Towards an Integrated Safety-Security Ontology for System of Systems

Nazakat Ali, Julieth Patricia Castellanos-Ardila, and Sasikumar Punnekkat
School of Innovation, Design and Engineering
Mälardalen University - Västerås, Sweden
{nazakat.ali, julieth.castellanos, sasikumar.punnekkat}@mdu.se

***Abstract*—In the modern world, connectivity and shared intelligence enable independent constituent systems (CS) to form systems of systems (SoS) capable of performing sophisticated missions. However, the sheer scale of an SoS can make it challenging to manage all components comprehensively, hiding potential security and safety concerns. These factors underscore the need for advancing conceptual models that permit a better understanding of the SoS intricacies. This paper presents a conceptual model for an integrated safety-security ontology for SoS, called SSO-SoS. Such a model is based on international standards, existing literature, and relevant conceptual models, where we pay special attention to safety, security, and mitigation for SoS. We also illustrate the SSO-SoS with a case study from the construction sector. Our conceptual model provides a hierarchical organization that permits stakeholders to navigate through different layers of information, enhancing their ability to identify, address, and understand the required SoS knowledge.**

***Index Terms*—system of systems, ontology, safety, security**

I. Introduction

In the era of connectivity, the System of Systems (SoS) concept is gaining popularity in many fields, including defense, aerospace, transportation, energy, and disaster management [1]. Systems of Systems (SoS) are arrangements of interconnected constituent systems (CS) configured to achieve desired systemic effects [2]. Often, the CS are independent entities [3], i.e., operationally (i.e., a CS is a functional entity on their own) and managerially (i.e., a CS has different owners). For instance, various CS work together in a smart city (e.g., intelligent transportation system) to enhance urban living through improved efficiency, safety, and security.

A SoS exploits opportunities by providing new services that a single CS cannot provide. However, an SoS might hide potential safety and security concerns due to its complexity, emergent behavior, unpredictability, and heterogeneity [4]. For instance, failures in a smart transportation system can cause severe consequences, such as traffic accidents and disruptions in public transit services, which can lead to both financial and human losses. These factors underscore the need for advancing conceptual models that permit a better SoS understanding.

This paper presents a conceptual model called SSO-SoS. Such a model aims to provide a structure for representing and managing the SoS knowledge focusing on safety and security. For this, we analyze the concepts presented on international standards, existing literature, and relevant conceptual ontological models. We pay special attention to those SoS aspects relevant to safety and security and model them. We also illustrate the SSO-SoS with a case study from the construction sector from which we derived essential benefits. First, SSO-SoS provides a structure for representing and managing the knowledge complexity inherent to the SoS context, facilitating the brainstorming required during design phases. Second, it provides a hierarchical structure that can be used to represent various levels of abstraction, from high-level configuration to risk mitigation strategies (aligned with regulatory frameworks) to deal with the safety and security concerns identified in such an SoS. This hierarchical organization allows stakeholders to navigate through different layers of information, enhancing their ability to understand the SoS from different perspectives.

The rest of this paper is organized as follows. In Section II, we present background and related work. In Section III, we describe our proposed ontology. In Section IV, we describe the evaluation of our proposed ontology. In Section V, we present a case study. In Section VI, we discuss the findings. Finally, in Section VII, we present conclusions and future remarks.

II. Background and Related Work

A. Ontology

An ontology is a formal, structured representation of knowledge within a specific domain that systematically captures concepts and relationships between them [5]. Systematic Approach for Building Ontologies (SABiO) [6] is considered an ontology engineering method that integrates practices from software engineering. It provides activities that are applied to develop domain reference ontologies as well as the design and coding of operational ontologies. SABiO process recommends five main steps: 1) purpose identification and requirements elicitation; 2) ontology capturing and formalization; 3) ontology designing; 4) implementation; and 5) testing. We aim to develop a reference ontology (a conceptual model); therefore, we only focus on the initial two steps. The first step recommends the formulation of Competence Questions (CQs), i.e., the questions that the ontology should be able to answer. We initiate the ontology-building process by engaging with domain experts and examining existing standards and literature to

This work is supported by SIMCON project-funded by Vinnova and DAISY project- funded by SSF (Swedish Foundation for Strategic Research).

construct our reference ontology. We also investigated relevant standards and existing literature in systems, SoS, and safety engineering. The reference ontology (SSO-SoS) is developed by extending the concepts of key reference ontologies [7], [8]. In addition, it is grounded on the Unified Foundational Ontology (UFO) [9] for capturing real-world semantics.

B. System of Systems (SoS)

A System of Systems (SoS) is a set of systems or system elements that "interact to provide a unique capability that none of the constituent systems (CS) can accomplish on its own [3]. Maier [2] has outlined five main characteristics of an SoS, which are: 1) *operational independence*, meaning that CS are independent and operate independently to achieve their own individual goal; 2) *managerial independence* meaning that CS are managed independently; 3) *evolutionary development* that means that SoS can evolve as a result of changes in the environment, changes in the CS itself, or changes in the purpose of SoS as a whole; 4) *distribution*, meaning that the CS are geographically distributed and communication channel is needed to exchange the required information; and 5) *emergent behavior*, meaning that the behavior of SoS emerges due to the interaction among/between CS.

