A Hybrid Ontology for Identifying Safety Hazards and Security Threats

Malina Adach, Alessio Bucaioni, Federico Ciccozzi
School of Innovation, Design and Engineering
Mälardalen University
Västerås, Sweden
{malina.adach, alessio.bucaioni, federico.ciccozzi}@mdu.se

***Abstract*—This paper introduces the Hazard and Threat Ontology, a hybrid ontology designed to illustrate safety hazards and security threats in complex systems of systems. Hazard Ontology and Combined Security Ontology are two ontologies with extensive terminology and complementary methodologies. They allow us to develop a hybrid approach that enables safety and security experts to analyze complex systems thoroughly. Combining these ontologies enhances the depth and scope of experts' analysis and decision-making process, and several tangible benefits are associated with using a hybrid approach across different industrial sectors. In this paper, an industrial use case illustrates the practical utility of the Hazard and Threat Ontology. Our approach facilitates the identification of hazards and threats, providing actionable insights into how to mitigate them. Consequently, assets and personnel can be protected, downtime can be reduced, and operational resilience can be enhanced.**

***Index Terms*—safety hazards, security threats, safety analysis, Hazard Ontology, Combined Security Ontology, system of systems**

I. Introduction

A system of systems (SoS) comprises independently owned and managed constituent systems (CSs) that collaborate to provide services such as air traffic management, smart grids, and medical applications [1]. These CSs evolve autonomously based on their environments and objectives [2]. Effective risk management in SoSs is crucial, focusing on safety, reliability, security, and sustainability to ensure successful operation and longevity [3].

Safety directly affects human lives and the environment, reducing the likelihood of failures and accidents. Security prevents unauthorized access and threats, enhancing system resilience and stakeholder trust. Integrating safety and security analyses early in SoS development helps identify technology gaps and improve feasibility. Safety analysis aims to prevent adverse environmental effects, while security analysis focuses on preventing environmental impacts on the system [4]. These analyses, performed during the concept and design stages, require interdisciplinary expertise due to complex system interactions.

Combined hybrid approaches to safety and security analysis are becoming increasingly important [5], [6]. A well-structured joint analysis process can identify attack potentials, explore failure scenarios, and streamline system design. Our previous work [7] reviewed security ontologies published in the time span 1988–2022; from the insights, we developed a Combined Security Ontology (CSO) [8], based on the Unified Foundational Ontology (UFO) [9]. Existing ontologies, presented using OWL [10] or UML Class Diagrams [11], focus on concepts such as asset, attack, countermeasure, threat, and vulnerability. Recent developments in integrating safety and security analysis methods have also been discussed [12]–[14].

This paper introduces the Hazard and Threat Ontology (HTO), a hybrid ontology for safety and security analyses, developed early in the engineering lifecycle to help identify safety hazards and security threats. By integrating safety and security concepts, the HTO enhances risk assessment and security management in SoSs. By establishing an integrated terminology, the paper facilitates the seamless integration of safety and security considerations in SoSs.

Key contributions of our work include:

1) A novel approach: HTO advances ontology-aided safety and security analysis, unmatched in current literature.
2) Extension to security analysis: Building on our previous work focused on safety analysis [15], our ontology now supports security analysis.
3) Broad applicability: Although illustrated with a transportation industry example, our method applies to various SoS scenarios. We validate HTO using a quarry site use case.

The remainder of this paper is organized as follows. Section II provides background information about SoS, a Hazard Ontology (HO) [16], and the Combined Security Ontology (CSO) [8]. Section III presents the steps for developing HTO. Section V introduces our hybrid ontology for safety and security analyses. Section VI presents the evaluation of the HTO, and Section VII describes a use case and provides a practical application of HTO in an SoS. Section VIII provides the related work, and Section IX discusses the results of applying the HTO and presents conclusions with directions for future work.

II. Background

The following section briefly the concept of a system of systems, the Hazard Ontology (HO), and the Combined Security Ontology (CSO), that were used to develop our approach.

A. System of Systems

There are many definitions of an SoS [17], and Maier's [18] description has become widely recognized today. In this paper, we use the definition derived from the ISO 21841 standard, which states that an SoS is "*a set of systems or system elements that interact to provide a unique capability that none of the constituent systems can accomplish on its own*". Maier identified the primary characteristics of SoSs as operational independence, managerial independence, evolutionary development, distribution, and emergent behavior [18]. According to Maier, an SoS can be classified as directed, acknowledged, collaborative, and virtual. We can categorize our use case as collaborative SoS since autonomous vehicles at a quarry site operate without authority and work together to accomplish a common goal [18].

B. Hazard Ontology

An ontological interpretation of hazards in the safety domain was provided by Zhou et al. [16]. The authors developed an HO identifying possible hazards and their causes, sources, and consequences within cyber-physical systems. The HO contains information about hazards in three forms: (i) concepts that describe entities crucial to the interpretation of hazard-related concepts, (ii) relationships between concepts that are clearly defined and directed, and (iii) axioms that provide accurate information. The HO includes 11 concepts and 14 relationships among them. A detailed description of the HO concepts is provided in [19]. HO includes concepts derived from UFO-A [9] and UFO-B [9]. Fig. 1 presents the HO [16], which includes 11 concepts and 14 relationships among them.

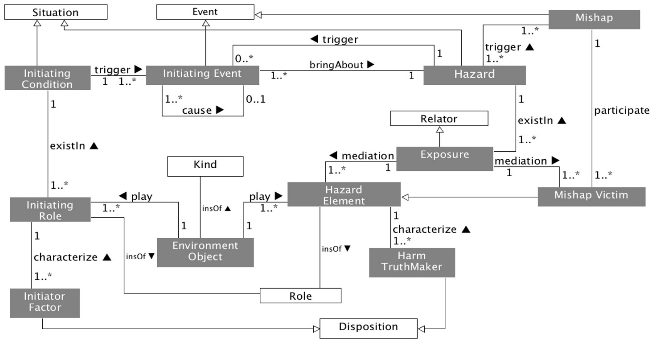

Figure 1. A UML diagram of a Hazard Ontology introduced in [16].

Fig. 1 represents a structured way to understand how various factors contribute to hazards and mishaps. Each gray rectangle corresponds to the HO concept, the white rectangle corresponds to the UFO concept, and arrows indicate the relationships between these concepts. The HO contains the following key concepts: **Situation** represents the context or scenario in which hazards occur; **Initiating Condition** describes the starting condition that triggers subsequent events; **Event** refers to an occurrence or happening; **Initiating Event** is the event that triggers further actions; **Hazard** represents a potential danger or harm; **Exposure** indicates the interaction with the hazard; **Mishap** represents an undesirable outcome or incident; **Mishap Victim** is the entity affected by the mishap; **Harm Truthmaker** represents critical dispositions or conditions that can lead to harm or mishaps; **Hazard Element** refers to an active or passive role that various environmental objects can play, and **Initiator Factor** represents an element that triggers or causes an event or action.

