An Approach for Safety Testing of Cooperative Cyber-Physical Systems Using STPA and Quality Attribute Scenarios

Yoona Heo
Konkuk University (Graduate)
hya1202@konkuk.ac.kr

I. Context

Cyber-Physical System (CPS) is a real-time system in which computation, communication, and control parts integrate to deal with the physical world [1], [2]. CPSs usually cooperate to achieve a common goal and when systems cooperate, various emergent properties can appear in the system. Safety is one of the emergent properties that can appear in cooperative CPSs and has to be treated significantly.

Safety testing is one of the ways to demonstrate that the system satisfies safety-specific requirements enough. This abstract proposes an approach for safety testing of cooperative CPSs using *Systems-Theoretic Process Analysis* (STPA) [3] and *Quality Attribute Scenarios* (QASs) [4]. Through this approach, we can identify QASs for safety testing of cooperative CPSs using STPA results and conduct safety testing with identified QASs.

Hazard analysis is an analysis to identify hazards and analyze the causes and effects of hazards [5] so that it can investigate accidents before their occurrence so that they can be eliminated or controlled [3]. STPA is one of the hazard analysis techniques based on an accident model called Systems-Theoretic Accident Models and Processes (STAMP) [6]. As shown in [7], there are four main steps in STPA as follows:

1) Define purpose of the analysis
2) Model the control structure
3) Identify unsafe control actions (UCAs)
4) Identify causal scenarios

In the first step, we identify the purpose of the analysis, which is to identify the losses, hazards, and safety constraints that can prevent the hazards. Then in the second step, we model the control structure which shows the control-feedback loop of the system. In STPA step 3, we identify UCAs, which are the control actions that can cause hazards. Four types of UCAs [7] are as follows:

- Not providing CA causes hazard
- Providing CA causes hazard
- Providing CA too late/too soon causes hazard
- Stop providing CA too soon/applying CA too long causes hazard

In STPA step 4, we identify causal (loss) scenarios, which are the causes of UCAs. There are four categories of causal scenarios (CSs) [7], which can be assigned to each part of the control structure:

- Unsafe controller behavior
- Causes of inadequate feedback and information
- Scenarios involving the control path
- Scenarios related to the controlled process

Quality Attribute is a measurable or testable property of a system that is used to indicate how well the system meets the needs of stakeholders beyond the basic functionality of the system [4]. QASs are quality attribute requirements specified in the form of scenarios and are used to evaluate the system's ability to satisfy these requirements. QAS is composed of six parts as follows:

- Source of stimulus: Some entity (human, computer system, and any other actor) that generates stimulus
- Stimulus: An event arriving at the system
- Environment: A certain condition that a stimulus occurs
- Artifact: A target of the stimulus. It can be a whole system or some part of the system
- Response: An activity that occurs when a stimulus arrives
- Response measure: Measurable criteria of the response that can be tested

There are general scenarios not specific to a system and concrete scenarios specific to a certain system in QAS. The safety general scenario is composed of six parts as mentioned above, and each part has some possible values. Some of the possible values from the safety general scenario in [4] are described later in ⟨Table I⟩. Due to the limitation of the number of pages, I cannot describe all the possible values for each part of the safety general scenario in this abstract. The point we focused on is that the safety general scenario [4] aims to prevent the system from entering into a hazardous state or to control the system even when the system is already in a hazardous state.

II. Gap

A. Related works

There are some previous researches on the safety of collaborative/cooperative CPS and system of systems (SoS). These previous researches are categorized and presented below according to each category.

1) Safety of Collaborative/Cooperative CPS: In [8], authors propose an extended version of three hazard analysis techniques, event tree analysis (ETA) [9], fault tree analysis (FTA) [10], and failure mode and effect analysis (FMEA) [11], considering the variability of collaborative CPSs. The authors develop a fault traceability graph (FTG) with variability (v_FTG) to trace the faults among multiple hazard analyses in collaborative CPSs and a tool to support the modeling of each analysis and the generation of v_FTG. Through this work, authors try to maximize the known-safe area and minimize a known-unsafe area and unknown-unsafe area. Unfortunately, as far as I know, hazard analysis techniques used in this study do not consider hazards caused by interactions among different CPSs while there can be interactions among various collaborative CPSs.