The level of independence of CS depends on the level of authority over them. Therefore, SoS can be classified into four categories [2], [10]: 1) *directed SoS* is type of SoS where SoS is centrally managed by an authority that is responsible for executing the operations; 2) *acknowledged* is a type of SoS which has recognized objectives and a central manager, however, CS maintain their independent ownership and objectives while operating within the SoS; 3) *collaborative* SoS has no central control and CS work together to achieve an agreed goal; and 4) *virtual* SoS is a type of SoS where there is no central control and no commonly established goals.

C. Related Work

HARA and TARA are the first steps in the safety analysis to identify potential hazards and threats, respectively. Zhou et al. [11] proposed an ontological approach to identify hazards in safety-critical systems. In [12], authors have applied their proposed hazard ontology in an SoS domain (query automation) to identify hazards SoS-related hazards. The authors found that hazard ontology can be used to identify hazards in SoS because it helps to identify hazards that emerge due to interactions among constituent systems (CS) within an SoS. The authors extended their hazard ontology [13] and incorporated concepts from ISO 26262 to make it exclusive to the automotive domain.

Ensuring safety and security in SoS is a challenging task due to their collaborative nature, which introduce significant complexity [14]. The challenges are amplified when it comes to the safety-critical SoS. Ali et al. [15] have investigated safety challenge where they proposed an ontology-based failure detection and prevention framework that utilizes a knowledge base to predict potential failures in the system and suggests recommended actions for those particular failures. In their another research [16], the authors have developed a tool called *SoCPSTracer* that considers hazard analysis for SoS at design time and provides a fault traceability graph to predict faults and their potential propagation in the network of SoS. However, the authors made an assumption that hazard the analysis artifacts for all the participating CS are provided, which is true in the case of directed SoS but can not be applied to another type of SoS.

Safety and security are important attributes of any system which bring trust to the system. The potential faults and security threats in SoS may lead to severe consequences. This is due to the fact that any fault in a CS may propagate into another CS, and sometimes, it accumulates and triggers hazards. Bhosale et al. [17] have highlighted the importance of safety, security, and risk assessment in industrial control systems. The authors have proposed an ontology-based approach for integrating safety and security risk assessment. The application of ontology is demonstrated to establish safety and security relations, which shows its potential to evaluate overall risk in industrial control systems. However, the proposed integrated ontology is too general and cannot provide information about the basic characteristics, i.e., SoS capabilities, constellation, emergent hazard and etc.

Reliability is another important system attribute that must be ensured for most software-based systems and even more important for SoS operating in the safety-critical domain. To address this challenge, Ferreira et al. [18] presented an ontology-based conceptual model for SoS reliability that captures key concepts and relationships to enhance the understanding of reliability in SoS. The proposed model has twenty-six concepts, their definitions, and relationships that are represented through a UML diagram. The authors have concluded that the proposed conceptual model can aid in the design, development, and management of reliable SoS. The authors investigated reliability in detail, however, safety and security attributes were not discussed.

III. Proposed Ontological Approach

We follow SABiO methodology [6] to develop the SSO-SoS ontological conceptual model. We first formulate a set of CQs that serve as functional requirements of the ontology. We have investigated standards in systems and engineering [3], [19], and functional safety standards [20]–[22] in order to elicit CQs. The CQs are listed as follows:

- CQ1: What is the *System of System (SoS)* composed of?
- CQ2: What is the *Common Mission* for the SoS? How can it be achieved?
- CQ3: What is a *Constituent System* of an SoS?
- CQ4: What is *SoS Capability*?
- CQ5: What is the *Configuration*?
- CQ6: What is a *Intended Functionality*?
- CQ7: What is a *Mediator*?
- CQ8: What is *Security Property*?
- CQ9: What is *Emergent Hazard*? and how it is identified?
- CQ10: What is *Threat*? and how it can be identified?

- CQ11: What is *Vulnerability*? and how it leads to *Damage Scenario*?
- CQ12: What is *Risk Factor*? and what is relationship between *Risk Factor* and *Risk Reduction Level*?
- CQ13: What are the *SoS Level Requirements* ? and how it related to the *Countermeasures* ?

Fig. 1 illustrates a core ontology for a System of Systems (SoS) that provides a high-level view and describes the complex relationships and interactions essential for achieving a common mission and managing hazards within an SoS.

In our ontology, one of the core concepts is the **Mission**, which represents the primary goal that needs to be accomplished by the SoS. A mission represents a set of objectives and goals to be achieved within a specific operational environment [23], [24]. The mission is composed of a **Mission Thread** that describes the mission in a high-level language. It summarizes the flow of activities required to complete a mission. The mission can be divided into a **Common Mission** and **Individual Mission**. A common mission is a type of mission that defines objectives and goals intended to be considered by Configuration. Individual mission is a type of mission that defines the objectives and goals intended to be accomplished by an individual CS to achieve a common mission in SoS. The mission has **Constraints** and **Priorities** for each task to execute the mission. The priority, often represented as an integer, defines the commitment level of the system within the mission while the constraints are the restrictions or limitations that impact how the mission is carried out [8]. These can include time constraint, resource availability, regulatory requirements, and environmental conditions. **Configuration** is "a composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to meet a capability " [25]. It is the property of an SoS where it determines SoS intended functionalities by considering the SoS Common mission and forms **Constellation** of various CS to achieve the common mission. When an entire set or subset of CS establishes an interaction link as a result of configuration, is called a constellation.