The following relationships are found in the HO: **Trigger** indicates causality between events; **BringAbout** describes how an initiating condition leads to an initiating event; **Cause** represents the relationship between events and hazards; **Participate** shows how an individual or entity can be involved in a hazard; **Meditation** mediates the relationship between exposure and the hazard; **Characterize** describes the relationship between exposure and harm; **Play** indicates the role or function of a component (such as a hazard element) within a specific context; and **ExistIn** associates elements with the environment or situation they exist in. It captures the context in which the hazard element and initiator factor operate. Each concept has cardinality indicators (e.g., 1, 1.., 0..) showing how many instances of one element can relate to another. This ontology can be used for hazard identification, risk assessment, safety management, and understanding cause-and-effect in hazardous situations. It provides a systematic framework for identifying, analyzing, and mitigating hazards.

In our previous paper [15], we successfully applied the HO to a quarry site using autonomous vehicles and identified different hazards, including interaction hazards. Using the HO, we analyzed potential interactions between systems and identified emergent hazards and components in a system of systems with autonomous vehicles. The results indicate that the HO can support single systems and SoS hazard analysis.

Throughout this paper, we focus on extending the HO with security concepts and relationships drawn from the CSO.

Zhou et al. [16] developed a Hazard Ontology (HO) based on the UFO, formalized it using UML, and applied it to a railway system. An application of the HO to an SoS is discussed in [15], and a comparison between the HO and the Combined Security Ontology (CSO), which is also grounded in the UFO, is provided in [19]. In this paper, we extend HO with the concepts and relationships of the CSO to create a hybrid ontology for analyzing safety and security concerns.

C. Combined Security Ontology

In CSO [8], we reused the security ontologies selected from our literature review [7] and extracted core security concepts and relationships from them to develop our Combined Security Ontology (CSO) based on the UFO. We evaluated CSO in the security domain and applied it to an SoS. CSO includes concepts from all three parts: UFO-A [9], UFO-B [9], and UFO-C [20]. Therefore, extending the HO with CSO concepts allows us to reuse all parts of the UFO.

Fig. 2 illustrates the concepts and relationships of the CSO [8], which contains 12 concepts and 37 relationships between them, 15 of which were derived from UFO. The rectangles represent the core concepts in this diagram, while the lines and arrows represent their relationships. Both ends of the relationships are labeled with cardinality constraints. As outlined

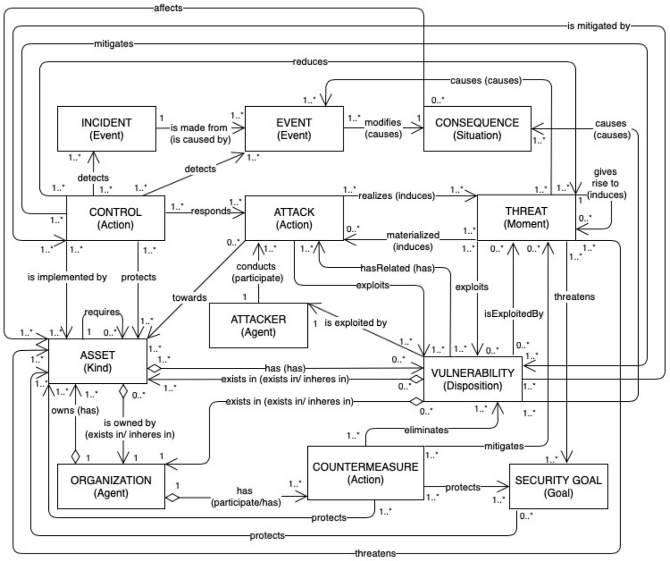

Figure 2. Fundamental concepts and relationships of the Combined Security Ontology [8].

above, the 12 core concepts for the security domain (in capital letters) are based on eight fundamental concepts within the UFO (in parentheses): "action," "agent," "disposition," "event," "goal," "kind," "moment," and "situation."

The CSO comprises the following concepts: **Incident** that represents a security-related event or occurrence; **Event** refers to any significant happening; **Consequence** indicates the outcome or result of an event; **Control** represents a specific response or activity; **Attack** refers to deliberate harmful actions; **Threat** represents potential danger or risk; **Asset**: refers to valuable resources or entities requiring protection; **Attacker** is responsible for attacks; **Vulnerability** is a Weakness or susceptibility; **Countermeasure** is a protective measure to mitigate threats; **Security Goal** is the desired outcome related to security; and **Organization**: represents entities such as companies, institutions, or groups.

In the CSO, various relationships are included: **affects** signifies that an incident can impact or affect something; **is mitigated by** indicates that detection mechanisms can help mitigate or reduce the impact of incidents; **detects** implies that detection mechanisms identify or detect attacks; **is implemented by** suggests that assets play a role in implementing detection mechanisms; **requires** signifies that organizations need specific assets to fulfill their requirements; **owns** and **is owned by** indicate ownership relationships between assets and organizations; **participates in** suggests that organizations can be involved in attack scenarios; **has** signifies that attackers target specific assets; **exists in** suggests that vulnerabilities exist within the context of attackers; **exploits** indicates that attackers exploit vulnerabilities; **eliminates** suggests that countermeasures affect information dissipation related to vulnerabilities; and **protects** means that a countermeasure can protect a security goal.

The application of our CSO at the quarry site, described in our previous paper [8], assisted in securing data communication among autonomous vehicles, smart devices, and the cloud infrastructure. Furthermore, the CSO identified possible paths for an attacker to exploit the GPS signals of autonomous vehicles to achieve their objectives and identify quarry site assets and technologies at risk (e.g., open GPS signal structure).

A use case demonstrated that this CSO can be applied to identify security issues by evaluating its completeness concerning existing security ontologies. Based on the findings of this paper, the CSO facilitates security analysis and improves the identification of potential vulnerabilities and threats.

