Advances in Reliability, Safety Advances in Reliability, Safety and Security, Part 2 – Kołowrocki, Magryta-Mut (eds.)
© 2024 Polish Safety and Reliability Association, Gdynia, ISBN 978-83-68136-14-2 (printed), ISBN 978-83-68136-01-2 (electronic)

Advances in Reliability, Safety and Security

ESREL 2024
Monograph Book Series

Mitigation Ontology For Analysis of Safety-Critical Systems

Nazakat Ali, Kristina Lundqvist, Kaj Hänninen

School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden

Abstract

This study introduces a Mitigation Ontology (MO) designed for the analysis of safety-critical systems. Recognizing the paramount importance of systematically addressing potential risks and hazards in complex systems, the proposed ontology serves as a structured framework for comprehensively modeling and analyzing mitigation strategies. Leveraging ontological principles, the framework enables a precise representation of safety-critical information, emphasizing the relationships and dependencies among various mitigation elements. To encapsulate the essence of safety-critical systems and support understanding of the mechanisms of situations, events, and associated hazards, we propose a hazard and mitigation domain ontology, i.e., the MO to provide a combined ontological interpretation of hazard and mitigation strategies. The MO facilitates a more thorough and standardized analysis of safety measures, contributing to enhanced understanding, communication, and implementation of mitigation strategies in software and hardware levels of safety-critical systems. The MO is grounded on Unified Foundational Ontology (UFO) and based on widely accepted standards, and scientific guides. We demonstrate our proposed ontology in the autonomous vehicle domain to check how it can help to analyze the safety of real-world safety-critical systems. Through the ontology instantiation process for a case study from the autonomous vehicle domain, we have verified that safety-critical related hazards, causes and consequences, and other entities contributing to hazards were well identified. we have seen that the MO offers a shared vocabulary that facilitates communication among diverse communities, preventing misunderstandings among engineers and stakeholders involved in safety-critical systems. Additionally, the conceptual model serves as a reference point for developers of safety-critical systems, enabling them to systematically extract and analyze safety requirements specifications and provide safety mechanisms.

Keywords: safety-critical systems, hazard ontology, mitigation ontology, iso 26262

1. Introduction

Safety-critical systems, including e.g., those found in aviation, automotive, healthcare, railways and smart industrial domains, play a vital role in our modern society (Knight, 2002). These systems are designed to perform essential functions with the utmost reliability, ensuring that potential failures do not lead to catastrophic consequences. Achieving the required level of safety in such systems demands a rigorous and comprehensive safety analysis process. Safety analysis encompasses the identification, assessment, and mitigation of potential hazards and risks throughout the system's lifecycle (Adach et al., 2023).

The constant evolution of technology and the increasing complexity of safety-critical systems pose significant challenges to safety engineers, making it imperative to adopt new approaches that enhance the safety analysis process (Knight, 2002). One such approach is the utilization of ontologies, which have gained prominence as powerful knowledge representation tools in various domains. In the context of safety-critical systems, a dedicated ontology can serve as a structured and formalized knowledge base, aiding in the identification and mitigation of potential hazards, thus contributing to the overall safety assurance process.

This paper introduces the concept of a Mitigation Ontology (MO) that is designed to facilitate safety analysis in safety-critical systems. The central aim of MO is to enhance safety engineers' ability to identify potential hazards, assess their associated risks, and prescribe effective mitigation strategies. The mitigation strategies may include elicitation, identification or modification of existing safety requirements in the system. The development and application of this ontology represent a significant advancement in the field of safety-critical systems, as it integrates the rich domain knowledge required for effective safety analysis and leverages modern ontology engineering techniques to create a comprehensive and versatile safety assurance framework.

In the sections that follow, we will explore the key elements of this MO and its role in augmenting safety analysis processes within safety-critical systems. We will discuss the MO´s design principles, its structure, and how it integrates with existing hazard ontologies. Additionally, we will delve into practical applications of the ontology, including case studies demonstrating its effectiveness in real-world safety-critical systems.

By addressing the critical challenges faced in safety analysis, this paper contributes to the ongoing efforts to enhance the safety and reliability of safety-critical systems. The MO, as presented here, offers a promising avenue for achieving a deeper understanding of potential hazards, evaluating associated risks, and prescribing robust mitigation strategies, ultimately reinforcing the foundation of safety in safety-critical systems. The MO is a reference ontology related to the safety-critical systems in the vehicle domain. MO is an extension of our previous work (Zhou et al., 2017b) where we presented a hazard ontology that captures entities, that form a causal chain from initiating events to a hazard/accident. In MO, we further enriched it by adding other concepts related to fault, failure, error, and their activation mechanisms. Along with this, MO also captures safety-critical concepts from a functional safety analysis perspective, which shows how safety goals are detailed into technical requirements for each identified hazard. It also shows how safety mechanisms can be implanted for hazards in the implementation details of software and hardware level requirements.

The MO has been developed by following the guidelines given in the Systematic Approach for Building Ontologies (SABiO). In order to get existing concepts, relevant standards (ISO 26262), and other existing literature in the domain of system and safety engineering, were analyzed. In particular, MO will be developed based on the set of some reference ontologies (Zhou et al., 2017b, Liu et al., 2022), by extending their concepts and relations while incorporating new entities. In addition, MO is grounded on the Unified Foundational Ontology (UFO) (Guizzardi, 2005, Guizzardi et al., 2022) to get real-world semantics.

