Artificial Intelligence for Engineering Design, Analysis and Manufacturing

www.cambridge.org/aie

An ontological framework for the integration of system design and FMEA

Haytham Younus[1], Felician Campean[1,2], Sohag Kabir[1], Pascal Bonnaud[3], David Delaux[3] and Unal Yildirim[1]

[1]School of Computing and Engineering, University of Bradford, Bradford, UK; [2]SAFI Verse Limited, Bradford, UK and [3]Valeo SE, France

Research Article

Cite this article: Younus H, Campean F, Kabir S, Bonnaud P, Delaux D and Yildirim U (2025). An ontological framework for the integration of system design and FMEA. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **39**, e31, 1–21
https://doi.org/10.1017/S0890060425100206

Received: 06 March 2025
Revised: 08 August 2025
Accepted: 21 October 2025

Keywords:
FMEA; ontology-based FMEA; system model; knowledge representation and management; FBS ontology

Corresponding author:
Haytham Younus;
Email: hiamoham@bradford.ac.uk

Abstract

This article proposes the Function–Behavior–Structure–Failure Modes (FBSFMs), a novel ontological framework for an enhanced representation of system knowledge, to address the integration gap between the system models and design risk analysis activities during the early product development phase. As a theoretical contribution, the FBSFM extends the well-established function–behavior–structure ontology for system design information representation in terms of functions, intended behaviors, and structure, with an ontology schema for the representation of the actual behavior as function failure modes, enriched with linkages to causes and effects across multiple levels of system abstraction. This integrated representation improves design risk analysis by facilitating the traceability between design decisions captured in system models and potential failure scenarios documented in Failure Mode and Effects Analyses (FMEAs). The framework was implemented using formal ontology engineering methods and implemented in Web Ontology Language using Protégé. A real-world automotive case study was conducted in collaboration with practicing engineers and domain experts from a global automotive manufacturer, to demonstrate the framework's applicability and its ability to support structured failure knowledge representation. The case study illustrates the capability of the ontology to consolidate multisource engineering knowledge, specifically design data derived from system modeling and structured risk artifacts from FMEA, into a coherent, machine-readable repository, supporting enhanced traceability from user goals to potential system failures. The use of ontological reasoning and structured querying facilitates the systematic review and validation of FMEA information against system models, with a positive impact on product development practice.

Introduction

The automotive industry is undergoing a transformative shift toward the development of increasingly complex, interconnected systems (Gavanas, 2019). Modern vehicles are no longer purely mechanical artifacts; they have evolved into intricate networks of advanced electronic control units, sensors, software, and mechanical components (Milakis et al., 2017). The convergence of cyber and physical elements has led to the emergence of complex systems with enhanced functionality and automation, offering significant benefits in safety, environmental performance, and user experience. However, these developments also introduce substantial challenges in risk identification, system validation, and failure management during the early stages of product development (Nikitas et al., 2021).

Parallel to these technical advancements, there is an increasing recognition of the vital role of knowledge management (KM) in navigating the complexity of modern engineering systems. Exploring the intersection between KM and Industry 4.0 offers opportunities for enhancing digital transformation efforts (Cabeza-Pullés et al., 2020). Yet, current research on KM practices within Industry 4.0 lacks a coherent, structured approach for analyzing trends, standardizing knowledge sharing, and integrating engineering data across domains (De Bem Machado et al., 2022).

Against this backdrop, Failure Mode and Effects Analysis (FMEA) remains a key methodology for identifying and mitigating potential risks within complex systems (Liu, 2019; Belu et al., 2022). FMEA systematically analyses system structures, functions, failure modes, and their effects on performance (AIAG and VDA, 2019). However, current FMEA practices face significant limitations: they are labor-intensive, heavily reliant on subjective brainstorming, and produce documentation that is often difficult to maintain, reuse, or systematically integrate into engineering workflows (Anugerah et al., 2022; Korsunovs et al., 2022; Hezla et al., 2023). Furthermore, inconsistencies in terminology, fragmented knowledge representation, and poor linkage between FMEA documentation and system models hinder its broader reuse and integration (Hodkiewicz et al., 2021; Razouk et al., 2023).

Efforts to address these challenges have introduced Information Technology (IT)-based solutions and model-driven approaches, yet many fall short in capturing system behavior–structure

© The Author(s), 2025. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

relationships or enabling cross-domain knowledge reuse (Rehman and Kifor, 2016). Particularly, the integration of FMEA within systems engineering practices remains hampered by inconsistencies between the tools and languages employed in both domains (Schindel, 2024). Additionally, the unstructured and fragmented nature of engineering data complicates attempts at knowledge integration across models and analysis frameworks (Mbaye et al., 2024; Nakajima, 2024). Despite these persistent challenges, integrating FMEA knowledge with system models remains a particularly promising strategy for enhancing KM during the design phase (Winton and Huang, 2021; Yazdi, 2023; Haytham et al., 2024).

While Model-Based Systems Engineering (MBSE) approaches have introduced methods for automatic FMEA generation (Husung et al., 2021; Lai et al., 2021; Korsunovs et al., 2022) and the embedding of failure analysis (FA) within system models (Godina et al., 2021; Liou et al., 2022; Uludağ et al., 2023), significant gaps remain in achieving a unified, machine-interpretable representation of functional and failure knowledge during early design stages.

To overcome the identified limitations in current FMEA-MBSE integration, this research introduces the novel Function–Behavior–Structure–Failure Mode (FBSFM) Ontological Framework. This framework integrates FMEA knowledge with function–behavior–structure (FBS)-based system models through a unique dual-layered ontological structure. It systematically captures, structures, and interrelates system functions, intended behaviors, actual behaviors (including failures), and structures with corresponding failure modes, causes, and effects by establishing formal semantic linkages. This explicit mapping between the FBS and FMEA layers provides a robust foundation for enhanced KM and interoperability, overcoming the scalability and automation challenges of previous approaches. The FBSFM framework supports both expert-driven analysis and machine-driven semantic reasoning, significantly improving knowledge storage, retrieval, reuse, and ultimately, risk assessment and decision-making throughout the product life cycle.

The proposed framework will be validated through a real-world automotive case study involving a Headlamp system, developed in collaboration with a global automotive manufacturer, to prove the practical application and value of the approach within an industrial setting.

The remainder of this article is organized as follows: section Background and related work establishes the theoretical foundation and context. The next two sections discuss on the methodology and framework and implementation on the case study. Finally, the sections Discussion and Conclusion present the synthesis of the findings and implications of this research.