III. Steps for Developing a Hybrid Analysis

This paper focuses on early safety and security analysis of complex systems of systems based on a hybrid ontology described as a safety and security model. This research extends previous work [16], which focused on safety analysis and included security analysis, also presented in another previous work [8]. We developed a hybrid analysis method based on ontology engineering, created it as a safety and security metamodel, and represented it using a UML diagram. We extend the HO presented in [16] used for safety analysis by incorporating security concepts of CSO (shown in Tab. I), including attack, attacker, control, countermeasure, security goal, and threat. To provide early safety and security analysis, we developed our hybrid ontology following these four steps:

- Step 1 - Harmonization of safety and security concepts: harmonize a hybrid ontology's basic safety and security concepts (Section IV).
- Step 2 - Hazard and Threat Ontology: develop a hybrid ontology based on the safety metamodel and add the known security concepts and relationships (Section V).
- Step 3 - Evaluation of the developed hybrid ontology: verification and validation of the developed hybrid ontology (Section VI).
- Step 4 - Application of the developed hybrid ontology: apply the hybrid analysis ontology to a complex SoS and identify safety hazards and security threats (Section VII).

IV. Step 1 - Harmonization of safety and security concepts

To develop a Hybrid Threat Ontology (HTO), this paper presents a unique approach that harmonizes safety and security concepts. Although safety and security share a number of methodologies and concepts, there are inherent differences in their terminologies and conceptual frameworks that make harmonization challenging. In order to meet these challenges, we have developed a framework that integrates the safety and security domains into an ontology developed specifically for Systems of Systems (SoS) analysis. Unlike our previous work, which separately examined the safety and security domains, this paper extends those foundations by creating a more cohesive structure that captures both domains. A novel approach is introduced by extending the Hazard Ontology (HO) [16] with Combined Security Ontology (CSO) [8] elements to address the complex relationships between the safety and security domains.

For example, the concept of "consequence" is harmonized to encompass both the "outcome of an event" in safety (ISO/IEC 16085:2021 [21], ISO/IEC 15026-3:2015 [22]) and the "outcome of an attack" in security (ISO/IEC 27001:2022 [23], [7]) (as shown in Tab. I). The new HTO refines these definitions to recognize that "consequence" represents "an effect (change or non-change) associated with an event," applicable across both safety (ISO/IEC 15026-3:2023 [24]) and security (NIST 800-160 [25]) domains.

Table I
HARMONIZATION OF SAFETY AND SECURITY CONCEPTS - PART 1

Concepts	*Safety*	*Security*
CONSEQUENCE is related to the kind object and represents	**effect** (change or non-change), usually associated with an event or condition or with the system and usually allowed, facilitated, caused, prevented, changed, or contributed to by the event, condition, or system. ISO/IEC 15026-1:2019	**effect** (change or non-change), usually associated with an event or condition or with the system and usually allowed, facilitated, caused, prevented, changed, or contributed to by the event, condition, or system. NIST 800-160
CONSEQUENCE is related to the asset and represents	**outcome** of an event affecting one or more stakeholders. ISO/IEC/IEEE 16085:2021. The outcome of an event affecting objectives. ISO/IEC 15026-3:2015	the possible **outcome** of an attack or an event (e.g., denial of services) affecting the properties (CIA) of an asset or a security incident caused by an attacker. ISO/IEC 27001.

While the "initiator factor" in safety and "vulnerability" in security can have different labels, they are both described in the HTO in terms of conditions that could result in adverse outcomes (as shown in Tab. II).

Table II
HARMONIZATION OF SAFETY AND SECURITY CONCEPTS - PART 2

Concepts	*Safety*
INITIATOR FACTOR is related to the hazard and represents	the property of the initiating role. It represents the **weakness** of the initiating role that makes it contribute to the initiating condition.
INITIATING EVENT is related to the hazard and represents	an undesirable or unexpected **event** that can bring about a hazard situation.
Concepts	***Security***
VULNERABILITY is related to the threat and represents	any **weakness** of an asset or the system that can be exploited by a threat (e.g., security flaws, defects, mistakes in software). It can be influenced directly (intentionally malicious) or indirectly (an unintentional mistake) by human behaviour. NIST 800-160, ISO/IEC 27001
INCIDENT is related to the asset and threat and represents	an anomalous or unexpected **event**, set of events, a condition, or situation at any time during the life-cycle of a project, product, service, or system. NIST 800-160, ISO/IEC 27001

Using our previous work and expanding on it, the developed HTO introduces robust tools for system engineers by providing innovative methodologies that effectively harmonize the safety and security domains. The following section provides a detailed description of this ontology, as well as its specific applications and advantages for SoS analysis.

V. STEP 2 - HAZARD AND THREAT ONTOLOGY

An ontology can be created by extracting and expanding elements from existing ontologies. Thus, safety ontologies can also be extended to include security concepts and relationships. We previously developed ontologies (i.e., Hazard Ontology [16], Combined Security Ontology [8]) to facilitate separate safety and security analyses in systems engineering and applied them to an SoS. We briefly described both ontologies in the previous sections.

In [19], we compared HO and CSO and identified their similarities and differences. According to our results, seven HO concepts are equivalent to eight CSO concepts, and four HO relationships are equivalent to six CSO relationships. The development of a well-structured hybrid ontology was motivated by several objectives:

- to ensure that documentation is arranged in a structured manner to avoid chaos;
- to provide a single source of integrated information;
- to improve the efficiency of the information input using guiding forms;
- to support automatic and semiautomatic documentation generation;
- to provide information at a higher level of detail within an SoS;
- to ensure a deeper understanding of the system analysis required by engineers;
- to provide an integrated terminology for the safety and security domains; and
- to facilitate reusing information required for safety and security analyses.

The developed Hazard and Threat Ontology (HTO) is based on existing safety and security ontologies. We intended to create a hybrid safety and security ontology to perform safety and security analyses. We conceptualized the HTO by integrating safety and security domains to support system engineers in performing an analysis during the concept stage. HTO facilitates the identification and understanding of how safety and security can impact the critical attributes of an SoS. Additionally, HTO guides systems engineers in developing safety and security scenarios. Fig. 3 illustrates the developed HTO. The diagram shows an overview of ontology and how its entities relate to one another.

a) Description of the HTO: Fig. 3 illustrates the Hazard and Threat Ontology (HTO), encompassing 17 concepts and 49 relationships among them, along with the following components:

- Six safety concepts are presented in yellow rectangles with bold text;
- Fourteen safety relationships are illustrated with yellow dashed lines and bold text;
- Six security concepts are presented in blue hexagons with italic text;

