2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Analysing the Safety and Security of a UV-C Disinfection Robot

Desiana Nurchalifah, Sebastian Blumenthal, Luigi Lo Iacono and Nico Hochgeschwender

***Abstract*— Safety is paramount for robots used in environments they share with humans. In such scenarios, security is also growing in importance. However, conventional approaches to analysing safety requirements are aimed at identifying hazards only. Security-related aspects such as cyber threats, cyber attacks and vulnerabilities have hardly been integrated into analysis and design methods to date. The methods available so far for the joint analysis of safety and security are based on established methods of safety engineering, where the amount of information is very large and usually stored in text- and table-based documents. This makes it challenging for engineers to systematically assess and maintain safety and security information. Thus, adequate tool support for robot engineers is required to cope with the increased complexity and to manage the safety and security risks. In this paper, we demonstrate that robot's safety and security information can be expressed, stored, analysed and queried in a knowledge graph representation paving the way to automated analysis. More specifically, we apply an integrated, systems-oriented safety and security co-analysis approach, namely STPA-Safesec, to a robot performing disinfection tasks in domestic environments. By querying the resulting graph of safety and security artefacts, we automatically retrieve hazardous scenarios, identify gaps in the analysis and increase our understanding of the overall risks of the robot.**

I. Introduction

Ensuring safe operation of robots interacting and collaborating with humans is of paramount importance for a responsible deployment of robots in the real world [1]. To prevent accidents, it is important to identify potential sources of danger, determine which humans in the robot's environment are most at risk, and assess the types of injuries the robot can inflict on these humans. This hazard analysis is already a complex task in controlled environments such as industrial workplaces [2], but it becomes even more complex when robots leave their cages and are brought into public spaces where they move around independently. The fact that robots are increasingly operated in dynamic environments is also reflected in current standards such as ISO 13482 [3]. However, ISO 13482 only considers the analysis of failures of individual components, which is insufficient as they are not able to capture emergent properties from the interaction between components and humans within the system.

System-based hazard analysis approaches such as System-Theoretic Process Analysis (STPA) [4] overcome these shortcomings by treating safety as a control problem, transforming the analysis to create safety measures in the system design facilitated by a control structure. It is shown in previous studies that STPA can find more potential hazardous scenarios compared to traditional methods such as FMEA [5], [4].

In addition to safety, robots operating in dynamic environments must also consider security, especially when used in public spaces and consisting of multiple units with independent sensing, communication, computing and decision-making functions, as this opens up a wide range of attack opportunities [6], [7], [8]. Security considerations contribute to the complexity of the analysis as one needs to consider threats, attack models and vulnerabilities. This is reinforced by the need to analyse the interactions between safety and security in relation to each other. However, current technical practice lacks methods and tools to support joint safety and security analysis and to manage the inherent complexity [9]. Moreover, the main concern in safety engineering to date is that security aspects may negate the safety acceptance of the system as both aspects are considered separately and integrated only in the later stages of the analysis [10]. The need to equally address safety and security becomes crucial as robots are highly connected nowadays and cannot be considered safe unless security is also assured [9]. Harmonised safety and security analysis with supporting tools are therefore required not only to decrease severe inconsistencies, but also to achieve higher coverage of hazards and to eliminate them. Initial approaches recently evolved in safety areas other than robotics, most notably STPA-Safesec [10], an extension of STPA to include security analysis capabilities in analysing both safety and security in the automation domain.

Fig. 1: A UV-C disinfection robot operating in a real hospital environment.

However, until now it remains unclear whether existing co-

Sebastian Blumenthal is with KELO Robotics GmbH (Germany), Desiana Nurchalifah, Luigi Iacono and Nico Hochgeschwender are with the Department of Computer Science, Bonn-Rhein-Sieg University of Applied Sciences, Germany. nico.hochgeschwender@h-brs.de

analysis approaches are applicable to robotics. In addition, there is a lack of tools, requiring engineers to manually identify the hazards and threats and their interactions in the form of text and spreadsheet documents. This approach is not only tedious and prone to errors, but also limits the analysis to the selected scenarios making it unsuitable for advanced service robots operating in open, dynamic and uncertain environments.

In this paper, we address this lack of insight with a bottom-up approach, i.e., making use of a case study. To this end, we formulate the following research questions:

RQ1: Is STPA-Safesec applicable to the domain of robotic systems engineering?

RQ2: Does safety and security co-design of robotic systems based on STPA-Safesec result in fewer flaws in the form of unidentified hazards compared to current design approaches?