Background and related work

FMEA integration within systems engineering: A review of methods and challenges

FMEA, one of the most widely recognized methods for reliability and safety, is a proactive tool developed to identify, evaluate, and prevent potential failures in products and processes. It aims to improve product and system design outcomes by enhancing reliability, reducing environmental impact, and achieving cost savings (Wu et al., 2021). FMEA has been widely adopted across industries like aerospace, nuclear, and automotive, following established methodologies (e.g., SAE-J1739, 2009]), with its versatile framework applicable to hardware, software, processes, human actions, and their interactions (IEC-60812, 2018), ultimately evolving into specialized variants, such as design, process, and system FMEA, to address specific process and system aspects (AIAG and VDA, 2019). Historically, FMEA has been approached from two perspectives:

i. Bottom-up approach: This approach, rooted in military standards like IEC-60812 (2018), starts at the component level, identifying failure modes and propagating their effects up the system hierarchy. This reactive approach can lead to significant design changes and increased costs.
ii. Top-down approach: The AIAG and VDA (2019) approach aligns FMEA with product development, adopting a top-down approach. It focuses on identifying potential failure modes at the system level, cascading them down to component levels, and defining countermeasures to mitigate critical risks.

On the other hand, integrating FMEA effectively within the systems engineering process is essential to ensure coherence between design decisions and risk identification, enhancing the reliability and robustness of complex systems, thereby minimizing late-stage errors and reducing overall system development time (Baklouti et al., 2019). Campean et al. (2013) explored the integration of FMEA into system engineering processes and, subsequently, Campean et al. (2014) proposed an integrated framework for aligning FMEA processes with systems engineering models for complex multidisciplinary systems. Their framework leverages a function-driven methodology, providing a coherent flow of information based on the analysis and cascading of functional requirements, failure modes, and design verification activities.

Several earlier efforts have similarly aimed to enhance the FMEA process through integration with system modelling frameworks. Research by Huang et al. (2018, 2017) focused on representing FMEA within MBSE models but faced challenges regarding automation capabilities. Hecht et al. (2014) and Girard et al. (2020) presented MBSE-based FMEA generation approaches. However, their scope was often limited in terms of semantic depth and extensibility, as they do not incorporate a formal ontology or support integration with external knowledge repositories for reuse and reasoning. Ontology-based approaches, such as those presented by Day et al. (2012), sought to formally structure risk information but encountered scalability and deployment challenges. These limitations align with broader findings by Campean et al. (2013, Campean et al. 2014), who noted the ineffective deployment of FMEA within systems engineering contexts across various Original Equipment Manufacturers (OEMs).

In addition, various research efforts have proposed different strategies to support the integration of risk identification and assessment within system design activities.

Table 1 provides a comparative overview of five distinct methodologies, each demonstrating diverse approaches for embedding risk assessment or functional failure reasoning at early stages of the design process.

Sierla et al. (2012) introduced the Functional Failure Identification and Propagation framework, facilitating early hazard identification using minimal design information, although subsequent quantitative analysis using traditional methods such as fault tree analysis (FTA) or probabilistic risk assessment (PRA) remains necessary. Mansoor et al. (2023) proposed a backward failure propagation methodology to strengthen conceptual design robustness by tracing causes of failure. However, its reliance on abstract models may limit fidelity without detailed system representations. Russomanno et al. (1993) embedded functional FMEA reasoning within an expert system (An Expert System for Failure Mode and Effects Analysis (XFMEA)), providing a structured simulation of system failure modes, albeit with ongoing challenges in knowledge formalization and intelligent system

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

Table 1. Comparative review of methods integrating risk analysis within the design process

Reference	Integration focus	Approach	Limitations
Sierla et al. (2012)	Risk identification and early hazard detection using minimal design information within system design processes	Functional failure identification and propagation (FFIP)	Requires follow-up with traditional methods (FTA and PRA) after detailed design for quantitative risk; FFIP alone is qualitative
Mansoor et al. (2023)	Integration of failure cause tracing for early risk identification within the conceptual system design	Backward failure propagation (extending Integrated System Failure Analysis)	Relies on functional, behavioral, and abstract system models; abstractness may limit fidelity without detailed models
Russomanno et al. (1993)	Risk reasoning (FMEA-based) integrated into the system design knowledge base	Functional FMEA integrated into expert system using blackboard architecture	Continuous research needed for fully intelligent systems; knowledge acquisition and formalism critical for operational success
Tumer and Stone (2003)	Design information (functional relationships) is used to avoid failures	Function–failure mode matrix method	Focused mainly on analytical support for design/redesign decisions; may not fully cover causal chain beyond function linkages
Stone et al. (2005)	Risk assessment (FMEA-style analysis during conceptual design) integrated into the design phase	Function–failure design method (FFDM)	Captures functional failures well but not all operational aspects unless expanded (e.g., transport, and maintenance failures)

extension. Tumer and Stone (2003) presented a matrix-based method linking component functionality to failure modes, offering analytical support for design improvements but limited by its focus on functional relationships alone. Finally, Stone et al. (2005) developed the Function–Failure Design Method, enabling FMEA-style analysis during conceptual design stages, although additional operational and maintenance aspects may require separate treatment.

These comparisons highlight that although significant progress has been made in integrating risk assessment analysis into system design, critical challenges persist. In particular, the complexity of modern automotive systems, the integration of diverse data sources, and multidisciplinary coordination continue to pose major hurdles that must be addressed to fully realize reliable, efficient system development.

The primary technical challenges in complex automotive systems include:

- High system complexity: Dealing with a large number of interconnected components and subsystems (Haytham et al., 2024).
- Data integration challenges: Integrating data from various sources (e.g., Computer-Aided Design (CAD), simulation, and testing) into the FMEA process (Filz et al., 2021).
- Multidisciplinary integration: Coordinating efforts across different engineering disciplines (e.g., mechanical, electrical, and software) (Jiménez López et al., 2022).
- Managing interfaces: Effectively managing the interfaces between different subsystems and components (Campean et al., 2014).

Overcoming these challenges requires a multipronged approach, including the development of more robust MBSE frameworks, leveraging Artificial Intelligence / Machine Learning (AI/ML) for automation and knowledge discovery, and improving knowledge sharing and collaboration among engineering teams.

Function modeling and its integration with FMEA

The cascade of functional requirements, from customer to shop floor, underpins systems engineering design by ensuring robust and reliable delivery of customer needs. However, within complex systems-of-systems, integrating FMEA into this functional cascade remains a significant challenge (Campean et al., 2013).

Function reasoning (FR) provides a structured approach to analyzing and representing complex systems. Hamraz et al. (2015) highlighted that various groups have developed FR schemes, typically structured around three domains: function, behavior, and structure (or state). These schemes offer systematic approaches for representing and analyzing system complexity. Extending the verb–noun approach, Tomiyama et al. (2013) introduce the transformation method, which emphasizes functional transformation processes, while Borgo et al. (2009) developed the physical behavior approach, integrating function–behavior–state relationships. Similarly, the FBS approach was introduced by Gero (1990), Gero and Kannengiesser (2007b), and Qian and Gero (1996). The FBS framework provides a comprehensive understanding of the interplay between functions and behaviors, essential for systematic design processes. Central to the FBS ontology is the clear delineation of key concepts: a "system" as an entity interacting with its environment with defined boundaries (Avizienis et al., 2004); "function" as the intended purpose or fundamental role of the system (Gero and Kannengiesser, 2007a); "behavior" as the actions performed to fulfil the system's function; and "structure" as the components and their configuration that enable the system's behavior. Gero's FBS ontology outlines an eight-step design process that begins with translating design requirements into functions, transforming these into expected behaviors, synthesizing solution structures, and iteratively refining the structure, behavior, and functions based on evaluations. This structured methodology ensures that the designed structure effectively implements the desired functions, addressing potential gaps between expected and actual behaviors.