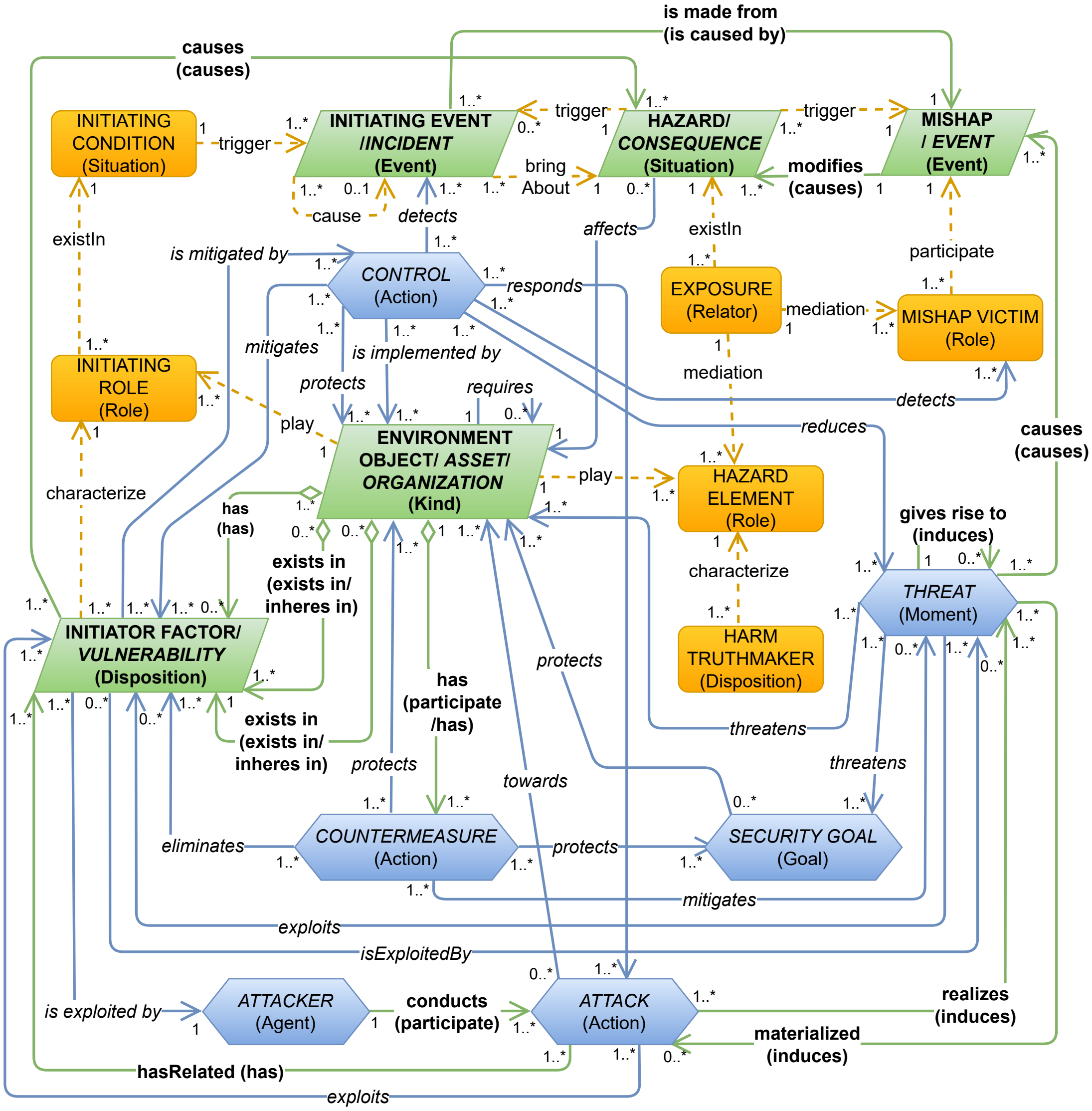

Figure 3. Hazard and threat ontology (HTO) - hybrid analysis method for safety and security

- Thirty-five security relationships are presented with blue lines and capital labels, and
- Five common concepts are illustrated in green parallelograms, whereas safety concepts are presented with bold text, and security concepts are presented with italic text.

In our previous paper [19], we provided detailed definitions of HO concepts and relationships related to the safety domain and CSO concepts and relationships related to the security domain.

VI. Step 3 - Evaluation of the developed hybrid ontology

This section evaluates the developed hybrid ontology through verification and validation methods. A preliminary evaluation was conducted after its construction to determine whether the hybrid ontology was feasible. As defined by Gómez-Pérez [26], ontology evaluation can be divided into two categories: verification and validation. In Gómez-Pérez’s words, ontology verification entails ensuring that the definitions of an ontology are aligned with its goals and competency questions (focusing on lexical and syntactic aspects) and apply to reality. The goal of ontology validation is to determine whether the definitions contained in an ontology accurately reflect the context within which they were intended [26]. According to Gómez-Pérez, it may be more convenient to evaluate whether the ontology was "built correctly" (verification) rather than whether the "right ontology" was built (validation).

a) Ontology verification: By verifying an ontology, we confirm that it meets certain quality criteria. The ontology taxonomy evaluation method is used to verify an ontology. To evaluate the taxonomy of the ontology, we followed the criteria described in [27]. Tab. I illustrates these criteria and their compatibility with the HTO.

b) Ontology validation: The validation process was carried out by evaluating the ontology content and answering competency questions. The ontology content was assessed based on the following main criteria: consistency, completeness, conciseness, expandability, and sensitivity [27].

Table III
TAXONOMY EVALUATION FOR THE HTO

Criteria	*Sub-criteria*	*Explanation*
Inconsistency	Circularity Errors	In the HTO, no concepts are classified as generalizations or specializations.
	Partition Errors	HTO does not contain any inappropriate definitions of disjoint concepts or incomplete concepts definitions.
	Semantic Errors	HTO does not include any concept that is a subclass of another concept.
Incompleteness	Incomplete Concept Classification	A complete set of domain concepts (safety and security) is included in the HTO.
	Partition Errors	Throughout the HTO, all relationships between concepts are clearly defined.
Redundancy	Grammatical redundancy	There are specific definitions for each concept in the HTO.
	Identical formal definitions of some classes	HTO does not contain any definitions that are identical to each other.
	Identical formal definition of some instances	HTO does not include instances with identical definitions.

Table IV
CONTENT EVALUATION OF HTO

Criteria	*Explanation*
Consistency	There is no contradiction between any of the concepts included in the HTO.
Completeness	HTO is completed according to the specifications established during the design phase.
Conciseness	There are no unnecessarily complex concepts or redundancies between concepts in the HTO, which makes it concise.
Expandability	A well-defined set of definitions in the HTO allows for easy expansion as there is no need to make major changes when new definitions are added.
Sensitiveness	Since the HTO is composed of well-defined concepts, small changes in definition will not affect the HTO.

The criteria and their compatibility with the HTO can be found in Tab. IV.