We organize this paper as follows. Section 2 provides a review of background and related work. Section 3 outlines the MO, detailing its development process. In Section 4, we provide the case study to implement our MO followed by the conclusion in Section 5.

1. Background and Related Work

1.1. Ontologies

Ontologies are formalized, structured representations of knowledge within a specific domain that capture concepts, relationships, categories, classifications, and properties in a systematic manner (Breitman et al., 2007, Guarino et al., 2009). Ontology provides a formalized and structured framework to represent knowledge about a system, its components, and interaction-related information. In the case of safety-critical systems, it provides safety-related concepts, relationships, and properties related to safety-critical systems. Traditionally, hazard analysis techniques such as Hazard and Operability Studies (HAZOP), Failure Modes and Effects Analysis (FMEA), Systems-Theoretic Process Analysis (STPA), and Fault Tree Analysis (FTA) have been used for hazard identification, causation analysis, elicitation of safety requirements, fault detection and safety analysis (Carniel et al., 2023, Ali, 2018, Polenghi et al., 2022). These techniques need expert knowledge and structured analysis to identify potential hazards and propose mitigations. Ontological hazard analysis is a technique that draws on ontologies, which are formal representations of knowledge about a specific domain. It involves capturing and representing knowledge about hazards, their causes, and their effects in a structured and formal way. This can help in organizing and reasoning about complex relationships among various elements in a system.

1.2. Ontology for Safety Analysis in Safety-critical Systems

Mokos et al.(Mokos et al., 2010) proposed an ontology-based model-driven engineering process for compositional safety analysis. The proposed approach automatically transforms models used as reusable artifacts and ensures early incorporation of safety assessment into the development cycle. By incorporating a comprehensive domain ontology and employing inference rules, the proposed ontology detects missing elements and identifies semantically inconsistent components within a system model. This not only empowers system designers to uncover potential safety hazards at the design level but also streamlines the model modification process by providing a more efficient means of identifying and rectifying inconsistency errors.

Aziz et al.(Aziz et al., 2019) introduced an ontological approach to model the system and quantify the most likely hazard scenarios associated with various system properties, operational factors, and environmental conditions. The developed ontology-based model is a knowledge-based system that aims to estimate risk in an automated manner for hazard identification. The proposed hazard identification model was validated by comparing it against previous accidents to check its effectiveness.

Provenzano et al.(Provenzano et al., 2017) presented an ontological approach for eliciting safety requirements grounded on the hazard ontology (Zhou et al., 2017b). The proposed approach introduces a heuristic approach called Safety Requirements Elicitation (SARE). SARE used hazard ontology to elicit safety requirements. The authors used a parking brake system to mitigate ‘‘Collision Hazard’’ of a high-speed train to see the applicability of SARE. It enables a more systematic analysis of safety requirements, allowing for a deeper understanding of the relationships between different safety elements. It also supports the integration of safety knowledge across various phases of the system development life cycle, contributing to more effective and robust safety engineering practices.

2. Mitigation Ontology for Safety-Critical Systems

As mentioned earlier, a safety-critical system essentially refers to a system that possesses the safety-critical property. A system is considered to be safety-critical only if it is in a safety-related system context.

MO is an ontological representation of mitigation concepts that are explicitly anchored in a foundation ontology which is also called Unified Foundational Ontology (UFO) (Guizzardi, 2005). Figure 1 shows our proposed MO which is represented using Unified Modelling Language (UML) class diagram and it is developed by following the SABiO methodology (de Almeida Falbo, 2014). The SABiO is an ontology engineering method that integrates different practices of software engineering. SABiO outlines five steps to develop the ontologies (Zhou et al., 2017b, Liu et al., 2022).