Answering these research questions is of high practical relevance. From the insights, researchers can derive further research to lay a broad and sound scientific ground in respect to the safety-security-codesign of robotic systems. To this end, developers involved in the design of robotic systems can obtain methods and tools to better manage the complexity of design tasks related to safety and security, and standardisation bodies can develop new standards and test methods to make them available to industry.

In summary, we make the following contributions:

- We propose a *co-analysis knowledge graph* which expresses the safety and security concepts and their interrelationships of STPA-Safesec [10].
- We develop means to populate the graph with safety and security artefacts paving the way from manual to tool-supported analysis of robotic systems.
- We evaluate the introduced approach through a case study in which a systematic safety and security co-analysis using STPA-Safesec is applied to a UV-C disinfection robotic system (see Fig. 1) operating in the real world and capable to autonomously disinfect surfaces of viruses in public spaces such as medical care facilities and universities.
- We show that with the help of semantic graph queries we can identify gaps in the analysis and improve the overall traceability of STPA-Safesec artefacts.

II. Background and Related Work

We treat both *safety* and *security* as attributes of *dependability* [11] where (i) safety is defined as the absence of catastrophic consequences on the users and the environment, and (ii) security is expressed as the composition of *availability*, *confidentiality* and *integrity* of information and assets.

A. Safety and Security Standards

Safety standards in robotics provide recommendations on how to identify, evaluate and mitigate safety risks (see Bozhinoski *et al.* [12] for a recent survey) by following general principles for design, risk assessment and risk reduction as, for example, defined in ISO 12100 [13]. One can distinguish between standards targeting robot applications where robots do not share the same workspace with humans and those where robots operate in shared environments and collaborate with humans by design. The latter and most challenging situation is reflected in (i) ISO 15066 [14] proposing different human-robot collaboration modes and associated force limits, and (ii) ISO 13482 [3] exemplifying typical hazards occurring in assistive robot scenarios. *Security standards* specifically targeting service robots do not exist [9]. Whether related standards such as IEC 62443 [15] targeting the secure development of automation and control systems are applicable for robotic applications remains to be investigated. Previous works in considering security in robotics include the assessment of attacks on robots [16], [17] and how security breaches could compromise the functionality of robots [18] and threaten the safety of humans [9]. Even though, security awareness is increasing among robot software developers, commonly used robot software frameworks such as ROS are vulnerable to cyber-attacks [19]. For example, DeMarinis *et al.* [20] showed how to gain unauthorised access to robotic sensors and actuators by robots running the ROS framework.

B. Risk Mitigation

Safety measures aim to mitigate the risks identified in the analysis and to conform to standards by numerous active or passive techniques [1], [21] ranging from monitoring safety requirements at run-time [22], [23], [24], [25] and implementing advanced control policies [26] for safely coping with contact situations [27] to inherently safe robot design [28]. *Security measures* tailored to robotics include approaches for detecting anomalies in data and control flows of robot control architectures [29], [30], resilient control policies which are capable to withstand malicious sensor attacks [31], [32] and tools and frameworks for hardening the Robot Operating System (ROS) [33].

C. Safety and Security Co-Analysis

Jointly analysing safety and security is rarely performed in robotics [9], yet there is an increased need of safety and security co-analysis in various domains with the automotive domain as the main driver in that direction [34]. However, even systematically analysing solely the safety of advanced robotic systems is already challenging due to the enormous amount of safety-relevant information [35]. To this end, Biggs *et al.* [36] proposed to employ model-based approaches and tools such as SafeML [37] to model the results of a safety analysis. In [38] another model-based approach is proposed where the HAZOP (HAZard OPerability) method is applied to UML models for the sake of performing safety risk assessment of service robots. However, to the best of our knowledge reports on employing safety and security co-analysis on robotic systems do not exist. Many co-analysis methods are either based on textual or model-based approaches [34], [9]. While textual approaches clearly have their limitations with respect to management and traceability of the produced and assessed artefacts, model-based approaches are usually tailored to a

specific formalism. For example, formalism combining attack and fault trees [39] or combinations of bow tie diagrams with attack trees [40]. Even though, specialised tools, in particular for architectural approaches [41], are available, typically no large case studies are performed to evaluate these approaches and the safety-security interactions and implications are still ill-understood [34].

III. Safety and Security Co-Analysis Approach

We present in the following paragraphs our safety and security co-analysis approach which is based on two ingredients, namely (i) the *co-analysis knowledge graph* for modelling safety and security concepts, and (ii) the STPA-Safesec co-analysis approach proposed in [10] both extended and employed for our work.