The systematic guidance provided by the FBS framework ensures alignment between functions, behaviors, and structures, facilitating continuous refinement that enhances system performance and reliability. This alignment is particularly relevant when integrating reliability analysis techniques, such as FMEA, into a system model represented as an FBS ontology. Furthermore, FMEA evolves to incorporate more sophisticated knowledge representation methodologies to handle system complexity; the FBS framework benefits from ontological enhancements to address real-world performance challenges. Frameworks such as FBS and the five-key-terms approach (Vermaas, 2013) exhibit limitations, particularly in explicitly addressing the system's actual behavior based on its real-world performance. This gap highlights the necessity of integrating ontological approaches to enrich the FBS framework, thereby complementing methodologies like FMEA that also seek to enhance reliability through structured knowledge representation.

Recent research efforts have explored the integration of the FBS framework into ontological models, each adopting distinct strategies

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

to formalize, structure, and enhance design knowledge representation. Cebrian-Tarrason et al. (2008) introduced OntoFaBeS, an ontology built upon the Behaviour-driven Function-Environment-Structure framework and the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) upper ontology, aiming to formalize product knowledge and infer structures directly from user-defined functional requirements. This work addresses the lack of unified criteria in existing FBS-based methodologies, although it notes that many current engineering ontologies remain predominantly taxonomic. Wang and Wang (2014) developed a product design ontology system by linking geometry application programming interfaces and surface behaviors to FBS elements, thereby strengthening the semantic connection among functions, behaviors, and structures. While this approach improves knowledge sharing and quantization, it faces challenges regarding instance creation, artificial support for function mapping, and the absence of unified standards for functional vocabulary and behavioral classification. Meanwhile, Galle (2009) critically reexamined the conceptual foundations of the FBS model itself, proposing two revised ontological models, the nominalist and the realist versions, focused on improving logical coherence and temporal clarity. Although the work by Galle (2009) does not target immediate practical application, it provides a more reliable conceptual infrastructure for future research and tool development in design theory. Together, these contributions demonstrate the diverse directions taken to advance FBS-based knowledge modeling, each addressing different aspects of conceptual rigor, semantic interoperability, and systematization challenges.

Beyond FBS-based frameworks, recent work has focused more specifically on integrating ontological approaches with function-based perspectives of failure and reliability analysis. Safont-Andreu et al. (2021) proposed a logic-based ontology for the FA domain, aiming to standardize terminology across engineers and software systems to enhance documentation clarity, information retrieval, and workflow verification. This ontology addresses the knowledge-intensiveness and ambiguity challenges inherent in fault analysis by offering a structured semantic framework, with further extensions envisioned for predictive task recommendation, causal relation mining, and ontology alignment with FMEA systems. Meanwhile, Kitamura et al. (2007) developed a middle-level reference ontology for functional knowledge interoperability, based on a generalized input–output model. His work categorizes functions to enable the mapping between different functional models and to facilitate the automatic generation and transformation of FMEA documents. Although both approaches show significant potential in advancing function-oriented reliability analysis, practical implementation challenges remain. The Safont-Andreu et al. (2021) approach requires further development to realize predictive capabilities and ontology alignment, while the Kitamura et al. (2007) model, although conceptually robust, demands careful mapping and categorization to ensure accurate integration across diverse engineering domains. Collectively, these contributions demonstrate the critical role of ontologies in bridging functional modeling and FA to support more consistent, interoperable, and automated system design practices.

Formal ontological approaches to function and failure

Several methodological frameworks have been proposed within the ontology engineering community. Prominent among these are NeOn (Suárez et al., 2011), Methontology (Poveda-Villalón et al., 2022), and DILIGENT (Pinto et al., 2004), each providing tools for ontology development and reuse. The NeOn methodology, in particular, offers guidance for nine distinct development scenarios, including ontology merging, reengineering, and alignment with external vocabularies, each of which is relevant for engineering domains involving both design and risk artefacts. Ontologies significantly enhance engineering and design by providing structured knowledge representations that improve machine comprehension and support complex system design (Ebrahimipour et al., 2010; Mikos et al., 2011; Rehman and Kifor, 2016; Hodkiewicz et al., 2021). In computer science, ontologies serve as formal specifications that define a common understanding of a domain, encompassing concepts, relationships, and attributes (Kumar, 2008; da Silva and Pereira, 2014). Unlike simple taxonomic hierarchies, ontologies offer rich definitions and elucidate intricate inter-concept relationships, enabling more profound and nuanced knowledge representation (Kumar, 2008). Tools like Protégé (Musen, 2015) have revolutionized ontology development by making it accessible and integral to advancements in engineering design. Protégé supports various ontology formats, including the Web Ontology Language (OWL), facilitating interoperability and integration with diverse systems. In engineering contexts, ontologies enable the formalization of design knowledge, supporting tasks such as function modeling, requirement analysis, and system synthesis (Dermeval et al., 2016). This formalization is particularly beneficial for enhancing FMEA processes, as ontological representations can improve the consistency, reusability, and integration of failure mode data within broader design and reliability frameworks.

In parallel, several top-level ontologies offer differing interpretations of "function" and its relation to behavior and failure, contributing valuable insights for structured engineering representations. DOLCE does not formally define function within its core ontology, but it has been extended in related work to represent functions as types of "perdurants," while roles are treated as anti-rigid concepts classified over endurants and perdurants (Borgo et al., 2022). It is therefore important to clarify these ontological categories. *Endurants* are entities that are wholly present at any given point in time, such as a person, a component, or a vehicle, whereas *perdurants*, in contrast, are entities that unfold over time and are only partially present at any moment, such as a manufacturing process, a failure event, or a design meeting. The distinction reflects the philosophical differentiation between static entities and dynamic processes, which is fundamental in modeling behaviors and functions within engineered systems (Borgo et al., 2022). Meanwhile, Basic Formal Ontology (BFO) (Otte et al., 2022) functions are defined as a subclass of dispositions, realizable entities inherent in material objects such as organs or components, which manifest through specific processes like pumping or vision (Spear et al., 2016). In the General Formal Ontology (Herre, 2010), functions are treated as attributives, defined as roles played by entities in the context of achieving a goal. This interpretation frames functions within a teleological structure based on the role–goal relationship (Loebe et al., 2022). More recently, in the MALFunction Ontology (MALFO) (Compagno and Borgo, 2024) designed to address the need to structure maintenance-related knowledge, functions are modeled as DOLCE-aligned concepts classifying certain classes of perdurants, enabling them to exist independently of execution. The ontology distinguishes between systemic and ontological functions based on the abstraction level and further separates engineering methods from proper functions to clarify functional ambiguity. While these ontologies do not always directly formalize failure or dynamic system behaviors, their treatment of functional roles and realizable processes provides foundational scaffolding for modeling causality, state change, and failure propagation within engineered systems.