An SoS is composed of multiple **Elements**, including two or more CS and **Mediators**, each providing specific capabilities that together form the overall SoS capability. Additional elements can be introduced when necessary to achieve a specific mission. Any CS is considered to be an independent system, having its own development, management goals and resources, but also interacts with others within a SoS to provide unique capabilities [18]. Capability is the ability of a CS to contribute to a common mission under specified standards and conditions by combining various methodologies to perform a set of tasks [7], [18], [26]. **SoS Capability** is the ability of the entire SoS to demonstrate unique behavior(s) through the capability configurations of its associated CS [7], [27]. **Mediator** is an element that facilitates communication, coordination, and collaboration among CS [28].

The **Operational Context** defines the environment and conditions in which the SoS operates, which brings about an **Emergent Hazard**. It also influences SoS configuration. The configuration considers a common mission to form a constellation. A constellation is formed by grouping two or more CS. The configuration determines SoS **Intended Functionality** that can be affected by **Vulnerability** (CS's vulnerability or SoS's Vulnerability). Vulnerability refers to weakness or control function when it is exploited, it compromises the security properties and may lead to **Damage Scenario**. **Security Property** is the SoS attribute that ensures protection against threats. The security property can be analyzed by **TARA** (Threat and Risk Assessment). This process identifies potential **Threats**, which in turn determine security requirements. These requirements are met by defining appropriate countermeasures. These threats include various **Risk Factors** e.g. potential impact, likelihood and etc. that directly impact the safety goals and corresponding countermeasures of SoS. The risk factors are provided by relevant standards. In safety, the risk factor [20] is the consequence of the severity, frequency of exposure, the possibility of avoidance, and demand rate. This combination of parameters determines the safety risk reduction level required for the safety countermeasure. Regarding security [21], the risk factors are potential impact (high) and likelihood (possible), which together determine a security level, which is required for the security countermeasure. A threat, according to IEC 62443 [21], is a circumstance or event having the potential to substantially impact organisation's operations.

Intended Functionality refers to specified functionality that is designed to achieve a specific mission. These functionalities set operation goals that SoS is supposed to achieve by integrating its CS. At design time, intended functionalities are analyzed with **HARA** (Hazard Analysis and Risk Assessment). This process identifies the **Emergent Hazards**. The HARA process also suggests safety **Countermeasures** to mitigate the identified hazards. These countermeasures are developed to conform to the risk reduction levels recommended by relevant standards. Moreover, the countermeasures fulfill **SoS-Level Requirements**, which are established to mitigate the identified emergent hazards effectively. **SoS-Level Requirement** represents a set of documents or artifact that outlines what an entire SoS must fulfill in terms of both functional and non-functional aspects. **Safety Requirement** is a type of SoS-level requirement that ensures that SoS operates within acceptable safety margins. It also defines safety-critical functions and ensures they are adequately mitigated through safety countermeasures. Likewise, **Security Requirement** is a type of SoS-level requirement that defines access control policies, encryption standards, or data breaches [21]. It also defines security-critical functions and ensures they are adequately mitigated through security countermeasures.

Hazard Analysis is conducted to identify and assess hazards, particularly those that emerge from the interactions among/between CS within the SoS, known as emergent hazards. An emergent hazard refers to a potential risk or danger that arises unpredictably from the interactions, dependencies, and behaviors among multiple interconnected CS within the SoS. Various types of emergent hazards can arise in SoS [29]. In particular, **Reconfiguration Hazards** is a type of emergent