A. Co-Analysis Knowledge Graph

We model safety and security concepts and their relations according to the definition of property graph [42], where a knowledge graph is a tuple of $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathcal{P}, \mathcal{U})$ and $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is a set of directed edges, and a function $e : E \to V \times V$ that maps each edge to the pair of nodes it connects. A function $l : V \cup E \to 2^L$ maps an edge or node to the set of labels $\mathcal{L}$, and the function $p : V \cup E \to 2^{P \times U}$ matches sets of property–value pairs, $\mathcal{P}$ and $\mathcal{U}$ respectively, to a node or edge.

Typically, a STPA-based analysis yields to numerous artefacts expressing hazards, losses and systems flaws, to name a few, which are conventionally stored in the form of tables. Links among these artefacts are either by specific columns or within the artefact descriptions [4]. Following how these artefacts are stored, we employed graph modelling [42] to transform the tabular representation to a graph representation. An excerpt of the resulting co-analysis knowledge graph is shown in Fig. 2. Here, a node represents an analysis concept, for example, the concept of a (:Hazard). Similarly to our prior work in employing knowledge graphs to model robot software architectures [43], each artefact contains a unique identifier to identify the instances it belongs to. Next, we established relations among the constructed concepts. We denote the relations among the nodes in coherence with the definition of each concept stated in [4], with the exception of unique artefacts to STPA-Safesec [10]. For example, (:Flaw)–:leads_to→(:HCA) expresses the flaws of the system which lead to hazardous control actions (cf., III-B). For implementing the knowledge graph we employed the neo4j[1]property-graph database and associated querying facilities. We also extended existing data loaders to process comma separated files into the database from traditional table-based analyses.

B. STPA-Safesec

STPA-Safesec is an extension of STPA [10], a qualitative (:Hazard) analysis technique based on the systems-theoretic accident model and processes (STAMP) approach. STAMP considers safety as a control problem rather than a failure prevention [4] activity. Safety in STAMP is based on system thinking where accidents are dynamic processes originating from insufficient (:Constraints) enforced to the system, which consists not only of technical, but also of socio-technical aspects. Examples of the latter are constraints imposed by users like operators of technical systems. The analysis employs a hierarchical control structure as a system modelling technique to describe loops of control composed of elements dictating commands to other elements and their associated responses.

The STPA analysis can be summarized into four steps, namely (i) defining the analysis purpose, (ii) modelling the control structure, (iii) identifying unsafe control actions, (iv) and lastly identifying loss scenarios. For the analysis purpose, (:Hazard) and (:Losses) are determined which are then systematically linked to the later steps in deriving unsafe control actions (cf. (:HCA)) and loss (:Scenarios). STPA-Safesec [10] extended the approach in STPA with the addition of guidance related to the security aspects of the system. To this end, STPA-Safesec integrates both safety and security supported by a (:Component) diagram and general causal factor table in the security domain. To this end, the hierarchical control structure is mapped to a component layer diagram that details the components of the system.

TABLE I: Losses and top-level hazards.

ID	Losses
L-1	Serious injury (irreversible) to humans requiring medical attention and leading to long-term or permanent physical effects
L-2	Slight injury (reversible) that does not require any medical attention and does not lead to long-term or permanent physical effects
L-3	Damage to the environment
L-4	Destruction or damage to the UVC robot
L-5	Non-achievement of disinfection task
L-6	Loss of IP and critical information
ID	**Hazards**
H-1	**Radiation overexposure (L1, L2)**. Any person is exposed to radiation from the robot(s) at hazardous levels.
H-2	**Robot(s) are subjected to collision (L3, L4)**. Collision with objects or other robots due to either internal or external factors of the robot system.
H-3	**Robot(s) entry of forbidden zone (L3, L4, L5)**. Either internal factor of the robot system or external disturbances leads the robot to enter non-intended area.
H-4	**Robot network security is compromised (L5,L6)**. Attacks on robot network availability, authentication, and confidentiality.

[1]https://neo4j.com/learning-neo4j-book/ Last accessed: 6th of March 2023

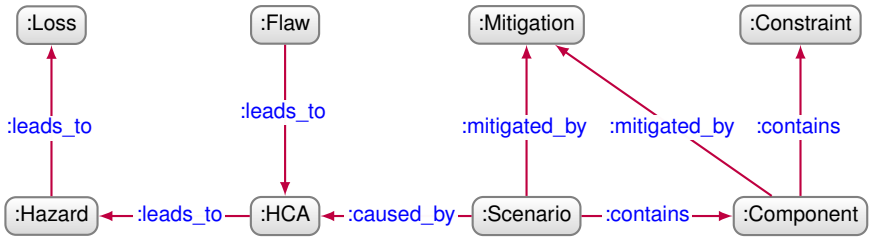

Fig. 2: Excerpt of a property graph schema expressing the STPA-Safesec concepts required to perform a safety and security co-analysis. It is an excerpt as some details (e.g., properties of nodes) are for the sake of readability omitted.

