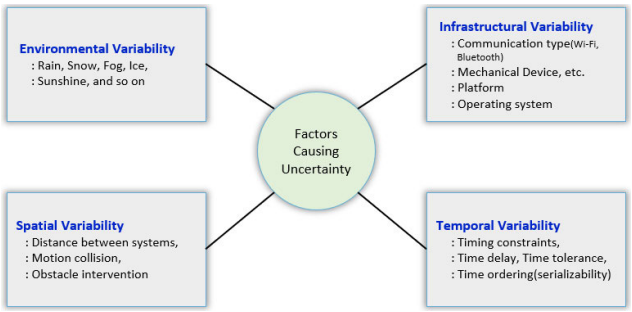


FIGURE 3. Variability factors causing uncertainty.

shortage in peak hours [13]. In contrast to the traditional systems, the variability in CPSs is difficult to model because they have to operate in a robust environment where variabilities are expected to occur frequently. Hence, we argue that one of the main reasons for the misbehavior of a collaborative CPS can be the variability in its environmental context, requirements, or operating scenarios. Considering collaboration among multiple CPSs, we classified variability factors into four types that can cause uncertainty in CPSs as shown in Fig. 3. In order to classify these variable factors, we first divide universal space with scope of CPS factors and scope of non-CPS factors. Then the CPS scope is divided into single CPS related factors and inter-CPSs related factors. As a result, environment variability was defined as the non-CPS scope, infrastructure variability as a single CPS scope, and spatial and temporal variabilities were defined as inter-CPSs scope. These variability types can be classified into four or more types; however, the proposed classification reflects the characteristics of mutually exclusive and collectively exhaustive that can contain all possible uncertainties. The variable factors include environmental factors which are the contextual variabilities in a CPS e.g. fog, rain and ice, etc. The context of a CPS can be changed frequently, and the system has to adapt its behavior accordingly to achieve its final goal while ensuring safety. The variabilities can also occur due to infrastructural variability factors such as communication type, or other heterogeneous hardware devices, operating systems of participant CPSs, etc. Variability can also occur due to variability in spatial variability factors e.g. distance between

two or more participant systems, signal coverage variabilities of two communicating systems, etc. The spatial variabilities occur due to the variabilities in the model perceptions e.g. unintended braking of a self-driving car could be caused by limitations in the perception system due to model uncertainty. The temporal variabilities can occur due to time of variability, meaning that a CPS can be adaptable in runtime or design time. In some CPSs, the variability can be adapted at design time. On the other hand, some CPSs have the ability to adapt itself during runtime [21].

The CPSs are heterogeneous in nature, as they consist of various hardware and software components. These heterogeneities originate due to lack of modeling a wide range of variable parts, variability representation differs in different CPSs, and safety interpretations differ in different systems. Therefore, we identified four variability factors that must be considered at the design time of a CPS. When multiple CPSs collaborate, these variabilities pose serious safety concerns. Therefore, hazard analysis for a CPS with variability is also a critical challenge. In order to cope with this challenge, we have extended hazard analysis techniques (FTA, FMEA, and ETA) to analyze systems with variability.

B. HAZARD ANALYSIS WITH VARIABILITY

Hazard analysis enables safety engineers to discover the potential failures, their consequences, and potential safety guards to mitigate the failures. Recently several hazard analysis techniques had been adopted for safety analysis [22]. We choose FTA, ETA, and FMEA for our study. These are well-proven and strong hazard analysis techniques that ensure the safety of the target systems. The drawback of these hazard analysis techniques is the analysis complexity [22]. The current FTA, ETA, and FMEA do not support an effective way to analyze hazards related to the variabilities. However, these techniques support hazards analysis in a general way but not as specific as our extended approach does. In our approach, variability models are maintained independently which can grow over time. This provides an effective way to analyze the hazards and reduce the analysis complexity of FMEA, ETA, and FTA.

1) EXTENDING FTA FOR VARIABILITY

FTA is a very popular technique to analyze the hazards related to safety. It is designed to analyze systems to find the root cause of potential failures [23]. FTA is comprised of a wide variety of modeling and analysis techniques, supported by a wide range of software tools [24]. We extended the FTA by introducing a new gate known as the variability gate. In a variability gate (Variability OR gate and Variability AND gate), the output comes about if an input, initiated by at least one or/and more variability factors, occur. We introduced the following new elements to model the variability in FTA.

- *Variability Point (VP) Node*: The VP is the top node of the variability model that connects the variable nodes.
- *Variable Node*: A node that represents a variable element that contributes to the failure of the system.
- *v-OR gate*: In variability OR gate, the output comes about if an input, initiated by at least one or more variability factors, occur.
- *v-AND gate*: In variability AND gate, the output comes about if an input, initiated by more variability factors at a time, occurs.

Fig. 4 shows an example of variability modeling in FTA.

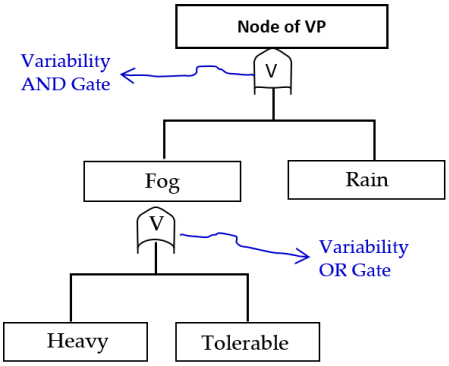


FIGURE 4. Variability model for v_FTA.

The advantage of maintaining a separate variability model reduces the complexity of an FTA for large systems. The variability gate in FTA can discover the exact cause of an event. For instance, the cause of *collision event* in robots can occur due to *broken of proximity sensor* or due to *inaccurate decision of proximity sensor*. However, the *inaccurate decision of proximity sensor* event needs more investigation in order to know its exact root cause. The *inaccurate decision of proximity sensor* event can occur due to a number of reasons. One of them can be environmental factors (fog, rain. snow, etc.). Therefore, the *inaccurate decision of proximity sensor* event is connected with environmental variability (variability point) to know the exact cause of the *inaccurate decision of proximity sensor* event. In this way, the variability in FTA enables us to discover more basic events of an intermediate event.

2) EXTENDING ETA FOR VARIABILITY

ETA is a top-down logical modeling technique that shows possible outcomes resulting from an initiating event [25]. ETA consist of an initiating event and a number of pivotal events. Pivotal events may include a sequence of basic events or intermediate events. The intermediate events need more investigation in order to know the possible outcomes. We introduce a new field i.e. *"Variable initiating event"* in the traditional ETA in order to accommodate the variability in ETA as shown in Fig. 5. This variability model for ETA is independent of general ETA and can grow our time. Hence, in the analysis time, we can connect general ETA to its variability model if necessary. For instance, a *collision event* (an initiating event) can occur if *broken of proximity sensor* event or/and the *inaccurate decision by proximity sensor* event occurs. The possible environmental factors that might affect the proximity sensor can be fog, sunshine, snow, and etc.