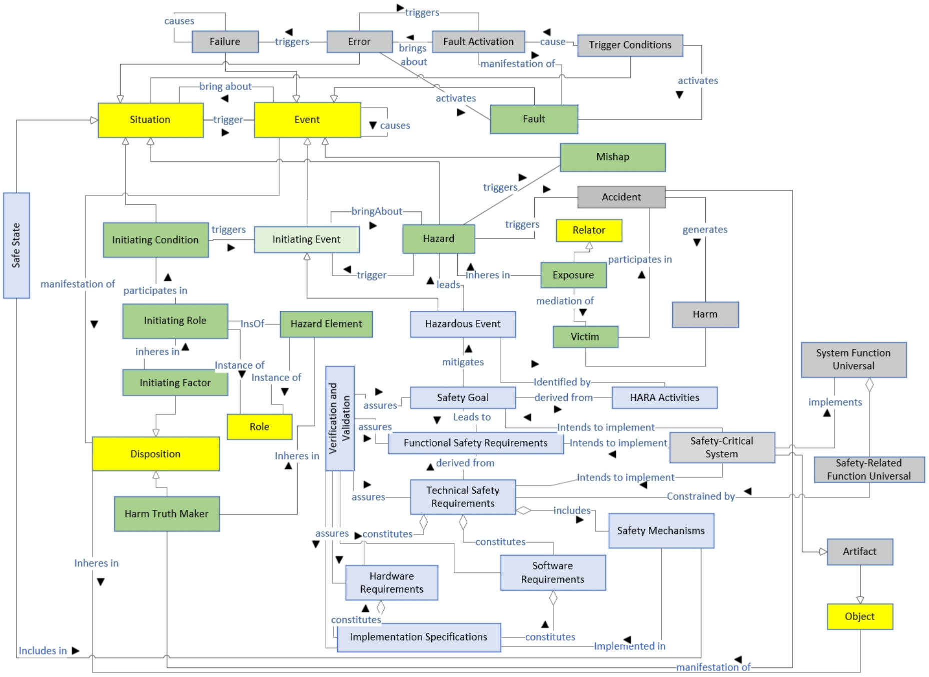


Fig. 1. The UML class diagram for Mitigation Ontology. Concepts are represented as rectangles. The UFO concepts (Guizzardi, 2005) are presented with yellow colored rectangles, hazard-related concepts from Hazard Ontology (Zhou et al., 2017b) are presented in light green color while mitigation-related concepts are presented with light blue colored rectangles. The concepts taken from Software Fault Ontology, and the Software-failure-induced Hazard Ontology (Liu et al., 2022) are represented in grey color. Typed relations are represented “▶” by lines with a reading direction pointed by from the open end to the aggregated end. Subsumption constraints are represented by open-headed arrow lines connecting a sub-concept to its subsuming super-concept.

These are: 1) purpose identification and requirements gathering; 2) ontology formulization; 3) ontology design; 4) implementation; and 5) test. These steps are facilitated by a set of processes i.e., acquiring knowledge, reuse, configuration, documentation, and evaluation. As our objective is to construct a reference ontology, we focus on accomplishing the initial two steps. We initiate the process by engaging with domain experts and examining existing standards, guides, and literature to extract requirements to construct MO. These requirements will be formulated as a set of Competency Questions (CQs)—questions that each ontology should be capable of answering. Following these CQs as a guide, we proceed to identify and structure the concepts, relations, properties, and axioms. In order to elicit the CQs, we have investigated (Bourque et al., 1999), and functional safety standards (2019, 2018).

Below are the listed CQs:

- CQ1: What is a *Hazard* and what kind of events lead to a *Hazard*?
- CQ2: Which kind of *Situations* lead to an *Accident*?
- CQ3: How *Errors*, *Faults*, and *Failures* lead to a *Hazard*?
- CQ4: What are the *Safety Goals* and how they are identified?
- CQ5: What are the *Technical Safety Requirements* and how are they identified?
- CQ6: What are *Safety Mechanism*s and how *Safety Mechanism*s are implemented in *Software* and *Hardware Requirements*?
- CQ7: What is Verification and Validation?

The term **Hazard** has been thoroughly investigated and defined in the literature e.g., (Leveson, 1995, 2018, 2019). Specifically, Zhou et al. (Zhou et al., 2017b) termed **Hazard** as a combination of system and **Situation** that encompasses four distinct entities: (1) **Hazard Element**, exemplified as a role instance (e.g., a vehicle); (2) **Harm TruthMaker**, classified as a subtype of **Disposition** (e.g., the kinetic energy of a vehicle) inheres in **Hazard Elements**; (3) **Victim**, a subtype of **Hazard Element** (e.g., a driver or pedestrian); (4) **Exposure Relator**, representing a relation in which at least one **Victim** is exposed to safety threats presented by **Hazard Elements** (e.g., "a person crossing a road is exposed to the threats posed by an out-of-control vehicle") (Zhou et al., 2017b). When these essential entities exist and lead to a **Hazard**, it can initiate an **Accident** resulting in **Harm** to both individuals and the environment. To induce such a **Hazard**, there must be an **Initiating Event** that is *triggered* by a preceding **Initiating Condition**. This **Initiating Condition** necessarily involves **Initiating Roles** and **Initiating Factors**. **Kind** and **Role** are categories of an object where a kind of an object (e.g., a pedestrian) plays different **Roles** (e.g., a driver). A **Relator** is a relational property that connects multiple objects. **Disposition** indicates a property that can characterize an object. The disposition is an intrinsic moment that can only be manifested by the occurrence of events.

The MO includes a set of foundational concepts incorporated from the UFO (Duarte et al., 2021, Guizzardi et al., 2022, Guizzardi, 2005). The **Event**, **Situation**, **Relator**, **Role**, and **Disposition** are concepts that are inherited from the UFO (Guizzardi et al., 2022). The **Event** is a perdurant type that unfolds in time accumulating temporal parts.

Situations are the endurant that exits in time with all their parts. The parts can be objects (kind/role), relators, and dispositions. For instance, "a vehicle is approaching to the pedestrian who is crossing the road" is a situation where three kind objects (i.e., a vehicle, a pedestrian, a road), two relators (i.e., being-approaching, being-crossing), and two kinetic energy dispositions that characterize a pedestrian and a vehicle, respectively. A **situation** can *trigger* an event and in turn, the event can *bring about* another situation. The event can also cause another event in the system.