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

Integrating the FBS ontology with ontological approaches provides a robust framework for engineering and design knowledge representation, enhancing systematic function modeling, supporting comprehensive design processes, and addressing critical limitations related to actual behavior. Ontologies offer the necessary structure to formalize the FBS concept's function, behavior, and structure, enabling sophisticated KM and automated reasoning. By leveraging ontologies, the FBS framework achieves greater precision and adaptability, providing clear guidelines for describing the actual behavior and aligning design processes with broader system goals. Furthermore, ontological approaches mitigate conceptual bypassing by ensuring that all relevant conceptual layers are adequately addressed, thereby enriching function-based reasoning during the design process (Eisenbart and Gericke, 2020). This synergy fosters a more comprehensive and flexible design methodology, supporting iterative refinements and ensuring that systems meet both functional requirements and real-world performance expectations. Additionally, such an integrated approach is essential for advancing FMEA methodologies, where detailed and structured knowledge representation is crucial for identifying and mitigating potential failure modes effectively. This integration not only advances the FBS methodology itself but also complements and enhances reliability analysis techniques, such as FMEA, highlighting the interconnectedness of structured knowledge representation in modern engineering practices.

The Knowledge Representation and Semantic Web communities have collaborated to standardize widely used knowledge representation languages, such as Resource Description Framework (RDF) and OWL (Pan et al., 2023). In this context, researchers argue that augmenting Large Language Models (LLMs) with ontologies offers several advantages (Yang et al., 2023) such as Knowledge Representation and Reasoning, High-Precision Methods, Long-tail Knowledge, and Explainability and Interpretability (Pan et al., 2023). Nevertheless, the development of Ontologies and Knowledge Graphs (KGs) heavily relies on human domain experts to define entities and relationships, establish hierarchies, and maintain domain relevance (Kommineni et al., 2024). A KG is a graph-based data model designed to accumulate and convey real-world knowledge. Its nodes represent entities of interest, while edges represent relationships between these entities. KGs are employed in applications requiring the integration, management, and extraction of value from diverse, large-scale data sources (Hogan et al., 2021).

Building upon these findings, it is evident that while integrating FMEA with MBSE shows significant promise, substantial challenges remain in effectively capturing and utilizing complex system behaviors within FMEA processes. The integration between risk analysis, represented by FMEA, and system design is crucial for ensuring robust product development and system reliability. However, the lack of a structured data representation continues to hinder efficient data sharing, reuse, and interoperability between different system models and analysis frameworks. To address these gaps, this research proposes the development of a novel framework that integrates FMEA within system models through ontological representations. The objective is to enhance the capture and utilization of system elements and behaviors within risk assessments, thereby enabling a more efficient and effective FMEA process, improving overall system reliability, and strengthening decision-making capabilities throughout complex systems engineering projects.

Framework and methodology

Methodology overview

The review of the literature showed that significant challenges persist in the integration of FMEA with MBSE and ontological frameworks. Existing FMEA-MBSE integration efforts lack robust automation and scalability, limiting their practical deployment in complex systems. While frameworks such as the FBS ontology offer structured design approaches, they often fail to explicitly represent real-world system behavior, leading to a disconnect between theoretical modeling and actual system operation. Moreover, data integration challenges and multidisciplinary inconsistencies further complicate the application of FMEA in engineering design. Ontology-based methods, despite offering structured knowledge representation, face scalability issues and are heavily reliant on human expertise for their development and maintenance.

In light of these gaps, this research is driven by the following central question:

How can an ontological framework integrating system design and FMEA support the structured representation and querying of failure knowledge in complex engineering systems?

The work aims to develop a structured, scalable, and integrative framework that bridges the gap between system design models and risk analysis through ontological knowledge representations.

The ontological framework put forward in this work is theoretically grounded in the FBS ontology, which is a well-established conceptual model for representing design knowledge. The proposed framework, henceforth referred to as FBSFM, extends the FBS model to include consideration of failure modes associated with the system functions and behavior. While leveraging the strengths of FBS in capturing the intended functional requirements, expected behaviors, and physical structure of a system, the FBSFM framework explicitly addresses the limitation of conventional FBS models, related to the inadequate capture and representation of the actual behavior of the system, particularly under faulty or failed conditions. Standard FBS primarily focuses on the intended behavior derived from function, often neglecting the actual behavior that manifests during operation, including deviations and failures.

The FBSFM framework extends the FBS model by incorporating a dedicated FMEA ontology layer and establishing explicit semantic links between the intended design representation (FBS layer) and the potential failure characteristics (FMEA layer). This integration introduces the concept of "actual behavior" as a bridge between intended function and observed failure modes. By formally mapping functions to both intended and actual (potentially faulty or failed) behaviors, and subsequently linking these behaviors to specific failure modes, causes, and effects, the FBSFM framework provides a more comprehensive theoretical model. Thus, FBSFM moves beyond representing just the "as-designed" system to encompass the "as-is" or "as-failed" system behavior, thereby offering a richer, more realistic foundation for integrating design synthesis with reliability analysis and risk assessment within a single, coherent ontological structure.

The key objectives of the proposed ontological framework, along with anticipated quantifiable contributions, are as follows:

1. To create a unified knowledge repository: This repository will integrate design data (functions, behaviors, and structures) and risk data (failure modes, causes, and effects) for systems.
2. To facilitate automated knowledge retrieval and reuse: The framework aims to reduce the time required for engineers to retrieve relevant design and risk information.
3. To support early-stage FA by explicitly linking intended functions to potential failure modes via behavior representations.
4. To improve decision-making: The framework will enable semantic analysis and automated reasoning, potentially leading reduction in design changes related to previously unidentified

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

risks and improving the effectiveness of risk mitigation strategies through more informed decision support.

The overall methodology for developing the integrated FBSFM Ontological Framework is supported by the conceptual structure illustrated in Figure 1. This figure presents the key components and their interactions, starting with the independent development of the FBS ontology (representing system design knowledge) and the FMEA ontology (capturing failure knowledge) from the design phase domain. These ontologies are then semantically integrated into a unified FBSFM ontology, which consolidates system functions, behaviors, structures, and failure data within a coherent formal representation. The resulting FBSFM ontology is subsequently deployed into a system repository to enable structured querying, reasoning, and validation. The implementation of this framework involves three main tasks: constructing the individual ontologies, establishing semantic mappings to integrate them, and deploying the combined ontology into a knowledge repository that supports enhanced traceability, knowledge reuse, and decision support.

The development of the FBSFM ontology followed established ontology engineering practices, specifically drawing on the NeOn methodology (Suárez et al., 2011). NeOn defines nine distinct scenarios for building ontologies and ontology networks, accommodating diverse development contexts such as creating new ontologies from scratch, reusing existing ontological or non-ontological resources, and merging or restructuring them. The construction of FBSFM applied a combination of Scenario 4 (reusing and reengineering ontological resources) and Scenario 5 (reusing and merging ontological resources). Scenario 4 was applied to extend the existing FBS ontology by reengineering its behavioral layer to include the concept of actual behavior represented as function failure modes. Scenario 5 was employed to merge the reengineered FBS ontology with a domain-specific FMEA ontology, forming an integrated framework that semantically aligns system design concepts with FA artifacts. Additionally, elements of Scenario 2 (reusing and reengineering non-ontological resources) were involved, particularly in the transformation of FMEA tables and functional requirement documents into structured ontology instances. These scenarios collectively enabled the systematic integration of both structured and semi-structured engineering knowledge into a coherent, formal ontology, ensuring consistency, extensibility, and reuse.