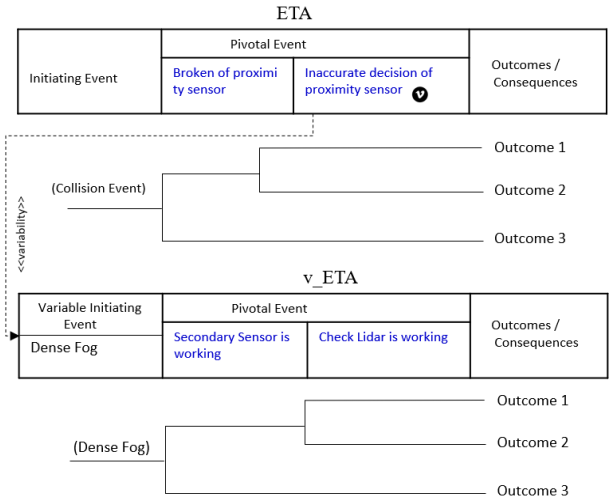


FIGURE 5. Variability model for ETA.

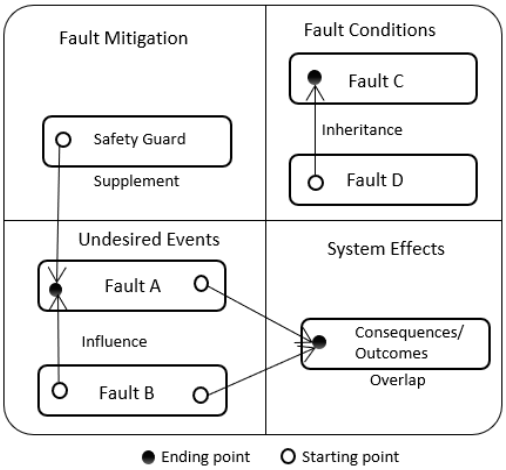


FIGURE 7. Relationship among hazard analyses techniques.

Therefore, *environmental factors* is the variability point and fog, snow, and sunshine, etc. can be variants of the variability point. Hence, an *inaccurate decision by proximity sensor* is connected with its variability point and each variant can be taken as a variable initiating event and constructed a new event tree.

3) EXTENDING FMEA FOR VARIABILITY

FMEA is an inductive hazard analysis technique that analyzes systems in detail [26]. FMEA is a tabular form that includes failure modes, causal factors, system effect, severity, etc. For large systems, the FMEA grows exponentially over time and thus makes it complex for the FMEA team to analyze the system. Furthermore, the current FMEA form does not allow to reflect variabilities of failure modes. Therefore, we made a new tabular form independent of FMEA for variability analysis as shown in Fig. 6. This variability management form can grow over time. Additionally, it can be used only when needed. In the newly introduced form, the *Failure Cause* of FMEA field is divided into *variability point* field and *variability* field in order to accommodate variability.

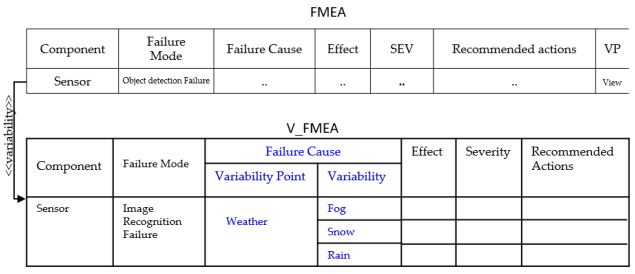


FIGURE 6. Variability model for FMEA.

C. HAZARD TRACEABILITY AND PROPAGATION

1) RELATIONSHIP BETWEEN HAZARD ANALYSIS TECHNIQUES

Hazard analysis techniques help to analyze the hazards that could lead to an undesired event resulting in, death, injury, damaging equipment, or environment. Several hazard analysis techniques [22] are available to analyze the system. Each hazard analysis technique has its own scope, purpose, focus, and methodology to address hazards in the system. Therefore, a composite approach that would consider the relationship among multiple hazard analysis techniques is required to perform them simultaneously. Hence, we defined relationships among FTA, ETA, and FMEA as shown in Fig. 7. We observed that four common elements such as an undesired event, fault effect, consequences/outcome of a fault, and fault mitigations or safety guards are usually present in FTA, ETA, and FMEA. These four common elements were used to identify the relationship among FTA, ETA, and FMEA as shown in Fig. 7. The definition of each relationship is as under:

Influence Relationship: A relationship in which a fault of one system/subsystem element, that participates in a common mission of multiple CPSs, affects the failure of another system/subsystem element. This relationship exists between the faults of the system(s).

Inheritance Relationship: Inheritance relationship exists when two or more collaborating CPSs share the same operational and functional constraints. This relationship exists between faults in collaborative multiple CPSs.

Overlap Relationship: Overlap relationship exists when the failure result of one system/subsystem element is the same as the failure result of another system/subsystem element. This relationship is proposed to be between faults and consequences in the collaborative CPSs.

Supplement Relationship: Supplement relationship exists when one system/subsystem has safety guards to cover a failure in another collaborative CPS, meaning that a safety guard for a failure in one system can be applied to another identical failure of a system. This relationship exists between safety guards and faults so that a safety guard for a failure will be used for the same failures in other collaborative CPSs.

Once, such relationships are established, an impact-analysis algorithm is used to generate an FTG.

2) FAULT TRACEABILITY GRAPH

Fault traceability provides a way to determine the flow of a fault in the system. Once, we established relationships among FTA, ETA, and FMEA, we developed FTG to trace the fault among multiple hazard analyses in CPSs [20]. In this paper, we extend FTG for collaborative CPSs with variability and named it v_FTG. Through v_FTG, we can see the source of a fault and its impact on the system. The v_FTG with variability gives more information on traceability of a fault and its criticality. The variability discovers more basic events that lead to the failure of the system. This information can be supplemented to other hazard analysis techniques to find the potential effects of that particular failure on the system and also to find potential safety guards to ensure the safety of the system. The extracted information from v_FTG can also be used to revise the safety requirements of a collaborative CPS [27].

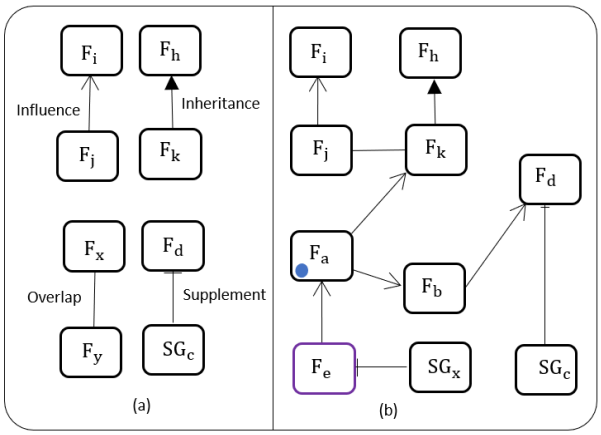


FIGURE 8. Notation (a) and example (b) of v_FTG.