The concepts i.e., **Fault**, **Failure**, and **Error** are considered to be the sub-concepts of an **Event** taken from software fault ontology (Liu et al., 2022). This is due to the fact that all these factors can bring about a situation that can trigger an event and an event can bring about a potential hazard. A **Fault** is characterized as a defect that, when triggered or activated, has the potential to lead to a **Failure**. A **Fault** is activated by a **Trigger Condition** and manifested as a **Fault Activation**. A **Fault Activation** further brings about another **Error**.

The **Error** is defined as an incorrect state within an executed program that shows a discrepancy between a computed, observed or measured value/condition and the true, specified, or theoretically correct value/condition (Duarte et al., 2021). **Error**s may lead to **Failures** if they propagate to a critical point in the system where they can cause functional or operational problems and lead to **Hazards**.

Failures are Perdurants (Events). **Failures** occur when a system, component, or device does not perform its intended function. It is the manifestation of a fault in the system. A **Failure** is only triggered when an **Error** is observed to be far beyond a specified threshold. As Events, **Failures** can cause other **Failures** in a chain of **Events** (Duarte et al., 2021). A **Mishap** denotes the unintentional occurrence of an **Event** that can subsequently result in injuries to individuals, harm to the environment, or substantial financial losses. A collision accident (the collision of a vehicle with a pedestrian) is a common example of such a **Mishap**. Two fundamental causal relationships are established between events and situations. In the first scenario, a situation has the capacity to *trigger* an event, and subsequently, the occurrence of that event can bring about another situation. The underlying concept behind these causal relations is twofold: 1) an event transpires as the expression of a set of dispositions within a given situation, and 2) an event has the potential to alter reality by transitioning the state of affairs from one situation to another.

2.1. The Concepts and Relations in the Mitigation Ontology

The MO includes concepts related to the interpretation of hazard (Liu et al., 2022), (Zhou et al., 2017a), the Software Fault Ontology, and the Software-failure-induced Hazard Ontology (Liu et al., 2022) along with mitigation concepts. The mitigation concepts are defined after the detailed investigation of ISO 26262 standard (2018) , IEC 61508 (2019) and standard for system and software engineering (Standardization, 2019).

Hazardous Event is defined as an event that can *lead* to a hazardous situation. It is a sub-concept of initiating an event that causes a hazard that may *trigger* a mishap. It involves identifying potential hazards, evaluating the associated risks, and implementing safety measures to mitigate or control those hazards. Braking system failure during vehicle operation on the road is a typical example of a hazardous event.

Safety Goal are high-level requirements that are intended to be implemented in a **Safety-Critical System**. The safety-critical System (System hereafter) is an **Artifact** with manual, automated, and abstract components, separating it from the concept of machine (Zave and Jackson, 1997). Safety goals lead to Functional Safety Requirements (FSR) that are needed to avoid hazardous events. Safety goals are determined for each hazardous event. They are not articulated in terms of technological solutions, but these are expressed in terms of functional objectives. The safety goals are *derived from* Hazard Analysis ad Risk Assessment Activities (HARA). Safety goals are established to achieve an acceptable level of risk in the operation of a system. For instance, "Ensure that the braking system brings the vehicle to a complete stop within a specified distance when the driver applies the brakes, even in the event of a single failure within the braking system." can be a safety goal for an autonomous vehicle.

HARA involve several activities to identify and evaluate hazards, assess risks, and define safety goals. The HARA estimates the probability of exposure, the controllability, and the severity of the hazardous events. These parameters determine the Automotive Safety Integrity Levels (ASILs) of hazardous events. The ASILs that are determined for each hazardous event are assigned to corresponding safety goals.

Further analysis of safety goals *leads to* **Functional Safety Requirements (FSR)**. They are derived by considering safe states, safety goals, and preliminary architectural assumptions. The FSR for each goal is derived, if applicable, by following operating modes, safe states, fault tolerance, fault-tolerant time interval, and emergency operation intervals.

In the overall system development lifecycle, the **Technical Safety Requirements (TSR)** are required to implement the functional safety requirements. These TSR are *derived from* FSR. The TSR refines the item-level functional safety requirements into the system-level TSR by considering both functional safety concepts and preliminary architecture assumptions. These requirements must *include* **Safety Mechanisms. TSR** also contains **Safety-Related Function Universal** which is included in **System Function Universal**. **System Function Universal** refers to functional requirements that represent the type of functions expected from the **System**.

Hardware Requirements are *derived from* TSR. They are further detailed into implementation specifications considering design constraints, and impact of these constraints on the hardware. It also Specifies ASIL for each hardware component.

Software Requirements are *derived from* TSR and system design specifications. Software safety requirements address software-based functions absence of failure could lead to a violation of technical safety requirements allocated to software. These requirements must consider specification and management of safety requirements, hardware-software interface requirements, communication, operating modes of a vehicle and timing constraints (Tabani et al., 2019). It also specifies ASIL for each software component.

Implementation Specifications include detailed design constraints, implementation details of software and software requirements, and implementation of safety mechanisms.