The roadmap emphasizes the structured flow of knowledge from distinct sources into a consolidated framework, culminating in an implemented system ready for advanced KM, cross-domain analysis, and decision support.

Ontological framework development

Figure 2 shows an FMEA extract from the AIAG and VDA (2019) standard, which illustrates a case for a window lifter system. It presents only a single row from the example to demonstrate the Design FMEA steps, beginning with step 2: structural analysis. However, it does not include the initial step of the FMEA process, "planning and preparation," which encompasses critical information regarding the organization, product/system, and the FMEA team.

The AIAG/VDA FMEA procedure establishes a linkage between system models through structure analysis, function analysis, and failure mode analysis as part of risk assessment. In structural analysis, the system is decomposed into three levels: (1) next higher level, (2) focus element, and (3) next lower level. Each level undergoes its corresponding function decomposition in step 3, that is, function analysis.

However, functional analysis remains a recognized weakness in current systems engineering practices, as it often relies heavily on brainstorming techniques (Campean and Henshall, 2012). Once functional requirements are identified, functional failure mode analysis follows to address potential design risks. Additionally, nonfunctional requirements (NFRs) represent the intended behaviors that support the fulfilment of these functional requirements (Eckhardt et al., 2016).

Despite the AIAG/VDA framework's strengths, its integration of behavior analysis with functional and risk analyses is limited. Although it incorporates P-diagrams to characterize system output behaviors (Su et al., 2014), a comprehensive behavioral modeling approach linking design intent to observed failure remains absent.

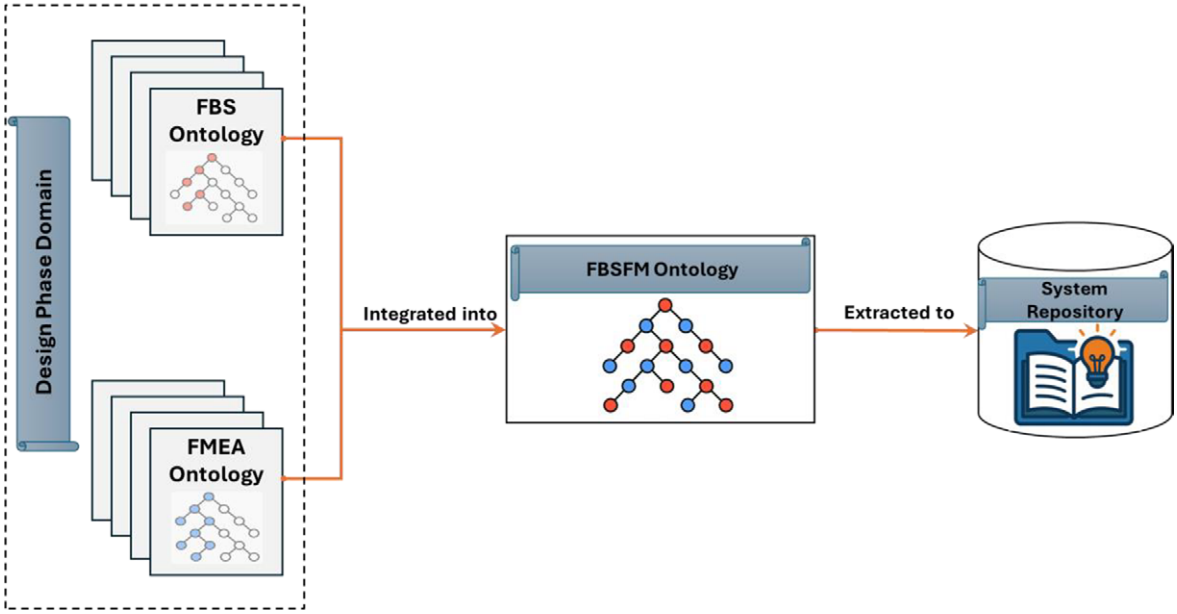


Figure 1. Methodology roadmap for developing and implementing the integrated FBSFM ontological framework.

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

Example of DFMEA with one instance for Window Lifter System from AIAG standard

Structure Analysis (Step 2)			Function Analysis (Step 3)			Failure Analysis (Step 4)			
1.Next Higher Level	2.Focus Element	3. Next Lowel Level or Characteristic Type	1. Next Higher-Level Function and Requirement	2.Focus Element Function and Requirement	3. Next Lowel Level Function and Requirement	1.Failure Effect (FE) to the Next Higher-Level Element and/or End User	Severity (S) of FE	2. Failure Mode (FM) of the Focus Element	3. Failure Cause (FC) of the Next Lower Element or Characteristic
Window Lifter Motor	Commutation System	Bursh Card Base Body	Convert electrical energy into mechanical energy according to parameterization	Commutation system transports the electrical current between coil pairs of the electromagnetic converter	Brush card body transports forces between spring and motor body to hold the brush spring system in X, Y, Z position	Torque and rotating velocity of the window lifter motor too low	6	angle deviation by commutation system intermittently connects the wrong coils	Bursh card body bends in contacts area of the carbon brush

Figure 2. Example of a design FMEA table for a window lifter system (AIAG-VDA, 2019).

Within this framework, failure modes are defined as deviations in system behaviors from intended outputs, while error states categorize these failure modes as unintended behaviors (Goktas et al., 2024).

FA, the fourth step in conducting FMEA, investigates failure modes, their effects, and causes in relation to their corresponding functional and behavioral elements. Thus, the FMEA analytical sequence progresses logically from system structure to functional requirements, to NFRs (intended behaviors), and finally to failure modes (unintended behaviors), as summarized in Table 2.

The FBSFM Ontology is developed by extending the foundational FBS ontology introduced by Gero (1990) and Gero and Kannengiesser (2007a, 2007b), which describes the relationships among system entities during the design phase. In the standard FBS ontology, system behavior is categorized into two types: expected behavior (Be), which represents the intended functionality derived from requirements, and structural behavior (Bs), which describes how the implemented design performs. The design process ideally concludes when structural behavior closely matches expected behavior. However, in current industry practice, as discussed by Campean et al. (2013), Campean et al., 2014), early consideration of design risks is mandated to identify potential behaviors and integration requirements that need to be considered within the conceptual design iteration, to avoid costly downstream engineering change iterations, associated with the post-design risk evaluations. Thus, the FBS extension to explicitly include consideration of failure modes, that is, FBSFM, is coherent with the current design practice.

The proposed FBSFM Ontological Framework comprehensively unifies system design knowledge and risk analysis within a single ontological structure. Unlike the methodologies discussed earlier, which either treat design and risk assessment as separate activities

Table 2. Mapping sequence analysis of system structure, functional requirements, nonfunctional requirements, and failure modes

System	Function requirements	Nonfunction requirements (intended behavior)	Failure modes (unintended behavior)
System structure	FR1	NFR1	FM1,…
System structure	FR1	NFR2	FM21,…

or do not explicitly provide adequate representation of knowledge from both domains, the proposed framework creates formal semantic bridges between them, thereby enabling unprecedented integration across system levels.