The v_FTG is composed of nodes and edges. Nodes represent faults, failures, or a safety guard and the edges represent the relationship between/among nodes. Fig. 8 (a) shows the notation for the relationship in FTG while Fig. 8 (b) shows an illustrative example for FTG with variability. Node F_a is the fault influenced by the variability fault node F_e. The fault node F_a is marked with a blue filled circle indicating that the respective node has a relationship with some variability. For example, Fault F_a is influenced by a variable fault Fe. F_a when trigged, influences fault $F_{b,}$ and fault F_b influence F_d. The fault F_d is mitigated by supplementing a safety guard SG_c.

The modified algorithm to generate v_FTG including variability is mentioned in algorithm 1. The v_FTG is generated based on our defined content relationships among FTA, ETA, and FMEA [17]. Furthermore, the concept of a graph theory is utilized to visualize v_FTG [28].

Let, X (faults from FTA, FMEA, and ETA) and VX (faults from v_FTA, v_FMEA, and v_ETA) are the sets of faults we are interested in discovering their relationships i.e. supplement relationship (R1), influence relationship (R2), inheritance relationship (R3), and overlap relationship (R4). Let Z is the disjoint union set of X and VX. Therefore, Z $\{x: x \in X \vee x \in VX\}$. Let I$\{i_1, i_2 \ldots, i_k\}$ is the set of hazard analysis artifacts for instance failure modes, causal factors, and recommended actions, etc. in FMEA, undesired events in FTA, initiating events, outcomes, and pivotal events in ETA without variability. Let VI$\{vi_1, vi_2 \ldots, vi_k\}$ is the set of hazard analysis artifacts in v_FTA, v_FMEA, and v_ETA. Let Y is the disjoint union set of I and VI. Therefore, Y$\{i: i \in I \vee i \in VI\}$ and Y $\subset$ Z.

Algorithm 1 Algorithm to Generate FTG With Variability

```
Z {x : x ∈ I}
R1 (x, i_k) ← ∅ of Supplement Relationship
R2 (x, i_k) ← ∅ of In fluence Relationship
R3 (x, i_k) ← ∅ of Inheritance Relationship
R4 (x, i_k) ← ∅ of Overlap Relationship
foreach i ∈ Y do
    foreach x ∈ Z do
        R1(x, i_k) ← {i ∈ Y : {x, i_k} ⊢ Θ}
        R2(x, i_k) ← {i ∈ Y : C ≥
        min_confidence ∧ S ≥ min_support}
        R3(x, i_k) ← {i ∈ Y : |(x∩i_k)/(x∪i_k)| ≥
        threshold ∧ i.createdDate > x.createdDate}
        R4(x, i_k) ← {i ∈ Y : |(x∩i_k)/(x∪i_k)| ≥ threshold}
        If x ∈ VX ∨ i ∈ VI then
            Node.color.green
        else
            Node.color.default
        end
    end
end
v_FTG ← R1(x, i_k) + R2(x, i_k) + R3(x, i_k) + R4(x, i_k)
```

As we see, confidence C $\geq$ *min_confidance* and support S $\geq$ *min_support* where *min_confidance* and *min_support* are the corresponding confidence and support thresholds defined by safety engineers. The Confidence C is calculated from equation (1) and support *S* is calculated using equation (2). Support *S* measures how frequent a fault occurs with another fault or how often a safety guard is used to mitigate a fault in the whole causal chain history *R*. Confidence *C* defines the likeliness of occurrence of consequent fault *x* in hazard analysis that contains i_k in the causal chain history. It measures the reliability of the inference made by rule. For instance, for a given rule $x \rightarrow i_k$, the higher the confidence, the more likely it is for i_k to be occur in hazard analysis that contains *x*. By calculating confidence, safety engineers can estimate the conditional probability of occurrence of a fault.

$$Confidance\ C = \frac{frequency(Z, i_k)}{frequency(Z)} \tag{1}$$

$$Support\ S = \frac{frequency(Z, i_k)}{frequency(R_{size})} \tag{2}$$

IV. EVALUATION

A. USE CASE – HUMAN RESCUE ROBOT SYSTEM

In a situation such as fires and earthquakes, rescue teams are supposed to do very dangerous and hazardous work. Therefore, rescue robots are extremely needed to perform rescue operations instead of human rescue workers. The human rescuing robot system is consisting of three types of robots i.e. searching for victims, removing obstacles, and lifesaving robots. Each robot is considered as a single CPS collaborating with each other to rescue disaster victims. All these robots are controlled by a control station (CS). The rescuing robots are expected to be able to perform safe and delicate operations in order to rescue disaster victims. All three CPSs (searching robots, removing obstacle robots, and lifesaving robots) collaborate with each other to save human life. The role and responsibilities of each robot are described below.

- *Searching robot (SR)*: The searching robot searches for victims on the ground and sends its location to the obstacle removing robot and lifesaving robot.
- *Obstacle removing robot(OR)*: The obstacle removing robot gets the location of a victim from searching robot on the ground and detects the obstacles around the victim and approaches to obstacles and remove the obstacles. After completing its mission, obstacle removing robot sends a clearance message to the lifesaving robot.
- *Lifesaving robot(LSR):* The lifesaving robot approaches the victim and evacuates from the place of disaster to the safe zone.

In order to analyze the real-time behavior of human recue robot system, we used timed automata in Behrmann [29] as shown in Fig. 9. Uppaal is a tool for verification of real-time systems and its modelling language offers features like bounded integer variables, time urgency, synchronous and asynchronous channels, data types, etc. It also has a query language for model-checking. To preserve space and simplicity, instead of presenting the whole Upppaal models for our collaborative CPS case study, we present just collaborative behavior to ensure the designed CPS works being in safe states.

Our main focus is to analyze hazards that emerged as a result of variabilities. Therefore, we investigated the collaborative behavior of each CPS to extract variable factors that can cause uncertainty at the runtime. For evaluating the proposed approach, we have implemented two simple use case scenarios including communication variability and environmental variability.

Scenario 1: In the human rescue robot system, the SR notifies its location and status (engaged, idle, etc.) to the CS. It also sends the location of victims to OR and LSR, if found, and gets acknowledgment to ensure the communication. The SR can wait up to 10 seconds for a confirmation response from OR and LSR. Fig. 10 shows the interactions of SR with other robots and the control station.