IV. Case Study

We employed our approach (see Sec. III) to analyse the safety and security of an autonomous UV-C disinfection robot (see Fig. 1). To this end, we obtained the overall analysis of the system, including :Hazard in both safety and security aspects, and collected :Mitigation that could be applied to the system to ensure safe and secure operation in a public space. Despite the diverse approaches in disinfecting environments, UV-C has proven effective in reducing viral or bacterial contamination of surfaces [44]. However, this poses increasing requirements to robotic systems as well as challenges in terms of both safety and security, as such robotic systems can approach and potentially injure humans. The investigated UV-C robot carries six 256 nm UV-C lamps combined with sensors to navigate autonomously within the environment. The robot is equipped with 2D laser scanners in the base compartment, 3D laser scanners and a set of 3D cameras in the top part of the lamp system. These sensors are used for object collision avoidance and localisation during navigation. Further, a set of RGB cameras is build into the top of the robot in order to detect persons. The purpose is to detect persons and switch off UV-C lamps, to avoid over-exposure with UV-C light. We selected this robotic system because it poses potential hazards to humans, as it operates autonomously in public spaces such as hospitals, emitting UV-C radiation, and because it has a variety of different communication interfaces that open up attack surfaces. This type of robot needs to be analysed and designed with safety and security in mind and is therefore suitable for answering our overall research question whether a knowledge-graph assisted STPA-Safesec co-analysis is beneficial.

A. Defining the Analysis Purpose

We started the co-analysis by deriving high-level :Losses. The list of losses is retrieved (i) from conventional regulation of machinery, namely ISO 14121, and (ii) based on the feedback from the engineers in particular with respect how the robot operates within public space. For the latter this includes the consideration of several varying environmental and mission-related features such as size and appearance of humans, dynamic and static objects, and different mission areas, such as hallways or floors with stairs. Table I summarises the identified losses and their relation to :Hazards causing the losses. It is worth to mention that the loss of non-achievement of the disinfection task is rather unique to the considered application of the UV-C robot as, for example, not turning UV-C lamps on would not expose potential humans to UV-C light, but would also not perform the disinfection task.

B. Modeling the Control Structure

We define top-level safety and security constraints, which are the negations of system-level hazards, followed by a design of control layer diagram that equals to hierarchical control structure in STPA as shown in Fig. 3. An example of a :Constraint is the fact that radiation must not be exposed to any person at a hazardous level. This constraint establishes a link between the first hazard (H-1) and the first and second loss (L-1 and L-2) as shown in Table I. It is important to note that the control structure in Fig. 3 is entailing the controllers and processes that deemed imperative for the disinfection task. It is not a physical or software architectural description. For example, the control action *Start disinfection* (cf., Fig. 3) turns among others the UV-C lamp on.

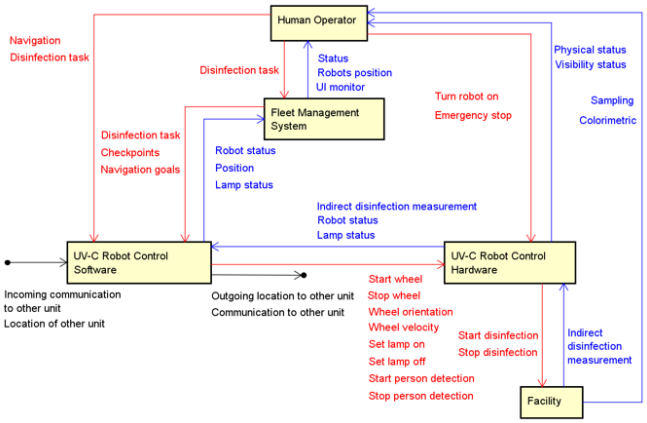

Fig. 3: Control layer diagram of the considered disinfection application. The boxes represent the controllers, while the very bottom box represents the controlled process. Red arrows represent control actions and blue arrows represent feedback. External process inputs and process outputs are represented in black.

C. Analysing Unsafe Control Actions

We identified and analysed control actions and their associated context. Control actions express how system modules interact with each other. To this end, the general control structure defined in Fig. 3 is further refined in concrete

software and hardware components of the system. Due to intellectual property rights we do not show the complete component architecture here. The architecture entails one master control loop which allows to command the mobile base (e.g., velocity and pose set points) as well as means to turn on and off the UV-C lights based on a module which is capable to identify persons in the surrounding. For further analysis we derived ten contextual variables impacting the control actions, including environmental conditions relevant for the robot operation (e.g., mission area). These ten variables are: (i) person detection module state, (ii) robot navigation module state, (iii) environment state, (iv) robot state, (v) robot navigation mode, (vi) robot position, (vii) high-level controller state, (viii) low-level controller state, (ix) UV-C lamp state, and (x) other robot states.