As a second method of evaluating the validity of the HTO structure, competency questions (CQs) were used. The answers to these questions allowed us to understand the scope of the HTO better. The CQs were answered and justified according to the HTO concepts. As shown in Tab. V, a subset of CQs that illustrate the main concepts of the HTO are presented, and their answers are provided. The HTO was evaluated using validation and verification methods after it was built, as shown in this section. According to the validation methods, the HTO content met all criteria for evaluation. Furthermore, all CQs were answered clearly and concisely, and the objective of building the ontology, outlined in the design phase, was accomplished. HTO was verified by evaluating the ontology taxonomy. All criteria were met, and no violations of the taxonomy were observed. Since the HTO addresses the required aspects of ontology development, these results indicate that the HTO is of high quality. Validation and verification were successfully achieved, enabling the HTO to be used in various applications.

Table V
ANSWERS TO CQS

Competency Questions (CQs)	*Answers*
CQ1. What is an asset/ environment object/ organization to be protected?	critical data owned by organization
CQ2. Which is an existing initiator factor/ vulnerability?	missing input validation
CQ3. Who is responsible for an attack?	hacker
CQ4. What attack can be conducted?	SQL injection
CQ5. What is a threat that threatens an asset/ environment object/ organization?	malformed input
CQ6. Which security goal can eliminate an initiator factor/ vulnerability?	confidentiality of critical data
CQ7. What countermeasure eliminates an initiator factor/ vulnerability?	input validation
CQ8. What initiating role can an asset/ environment object/ organization play?	provider
CQ9. What is an initiating condition that triggers an initiating event/ incident?	interception
CQ10. What is a harm truthmaker that characterizes a hazard element?	outdated application
CQ11. What hazard element can be played by an asset/ environment object/ organization?	receiver
CQ12. What exposure exists in a hazard/ consequence?	critical data being stolen
CQ13. What a control reduces a threat?	machine learning
CQ14. What initiating event/ incident can be detected by a control?	unauthorized access
CQ15. What is a hazard/ consequence that affects an asset/ environment object/ organization?	massive data breach
CQ16. What a mishap/ event is caused by a threat?	data exposure
CQ17. What mishap victim can participate in a mishap/ event?	organization X

VII. STEP 4 - APPLICATION OF THE DEVELOPED HYBRID ONTOLOGY

This section presents the use case of a quarry site with autonomous vehicles and the application of the HTO to an SoS. An industrial scenario demonstrated the HTO's applicability for identifying safety hazards and security threats in a real-life application context. A description of a use case is provided below.

A. A use case - Quarry site with autonomous vehicles

A quarry site with autonomous vehicles integrated new technologies with logical solutions for employing electric machines, which can be considered a system of systems (as shown in Fig. 4).

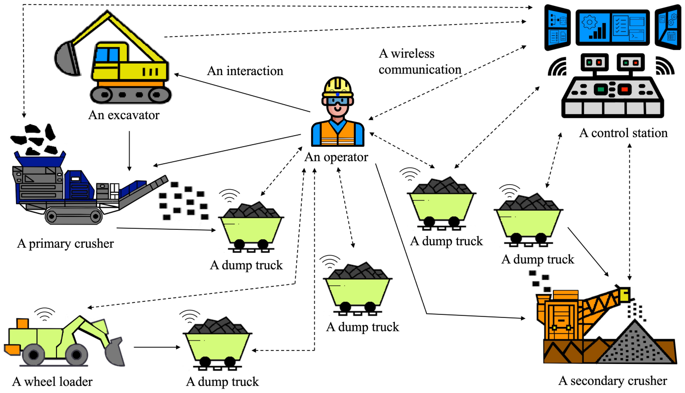

Figure 4. A quarry site with autonomous vehicles.

The implementation of autonomous electric vehicles in traditional quarries presents several challenges. Generally, crushing, aggregation and transportation of materials are the primary operations at a site. The transportation and dumping of aggregated material between the primary crusher and secondary crusher can pose an issue when autonomous vehicles such as dump trucks are used. All operating systems are monitored, managed, and controlled by a control station at the quarry site that facilitates wireless communication. Since all systems work together to accomplish the production goals of a quarry site, it is considered a *directed* SoS.

Over time, a central control station is responsible for managing the system to meet previously identified objectives and adding new ones as necessary. The systems are usually controlled by centralized objectives, but they can also be operated independently. During the transportation and unloading of materials at the quarry site, autonomous vehicles must follow a defined path and instructions without human assistance. As part of the SoS, personnel perform independent or collaborative tasks with autonomous vehicles. The SoS is intended to collaborate at all production process phases to accomplish its goals. Depending on the weather and terrain conditions, this process can adversely impact remote control, communication, and collaboration between systems.

Moreover, there is a possibility that any irrational behavior at the quarry site can negatively impact the life cycle of the SoS and pose hazards to both vehicles and people. Some harmful and undesirable emergent behaviors can result from the operation of constituent systems and their interactions. Constituent systems (CSs) included in an SoS can change over time and be added or removed as necessary. Potential scenarios and processes with potential hazards must be considered to ensure a safe and secure operation at the quarry site. When a hazardous scenario occurs, the CSs can change their states, assume new functions, or encounter new challenges (such as aggregating material, coordinating movements, and changing workplace conditions). For this reason, hazards and threats cannot be identified when only a single CS is analyzed.

B. Application scenario - Explosion of battery in the EV.

Fig. 5 illustrates how the HTO can be applied to identify potential hazards and threats associated with batteries in autonomous and remote-controlled vehicles deployed in a quarry. Accordingly, the application scenario was structured as follows:

A hacker conducts a false data injection attack (FDIA) that exploits the combustion of an electric vehicle (EV) battery, which leads to a battery management system (BMS) malfunction, resulting in a battery explosion and the injury of an operator. The identified concepts in Scenario 1 are as follows: *EV battery* is identified as an environment object/asset/organization. *EV battery can play the role of energy provider*, which can be considered an initiating role, and the role of *beingIgnited*, which can also be a hazard element. *A hacker* is identified as an attacker who conducts *False data injection attack (FDIA)* considered an attack. *Intrusion detection system (IDS)* is identified as a countermeasure that protects *integrity* considered as a security goal. In this scenario, *threshold monitoring* was identified as a control that can reduce a threat considered *EV battery overheating*. The occurrence of *combustion of EV battery* is identified as an initiator factor and vulnerability, while *BMS malfunction* is considered an initiating condition. *An unexpected conflagration of EV battery* is an initiating event/incident. *Flammability of a vehicle* is a harm truthmaker, and *Ignition of an EV* is an exposure. A hazard/ consequence is the *EV battery explosion*. A mishap/event is when *an operator is injured* and *an operator* is a mishap victim.