Hazardous Case: Communication between SR and OR failed due to communication (infrastructural) variability:

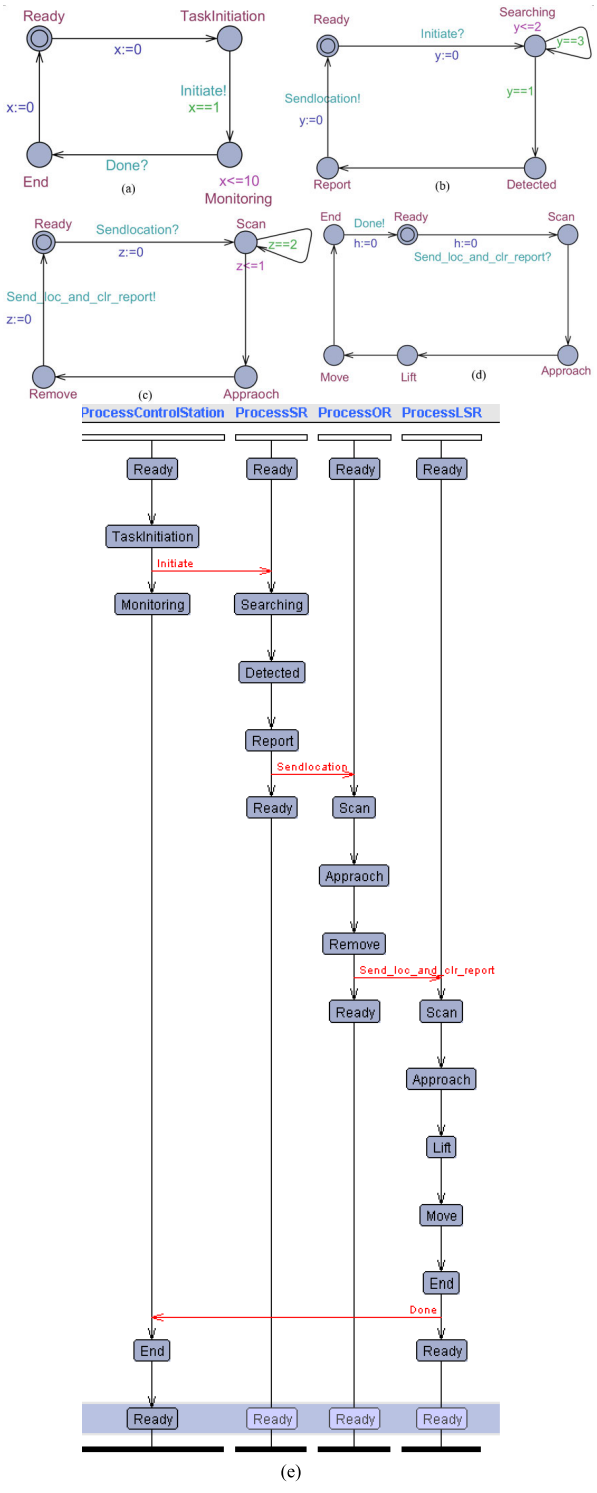


FIGURE 9. (a) CS, (b) SR, (C) OR, (d) LSR, and interaction (e).

We assume that the SR and LSR use the same communication type of the same manufacturer (same protocols with the same frequency). However, the OR has a different communication type (different frequency) manufactured by another company. When SR sends the location of victims to OR and LSR, it waits for 10 seconds to get a confirmation message from each robot. If response time exceeds 10 seconds, the SR sends

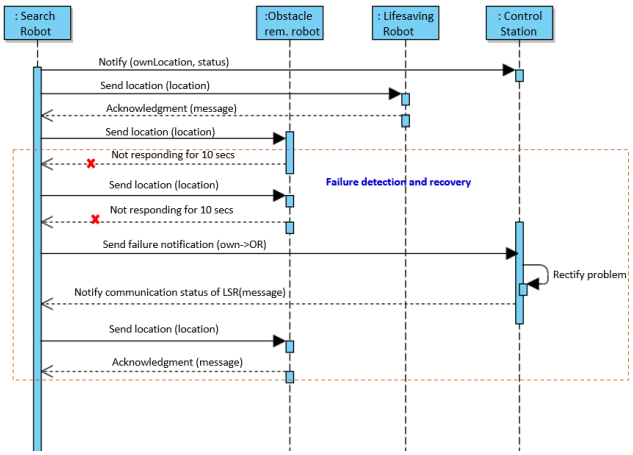


FIGURE 10. Interaction of SR with other robots.

a communication failure notification to the CS. An operator in the control station rectifies the problem (frequency difference). We assume that the operator found that SR and OR have communication variability meaning that both robots have different types of communication mode (difference in frequency). The operator reconfigures the communication protocols in OR and enables it for communication with other robots. The operator then sends a message to SR that the respective robot is now available for communication. The SR then start successful communication with OR and so on.

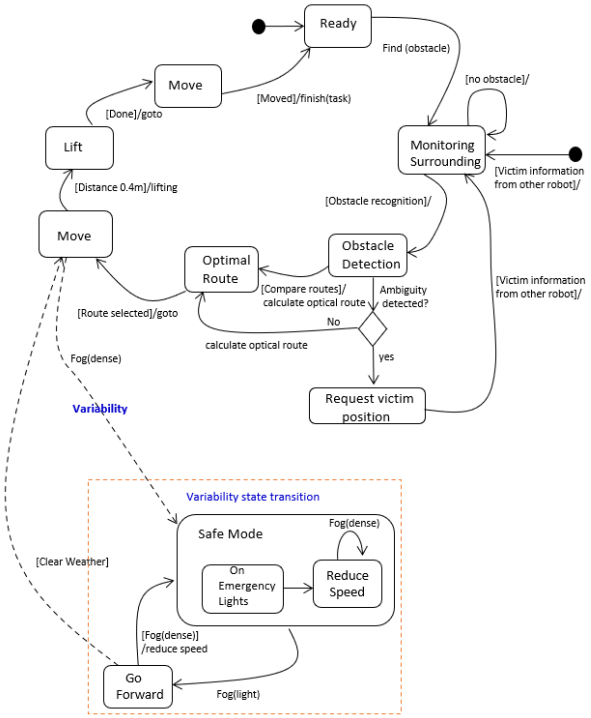


FIGURE 11. State transition diagram for OR with variability.

Scenario 2: OR is responsible for removing obstacles from the surroundings of a victim so that LSR would quickly rescue the victim without any hurdle. The state transitions for OR is shown in Fig. 11 where it receives victim location from SR and scans for obstacles surrounding the victim. When OR finds obstacles around the victim, it estimates the shortest path and starts moving towards the victim.