The FBSFM framework, shown in Figure 3, represented as a KG, introduces a critical distinction in behavior types after structure determination. Specifically, the established structure "Delivers" both intended behavior (aligned with expected behavior) and unintended behavior (representing potential deviations or failures). The intended behavior "Fulfils" the system's function, which in turn "Achieves" the user's goals, while unintended behavior forms the foundation for systematic risk analysis.

The FBSFM ontology, illustrated in Figure 3, extends the FBS ontology framework by explicitly incorporating unintended behavior alongside intended behavior. This represents a fundamental advancement over existing approaches. The framework establishes formal semantic connections between design elements (functions, behaviors, and structures) and risk elements (failure modes, effects, and causes), creating a goal-oriented traceability from user needs through to potential system failures. By combining the extended

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

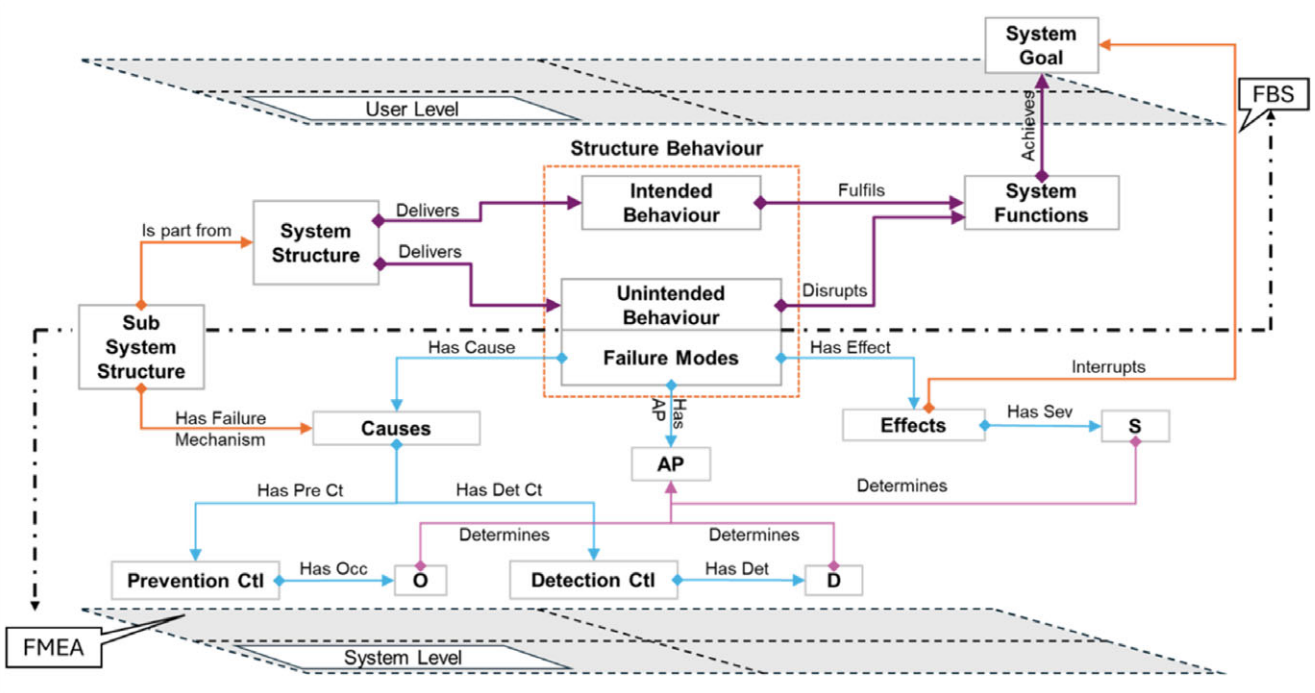


Figure 3. The knowledge graph of the FBSFM ontology for the system level.

system ontology concepts shown in Figure 2 (upper side) with FMEA knowledge shown in Figure 3 (lower side), the framework enables cross-domain knowledge representation and reasoning, overcoming the limitations of conventional, siloed approaches to engineering design and risk management.

Identification of classes and relations within the framework

The proposed FBSFM can be deployed across multiple system levels, as illustrated in Figure 2, which shows the system structure divided into higher, focus, and lower levels. Within this framework, the focus level structure "is a part of" the higher-level structure, and similarly, the focus level function "is a part of" the higher-level function. Regarding failure modes and their consequences across multiple system levels, this can be demonstrated through the cause-and-effect chain shown in Figure 4, introduced in the AIAG/VDA standard. In this chain, the failure cause of a system's top level "is a" failure mode for system's lower level. With this explanation, the FBSFM ontology can be deployed to any number of system-level decompositions.

Central to the **FBSFM** ontology, as shown in Figure 3, is the concept of failure modes as "Unintended Behaviors." These unintended behaviors disrupt the performance in achievement of the system functions, resulting in potential failure modes defined as states in which the system fails to deliver the expected or specified performance. Within this work, reflecting our focus on early system design analysis, and coherent with the AIAG and VDA standard, we are concentrating on function failure modes, which are explicitly conceptualized in relation to the functional specifications. While related, this approach is distinctly different from other prominent recent ontological approaches focused on failure mode knowledge representation, such as Compagno (2025), Compagno and Borgo (2024), and Hodkiewicz et al. (2021), which focused on a bottom-up approach, starting with the failure mechanism leading to component failure modes. While similar in the objective of formalizing knowledge representation for future design tasks, the approaching viewpoints and the requirements of the domain experts are fundamentally different. Specifically, systems engineering designers are expected to provide solution-agnostic system models in the early stages of the design, which need to be evaluated using function failure frameworks.

The proposed ontological framework provides a structured representation of the classes and their relationships within the FMEA and system model for the system's focus elements. These classes and relationships are outlined in Table 3. Specifically, the parameters governing FMEA-related relationships were adopted from the AIAG and VDA (2019) standard, while the parameters defining system design relationships were derived from established FBS ontology frameworks. This dual sourcing ensures that the framework accurately reflects both risk assessment standards and design modeling principles.

Since the FBSFM ontology is built upon the top-level FBS and FMEA ontologies, both of which align with the definition provided by Ye et al. (2011), describing a top-level ontology as one that captures common semantics across dimensions of information, it inherently integrates the semantic relationships and structural compositions of these foundational models. This integration enables FBSFM to offer a unified semantic structure for representing system knowledge.

Implementation

FBSFM ontology development and tools

This section details the development process of the FBSFM ontology, focusing on the methodology employed to integrate the extended FBS concepts with FMEA knowledge, along with the supporting tools used throughout the process.