Authors of [12] present the platooning research within the Safe Cooperating Cyber-Physical Systems using Wireless Communication (SafeCOP) project. They use safety cases for safety assurance in design time, using Goal Structuring Notation (GSN) [13]. And they use a runtime manager for continuous safety assurance in runtime, to check strong and weak contract violations. Although it mentions that it uses safety analysis to derive system contracts, as far as I understood, I could not find a mention of how to perform the analysis and derive contracts from the analysis results.

2) Safety of System of Systems: In [14], authors propose a process of identifying hazards of SoS using the hazard analysis and risk assessment (HARA) [15] method and exemplify it in a quarry site automation context. The objective of this process is to fill a HARA table with relevant information about the hazardous event associated with SoS and the causes of potential accidents. An operating phase, the first step of this process, is a typical operation scenario at the quarry site level and a more detailed description should be provided by distinguishing which geographical zones of the site are used. After distinguishing each zone, they create an impact matrix using a factor of approaching autonomous machines entering this zone, and this machine can interact with humans, other machines, and infrastructures. Then they identify hazards and classify them into various categories using the impact matrix and the HARA table. Unfortunately, the scope of this study is limited to the identification of hazards and does not mention how to deal with the identified hazards.

Authors of [16] propose a tailored version of the safety lifecycle for SoS. They first define items, and in this study, items are defined at the vehicle and connected vehicle levels. Then they perform HARA on each level and derive safety goals for each level. Next, they develop a safety concept for each level to discover a required safety measure that needs to mitigate hazards at each level, through performing safety analyses. After developing a safety concept, they follow parts 5 and 6 of ISO 26262. They also evaluate the proposed process by performing a comparative study on several project contexts. The main difference between this study and my study is that this study focuses on the functional safety of SoS, while my study focuses on the control safety of cooperative CPSs.

Authors of [17] focus on the SoS HA for the electric quarry site using two hazard analysis techniques, hazard and operability study (HAZOP) [18] and FTA, to deal with emergent behaviors of different machines. The authors first present a set of guidewords and perform HAZOP on the quarry site to identify hazards and their potential effects. Analysis performed using HAZOP focuses on the communication failures in SoS, external malfunctions and internal system failures. Then they apply FTA to the system using the identified hazards as the top undesired events. Finally, they derive preventive measures from FTA results to eliminate or control the identified hazards to demonstrate acceptable safety at the quarry site. The merit in cost and time of my study compared to this study is that my study only includes one hazard analysis technique, STPA, while this study includes two hazard analysis techniques, HAZOP and FTA.

What we have in common with other studies is that we want to eventually create a safer cooperative/collaborative CPS. The term safety includes physical safety, functional safety, and control safety [19]. By using methodologies proposed in these previous studies, functional safety which standards including ISO 26262 [15] focus on can be satisfied at a higher level. However, in this study, we focused on testing the control safety of the cooperative CPSs, which can be done by using QAS based on STPA results, since STPA models control structure composed of control-feedback loops. Even though there are many previous works done about the safety of collaborative/cooperative CPSs and SoSs, we could not find any research that mentioned identifying the QAS for safety testing of cooperative CPSs from the results of hazard analysis and using it to test the safety of the system. So, in the next subsection, we propose an approach using STPA to identify QASs from cooperating CPSs.

B. An Approach to Identify Quality Attribute Scenarios Using STPA

In this subsection, I propose an approach to identify QASs for cooperative CPSs from STPA results. Through this approach, we can identify QASs of each and overall CPS, and we can conduct safety testing with identified QASs. We first need to perform a typical STPA on a cooperative CPS before we identify QAS. Three main steps of this approach are described in ⟨Fig. 1⟩.