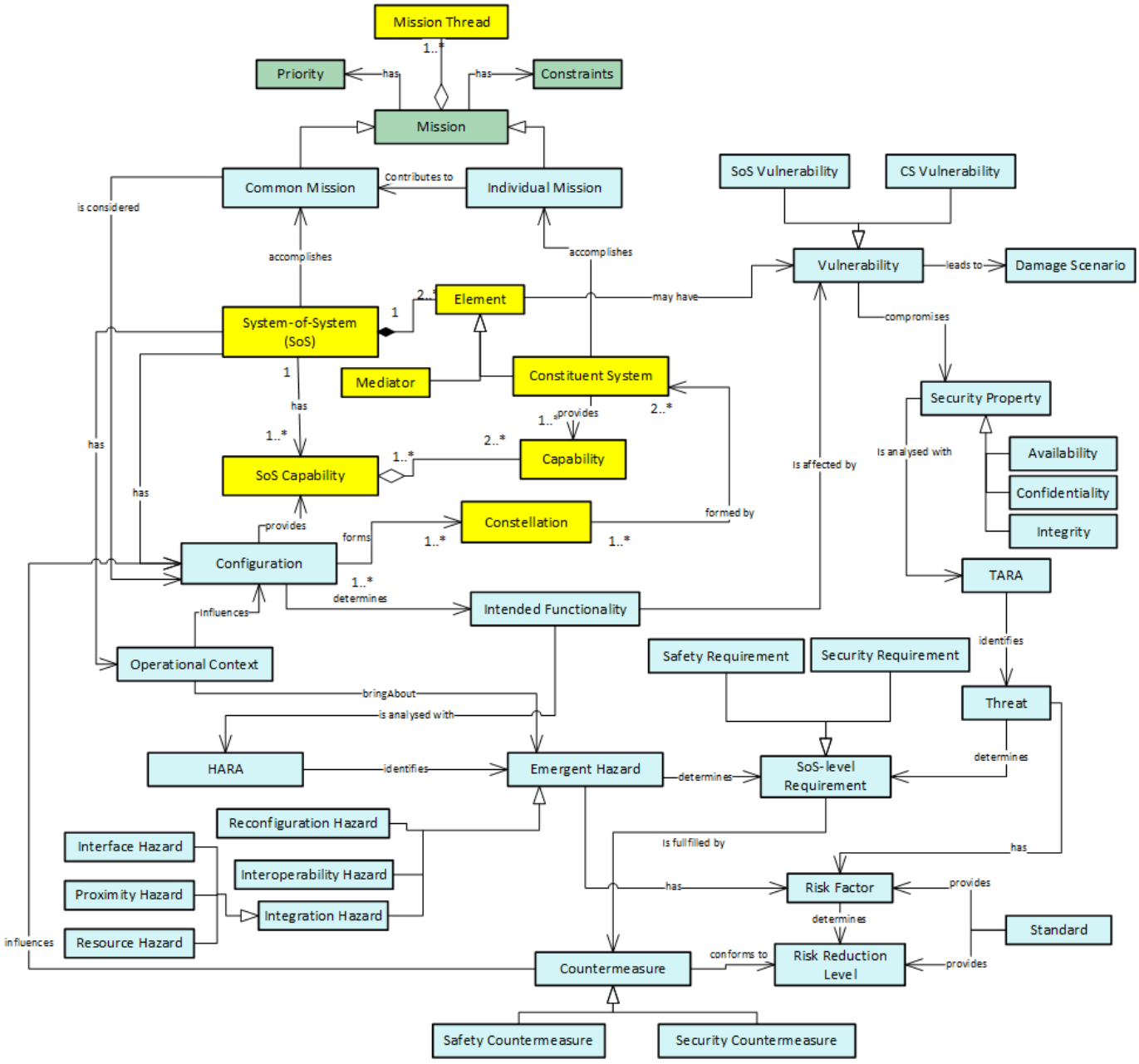

Fig. 1. SSO-SoS: Concepts are represented as rectangles. Concepts taken from [7] are presented in yellow color and some concepts related to Mission are taken from [8] and represented in green color. The concepts represented in light blue are gathered after investigating existing literature in detail

hazard that stems from changes in the SoS configuration; **Integration Hazard** is a type of emergent hazards that arise due to the heterogeneity in the SoS. **Interoperability Hazard** is a type of emergent hazard that arises from interpreting the information from one CS by another CS in a way that the first CS did not intend. It can also occur due to synchronization issues, e.g., response time etc. **Interface Hazard** is a type of **Integration Hazard** that may arise from sharing faulty data with another CS through a defined channel. **Proximity Hazard** is a type of integration hazard that arises when CS operate within a short distance; **Resource Hazard** involves competition for or degradation of shared resources.

The SoS elements may have vulnerabilities. This is because the interconnected nature of SoS may increase the point of vulnerabilities, exposing the entire SoS to cyber-attacks leading to potential damage scenarios. These vulnerabilities can compromise the security property i.e., confidentiality, integrity, or availability of both the asset itself and the overall SoS.

IV. Ontology Evaluation

A. Ontology Verification

The purpose of ontology verification is to verify that the ontology is built accurately to ensure no inconsistency and

coherence issues. To meet this primary goal, we have done verification in a competency-question-driven manner and created a table indicating SSO-SoS elements (concepts, relations and axioms) to answer each CQ. Table I shows the verification of SSO-SoS where we see that SSO-SoS is capable of responding to all defined CQs.

B. Ontology Validation

The validation of ontology aims to show that SSO-SoS meets its purpose: 1) it manages the domain knowledge of safety, security and mitigation for SoS; 2) it supports the identification of emergent hazards, security threats, derives SoS level safety and security requirements, and designs safety and security countermeasures. As mentioned earlier, We follow the SABiO methodology and take a case study construction domain to investigate whether the built SSO-SoS can be applied to demonstrate the mentioned case study. Section V explains the ontology validation in detail.

V. Case Study: Mass Removal Operation

In this section, we use the ontology (see Section III) to represent an SoS for mass removal in the construction domain.