Next, we derive hazardous control actions (:HCA) which are control actions that lead under certain contextual conditions to :Hazards. These control actions violate the constraints identified in Sec. IV-B and are composed of five elements, namely (i) source (e.g., the controller capable of performing the control action), (ii) type, (iii) control action, (iv) context (e.g., information when or where an control action is applied or not applied by the controller), and (v) link to hazard. As an example we derived the following hazardous control action:

> When an operator (cf., source) would command the robot to stop its motion (cf., control action) when the robot is on uneven terrain (cf., context) which would yield to hazard 2 (cf., Table I).

We employed the proposed STPA-Safesec methodology to derive hazardous control actions by investigating the following conditions, namely (i) control action is not performed by the robot, (ii) control action is performed in hazardous situations, (iii) control action is provided at the wrong time or scheduling, and (iv) control action is performed in the wrong duration. To achieve the disinfection task within a public space, the robot needs to ensure a state where the UV-C lamp is off while a person is detected and the UV-C lamp is on while no person is present. Overall, this is represented in 16 hazardous control actions. Another case is the regulation of robot movement such that safe distance to objects is maintained while ensuring at the same time that disinfection is effective, which is described in 25 hazardous control actions. Case consideration also includes the disinfection task sent from external sources such as fleet and human operators. We also identified two cases where unidentified access is granted to the robot, totalling 70 hazardous control actions.

In the next step we identify the security aspects of the system, which are then refined in the later step to further analyse the constraints of safety and security of each component. To this end, we extended the component layer analysis proposed by STPA-Safesec with a trust boundary which is defined as the place where more than one principle interacts. Thus, where threats are more clearly visible [45]. The reason why a system has trust boundaries is because the entity on the other side of the boundary is less trusted. We combined the component layer diagram and threat boundary concept, in which we found that the application of threat boundary could guide the analysts in how to group the details of components of the system in a component layer diagram based on different levels of trust. From the trust boundary, generation of loss scenarios (cf., Sec. IV-D) are better visible as level of trust could be the causal factor of why a scenario would happen. However, threat boundary produced different insights to STPA-Safesec analysis, bringing forth a contradiction to be determined by the analysts. For example, threat boundary suggests a grouping of different high-level controllers in one representation, such that security can be affirmed. This grouping is not in accordance with STPA-Safesec that suggests a high-level controller to be positioned near the low-level controller with which it is associated. Hence, the final decision considering the trade-off of the architecture is left to the analysts' decision. In this refining step, we utilise the general integrity and availability threats provided by STPA-Safesec (cf., Table II). As an example, we refine safety and security constraints by looking at the high-level controller. Critical points in a high-level controller are: (i) combined with the variability of person detection module state one can cause radiation overexposure (cf. H-1), (ii) combined with the variability of navigation module state one can cause collisions (H-2) Hence, in this controller possibility of threats from integrity are both commands and measurement injection, drop, manipulation, and delay.

TABLE II: Excerpt of STPA-Safesec security constraints utilised in the case study. 'CSTR' define security constraints, 'A' define availability and 'I' define integrity.

ID	Description
CSTR-A-1	Communication delay
CSTR-A-2	Communication dropped
CSTR-A-3	Node overloaded (delay)
CSTR-A-4	Node overloaded (drop)
CSTR-I-5	Measurement injection
CSTR-I-6	Measurement drop
CSTR-I-7	Measurement manipulation
CSTR-I-8	Measurement delay

D. Analysing Hazardous Scenarios

Next, hazardous scenarios are described. A scenario is a specific case during the operation of the UV-C robot in which a flaw could lead to hazardous control action. Here, a flaw is a process model of a controller, where the controller believes the state of the outside world is incorrect or inadequate. Hazardous control actions, as a result of the system flaw, are correlated to system hazards and tend to be very long. An example scenario is described in the following paragraph.

> High-level controller incorrectly believes person detection is adequate while robot is idle and incorrect, delayed, or missing feedback from person detection module is provided, as a result, command set lamp on

(cf. control action) is provided. This example scenario may be due to (i) undetected physical cable disturbance, (ii) wrongly configured camera, (iii) delayed transmission due to signal or cyber-attack, (iv) person detection module sent compromised data, (v) and many more.