The dotted line in Fig. 11 shows a transition to safe mode in case of uncertainty. For instance, when OR was moving towards the victim with some speed and detect the fog on its way. As a result, OR goes to safe mode by turning on emergency lights and reduces its speed. The robot remains in a safe mode state as long as the dense fog does not vanish and returns to the normal state (move with the same speed) once the foggy situation ends. This kind of functional modeling can achieve sustainability in collaborative CPSs.

Hazardous Case: Potential collision event due to environmental variability: On the way to the victim's position to remove the obstacles around the victim, we assume that OR faces a dense fog which can decrease the perception of obstacle recognition, thus failing to brake and cause a collision. To ensure robot-safety at runtime, the behavior of the robot must become adaptable, depending on the environmental variabilities. Therefore, in case of dense fog, the OR can go to a safe mode and comes back to its normal state (move state with the same speed) when fog vanishes as shown in Fig. 11.

The victim's safety is ensured based on the victim's proximity to the operating robot. A collision risk detection module can support for activating the appropriate collision avoidance strategy.

B. HAZARDS ANALYSIS WITH CPSTracer

In this subsection, we analyze the hazards including hazardous cases in evaluation scenarios 1 and 2 with our developed tool. CPSTracer helps to analyze the potential hazards with variability. We use our extended FTA, FMEA, and ETA to analyze the hazards in our defined use case scenarios. For instance, several factors cause the failure of a human rescue robot system including environmental variabilities, the damage of one of the participant robots, the information flow of faults, infrastructural variabilities, and the collision of robots. We determined five intermediate events that can cause communication failure and robot collision within scenarios 1 and 2 as shown in Fig. 12. (level 1 of FTA).

Let's take the *Robot Crash* intermediate event as an example and analyze it in detail. The *Robot Crash* event can occur due to a *mechanical failure* or due to a *collision event*. The *collision event* can occur due to several reasons i.e. *broken of proximity sensor* event, *obstacle prediction uncertainty* and *inaccurate decision of proximity sensor* event mentioned in Fig. 12 (FTA_0). In general hazard analysis techniques, we usually do not consider the variable factors that lead to unexpected events at run time. For instance, unintended braking could be caused by the limitations in the perception system. The limitations for perception potentially would come from weather conditions (rain, for, snow, etc.). Therefore, we need to analyze the hazard of unexpected events. For instance, in Fig. 12 (FTA_0), the *Collison Event* occurs due to the *broken of proximity sensor*, *obstacle prediction uncertainty*,

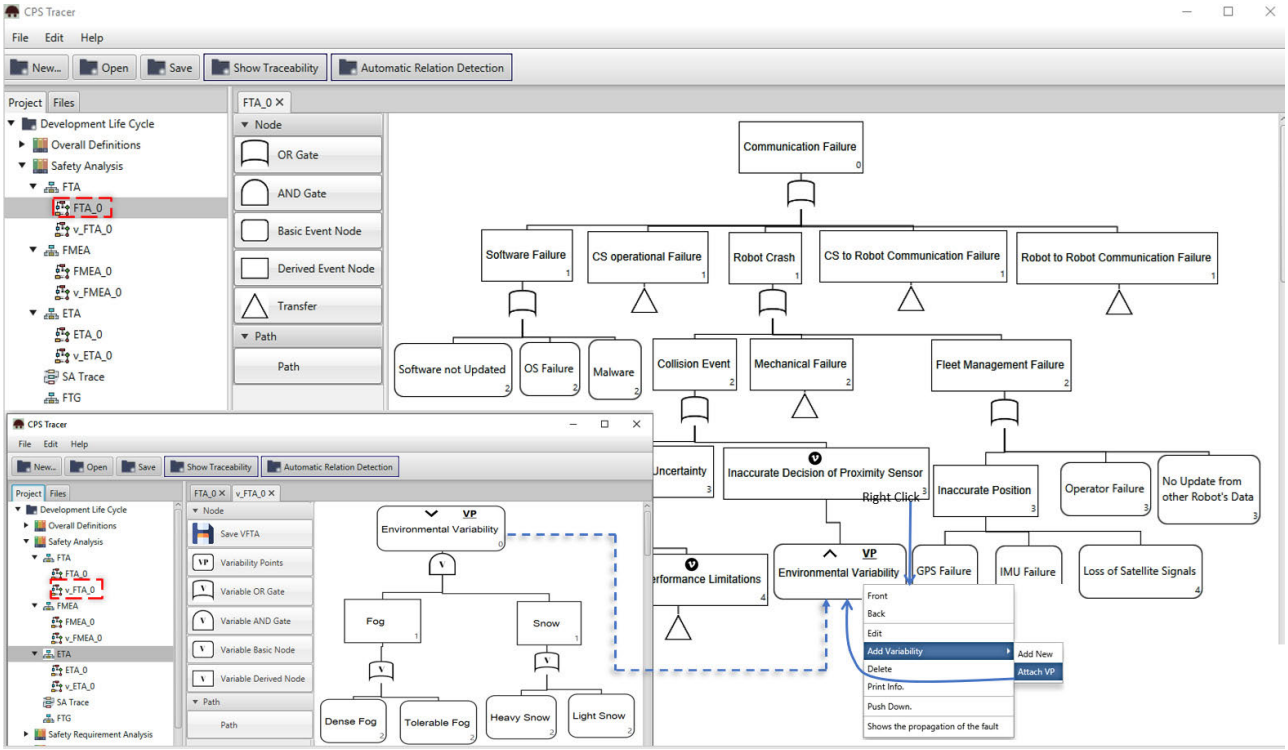


FIGURE 12. Excerpt of hazard analysis with FTA and v_FTA.

or *inaccurate decision of proximity sensor* event. However, we had not known what actually has caused the *inaccurate decision of proximity sensor* event. Similarly, when analyzing the *obstacle prediction uncertainty* event, we ended up with basic events such as *model parameter limitation* and *model performance limitation* events. The proposed variability modeling copes with such kind of problems and brings more accurate causes of the events. The *inaccurate decision by proximity sensor* event is further investigated and we found potential causes for that event. The *environmental variabilities* can influence the sensors of the robots for instance, fog can cause misinterpretation of obstacles in front of robots thus leads to a collision event as shown in Fig. 12 (v_FTA_0). Therefore, we attached environmental variability to the *inaccurate decision by proximity sensor* event which is indicated by ❼. Similarly, in Fig. 13, variability is shown for *Object Detection Failure* and *Robot to Robot Communication Failure*, which are failure modes of FMEA. Both *Object Detection Failure* and *Robot to Robot Communication Failure* are caused by weather conditions (environmental variability) and infrastructural variability respectively. The v_FMEA in Fig. 13 shows variability for *Object Detection Failure*. Likewise, variability *for inaccurate decisions by proximity sensor* is shown in Fig. 14 (v_ETA).