I focus on steps 3 and 4 of STPA since the results of these steps are used while identifying QASs. After UCAs and

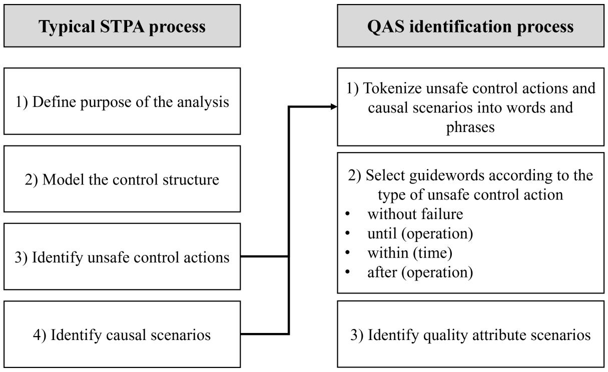
Typical STPA process
QAS identification process
1) Define purpose of the analysis
2) Model the control structure
3) Identify unsafe control actions
4) Identify causal scenarios
1) Tokenize unsafe control actions and causal scenarios into words and phrases
2) Select guidewords according to the type of unsafe control action
• without failure
• until (operation)
• within (time)
• after (operation)
3) Identify quality attribute scenarios

Fig. 1: QAS identification process

CSs are identified through STPA, we can identify QASs from the UCAs and CSs using some guidewords, such as 'without failure', 'until (certain operation)', 'within (time)', and 'after (certain operation)'. As a first step, we first tokenize UCA and CS into words and phrases. After that, we select appropriate guidewords according to the type of UCA. Then, we identify QASs that correspond to each CS using tokenized words and phrases, guidewords, and possible values described in ⟨Table I⟩.

TABLE I: Six parts of QAS from STPA results

Portion of scenario	Possible values
Source of stimulus	• Sensor • Controller • (internal/external) Data source
Stimulus	Data or Control from the source
Environment	(from safety general scenario [4]) • Normal operation • Degraded operation • Manual operation • Recovery mode
Artifact	• Controller • Controlled process • Hardware component
Response	Values can be either the opposite of the tokenized phrases from STPA results or from the possible values of the response part of the safety general scenario [4] as described below. Recognize the unsafe state and one or more of the following: • Avoid the unsafe state • Recover • Continue in degraded or safe mode • Shut down • Switch to manual operation • Switch to a backup system • Notify appropriate entities (people or systems) • Log the unsafe state (and the response to it)
Response measure	Depends on the analysts' decision using the given guidewords

III. Innovation

This abstract proposes an approach to identify quality attribute scenarios for the safety testing of cooperative CPSs based on STPA. By conducting the safety testing and evaluating the results of safety testing, we can demonstrate that the system satisfies safety-specific requirements. The reason we use STPA to identify QASs is that STPA can deal with cooperative systems when various interactions occur among systems and hazardous behavior exists between components [20], [21]. STPA can also deal with the timing issue of cooperative CPSs since CPSs are real-time systems.

Since the safety of the cooperative CPSs is the main concern of this abstract, test cases for safety have to be identified as a form of scenario considering the interactions among various cooperative CPSs. In this perspective, QAS can be treated as a scenario that can be used to test the safety of the system, because they are originally used to evaluate the system's ability to satisfy quality attribute requirements. By using this approach, we can test the safety of the cooperative CPSs with identified QASs.

IV. Evaluation Plan

With my co-researchers, I implemented a testbed consisting of several cooperative CPSs which can interact through WiFi network and Radio Frequency communication. The software controllers of each CPS in the testbed are developed systematically following the software development life cycle (SDLC) [22]. This testbed is composed of software controllers on Arduino D1 R1 boards and several hardware components such as communication modules, some sensors, LEDs, and LCDs, and will be further improved. It is part of a small smart city example, which is a downscaled intelligent transportation system with several subsystems. It has some functions to collect external environmental data such as temperature, humidity, and data from vehicles on the virtual road, and to handle such data to give information to control the traffic light.

By using this testbed, I will conduct a case study following the steps presented in ⟨Fig. 1⟩. I will also evaluate the effectiveness of this approach by evaluating the pass/fail rate of these safety test cases to be generated. If STPA is performed thoroughly, the pass rate of the generated safety test cases will be high since the software controllers are developed systematically. Coverage of these safety test cases will also be high enough since the test cases are generated from the results of STPA, and STPA covers the whole safety-related part of the system if performed thoroughly. Also, using QASs to identify safety test cases is a merit of this approach since I can identify well-formatted safety test cases with measurable criteria. I expect that this approach can be sufficiently effective in identifying safety test cases.