From the initial set of hazardous control actions (cf., Sec. IV-C) we identified 421 scenarios where 246 relate to the UV-C mobility, 180 relate to the UVC-C lamps, and the remaining refer to the disinfection task itself. Formulated in STPA-Safesec mitigation's can be done in different stages of the analysis. These stages include the identification of the component layer diagram and the scenarios. In the context of this work we developed 16 mitigation plans combating the 421 scenarios either on the component-level as exemplified in Table III or on a scenario-level (e.g., specifically MP-1 in Table III).

TABLE III: Excerpt of the implemented mitigation plans for the UV-C disinfection robot.

ID	Description
MP-1	Robot denies commands while processing a disinfection task
MP-2	Distributed clock [46]
MP-3	ROS-based runtime monitoring [47]
MP-4	Port blocking
MP-8	Safety over EtherCat

V. Querying the Co-Analysis Knowledge Graph

We populated the analysis results from Sec. IV into a graph database conforming to the schema shown in Fig. 2. To this end, we developed a set of Cypher[2] scripts to automatically convert comma sepearated files obtained from spreadsheets to the graph representation. Overall, the resulting graph contains > 3000 nodes expressing various artefacts such as (:Hazards), (:Components) and so forth.

A. Results

We answer the research questions formulated in Sec. I. STPA-Safesec is applicable to analyse robotic systems (cf., **RQ1**). No application or domain-specific assumptions are formulated in the STPA-Safesec methodology, yet domain-specific concerns (e.g., context variables of hazardous control actions) can be integrated. In fact, the definition of control actions and controlled processes helped us to retrieve hazardous situations originating from the human operator (e.g., wrongly commanding actions during disinfection execution) at an early stage in development.

To answer **RQ2** and to support the analysis we developed a set of queries in Cypher capable to infer knowledge from the co-analysis knowledge graph. The first query (cf., Fig. 4) demonstrates the ability of the developed knowledge graph to find hazardous scenarios based on the cause of delayed response. The second query (cf., Fig. 5) investigates the components that contain both safety and security constraints. For our case study we found that 34 out of 39 components are connected to both aspects in our STPA-Safesec analysis demonstrating the importance of a joint analysis of safety and security. With a similar query not shown here we identified whether mitigation have been applied to all hazardous scenarios in the context where UV-C lamp is on and person is present. We found that 6 out of 21 hazardous scenarios are mitigated on the top-level while the other 15 hazardous scenarios are mitigated on a component-level. With the query shown in Fig. 6 we retrieve all the artefacts involved in mitigating hazardous scenarios on a scenario-level (see Sec. IV-D) where the cost of development are rather low. Here, costs are associated to mitigation during the analysis [10]. Similar queries can be formulated to infer all components required for particular mitigation plans. This enables developers to make their development more effective (cf., **RQ2**) as one can focus, for example, to test the components appearing in several mitigation plans or to identify hazard which are caused by flawed control actions rather than flawed by component-level behaviour.

```
MATCH
(s:Scenario)->[:CORRELATE_WITH]->(c:Security_Constraint)
WHERE c.desc CONTAINS "delay"
RETURN s
```

Fig. 4: Retrieving hazardous scenarios caused by delay.

```
MATCH
 (safe:Safety_Constraint)->[r:CONTAINS]->(c:Component)
WITH c
MATCH c<-[r:CONTAINS]<-(sec:Security_Constraint)
RETURN c
```

Fig. 5: Retrieving components impacted by both safety and security constraints.

VI. Conclusion

We performed a safety and security co-analysis for a real world robot application. The analysis is supported by a knowledge graph resembling concepts from STPA-Safesec and which paves the way to manage the huge amount of safety and security relevant information. Our approach clearly inherits the limitations of STPA-Safesec, namely the analysis is exhaustive. For example, the enumeration of contextual factors impacting the hazardous control actions requires significant domain knowledge as guidance about

```
MATCH
(s:Scenario)->[:MITIGATED_BY]->(m:Mitigation)
WITH COLLECT s.id AS excluded
MATCH path=s:Scenario->[*2]->(m:Mitigation)
WHERE
(NOT s.id in excluded) AND relationships(p)[1].cost < 3
RETURN p
```

Fig. 6: Retrieving high-priority scenarios and mitigations.

[2] https://neo4j.com/developer/cypher/ Last accessed 6th of March 2023.

terminology and ranges of specific contextual variables is missing. In the future we plan to investigate whether existing robot ontologies could support the definition of contextual factors. We conclude that the high-level approach propagated by STPA-based methods is beneficial, because to start with high-level control actions enables one to generalise, for example, mitigation plans among different software and hardware architectures. Even though, the proposed approach is only evaluated to the case study of disinfection robot we believe that the presented insights are helpful to support future co-analysis activities in robotics.