Compared to previous studies, our extended hazard analysis techniques provide modeling flexibility and integrated hazard analysis in addition to existing hazards analysis techniques (FTA, FMEA ETA, etc.). The modeling flexibility offers advantage to safety engineers that it can be used without changing the existing hazard models by easily adding and modeling unexpected events that were not considered in the initial hazard analysis process. Also, even though the safety analysis was performed using several hazard analysis techniques, our technique combines those analysis models to create a single fault traceability graph. This provides the advantage of being able to identify hazardous states that may arise due to interactions among multiple CPSs.

Based on our defined relationships, we have generated FTG with variability also known as v_FTG using algorithm 1 as shown in Fig. 15. The influence relationship, inheritance relationship, supplement relationship, and overlap relationship has been reflected with green, red, black, and blue color respectively. The variability node in v_FTG is indicated with a purple outline. For instance, *dense fog* influences *perception uncertainty* and an *inaccurate decision by proximity sensor* faults in the system. The rain, which is an environmental variability, can cause *perception uncertainty* and *primary camera failure* which can finally lead to robot crash hazards. Similarly, *communication protocols vary*, which is an infrastructural variability causes *interoperability problem* that leads to *communication failure* in Human Rescue Robot System. On the other hand, *software failure* is influenced by *virus/malware*, *software not updated*, and *operating system problem* faults. Furthermore, the safety guard

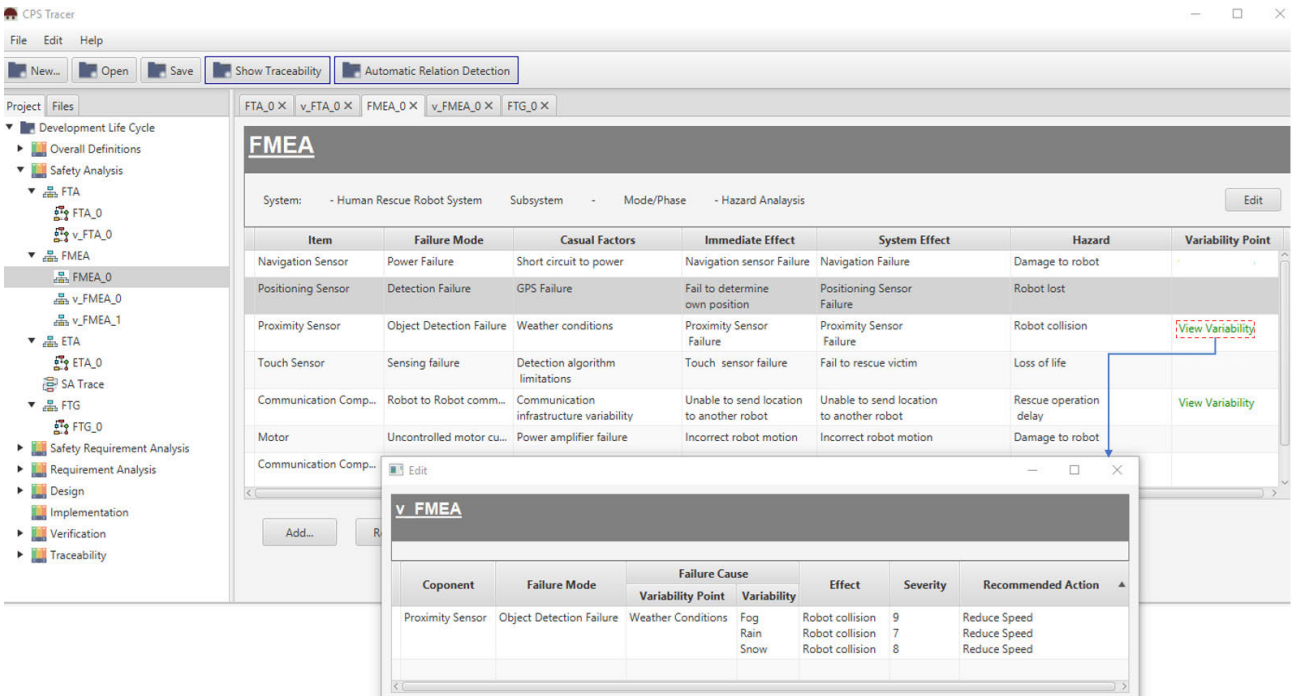


FIGURE 13. Excerpt of hazard analysis with FMEA and v_FMEA.

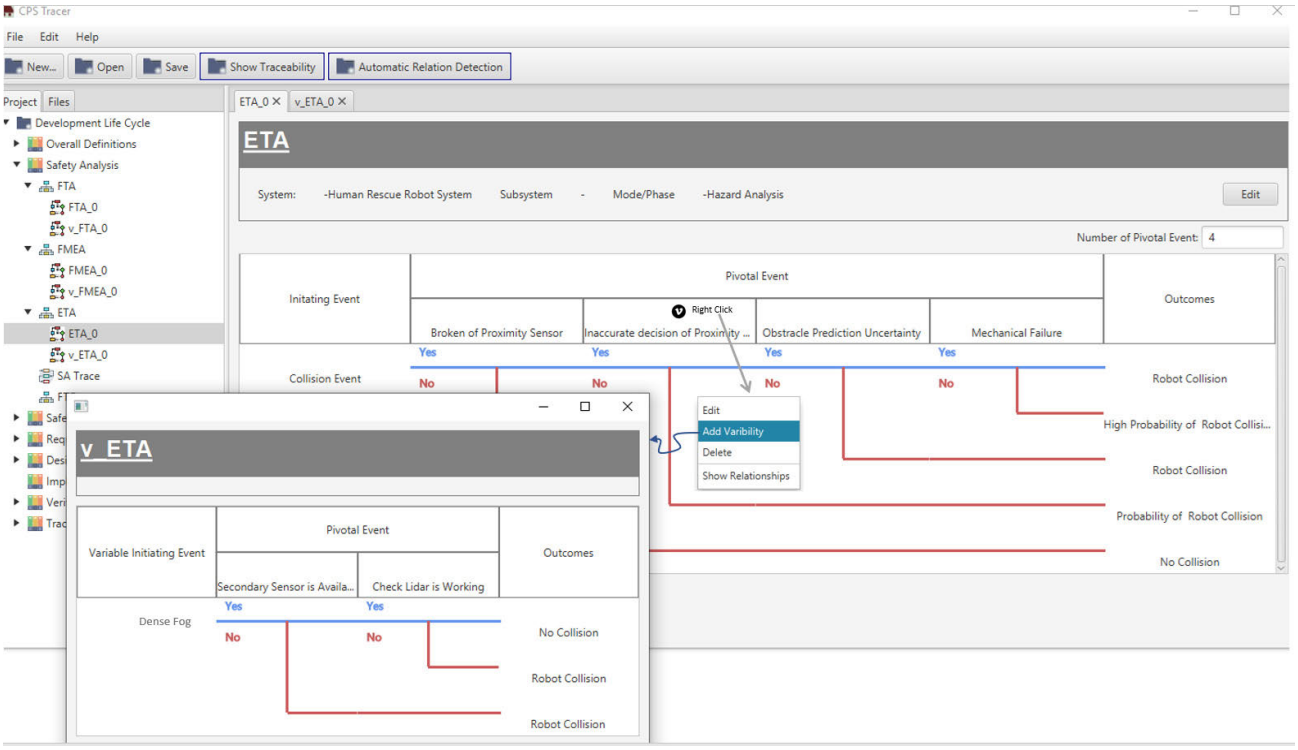


FIGURE 14. Excerpt of hazard analysis with ETA and v_ETA.