Verification and Validation represent concepts that *assure* the safety goals and requirements are adequately addressed (Mokos et al., 2010, Honour, 2013). Verification in ISO 26262 (2018) involves confirming that work products and activities in the development process meet the specified requirements for functional safety. The goal is to ensure that the safety-related elements (FSR, TSR) of the system are implemented correctly, and that the development process follows the defined safety plan. Validation is concerned with ensuring that the safety requirements for the entire system are met and that the system behaves correctly in the operational environment. It focuses on confirming that the safety goals are achieved and that the system is fit for its intended purpose.

Safety Mechanisms refer to the measures and features integrated into the design of automotive systems to ensure functional safety. These are implemented based on the FSR derived from HARA and safety goals. Safety mechanisms may include monitoring and diagnostics for fault detection, fault isolation, fault mitigation, fail-safe design, fail-operational design, etc. Safety mechanisms are ***implemented in*** the implementation specifications of software and hardware requirements.

Safe State refers to a condition or mode in which the system is designed to be safe and able to respond appropriately to potential failures or unexpected situations. The identification and implementation of a safe state are key aspects of the design and engineering process for safety-critical systems. It is part of the broader safety architecture that includes risk analysis, fault tolerance, and other safety mechanisms to ensure the system's dependability and resilience in the face of potential failures.

3. Ontology Evaluation

In This section, we evaluate the MO using guidelines from SABiO in terms of its verification and validation.

3.1. Ontology Verification

The main objective of ontology verification is to ensure that terms are defined correctly and consistently in a sense that there is no inconsistency and coherence issue and that produced artifacts adhere to the specifications established earlier (de Almeida Falbo, 2014). In order to achieve this objective, verification can be conducted in a manner driven by competency questions (CQs). In this method, we draw a table that indicates which ontology elements (concepts, relations, and axioms) are able to answer each defined CQ. Table 1 shows the verification of MO.

Table 1. MO Verification and Validation based on Competency Questions.

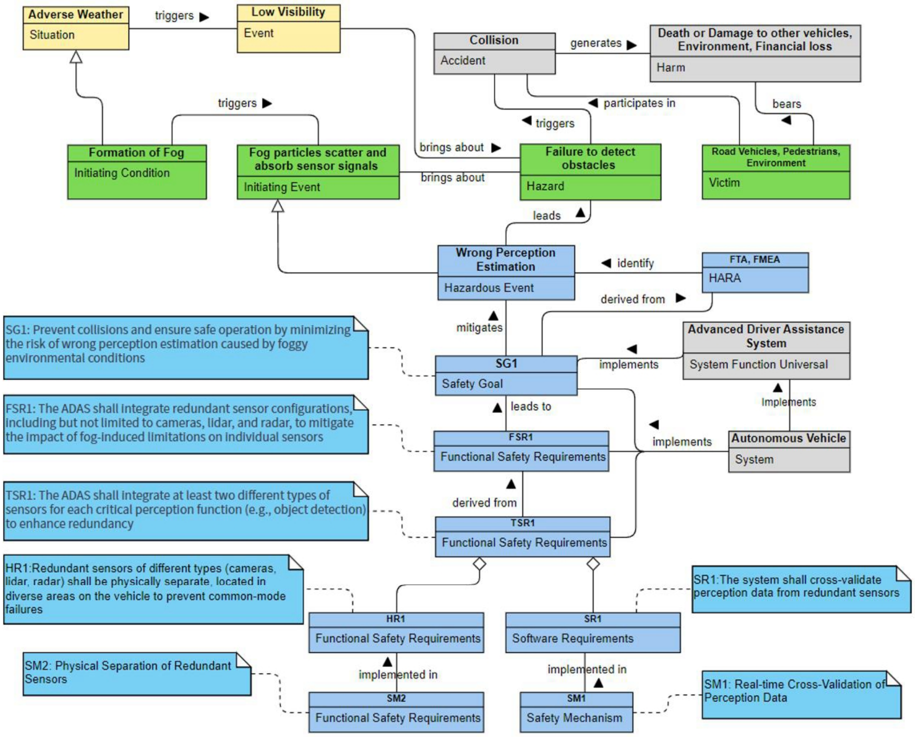
CQs	Concepts and Relationships in MO
CQ1	Hazard as a combination of system and Situation that encompasses four distinct entities: (1) Hazard Element, Harm TruthMaker, Victim, and Exposure. Initiating Event and Hazardous Event which is a type of Initiating Event can trigger Hazards.
CQ2	Accident is the manifestation of Harm TruthMaker. It generates Harm to the Victims. When necessary, entities exist and constitute a Hazard, the Hazard leads to an Accident
CQ3	Errors may lead to Failures if they propagate to a critical point in the system where they can trigger an Initiating Event which is a subtype of Event that can bring about Hazards. Fault is activated by a Triggering Condition, when activated, can trigger and lead to a Hazardous Event which is a subtype Initiating Event which is a subtype of Event that can bring about Hazards.
CQ4	Safety Goals are high-level requirements that are intended to be implemented in a Safety-Critical System. They are derived by conducting HARA.
CQ5	TSR are derived from FSR that refine the item-level TSR into the system-level TSR by considering both functional safety concepts and preliminary architecture assumptions. TSR are intended to address potential hazards and mitigate risks associated with the system.
CQ6	Safety Mechanisms are the measures and features that are integrated into the design of automotive systems to ensure functional safety. Safety mechanisms include monitoring and diagnostics for fault detection, fault isolation, fault mitigation, fail-safe design, fail-operational design, etc. They are implemented in the implementation specifications of software and hardware requirements to achieve a Safe State.
CQ7	Verification ensures that the safety-related elements (FSR, TSR) of the system are implemented correctly, and that the development process follows the defined safety plan. Validation ensures that the safety requirements for the entire system are met and that the system behaves correctly in the operational environment. It focuses on confirming that the safety goals are achieved, and that the system is fit for its intended purpose.