In the application scenario, the EV's battery can be attacked by a hacker, resulting in an explosion that can cause serious injury to the operator. Moreover, as illustrated in the scenario, there may be a threat to the BMS in electric vehicles that operate autonomously at a quarry site, which is critical for the battery's safety, performance, and reliability. During the development of autonomous vehicles, the BMS plays an important role in the safety of SoSs, along with communication channels, sensors, cameras, and radars. The BMS also facilitates communication between battery systems (including controllers) and external systems (including the control station). BMSs are becoming more interconnected, making them more hazardous and more likely to pose safety and security threats. However, system designers and BMS developers often fail to consider security threats when designing a system. Due to the increasing dependence of SoSs and BMSs on the Internet for operational functions such as performing, monitoring, and controlling maintenance, security must be considered. This scenario outlines general security vulnerabilities that may be caused by potential cyber-attacks, along with possible countermeasures and control methods for BMS developers. The malfunction or failure of a BMS can result in injuries to operators or damage to machines and vehicles working nearby. Identifying threats and hazards protects BMSs from malicious cyber-physical attacks and can be used safely for various SoS applications. The scenario presents an opportunity to identify both safety hazards and security threats simultaneously by utilizing our HTO method.

VIII. Related Work

The Combined Harm Assessment of Safety and Security for Information Systems (CHASSIS) was developed by Raspotnig et al. to integrate safety and security techniques and provide a united approach to the early stages of developing a system [28]. In this approach, requirements are defined using HAZOP tables [3] along with Boolean logic Driven Markov Processes (BDMP) [29] and ISO 26262 [18]. In contrast to our approach, CHASSIS requires advanced expert knowledge and detailed analysis (i.e., alignment with existing safety processes and detailed guidelines).

Silva and Lopes introduced Failure Modes, Vulnerabilities, and Effect Analysis (FMVEA) [30], based on the safety approach FMEA, which is presented in IEC 60812 [31]. This method combines the failure mode and failure effect model

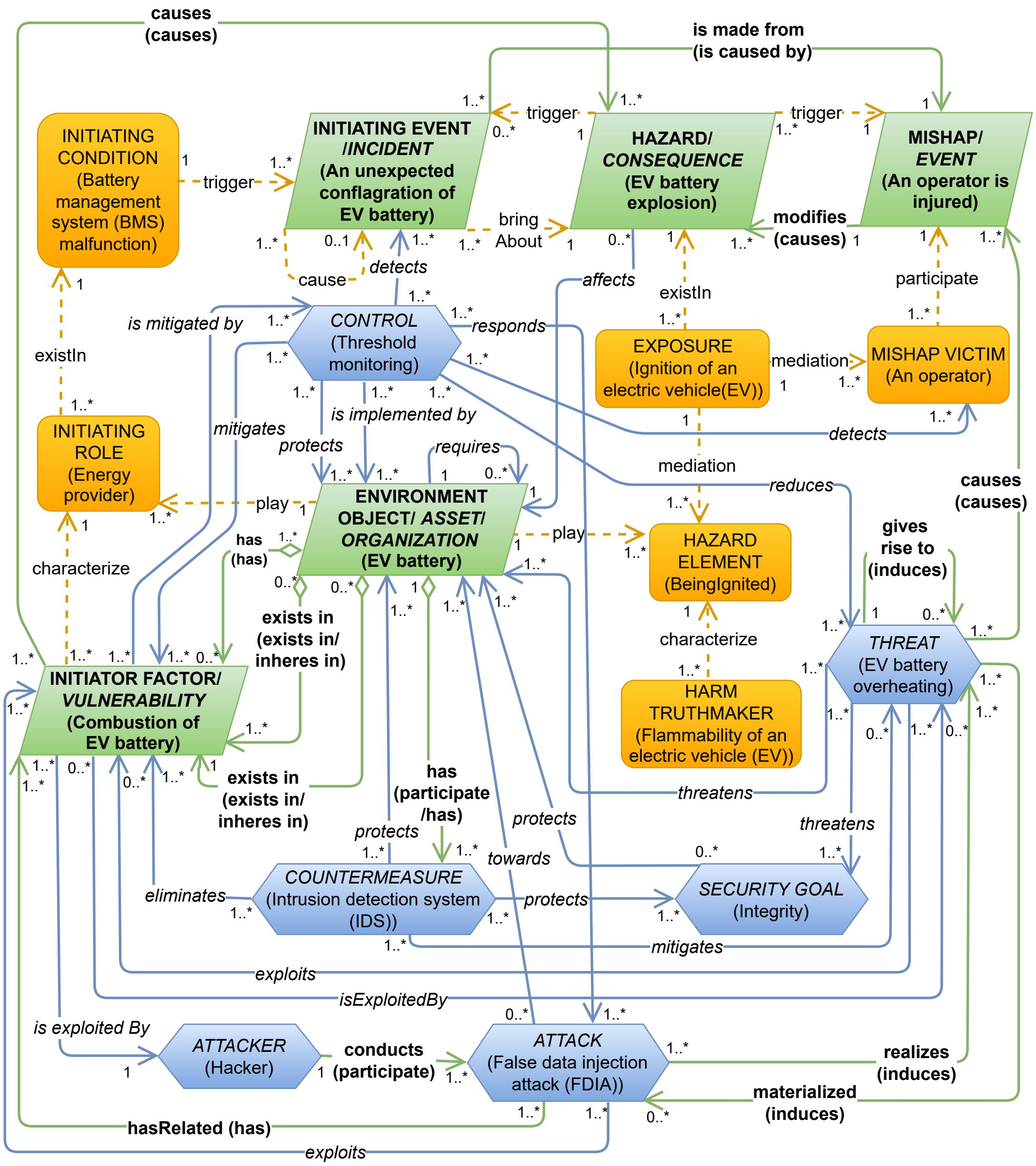

Figure 5. HTO - Scenario 1 - A battery in an electric vehicle overheats and explodes, causing injury to the operator.

to analyze cause and effect relationships in the safety and security domain. FMVEA is appropriate for the design and verification stages of system development and for analyzing single cases. It can reuse previously gathered results and be repeated if a new vulnerability or threat arises. However, this method does not constitute a complete approach and may only partially overlap with ours (i.e., FMVEA primarily identifies the hazards and threats associated with a single component). Using the FMVEA, safety and security analyses can be performed independently, but the interactions between the two may be overlooked.