"*resend command*" is supplemented to *command failure*, and "*check the distance from CS*" safety guard is supplemented to *communication with CS failure* node in v_FTG.

This v_FTG helps to improve the efficiency of fault diagnoses in the system. The relationships provide more depth knowledge of faults and their impacts in collaborative CPSs.

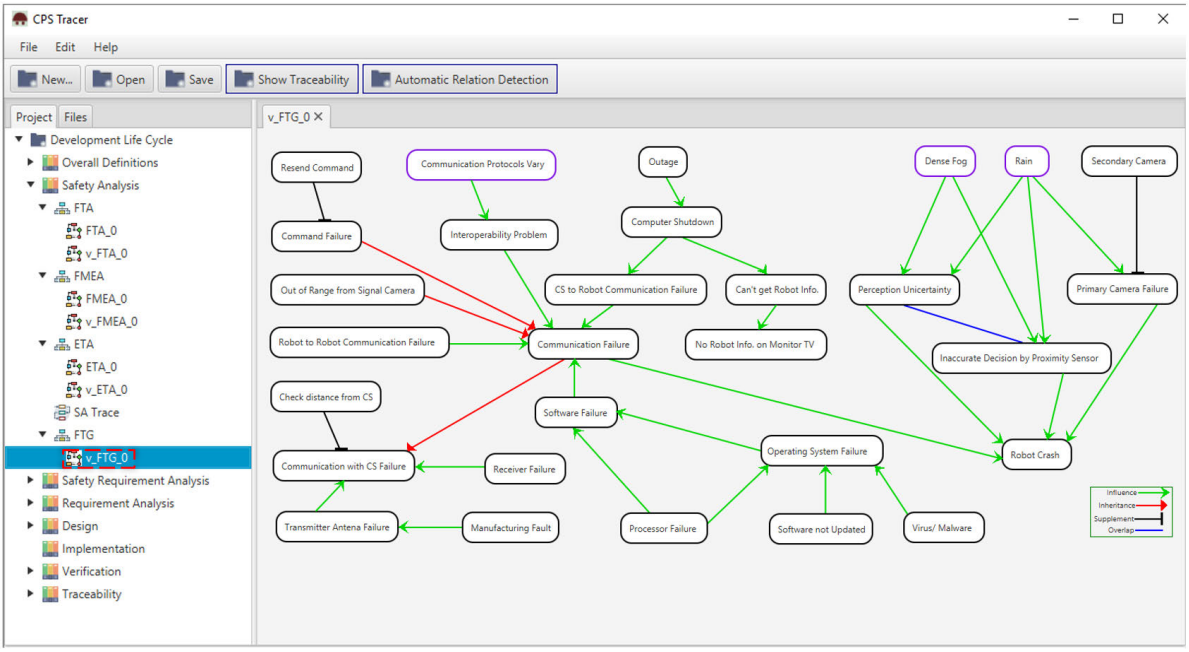


FIGURE 15. Excerpt of Fault Traceability with variability a.k.a. v_FTG.

It also helps to identify what kind of safety guards can be applied to a particular fault in the system.

V. CONCLUSION

CPSs are interconnected and heterogeneous combination of cyber and physical world. This interconnected and heterogeneous combination of behaviors makes system difficult to design and analyze. For example, safety and reliability specifications foisted in CPS applications are very precise and stringent and the robust standards makes the design of those systems very complicated. Indeed, the state-of-the-art tools for CPS analysis and design can not fully deal with intrinsic complexity in CPSs. These provided tools need to guarantee that the behavior of system is as desired even under combination of physical, which are continuous dynamic, and the cyber or computational parts which are discreet dynamics. The safety even suffers more when multiple CPSs collaborate to achieve a common goal.

Therefore, developing a CPS is one of the challenging tasks due to the variable operating environment and a diverse set of heterogeneous computing and communicating devices. CPSs may not be working in a controlled environment and must operate in a robust way to cope with uncertainties. The uncertainties may occur either from the unintended behavior of a failure-free system due to its performance limitations, lack of robustness with respect to environmental influences that might disturb sensors or due to insufficient situational awareness.

We present an approach that addresses the variability aspects of CPSs. We developed a tool named CPSTracer to analyze collaborative CPS with variabilities. First, we identified four variability factors that can cause uncertainty in the system, to analyze the systems with variability, we extended hazard analysis techniques (FTA, FMEA and ETA) for variability (v_FTA, v_FMEA, and v_ETA). We took a collaborative CPS case study of human rescue robot system to validate our proposed approach. We analyzed collaborative CPS with our CPS Tracer and generated a v_FTG that enables us to trace faults across the collaborative CPSs.

In the future, we want to apply our CPS Tracer to analyze hazards in autonomous cars especially on focusing on environmental variabilities. To cope with uncertainties emerging due to environmental variabilities, we are working to develop a learning-based algorithm as part of a healing strategy to ensure safety of autonomous cars.

ACKNOWLEDGMENT

The authors would like to thank our CPS team for their support and cooperation.

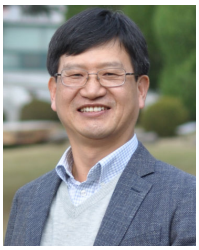
REFERENCES

[1] S. Mouelhi, M. Laarouchi, D. Cancila, and H. Chaouchi, "Predictive formal analysis of resilience in cyber-physical systems," *IEEE Access*, vol. 7, pp. 33741–33758, 2019.
[2] N. Ali and J.-E. Hong, "Failure detection and prevention for cyber-physical systems using ontology-based knowledge base," *Computers*, vol. 7, no. 4, p. 68, Dec. 2018.
[3] X. Lyu, Y. Ding, and S.-H. Yang, "Safety and security risk assessment in cyber-physical systems," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 4, no. 3, pp. 221–232, Sep. 2019.
[4] H. Daneth, N. Ali, and J.-E. Hong, "Automatic identifying interaction components in collaborative cyber-physical systems," in *Proc. 26th Asia–Pacific Softw. Eng. Conf. (APSEC)*, Putrajaya, Malaysia, Dec. 2019, pp. 197–203.
[5] F. Platbrood and O. Gornemann, "Safe robotics-safety in collaborative robot systems," SICK AG, Berlin, Germany, White Paper, 2017, pp. 3–7.
[6] O. Kirovskii and V. Gorelov, "Driver assistance systems: Analysis, tests and the safety case. ISO 26262 and ISO PAS 21448," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, 2019, p. 534.