A. Systems of System (S0S) Configuration

The **SoS** (see the top-left side in Fig. 2) refers to the integration of systems for mass removal in a construction site. It exists to accomplish a **Common Mission**, i.e., prepare the construction site for subsequent construction activities. The mission has **Priority** high and a **Constraint**, i.e., it requires electrified systems. Several **CS** are included, i.e., excavators, loaders, trucks, and human operators, with specific **Capabilities**. For example, beyond the earthwork excavation, CS1 provides direct human control and decision-making, while CS2 provides less human exposure and potential for automation but is connectivity-dependent. The **Configuration** in the SoS manages the relationships, interactions, and interoperability among CS. It considers the **Mission Threads**, i.e., excavation (E), material loading (ML), and transportation and disposal (TD), and the **Operational Context** in which the SoS operate, i.e., the terrain, soil, and weather variation. The configuration provides the **SoS Capability** of leveraging electric-powered equipment and machinery to enhance efficiency, safety, and sustainability by forming thread-oriented **Constellations**. For instance, the mission thread excavation (E) can be accomplished by forming a constellation composed of one of the excavators (C1 Or C2) and the human team (CS8). An initial set of **Mediators** is also configured to facilitate the flow of information and resources, e.g., the site control system, which is in charge of the operational management, and the fleet management system, which focuses on vehicle monitoring.

B. SoS Safety

The SoS configured in Section V-A is arranged to have a **SoS Intended Functionality**, i.e., CS synchronized and perform safe movements to optimize mass excavation, material loading, transportation, and disposal. This functionality is essential to reach the SoS's full potential. In addition, the functionality is safety-related since humans are involved in the operation. Therefore, it requires to be analyzed with the **HARA - Hazard Analysis and Risk Assessment** (see the bottom-left side in Fig. 2). For this, we used STPA (System Theoretic Process Analysis) [30], an analysis technique aiming to accumulate information regarding system safety constraints to enforce them during the system lifecycle. In STPA, the initial step is determining the loss we want to avoid when the intended functionality is in operation. We focus on avoiding the loss of human lives or injuries. As a result, we identify the **Proximity Hazard**, i.e., machines infringe on the safe space of workers or other machines in the work area during operations. The hazard is used to determine the **Safety Requirement**, i.e., machines shall not infringe on the safe space of workers or other machines in the work area during operations.

C. SoS Security

The **SoS Intended Functionality** also requires protection in terms of security (see the top-right side in Fig. 2) due to the required interconnections needed in its configuration. In particular, the SoS intended functionality could be affected by a **SoS Vulnerability**, e.g., lack of secure authentication for communication between excavation machinery and the site control system. This vulnerability (which is only one example of the multiple vulnerabilities in an industrial system) may compromise a **Security Property**, e.g., the integrity of operational data, which can no longer be trusted to be accurate, complete, or unaltered. Such property is analyzed with a **TARA - Threat Analysis and Safety Assessment**. In particular, we use the STRIDE security threat model [31], which includes threat categories such as spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. One of the **Threats** identified tampering, where an attacker can alter sensor data to misrepresent the position of equipment. We determine a **Security Requirement** to prevent such a threat, i.e., use secure communication protocols with cryptographic integrity checks.

D. Safety and Security Mitigation

In Sections V-B and V-D, SoS-level requirements for safety and security have been identified. Each requirement is then fulfilled by a specific **Countermeasure**, which conforms to a **Risk Reduction Level**, commonly determined by using a combination of **Risk Factors** provided by an industry/concern-specific standard (see the bottom-right side in Fig. 2). In our example, we used IEC 61508 [20] for safety and IEC 62443 [21] for security. In safety, the risk factors are the consequence of the severity (C4, which means that a failure has a critical impact on safety), frequency of exposure (F2, which means that there is frequent to continuous exposure), possibility of avoidance (P2, which means that the probability of avoiding or limiting the harm is very low), and demand rate (W2, which indicates that the hazardous event is likely to occur frequently). This combination of parameters determines the safety integrity level (SIL) 4, the risk reduction level