ACKNOWLEDGEMENT

This work was partially supported by the European Union's Horizon 2020 project SESAME (grant agreement No 101017258).

REFERENCES

[1] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," *Robotics and Autonomous Systems*, vol. 94, pp. 43–52, 2017.

[2] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[3] "Robots and robotic devices — Safety requirements for personal care robots," International Organization for Standardization, Geneva, CH, Standard, Feb. 2014.

[4] N. G. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. The MIT Press, 01 2012.

[5] J. Chen, S. Zhang, Y. Lu, and P. Tang, "Stpa-based hazard analysis of a complex uav system in take-off," in *International Conference on Transportation Information and Safety (ICTIS)*, 2015, pp. 774–779.

[6] F. Jahan, W. Sun, Q. Niyaz, and M. Alam, "Security modeling of autonomous systems: A survey," *ACM Comput. Surv.*, vol. 52, no. 5, sep 2019.

[7] G. Cornelius, P. Caire, N. Hochgeschwender, M. A. Olivares-Mendez, P. Esteves-Verissimo, M. Völp, and H. Voos, "A perspective of security for mobile service robots," in *ROBOT 2017: Third Iberian Robotics Conference*, 2018, pp. 88–100.

[8] D. M. Gage, "Security considerations for autonomous robots," in *IEEE Symposium on Security and Privacy*, 1985, pp. 224–224.

[9] M. Gleirscher, N. Johnson, P. Karachristou, R. Calinescu, J. Law, and J. Clark, *Challenges in the Safety-Security Co-Assurance of Collaborative Industrial Robots*. Springer International Publishing, 2022, pp. 191–214.

[10] I. Friedberg, K. McLaughlin, P. Smith, D. Laverty, and S. Sezer, "Stpa-safesec: Safety and security analysis for cyber-physical systems," *Journal of Information Security and Applications*, vol. 34, pp. 183–196, 2017.

[11] A. Avizienis, J. Laprie, and B. Randell, "Dependability and its threats - a taxonomy," in *IFIP Congress Topical Sessions*, 2004.

[12] D. Bozhinoski, D. Di Ruscio, I. Malavolta, P. Pelliccione, and I. Crnkovic, "Safety for mobile robotic systems: A systematic mapping study from a software engineering perspective," *Journal Systems and Software*, vol. 151, no. C, p. 150–179, may 2019.

[13] "Safety of machinery — General principles for design — Risk assessment and risk reduction," International Organization for Standardization, Geneva, CH, Standard, Nov. 2010.

[14] B. Group *et al.*, "Robots and robotic devices—collaborative robots (iso/ts 15066: 2016)," *BSI Standards Publication: London, UK*, 2016.

[15] "Industrial communication networks – Network and system security," International Electrotechnical Commission, Standard, Feb. 2009.

[16] F. Jahan, W. Sun, Q. Niyaz, and M. Alam, "Security modeling of autonomous systems: A survey," *ACM Comput. Surv.*, vol. 52, no. 5, sep 2019.

[17] N. Hochgeschwender, G. Cornelius, and H. Voos, "Arguing security of autonomous robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7791–7797.

[18] A. Giaretta, M. De Donno, and N. Dragoni, "Adding salt to pepper: A structured security assessment over a humanoid robot," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, 2018.

[19] N. Goerke, D. Timmermann, and I. Baumgart, "Who controls your robot? an evaluation of ros security mechanisms," in *7th International Conference on Automation, Robotics and Applications (ICARA)*, 2021, pp. 60–66.

[20] N. DeMarinis, S. Tellex, V. P. Kemerlis, G. Konidaris, and R. Fonseca, "Scanning the internet for ros: A view of security in robotics research," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8514–8521.

[21] M. Valori, A. Scibilia, I. Fassi, J. Saenz, R. Behrens, S. Herbster, C. Bidard, E. Lucet, A. Magisson, L. Schaake, J. Bessler, G. B. Prange-Lasonder, M. Kühnrich, A. B. Lassen, and K. Nielsen, "Validating safety in human–robot collaboration: Standards and new perspectives," *Robotics*, vol. 10, no. 2, 2021.

[22] M. Machin, F. Dufossé, J.-P. Blanquart, J. Guiochet, D. Powell, and H. Waeselynck, "Specifying safety monitors for autonomous systems using model-checking," in *Computer Safety, Reliability, and Security*, 2014, pp. 262–277.

[23] M. Machin, J. Guiochet, H. Waeselynck, J.-P. Blanquart, M. Roy, and L. Masson, "Smof: A safety monitoring framework for autonomous systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 5, pp. 702–715, 2018.