3.2. Ontology Validation

In this section, we will present a case study to ensure that the right MO is developed. Validation aims to demonstrate that MO meets the designated goals: 1) it manages the domain knowledge of safety-critical systems; 2) supports the identification of safety goals, and hazards, derives safety requirements, and designs safety mechanisms for identified hazards. We adopt the SABiO method and take a case study from the autonomous vehicles´ domain to investigate whether the built MO can be applied to represent the chosen case study.

Autonomous vehicles, along with Advanced Driver Assistance Systems (ADAS), offer a transformative approach to transportation, promising benefits such as improved road safety, increased accessibility, and enhanced traffic flow (Hussain and Zeadally, 2018). ADAS plays a pivotal role in providing real-time assistance to drivers and facilitating the transition to full autonomy. These systems, equipped with sensors and learning algorithms, can enhance vehicle control, monitor surroundings, and mitigate potential risks. However, challenges persist, especially collision hazards due to perception problems. Despite advancements, both autonomous vehicles and ADAS may face difficulties in interpreting complex scenarios correctly, posing potential risks on the road. In case of Adverse weather conditions (Situation), e.g., the formation of fog (Initiating Condition) may trigger an event where fog particles scatter and absorb the sensor signals bring about an obstacle detection Hazard. The Hazard can also be triggered due to other factors of adverse weather conditions that can trigger a low visibility (Event) that can also bring about obstacle detection Hazard. When conditions are met and Hazard is triggered, it leads to a collision (Accident) which further generates Harm i.e. death or damage to vehicles on the road. Victims are the ones who can be involved in the Accident and bear the Harm.

As mentioned, adverse weather conditions can affect the perception of the autonomous vehicle (System) leading to wrong perception estimation of Hazardous Event which again leads to the obstacle detection Hazard. In the designing phase of autonomous vehicles (System), safety engineers investigate the potential hazards in detail using some HARA e.g., FTA and FMEA. These hazard analysis techniques uncover the potential Hazardous Events, for instance, wrong perception estimation. After hazard analysis, safety goals are identified for potential hazards. For instance, in the hazard analysis with our MO for autonomous vehicles, we see that (Fig. 2) a safety goal i.e. “SG1: Prevent collisions and ensure safe operation by minimizing the risk of wrong perception estimation caused by foggy environmental conditions” is developed for “Wrong Perception Estimation” event. The SG1 was further broken down into functional safety requirements e.g. “FSR1: The ADAS shall integrate redundant sensor configurations, including but not limited to cameras, lidar, and radar, to mitigate the impact of fog-induced limitations on individual sensors” and technical safety requirements e.g. “TSR1: The ADAS shall integrate at least two different types of sensors for each critical perception function (e.g., object detection) to enhance redundancy”.



Adverse Weather
Situation
triggers ▶
Low Visibility
Event
Collision
Accident
generates ▶
Death or Damage to other vehicles, Environment, Financial loss
Harm
◀ participates in
bears
◀
triggers ▶
◀ triggers
Formation of Fog
Initiating Condition
Fog particles scatter and absorb sensor signals
Initiating Event
brings about ▶
brings about
Failure to detect obstacles
Hazard
Road Vehicles, Pedestrians, Environment
Victim
leads ▲
Wrong Perception Estimation
Hazardous Event
◀ identify
FTA, FMEA
HARA
derived from ▶
SG1: Prevent collisions and ensure safe operation by minimizing the risk of wrong perception estimation caused by foggy environmental conditions
▲ mitigates
SG1
Safety Goal
◀ implements
Advanced Driver Assistance System
System Function Universal
▲ Implements
FSR1: The ADAS shall integrate redundant sensor configurations, including but not limited to cameras, lidar, and radar, to mitigate the impact of fog-induced limitations on individual sensors
▲ leads to
FSR1
Functional Safety Requirements
◀ implements
Autonomous Vehicle
System
TSR1: The ADAS shall integrate at least two different types of sensors for each critical perception function (e.g., object detection) to enhance redundancy
▲ derived from
TSR1
Functional Safety Requirements
HR1:Redundant sensors of different types (cameras, lidar, radar) shall be physically separate, located in diverse areas on the vehicle to prevent common-mode failures
SR1:The system shall cross-validate perception data from redundant sensors
HR1
Functional Safety Requirements
SR1
Software Requirements
▲ implemented in
implemented in ▲
SM2: Physical Separation of Redundant Sensors
SM2
Functional Safety Requirements
SM1
Safety Mechanism
SM1: Real-time Cross-Validation of Perception Data

Fig. 2. The perception failure of an autonomous vehicle as an instance of MO. Each box shows a instance; each solid line an annotation represents a relation. The UFO concepts (Guizzardi, 2005) are presented with yellow colored rectangles, hazard-related concepts from Hazard Ontology (Zhou et al., 2017b) are presented in light green color while mitigation-related concepts are presented with light blue colored rectangles. The concepts taken from Software Fault Ontology, and the Software-failure-induced Hazard Ontology (Liu et al., 2022) are represented in grey color. The dotted lines point explanation corresponding box.