TABLE I
SSO-SoS VERIFICATION BASED ON COMPETENCY QUESTIONS

CQ	Concepts and relations
CQ1	An SoS is composed of multiple *Elements*, including two or more *CS* and *Mediators*, each providing specific capabilities that together form the overall SoS capability. Through the *Configuration* of SoS capability, SoS forms a *Constellation*
CQ2	*Common Mission* is a type of *Mission* that defines objectives and goals intended to be considered by Configuration. Individual Mission of CS along with capabilities of other *Elements* help to achieve *Common Mission*. The mission is composed of a *Mission Thread* that summarizes the overall purpose and flow of activities required to complete a *Mission*.
CQ3	A *Constituent System* is an an independent system which has its own development, management goals and resources, but also interacts with others constituent systems within an SoS to provide unique capabilities.
CQ4	*SoS Capability* is the ability of the entire SoS that demonstrates a unique behavior(s) through the capability configurations of its associated CS
CQ5	*Configuration* is a composite structure that represents the physical and human resources (and their interactions) in the system, assembled to provide an SoS capability.
CQ6	Intended Functionality is a specified functionality that is designed to achieve a specific mission
CQ7	*Mediator* is one of the elements in SoS that facilitate communication and collaboration among CS
CQ8	*Security Property* is the SoS attribute that ensures protection against threats. It can be analyzed by *TARA*
CQ9	*Emergent Hazard* refers to a potential risk or danger that arises unpredictably from the interactions, dependencies, and behaviors among multiple interconnected CS within the SoS. It can be identified through *HARA* process.
CQ10	*Threat* is a circumstance or event having the potential to significantly impact systems's operations. The threats are identified through the TARA process.
CQ11	*Vulnerability* refers to weakness or control function when it is exploited, it compromises the security properties and may lead to *Damage Scenario*.
CQ12	From a safety point of view, the risk factor is the consequence of the severity, frequency of exposure, possibility of avoidance, and demand rate. This combination of parameters determines the safety risk reduction level required for the safety countermeasure. Regarding security, the risk factor is the potential impact (high) and likelihood, which determines a security risk reduction level, which is needed for the security countermeasure. Standards guide how Risk factors and risk reduction levels can be determined
CQ13	*SoS-Level Requirement* is a detailed specification that elaborate what an entire SoS must meet in terms of both functional and non-functional aspects. These requirements must consider the SoS capabilities, interactions, and emergent behaviors that arise when CS interact with each other. Countermeasures are defined to mitigate the identified emergent hazards and threats. These countermeasures are implemented in SoS-Level requirements

required for the safety countermeasure. Regarding security, the risk factors are potential impact (high) and likelihood (possible), which together determine a security level (SL) 3, which is required for the security countermeasure. Both countermeasures, i.e., safety and security countermeasures, influence the initial configuration created for the SoS.

VI. Discussion

Our proposed ontology (see Section III) is an attempt towards defining a unified model capable of characterizing safety and security concerns in SoS. It provides an organization that facilitates navigability through different layers of information, In addition, it enhances stakeholders' ability to identify SoS-related knowledge in terms of its configuration, safety, and security concerns, as well as mitigation strategies for those concerns (see Fig. 2). Still, it can be incremented with more refined aspects or concepts related to a variety of concerns, e.g., perspectives on productivity, which can have an impact on the SoS configuration and, therefore, safety and security. With that in mind, we can consider SSO-SoS as a starting point in the provision of tools for knowledge management where concern-specific views can be configured and analyzed.

Eliciting requirements for SoS is a complex process due to the heterogeneous and dynamic nature of the constituent systems. An ontology can significantly enhance this process, allowing SoS integrators to identify key concepts. For example, as presented in our case study (see section V), we departed from an initial requirement expressed by a project owner responsible for overseeing the project's direction. The requirement says: *"We must perform mass removal in the construction site by only using machines powered by electricity."* From this requirement and the set of concepts and relationships provided by the ontology, we could elicit the information required to create an initial SoS configuration to resolve the expressed requirement. As a result, we provided the information to determine the SoS synergy collaboration, represented as the intended functionality (i.e., the higher-order functionality that individual systems cannot accomplish on their own). There were some ontology concepts not used in this case study, e.g., individual mission (which can contribute to the common mission) as well as CS vulnerability (which can be caused by an element and also compromise the security properties of the SoS). However, such concepts are useful where the analysis starts from the CS perspective. In that sense, the ontology can also be used as a requirements elicitation tool to facilitate the concept analysis required in the SoS lifecycle.

The ontology is also a tool to facilitate safety and security risk analysis. In particular, it provides a set of conceptual elements that serve to identify standardized methods for risk analysis and assessment (i.e., HARA for safety and TARA for security), which can lead to the identification of potential emergent hazards with the former as well as the threats on the security properties affected by vulnerabilities across the SoS with the latter. This standardization is crucial for maintaining uniform practices and supporting, at the same time, the informed decision-making process required in the configuration of SoS. In particular, stakeholders can use the ontology to evaluate the trade-offs between safety and security