[24] N. Hochgeschwender, "Adaptive deployment of safety monitors for autonomous systems," in *Computer Safety, Reliability, and Security*, 2019, pp. 346–357.

[25] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, pp. 73–88, 2005.

[26] S. Haddadin, A. Albu-Schaffer, A. De Luca, and G. Hirzinger, "Collision detection and reaction: A contribution to safe physical human-robot interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3356–3363.

[27] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[28] S. Haddadin, A. Albu-Schäffer, O. Eiberger, and G. Hirzinger, "New insights concerning intrinsic joint elasticity for safety," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2181–2187.

[29] D. Azzalini, A. Castellini, M. Luperto, A. Farinelli, and F. Amigoni, "Hmms for anomaly detection in autonomous robots," in *Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems*, 2020, p. 105–113.

[30] P. J. Bonczek and N. Bezzo, "Detection of hidden attacks on cyber-physical systems from serial magnitude and sign randomness inconsistencies," in *American Control Conference (ACC)*, 2021, pp. 3281–3287.

[31] N. Bezzo, J. Weimer, M. Pajic, O. Sokolsky, G. J. Pappas, and I. Lee, "Attack resilient state estimation for autonomous robotic systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3692–3698.

[32] V. Marquis, R. Ho, W. Rainey, M. Kimpel, J. Ghiorzi, W. Cricchi, and N. Bezzo, "Toward attack-resilient state estimation and control of autonomous cyber-physical systems," in *Systems and Information Engineering Design Symposium (SIEDS)*, 2018, pp. 70–75.

[33] V. Mayoral-Vilches, R. White, G. Caiazza, and M. Arguedas, "Sros2: Usable cyber security tools for ros 2," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 253–11 259.

[34] E. Lisova, I. Šljivo, and A. Čaušević, "Safety and security co-analyses: A systematic literature review," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2189–2200, 2019.

[35] G. Biggs, T. Sakamoto, K. Fujiwara, and K. Anada, "Experiences with model-centred design methods and tools in safe robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3915–3922.

[36] G. Biggs, T. Ogure, K. Fujiwara, Y. Nakabo, and T. Kotoku, "Modelling the safety of a semi-autonomous wheelchair," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4664–4671.

[37] G. Biggs, Y. Nakabo, T. Kotoku, and K. Ohba, "On the development of dependable intelligent systems," in *Advanced Robotics and its Social Impacts*, 2011, pp. 16–19.

[38] J. Guiochet, D. Martin-Guillerez, and D. Powell, "Experience with model-based user-centered risk assessment for service robots," in *2010 IEEE 12th International Symposium on High Assurance Systems Engineering*, 2010, pp. 104–113.

[39] "Integrating cyber attacks within fault trees," *Reliability Engineering System Safety*, vol. 94, no. 9, pp. 1394–1402, 2009.

[40] H. Abdo, M. Kaouk, J.-M. Flaus, and F. Masse, "A safety/security risk analysis approach of industrial control systems: A cyber bowtie – combining new version of attack tree with bowtie analysis," *Computers Security*, vol. 72, pp. 175–195, 2018.

[41] P. Dissaux, F. Singhoff, L. Lemarchand, H. N. Tran, and I.-H. Atchadam, "Combined Real-Time, Safety and Security Model Analysis," in *9th European Congress ERTSS Embedded Real Time Software and System*, Toulouse, France, Feb. 2020.

[42] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, "Knowledge graphs," *ACM Comput. Surv.*, vol. 54, no. 4, jul 2021.

[43] N. Hochgeschwender, S. Schneider, H. Voos, H. Bruyninckx, and G. K. Kraetzschmar, "Graph-based software knowledge: Storage and semantic querying of domain models for run-time adaptation," in *Proc. of the IEEE Intl. Conf. on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2016, pp. 83–90.

[44] S. K. Bhardwaj, H. Singh, A. Deep, M. Khatri, J. Bhaumik, K.-H. Kim, and N. Bhardwaj, "Uvc-based photoinactivation as an efficient tool to control the transmission of coronaviruses," *The Science of the total environment*, vol. 792, October 2021.

[45] A. Shostack, *Threat Modeling: Designing for Security*. Wiley Publishing, 2014.

[46] F. Sygulla, R. Wittmann, P. Seiwald, T. Berninger, A. Hildebrandt, D. Wahrmann, and D. Rixen, "An ethercat-based real-time control system architecture for humanoid robots," in *IEEE 14th International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 483–490.

[47] J. Huang, C. Erdogan, Y. Zhang, B. Moore, Q. Luo, A. Sundaresan, and G. Rosu, "Rosrv: Runtime verification for robots," in *Runtime Verification*, 2014, pp. 247–254.