Using System-Theoretic Process Analysis (STPA), Young and Leveson [32] introduced the STPA-Sec approach, designed for the concept stage of system development. To identify and prevent vulnerable states in a system, STPA-Sec relies on collaboration between experts in security, domain, and operations and can be used to consider both security and safety properties simultaneously. Our approach is similar to STPA-Sec as it can be used during the concept stage; however, unlike STPA-Sec, it covers the entire development process.

Macher et al. [33] presented a Security-Aware Hazard and Risk Analysis (SAHARA) which integrates the approaches of STRIDE (Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, Elevation of privilege) [34] from the security domain and HARA (Hazard analysis and risk assessment) [18] from the automotive domain. A safety analysis is performed using ISO 26262 [18] and the HARA method, whereas a security analysis is conducted independently using the STRIDE approach. Based on the security analysis results, security levels are calculated using the automotive safety integrity level (ASIL) quantitative concept. The authors introduced information regarding the available resources allocated for addressing security threats under the SAHARA approach. Our paper presents a systematic approach to incorporating security into safety analysis using a specific security method CSO and begins at an early stage of system development.

The Systems-Theoretic Likelihood and Severity Analysis (STLSA) is an approach presented by Temple et al. [35] that merges STPA-Sec [32] and FMVEA [30]. STLSA considers semi-quantitative risk assessment aligning with EN 50126, system-level functional control, and human-in-the-loop. For each unsafe control action, STPA-Sec evaluates the loss scenarios, while FMVEA evaluates the level of risk associated with the action. However, no guidelines exist for addressing and evaluating the severity of the failures and threats identified in scenarios. Despite combining STPA-Sec and FMVEA, this approach falls short of our hybrid ontology (i.e., it focuses on cybersecurity concerns but lacks a graphical representation and risk assessment).

IX. Discussion and Conclusion

This paper introduced the Hazard and Threat Ontology (HTO) for modeling and identifying safety hazards and security threats. A hybrid approach to safety and security requires appropriate handling of complex scenarios. While developing our hybrid ontology, we discovered that different ontologies (Hazard Ontology from the safety domain provided by Zhou et al. [16] and our Combined Security Ontology from the security domain) used different concepts to describe the same concepts. Among the definitions provided by Zhou et al. [16], the *Initiating Event* is defined as an undesirable or unexpected event that can trigger a hazard. In our work, the terms *Initiating Event* (safety) and *Incident* (security) are equivalent and interchangeable, as well as *Initiator Factor* (safety) and *Vulnerability* (security). Moreover, we introduced the *Environment Object/Organization* (safety) as an equivalent entity to *Asset* (security). Our ontology includes the concept *Consequence*, which is equivalent to the *Hazard* concept described by Zhou et al. [16]. The concept of *Mishap* (Zhou et al. [16]) in our ontology has the same meaning as the concept of *Event*, and they are considered equivalent. The hybrid ontology incorporates safety and security assessment into the design phase, whereas the approach presented by Zhou et al. [16] addressed only safety issues.

Through our approach, we can identify hazards and threats to security and safety within a system, allowing us to analyze them jointly. Like other safety and security analysis methods, our approach relies on and is limited by the assumptions and knowledge encoded in ontologies and models. Furthermore, HTO has been applied at the system level; thus, additional concepts to prioritize failure modes and attack mechanisms can be added in the future.

HTO inherits constructs from HO [16], which assists engineers in generating safety-related scenarios. Using our approach, system engineers can identify security threats in scenarios not covered by the HO, as demonstrated in our use case. An explanatory example of this is shown in Fig.5, where an attack results in an EV battery's explosion and an operator's injury are identified. The HTO can also assist system engineers in determining the system's appropriate security measures (i.e., control and countermeasure). Based on the stored knowledge of attacks, threats, hazards, assumptions, and elaborated models, the hybrid ontology can be applied to all possible scenarios. In our experience, establishing a causal relationship between attack and potential factors is critical to the approach. Cybersecurity experts need to evaluate attack scenarios to identify which attack may indicate a causal factor. To accomplish this task, they should have a broader understanding of the system than one focusing solely on security. Based on this knowledge, the HTO can support the development of threat scenarios and recommendations. It was applied to an industrial use case, where it was observed that identifying security threats is more complex than identifying safety hazards. Safety hazards typically result from systematic or well-known random faults, so safety analysis becomes more systematic. Security threats are primarily posed by malicious activity (i.e., adversarial attackers, including trusted individuals, seek to cause harm, making it difficult to understand and assess all potential threats. Furthermore, the application demonstrates that the HTO can identify potential safety hazards and security threats early in developing systems. Our application shows that our HTO works similarly to HAZOP, FMEA, and STPA methodologies but with a particular focus on security threats. Domain experts can successfully perform analysis to protect against hazards and threats using the information contained in the HTO. Our hybrid approach can be applied to other scenarios and systems, allowing for a deeper understanding and a systematic way to identify and analyze safety hazards, security threats, and their dependencies. We envision the following future works:

- Examine scenarios associated with platooning or cooperative driving that pose potential safety hazards and security threats;
- Apply HTO in other related domains, such as renewable energy, energy management, healthcare, and the Internet of Things (IoT);
- Extend HTO to include additional concepts related to attack mechanisms, failure modes, dependability, or risk; and
- Involve Explainable Artificial Intelligence (XAI) methods together with HTO to support safety and security analyses.

Acknowledgment

This work is partially supported by the Swedish Foundation for Strategic Research through the project Serendipity—Secure and Dependable Platforms for Autonomy (RIT17-0009), by the Knowledge Foundation through the projects DPAC (20150022) and MoDEV (20200234), by Vinnova through the project iSecure (202301899), and by the Swedish Knowledge Foundation through the project ORPHEUS (202203408).

References

[1] C.B. Nielsen, P.G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, "Systems of systems engineering: basic concepts, model-based techniques, and research directions," ACM Computing Surveys (CSUR), 48 (2) (2015):1-41.

[2] J. Fitzgerald, P.G. Larsen, and J. Woodcock, "Foundations for Model-Based Engineering of Systems of Systems," Springer International Publishing, Cham, 2014, pp. 1–19.

[3] J. Axelsson, "A systematic mapping of the research literature on system-of-systems engineering," 10th System of Systems Engineering Conference (SoSE), San Antonio, TX, USA, 2015, pp. 18-23.

[4] D.P. Pereira, C.M. Hirata, and S. Nadjm-Tehrani, "A STAMP-based ontology approach to support safety and security analyses," J. Inf. Secure. Appl., 47 (2019):302-319.

[5] L. Piètre-Cambacédès, and M. Bouissou, "Cross-fertilization between safety and security engineering," Reliab. Eng. Syst. Saf. 110 (2013):110-126.