As per ISO 26262 and IEC 61508 standards, the TSR1 is further detailed into software level requirements (SR1: The system shall cross-validate perception data from redundant sensors) and hardware levels requirements (HR1:

Redundant sensors of different types (cameras, lidar, radar) shall be physically separate, located in diverse areas on the vehicle to prevent common-mode failures) as shown in Fig.2.

After detailed requirements implementation at the software and hardware level, safety mechanisms are implemented to make the system safe. In our case study, we suggested implementing “SM1: Real-time Cross-Validation of Perception Data” as a safety mechanism for “SR1: The system shall cross-validate perception data from redundant sensors”. Similarly, we also suggested implementing “SM2: Physical Separation of Redundant Sensors” as a safety mechanism for “HR1: Redundant sensors of different types (cameras, lidar, radar) shall be physically separate, located in diverse areas on the vehicle to prevent common-mode failures”.

As shown in Fig.2, the above case study about autonomous vehicles (System) is represented as an instantiation model of MO. We observe that the represented MO model establishes a causal chain from the root cause (low visibility due to adverse weather and fog formation) to consequences (Hazards/Accidents) by connecting events in the System. The instantiation process offers a more comprehensive interpretation of the hazard experience compared to Hazard Ontology (Zhou et al., 2017b) and SCSO, SFO, and SFIHO ontologies presented in (Liu et al., 2022). This is because MO includes concepts related to mitigation for identified hazards as well. This makes MO a complete ontology for the identification of hazards and their mitigations.

3.3. Discussion

As we see that, ontologies play an important role in analyzing safety-critical systems by providing structured framework for representing, reasoning, and managing safety-related knowledge. Ontologies can be implemented and used in practice in the following three categories.

- **Ontology as domain model in model-driven engineering:**
  *Implementation***:** Ontologies be structured to represent the domain-specific concepts, relationships, and constraints relevant to the engineering domain. This could involve defining classes to represent entities like components, subsystems, interfaces, and relationships to capture dependencies, hierarchies, and system behavior.
  Usage in Practice: Safety engineers and developers can utilize the ontology within model-driven engineering tools to create models of systems, architectures, or processes. These models can be generated, analyzed, and manipulated using the ontology's defined vocabulary and semantics.
- **Incorporated into a larger safety knowledge base:**
  Implementation: The ontology can be expanded to comprise safety-related concepts, including hazards, risks, safety mechanisms, and safety standards. It can provide a structured representation of safety knowledge within the domain, define relationships between safety concepts and enable reasoning about safety implications.
  Usage in Practice: Safety professionals, and stakeholders can utilize the ontology as part of a comprehensive safety knowledge base. They can leverage it to organize safety information, conduct risk assessments, identify hazards, and develop mitigation strategies. For example, ontology could support the retrieval of relevant safety guidelines, facilitate compliance with safety standards, and aid in decision-making processes related to risk mitigation and safety measures.
- **Integrated with other hazard analysis tools:**
  Implementation: The ontology can be designed to align with the data structures and semantics used in existing hazard analysis tools. This might involve defining mappings between ontology concepts and the terminology used in hazard analysis techniques, such as FMEA or HAZOP.
  Usage in Practice: Safety engineers can also integrate the ontology with their preferred hazard analysis tools to enhance their capabilities. It can also support automated reasoning and inference, aiding in the identification of potential hazards, their causes, and associated risks.

4. Conclusion and Future Work

Safety-critical systems are an indispensable part of our modern society. The increasing complexity of these systems, coupled with technological advancements, necessitates continuous innovation in safety analysis processes. We propose MO for enhancing safety analysis in safety-critical systems. The main objective of MO is to empower safety engineers in identifying potential hazards, assessing associated risks, and prescribing effective mitigation strategies. By serving as a structured and formalized knowledge base, MO facilitates a comprehensive safety assurance framework. The ontology not only captures entities in the causal chain leading to hazards but also enriches the understanding of faults, errors, and failure activation mechanisms. Additionally, MO incorporates safety-critical concepts from a functional safety analysis perspective, illustrating how safety goals are refined into technical requirements and how safety mechanisms are integrated into software and hardware implementations. Moreover, the development of MO adheres to the SABiO guidelines, ensuring a rigorous and systematic construction process. Leveraging existing standards such as ISO 26262, relevant literature, and reference ontologies, MO extends and integrates concepts while introducing new entities. We verified and validated MO with the help of a use case in the autonomous vehicle domain. From the results, we have seen that the MO can offer a shared vocabulary that facilitates communication among diverse communities, preventing misunderstandings among engineers and stakeholders involved in safety-critical systems. Additionally, the MO may serve as a reference point for developers of safety-critical systems, enabling them to systematically extract and analyse safety requirements specifications and provide safety mechanisms.