[7] A. Causevic, A. V. Papadopoulos, and M. Sirjani, "Towards a framework for safe and secure adaptive collaborative systems," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Milwaukee, WI, USA, Jul. 2019, pp. 165–170.
[8] S. Medawar, D. Scholle, and I. Sljivo, "Cooperative safety critical CPS platooning in SafeCOP," in *Proc. 6th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2017, pp. 1–5.
[9] A. A. Nazarenko and L. M. Camarinha-Matos, "Towards collaborative cyber-physical systems," in *Proc. Int. Young Engineers Forum (YEF-ECE)*, May 2017, pp. 12–17.
[10] J. Brings, M. Daun, T. Weyer, and K. Pohl, "Goal-based configuration analysis for networks of collaborative cyber-physical systems," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 1387–1396.
[11] F. L. Leite, R. Adler, and P. Feth, "Safety assurance for autonomous and collaborative medical cyber-physical systems," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.*, 2017, pp. 237–248.
[12] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, Jan. 2004.
[13] S. Torsleff, C. Hildebrandt, M. Daun, J. Brings, and A. Fay, "Developing ontologies for the collaboration of cyber-physical systems: Requirements and solution approach," in *Proc. 4th Int. Workshop Emerg. Ideas Trends Eng. Cyber-Phys. Syst. (EITEC)*, Porto, Portugal, Apr. 2018, pp. 25–32.
[14] A. Khalid, P. Kirisci, Z. Ghrairi, K.-D. Thoben, and J. Pannek, "A methodology to develop collaborative robotic cyber physical systems for production environments," *Logistics Res.*, vol. 9, no. 1, p. 23, Dec. 2016.
[15] M. Daun, J. Brings, T. Bandyszak, P. Bohn, and T. Weyer, "Collaborating multiple system instances of smart cyber-physical systems: A problem situation, solution idea, and remaining research challenges," in *Proc. IEEE/ACM 1st Int. Workshop Softw. Eng. Smart Cyber-Phys. Syst.*, Florence, Italy, May 2015, pp. 48–51.
[16] A. Y.-Z. Ou, M. Rahmaniheris, Y. Jiang, L. Sha, Z. Fu, and S. Ren, "Safetrace: A safety-driven requirement traceability framework on device interaction hazards for MD PnP," in *Proc. 33rd Annu. ACM Symp. Appl. Comput. (SAC)*, 2018, pp. 1282–1291.
[17] E.-S. Kim, D. Lee, J. Sejin, J. Yoo, J. Choi, and J.-S. Lee, "NuDE 2.0: A formal method-based software development, verification and safety analysis environment for digital I&Cs in NPPs," *J. Comput. Sci. Eng.*, vol. 11, no. 1, pp. 9–23, 2017.
[18] N. Nikolakis, V. Maratos, and S. Makris, "A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace," *Robot. Comput.-Integr. Manuf.*, vol. 56, pp. 233–243, Apr. 2019.
[19] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of Cyber-Physical Systems," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nanjing, China, Dec. 2011, pp. 1–6.
[20] D. Horn, N. Ali, and J. E. Hong, "Towards enhancement of fault traceability among multiple hazard analyses in cyber-physical systems," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Milwaukee, WI, USA, Jul. 2019, pp. 458–464.
[21] M. Garcia-Valls, D. Perez-Palacin, and R. Mirandola, "Time-sensitive adaptation in CPS through run-time configuration generation and verification," in *Proc. IEEE 38th Annu. Comput. Softw. Appl. Conf.*, Västerås, Sweden, Jul. 2014, pp. 332–337.
[22] C. A. Ericson, *Hazard Analysis Techniques for System Safety*. Hoboken, NJ, USA: Wiley, 2015, pp. 56–59.
[23] W. S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie, "Fault tree analysis, methods, and applications? A review," *IEEE Trans. Rel.*, vol. R-34, no. 3, pp. 194–203, Aug. 1985.
[24] E. Ruijters and M. Stoelinga, "Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools," *Comput. Sci. Rev.*, vols. 15–16, pp. 29–62, Feb. 2015.
[25] J. D. Andrews and S. J. Dunnett, "Event-tree analysis using binary decision diagrams," *IEEE Trans. Rel.*, vol. 49, no. 2, pp. 230–238, Jun. 2000.
[26] D. H. Stamatis, *Failure Mode and Effect Analysis: FMEA From Theory to Execution*. Milwaukee, WI, USA: Quality Press, 2003.
[27] O. T. Arogundade, S. Misra, O. O. Abayomi-Alli, and L. Fernandez-Sanz, "Enhancing misuse cases with risk assessment for safety requirements," *IEEE Access*, vol. 8, pp. 12001–12014, 2020.
[28] D. B. West, *Introduction to Graph Theory*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[29] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on Uppaal 4.0," Dept. Comput. Sci., Aalborg Univ., Aalborg, Denmark, Tech. Rep., 2006, pp. 2–6.

NAZAKAT ALI (Graduate Student Member, IEEE) received the M.S. degree in computer science from Chungbuk National University, South Korea, in 2017, where he is currently pursuing the Ph.D. degree with the Software Engineering Laboratory, Department of Computer Science, School of Electrical and Computer Engineering. His research interests include software requirements engineering, data mining, ontology, software architecture, software process improvement, DevOps, software quality, system safety, system of systems, and cyber-physical systems.

MANZOOR HUSSAIN received the B.S. degree in software engineering from The University of Azad Jammu and Kashmir, Muzaffarabad, in 2019. He is currently pursuing the Integrated (M.S. leading to Ph.D.) degree with the Department of Computer Science, School of Electrical and Computer Engineering, Chungbuk National University, South Korea. He worked as a Software Developer with GIKI, Pakistan. His research interests include software engineering, data mining, cyber-physical systems, and machine learning.



JANG-EUI HONG received the Ph.D. degree in computer science from KAIST, South Korea, in 2001. He served as a Research Member with ADD (Agency for Defense Development), from 2000 to 2002, and also served as a Principal Consultant with Solution Link Company Ltd. He is currently a Professor with the Computer Science Department, School of Electrical and Computer Engineering, Chungbuk National University, South Korea. His research interests include software quality, embedded software architecture, low-energy software development, and software system safety.