[6] Z. Enrico, "The future of risk assessment," Reliab. Eng. Syst. Saf. 177 (2018):176-190.

[7] M. Adach, K. Hänninen, and K. Lundqvist, "Security Ontologies: A Systematic Literature Review," In: Almeida, J.P.A., Karastoyanova, D., Guizzardi, G., Montali, M., Maggi, F.M., Fonseca, C.M. (eds) Enterprise Design, Operations, and Computing (EDOC 2022). LNCS, vol 13585. 2022. Springer, Cham.

[8] M. Adach, K. Hänninen, and K. Lundqvist, "A Combined Security Ontology based on the UFO ontology," 16th IEEE International Conference on Semantic Computing (ICSC), Laguna Hills, CA, USA, 2022, pp.187-194.

[9] G. Guizzardi, "Ontological foundations for structural conceptual models," PhD thesis, University of Twente, The Netherlands, 2005.

[10] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition) - W3C Recommendation 11 December 2012.

[11] Object Management Group. Unified Modeling Language (UML). OMG, Milford (2017). Available online: www.omg.org/spec/UML (accessed October 20, 2023).

[12] N.H. Guzman, J. Zhang, J. Xie and J.A. Glomsrud, "A Comparative Study of STPA-Extension and the UFoI-E Method for Safety and Security Co-analysis," Reliab. Eng. Syst. Saf. 211 (2021): 107633.

[13] S. Verma, T. Gruber, Ch. Schmittner and P. Puschner, "Combined Approach for Safety and Security," In: A., Romanovsky, E., Troubitsyna, I., Gashi, E., Schoitsch, F., Bitsch,(eds) Computer Safety, Reliability, and Security. SAFECOMP Workshops, LNCS, vol. 11699. Springer, Cham, 2019, pp.87-101.

[14] J. Dobaj, C. Schmittner, M. Krisper and G. Macher, "Towards Integrated Quantitative Security and Safety Risk Assessment," In: A., Romanovsky, E., Troubitsyna, I., Gashi, E., Schoitsch, F., Bitsch, (eds) Computer Safety, Reliability, and Security. SAFECOMP Workshops, LNCS, vol 11699. Springer, Cham, 2019, pp.102-16.

[15] M. Adach, A. Nazakat, K. Hänninen, and K. Lundqvist, "Hazard Analysis on a System of Systems using the Hazard Ontology," 18th Annual System Of Systems Engineering Conference (SoSE), Lille, France, 2023.

[16] J. Zhou, K. Hänninen, K. Lundqvist, and L. Provenzano, "An ontological interpretation of the hazard concept for safety-critical systems," 27th European Safety and Reliability Conference (ESREL), Portoroz, Slovenia, 2017, pp.183-185.

[17] C. Nielsen, P. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, "Systems of systems engineering: Basic concepts, model-based techniques, and research directions," ACM Comput. Surv. 48 (2) (2015):1–41.

[18] M. Maier, "Architecting principles for systems-of-systems," Syst. Eng.: J. Int. Counc. Syst. Eng. 1, (4), (1998):267–284.

[19] M. Adach, K. Hänninen, and K. Lundqvist, "Concepts and Relationships in Safety and Security Ontologies: A Comparative Study," 6th International Conference on System Reliability and Safety (ICSRS), Venice, Italy, 2022, pp.357-364.

[20] G. Guizzardi, R. Falbo, and G. Guizzardi, "Grounding software domain ontologies in the Unified Foundational Ontology (UFO): the case of the ODE software process ontology," Iberoamerican Conference on Software Engineering (CIbSE), Recife, Pernambuco, Brasil, 2008, pp.244-251.

[21] ISO/IEC/IEEE 16085 (2021). "Systems and software engineering - Life cycle processes - Risk management," Geneva: International Organization for Standardization and International Electrotechnical Commission.

[22] ISO/IEC 15026-1 (2015). "Systems and software engineering - Systems and software assurance," Geneva: International Organization for Standardization and International Electrotechnical Commission.

[23] ISO/IEC 27001 (2022). "Information security, cybersecurity, and privacy protection - Information security management systems - Requirements," Geneva: International Organization for Standardization and International Electrotechnical Commission.

[24] ISO/IEC 15026-3 (2023). "Systems and software engineering - Systems and software assurance," Geneva: International Organization for Standardization and International Electrotechnical Commission.

[25] R. Ross and M. McEvilley and J.C. Oren, "NIST SP 800-160. Systems Security Engineering Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure System," National Institute of Standards Technology, US Department of Commerce, Gaithersburg, MD, USA, Tech Report NIST SP, 2016.

[26] A. Gómez-Pérez, "Ontology evaluation," in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, Springer, Berlin, Heidelberg, 2004, pp. 251–274.

[27] S. Lovrencic and M. Cubrilo, "Ontology evaluation-comprising verification and validation," in Proc. 19th Central Eur. Conf. Inf. Intell. Syst. 2008, pp. 657-663.

[28] C. Raspotnig, P. Karpati, and V. Katta, "A Combined Process for Elicitation and Analysis of Safety and Security Requirements," 13th International Conference Enterprise, Business-Process and Information Systems Modeling, vol. 113, Springer, Berlin, Heidelberg, 2012, pp.347-361.

[29] L. Piètre-Cambacédès and M. Bouisou, "Modeling safety and security interdependencies with BDMP (Boolean logic Driven Markov Processes)," IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, 2010, pp.2852-2861.

[30] N. Silva and R. Lopes, "Practical experiences with real-world systems: Security in the world of reliable and safe systems," 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W), Budapest, Hungary, 2013, pp.1-5.

[31] International Electrotechnical Commission, "IEC 60812: Analysis techniques for system reliability - procedure for failure mode and effects analysis (FMEA)," 2006.

[32] W. Young and N. Leveson, "Systems thinking for safety and security," 29th Annual Computer Security Applications Conference, ACM, New York, USA, 2013.

[33] G. Macher, A. Höller, H. Sporer, E. Armengaud, and C. Kreiner, "A Combined Safety-Hazards and Security-Threat Analysis Method for Automotive Systems," 34r34thternational Conference on Computer Safety, Reliability, and Security (SAFECOMP), vol. 9338, Springer, Cham, 2014, pp.237-250.

[34] Microsoft Corporation, "The STRIDE threat model," 2005.

[35] W. G. Temple, Y. Wu, B. Chen, and Z. Kalbarczyk, "Systems-theoretic likelihood and severity analysis for safety and security co-engineering," in Reliability, Safety, and Security of Railway Systems. Modeling, Analysis, Verification, and Certification. vol 10598. Springer, Cham, 2017, pp.51-67.