References

[1] Liguo Zhang, Yaser P Fallah, and Rezgui Jihene. Cyber-physical systems: computation, communication, and control. *International Journal of Distributed Sensor Networks*, 2013.

[2] Wayne Wolf. Cyber-physical systems. *Computer*, 42(03):88–89, 2009.

[3] Nancy G Leveson. *Engineering a safer world: Systems thinking applied to safety*. The MIT Press, 2016.

[4] Len Bass, Paul Clements, and Rick Kazman. *Software architecture in practice*. Addison-Wesley Professional, fourth edition, 2021.

[5] Clifton A Ericson et al. *Hazard analysis techniques for system safety*. John Wiley & Sons, 2015.

[6] Nancy G Leveson. *Safeware: system safety and computers*. ACM, 1995.

[7] Nancy G Leveson and John P Thomas. Stpa handbook. *Cambridge, MA, USA*, 2018.

[8] Nazakat Ali, Manzoor Hussain, and Jang-Eui Hong. Analyzing safety of collaborative cyber-physical systems considering variability. *IEEE Access*, 8:162701–162713, 2020.

[9] Marko Čepin and Marko Čepin. Event tree analysis. *Assessment of Power System Reliability: Methods and Applications*, pages 89–99, 2011.

[10] Clifton A Ericson and Clifton Ll. Fault tree analysis. In *System Safety Conference, Orlando, Florida*, volume 1, pages 1–9, 1999.

[11] Donald J Reifer. Software failure modes and effects analysis. *IEEE Transactions on reliability*, 28(3):247–249, 1979.

[12] Samer Medawar, Detlef Scholle, and Irfan Šljivo. Cooperative safety critical cps platooning in safecop. In *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–5. IEEE, 2017.

[13] Tim Kelly and Rob Weaver. The goal structuring notation–a safety argument notation. In *Proceedings of the dependable systems and networks 2004 workshop on assurance cases*, volume 6. Citeseer, 2004.

[14] Stephan Baumgart, Joakim Fröberg, and Sasikumar Punnekkat. Analyzing hazards in system-of-systems: Described in a quarry site automation context. In *2017 Annual IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2017.

[15] ISO. Road vehicles – Functional safety, 2018.

[16] Arash Khabbaz Saberi, Eric Barbier, Frank Benders, and Mark Van Den Brand. On functional safety methods: A system of systems approach. In *2018 Annual IEEE International Systems Conference (SysCon)*, pages 1–6. IEEE, 2018.

[17] Faiz Ul Muram, Muhammad Atif Javed, and Sasikumar Punnekkat. System of systems hazard analysis using hazop and fta for advanced quarry production. In *2019 4th International Conference on System Reliability and Safety (ICSRS)*, pages 394–401. IEEE, 2019.

[18] Thomas C McKelvey. How to improve the effectiveness of hazard and operability analysis. *IEEE Transactions on Reliability*, 37(2):167–170, 1988.

[19] Xiaorong Lyu, Yulong Ding, and Shuang-Hua Yang. Safety and security risk assessment in cyber-physical systems. *IET Cyber-Physical Systems: Theory & Applications*, 4(3):221–232, 2019.

[20] Victor Bolbot, Gerasimos Theotokatos, Luminita Manuela Bujorianu, Evangelos Boulougouris, and Dracos Vassalos. Vulnerabilities and safety assurance methods in cyber-physical systems: A comprehensive review. *Reliability Engineering & System Safety*, 182:179–193, 2019.

[21] Jakob Axelsson and Avenir Kobetski. Towards a risk analysis method for systems-of-systems based on systems thinking. In *2018 Annual IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2018.

[22] Vanshika Rastogi et al. Software development life cycle models-comparison, consequences. *International Journal of Computer Science and Information Technologies*, 6(1):168–172, 2015.