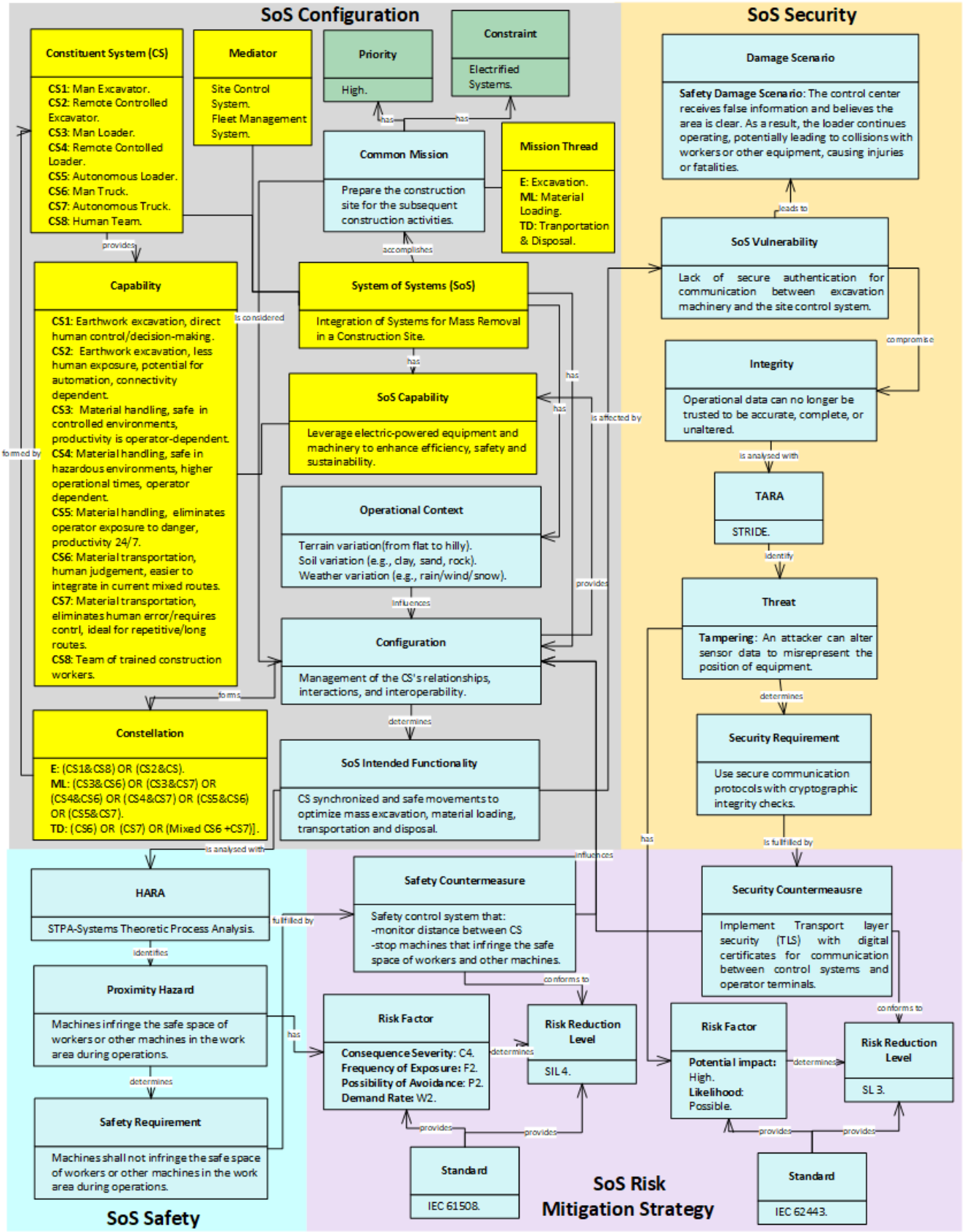

Fig. 2. SoS Integration of Electrified Systems for Mass Removal in a Construction Site

countermeasures, prioritize actions, and allocate resources more effectively. The ontology also incorporates some concepts related to industry standards, i.e., risk factor and risk reduction level. This characteristic of the mitigation strategy facilitates the SoS integrator in thinking about compliance with relevant industry-specific regulations, reducing legal risks, and enhancing stakeholder confidence in the SoS.

Finally, our ontological approach also has other capabilities that can be considered advantageous in the SoS description. For example, ontologies are designed to be scalable, allowing to accommodate the growth and evolution of the SoS. As new systems are integrated or existing systems are updated, the ontology can be expanded and modified to reflect these changes, ensuring that it remains relevant and effective. Such a dynamic nature of SoS requires adaptive safety and security strategies. The ontology supports real-time updates and adaptations, enabling the SoS to respond quickly to emerging safety and security risks and changes in the operational environment.

VII. Conclusions and Future Work

This paper presents a conceptual model for an integrated safety-security ontology for SoS, called SSO-SoS. This model provides a structure for representing and managing the knowledge complexity inherent to the SoS context. It provides a hierarchical organization that permits stakeholders to navigate different levels of abstraction, elicit requirements for SoS, and facilitate safety and security analysis and mitigations.

Future work includes further validation of our proposed conceptual model by considering more case studies in different domains. We also plan to evaluate the usability and effectiveness of the identified concepts and relationships by gathering experts' opinions via surveys and interviews. In addition, we plan to define and formalize rules and constraints that ensure the relationships and concepts are logically consistent when used. Finally, tool support for utilizing the conceptual model while providing essential customized views is also considered.

References

[1] J. Axelsson, "A systematic mapping of the research literature on system-of-systems engineering," in *2015 10th System of Systems Engineering Conference (SoSE)*. IEEE, 2015, pp. 18–23.

[2] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering: The Journal of the International Council on Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[3] *Systems and software engineering — System of systems (SoS) considerations in life cycle stages of a system*, https://www.iso.org/standard/71955.html, Std., [Online; Accessed: 2024-07-5].

[4] C. Harvey and N. A. Stanton, "Safety in system-of-systems: Ten key challenges," *Safety science*, vol. 70, pp. 358–366, 2014.

[5] N. Guarino, D. Oberle, and S. Staab, "What is an ontology?" *Handbook on ontologies*, pp. 1–17, 2009.