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

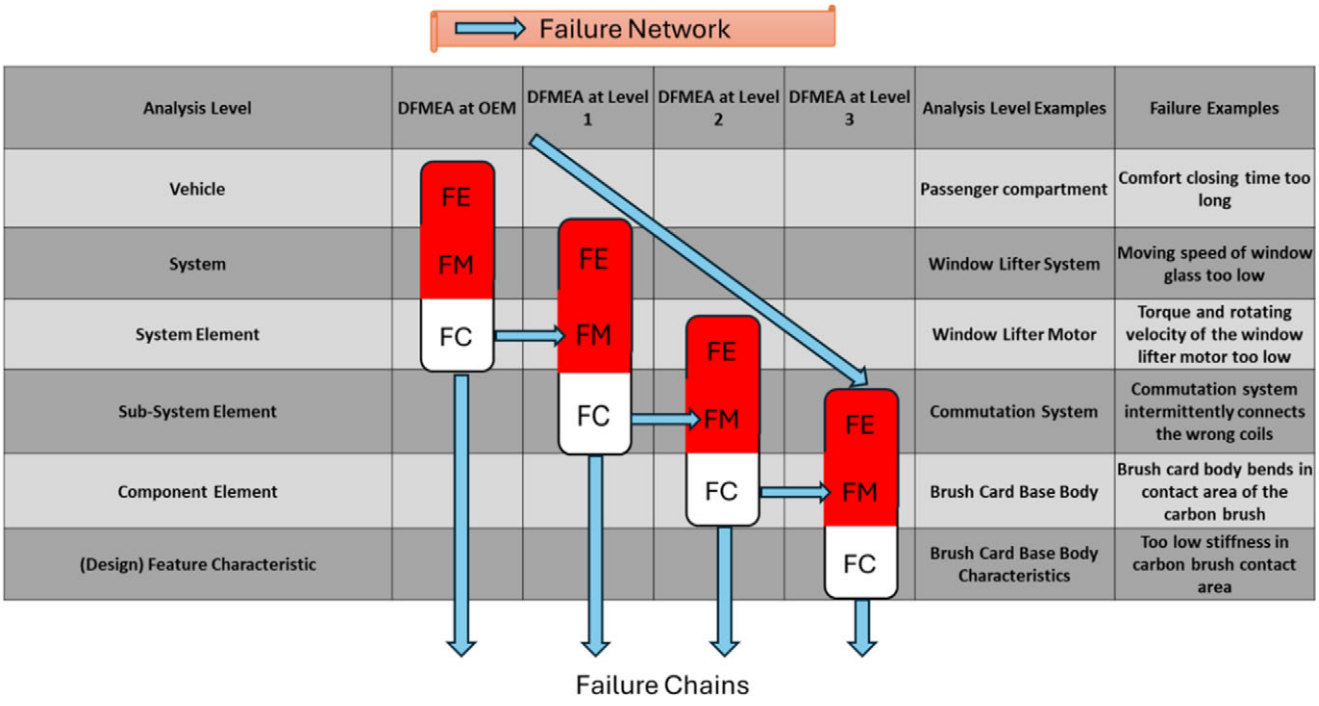


Figure 4. Failure modes and cause relation at different level.

Table 3. Classes and their relationships in FBSFM ontology

Entity (source)	Relationship	Connected entity (target)
System functions	Achieve	System goal
System structures	Deliver	Intended behaviors
		Failure modes
Failure mode	Disrupts function	System functions
	Has cause	Causes
	Has effect	Effects
Cause	Has detection control	Detection controls
	Has prevention control	Prevention controls
	Has AP	Action priority number (AP)
Detection controls	Has detection	Detection
Prevention controls	Has occurrence	Occurrence
Effects	Interrupt	System goal
	Has severity	Severity
Severity, occurrence, and detection	Calculate	Action priority number (AP)
Next lower system structure	Has failure mechanism	Cause

Methodology for FBS-FMEA integration

The construction of the FBSFM ontology addressed the conceptual integration between system design knowledge (FBS) and risk analysis knowledge (FMEA) within a unified ontological structure. A systematic mapping methodology was followed:

- Knowledge structuring: Core classes were developed separately for the system model layer (function, expected behavior, unintended behavior, and structure) and the FMEA layer (failure mode, failure cause, and failure effect). Special emphasis was placed on explicitly representing unintended behaviors (Bu) as a critical link to potential failure modes.
- Cross-domain mapping: Semantic relationships were defined between the two domains. For example:
  - Unintended behavior instances are linked to corresponding failure modes.
  - Structural elements are associated with failure causes that could originate within their physical configuration.

This integration process enabled the capture of both the "as-designed" and "as-fails" perspectives of system behavior, providing traceability from user goals through to potential system failures.

Tools and formalization approach

The ontology was developed using Protégé, a widely adopted open-source ontology editor (Musen, 2015), supporting modeling in the OWL. Protégé facilitated:

- Definition of classes, properties, and axioms reflecting the integrated FBSFM conceptual model.
- Management of complex relationships through the "Classes" and "Object Properties" tabs.
- Logical consistency checking using the HermiT reasoner, which was employed to validate that the ontology structure was free from logical contradictions and semantic errors.

During ontology formalization, Manchester Syntax was preferred for expressing axioms. Its more human-readable format compared to traditional OWL XML-based syntaxes significantly improved validation sessions, particularly during expert reviews and ontology debugging stages. Relationships, such as "Structure delivers Unintended Behavior" and "Unintended Behavior