In the future, we want to refine MO to incorporate practical feedback to ensure its alignment with evolving safety requirements in the vehicle domain.

Acknowledgement

This work was supported by the research profile DPAC - Dependable Platform for Autonomous Systems and Control project, funded by the Swedish Knowledge Foundation, grant number: 20150022 and by the Swedish Foundation for Strategic Research through the Serendipity project.

References

2018. ISO 26262-1:2011Road vehicles — Functional safety — Part 1 - 12.

2019. Functional Safety and IEC 61508.

Adach, M., Ali, N., Hänninen, K., Lundqvist, K. Hazard Analysis on a System of Systems using the Hazard Ontology. 2023 18th Annual System of Systems Engineering Conference (SoSe), 2023. IEEE, 1-6.

Ali, N. H., J.-E. 2018. Failure detection and prevention for cyber-physical systems using ontology-based knowledge base. 7.

Aziz, A., Ahmed, S., Khan, F. I. 2019. An ontology-based methodology for hazard identification and causation analysis. *Process Safety and Environmental Protection,* 123**,** 87-98.

Bourque, P., Dupuis, R., Abran, A., Moore, J. W., Tripp, L. 1999. The guide to the software engineering body of knowledge. *IEEE software,* 16**,** 35-44.

Breitman, K. K., Casanova, M. A.,Truszkowski, W. 2007. Ontology in computer science. *Semantic Web: Concepts, Technologies and Applications***,** 17-34.

Carniel, A., Bezerra, J. D. M., Hirata, C. M. 2023. An Ontology-Based Approach to Aid STPA Analysis. *IEEE Access,* 11**,** 12677-12697.

De Almeida Falbo, R. 2014. SABiO: Systematic Approach for Building Ontologies. *Onto. Com/odise@ Fois,* 1301.

Duarte, B. B., De Almeida Falbo, R., Guizzardi, G., Guizzardi, R., Souza, V. E. S. 2021. An ontological analysis of software system anomalies and their associated risks. *Data & Knowledge Engineering,* 134**,** 101892.

Guarino, N., Oberle, D., Staab, S. 2009. What is an ontology? *Handbook on ontologies***,** 1-17.

Guizzardi, G. 2005. *Ontological foundations for structural conceptual models.* PhD.

Guizzardi, G., Botti Benevides, A., Fonseca, C. M., Porello, D., Almeida, J. P. A., Prince Sales, T. 2022. UFO: Unified foundational ontology. *Applied ontology,* 17**,** 167-210.

Honour, E. 2013. Verification and Validation Issues in Systems of Systems. *arXiv preprint arXiv:1311.3626*.

Hussain, R., Zeadally, S. 2018. Autonomous cars: Research results, issues, and future challenges. *IEEE Communications Surveys & Tutorials,* 21**,** 1275-1313.

Knight, J. C. Safety critical systems: challenges and directions. Proceedings of the 24th international conference on software engineering, 2002 Orlando, FL, USA,. 547-550.

Leveson, N. G. 1995. *Safeware: system safety and computers,* ACM.

Liu, H., Jin, Z., Zheng, Z., Huang, C., Zhang, X. An Ontological Analysis of Safety-Critical Software and Its Anomalies. 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS), 2022. IEEE, 311-320.

Mokos, K., Meditskos, G., Katsaros, P., Bassiliades, N., Vasiliades, V. Ontology-based model driven engineering for safety verification. 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 2010. IEEE, 47-54.

Polenghi, A., Roda, I., Macchi, M., Pozzetti, A. 2022. An ontological modelling of multi-attribute criticality analysis to guide Prognostics and Health Management program development. *Autonomous Intelligent Systems,* 2**,** 2.

Provenzano, L., Hänninen, K., Zhou, J., Lundqvist, K. An ontological approach to elicit safety requirements. 2017 24th Asia-Pacific Software Engineering Conference (APSEC), 2017. IEEE, 713-718.

Standardization, I. O. F. 2019. ISO/IEC/IEEE 21841 Systems and software engineering — Taxonomy of systems of systems.

Tabani, H., Kosmidis, L., Abella, J., Cazorla, F. J., Bernat, G. Assessing the adherence of an industrial autonomous driving framework to iso 26262 software guidelines. Proceedings of the 56th Annual Design Automation Conference 2019, 2019. 1-6.

Zave, P., Jackson, M. 1997. Four dark corners of requirements engineering. *ACM transactions on Software Engineering and Methodology (TOSEM),* 6**,** 1-30.

Zhou, J., Hänninen, K., Lundqvist, K., Provenzano, L. An ontological approach to hazard identification for safety-critical systems. 2017 Second International Conference on Reliability Systems Engineering (ICRSE), 2017a. IEEE, 1-7.

Zhou, J., Hänninen, K., Lundqvist, K., Provenzano, L. An ontological interpretation of the hazard concept for safety-critical systems. The 27th European Safety and Reliability Conference ESREL'17, 18-22 Jun 2017, Portoroz, Slovenia, 2017b. 183-185.