[6] R. de Almeida Falbo, "Sabio: Systematic approach for building ontologies." *Onto. Com/odise@ Fois*, vol. 1301, 2014.

[7] J. Martin, J. Axelsson, J. Carlson, and J. Suryadevara, "Towards a core ontology for missions and capabilities in systems of systems," in *18th Annual System of Systems Engineering Conference (SoSe)*. IEEE, 2023.

[8] E. Silva, T. Batista, and F. Oquendo, "A mission-oriented approach for designing system-of-systems," in *2015 10th system of systems engineering conference (SoSE)*. IEEE, 2015, pp. 346–351.

[9] G. Guizzardi, "Ontological foundations for structural conceptual models," *PhD Thesis, University of Twente*, 2005.

[10] J. S. Dahmann and K. J. Baldwin, "Understanding the current state of us defense systems of systems and the implications for systems engineering," in *2008 2nd Annual IEEE Systems Conference*. IEEE, 2008, pp. 1–7.

[11] J. Zhou, K. Hänninen, K. Lundqvist, and L. Provenzano, "An ontological approach to hazard identification for safety-critical systems," in *2017 Second International Conference on Reliability Systems Engineering (ICRSE)*. IEEE, 2017, pp. 1–7.

[12] M. Adach, N. Ali, K. Hänninen, and K. Lundqvist, "Hazard analysis on a system of systems using the hazard ontology," in *2023 18th Annual System of Systems Engineering Conference (SoSe)*. IEEE, 2023, pp. 1–6.

[13] N. Ali, K. Lundqvist, and K. Hänninen, "Mitigation ontology for analysis of safety-critical systems," in *the 34-th European Safety and Reliability Conference (ESREL)*. Polish Safety and Reliability Association, 2024, pp. 9–17.

[14] M. Jamshidi, "System of systems engineering - new challenges for the 21st century," *IEEE Aerospace and Electronic Systems Magazine*, vol. 23, no. 5, pp. 4–19, 2008.

[15] N. Ali and J.-E. Hong, "Failure detection and prevention for cyber-physical systems using ontology-based knowledge base," *Computers*, vol. 7, no. 4, p. 68, 2018.

[16] N. Ali, M. Hussain, and J.-E. Hong, "Safesocps: a composite safety analysis approach for system of cyber-physical systems," *Sensors*, vol. 22, no. 12, p. 4474, 2022.

[17] P. Bhosale, W. Kastner, and T. Sauter, "Integrated safety-security risk assessment for industrial control system: An ontology-based approach," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2023, pp. 1–8.

[18] F. H. C. Ferreira, E. Y. Nakagawa, and R. P. dos Santos, "Towards an understanding of reliability of software-intensive systems-of-systems," *Information and Software Technology*, vol. 158, p. 107186, 2023.

[19] ISO, "Systems and software engineering — taxonomy of systems of systems," https://www.iso.org/standard/71957.html, 2019, [Online; accessed 20-June-2024].

[20] *IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems*, https://webstore.iec.ch/publication/5520, Std., [Accessed: 2024-06-24].

[21] *IEC 62443: Security for Industrial Automation and Control Systems*, https://webstore.iec.ch/publication/30727, Std., online; Accessed: 2024-06-24].

[22] *Road vehicles — Cybersecurity engineering*, https://www.iso.org/standard/70918.html, Std., [Online; Accessed:2024-07-5].

[23] R. Giachetti, S. Wangert, and R. Eldred, "Interoperability analysis method for mission-oriented system of systems engineering," in *IEEE International Systems Conference (SysCon)*. IEEE, 2019, pp. 1–6.

[24] I. Cherfa, N. Belloir, S. Sadou, R. Fleurquin, and D. Bennouar, "Systems of systems: From mission definition to architecture description," *Systems Engineering*, vol. 22, no. 6, pp. 437–454, 2019.

[25] OMG. (2022) Unified Architecture Framework Modeling Language (UAFML) . [Accessed: 2024-07-3]. [Online]. Available: https://www.omg.org/spec/UAF/1.2

[26] J. Dahmann, J. Lane, G. Rebovich, and K. Baldwin, "A model of systems engineering in a system of systems context," in *Proceedings of the Sixth Conference on Systems Engineering Research*, 2008.

[27] G. A. Lewis, E. Morris, P. Place, S. Simanta, and D. B. Smith, "Requirements engineering for systems of systems," in *2009 3rd annual IEEE systems conference*. IEEE, 2009, pp. 247–252.

[28] F. Oquendo, "Formally describing the software architecture of systems-of-systems with sosadl," in *2016 11th system of systems engineering conference (SoSE)*. IEEE, 2016, pp. 1–6.

[29] P. J. Redmond, "A system of systems interface hazard analysis technique," Ph.D. dissertation, Monterey, California. Naval Postgraduate School, 2007.

[30] N. G. Leveson and J. P. Thomas, *STPA Handbook*, 2018.

[31] Microsoft. (2022) STRIDE Model. [Accessed: 2024-06-24]. [Online]. Available: https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats#stride-model