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

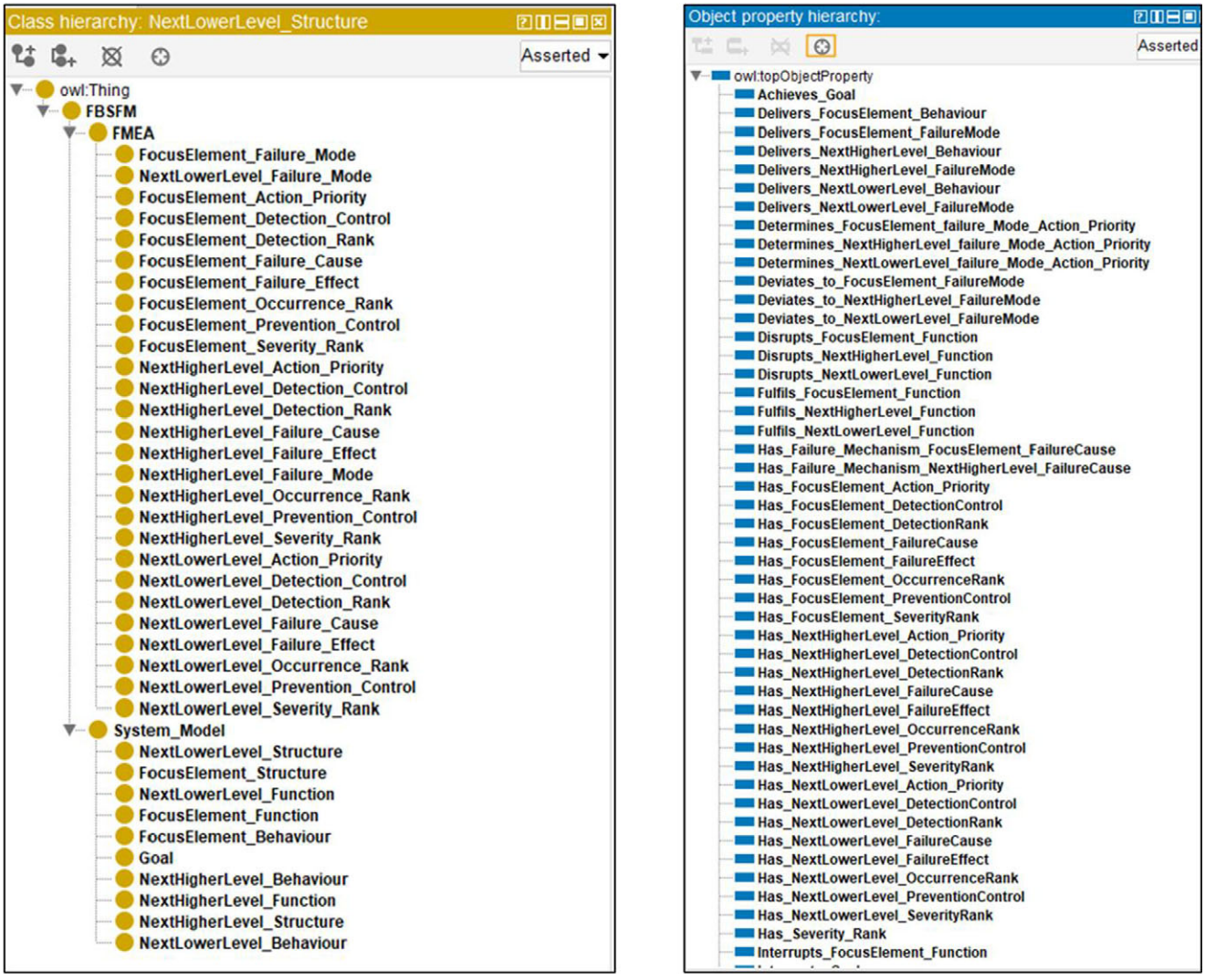


Figure 5. FBSFM ontology representation in Protégé.

manifests Failure Mode," were articulated and verified more easily using Manchester Syntax.

Figure 5 shows a snapshot of the FBSFM ontology structure within Protégé, illustrating the developed hierarchy and semantic linkages between design and risk elements.

To validate the integrity of the proposed ontology and identify any potential implementation pitfalls, we employed the OntOlogy Pitfall Scanner (OOPS) (Poveda-Villalón et al., 2014), which is a commonly used open-source tool to assess the quality of an ontology (Alkhariji et al., 2023). Figure 6 presents the results of the evaluation performed by the OOPS tool (Poveda-Villalón et al., 2014; Alkhariji et al., 2023). As can be seen, the tool reported three minor pitfalls related to annotations, relationship declarations, and naming conventions. While none of the issues were critical, the feedback supported targeted refinements to improve the overall structure and consistency of the ontology.

KG representation

Following formalization, the FBSFM ontology was visualized as a KG using OWLGrEd (Fig. 7), enhancing the understanding of the system design-risk analysis integration. Figure 7 depicts the KG representation, highlighting:

- Key classes (e.g., function, behavior, structure, failure mode, failure cause).
- Semantic interconnections between design knowledge and failure knowledge.

This graphical representation offers an intuitive overview of the integrated knowledge, demonstrating how potential failure mechanisms are traced back to the system's functional and structural elements.

The following section outlines the implementation of the FBSFM ontological framework through a real-world case study from the automotive industry. This case study demonstrates the ontology framework's capability to collect, represent, manage, and maintain knowledge that will be used, synthesized, inferred, and verified during the design phase.

Case study implementation

The validation of the developed framework's applicability and its ability to support structured failure knowledge representation was conducted through a real-world industrial case study of a headlamp system. This case study was undertaken in collaboration with practicing engineers and domain experts from a global automotive

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

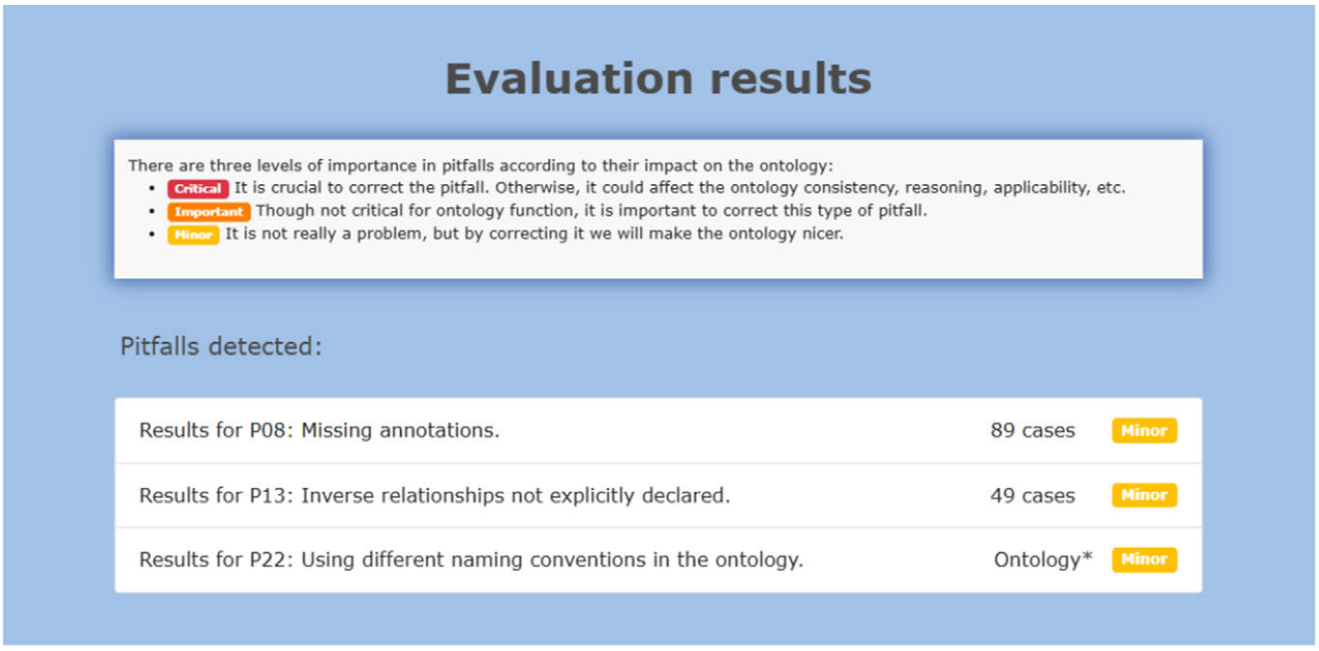


Figure 6. Evaluation results of the FBSFM ontology using the OOPS.

manufacturer, ensuring that the ontology was grounded in authentic engineering artefacts and practices. The implementation would demonstrate the capability of the FBSFM framework to consolidate multi-source engineering knowledge into a coherent, machine-readable repository, linking design entities from user goals to potential system failures and enabling systematic validation of FMEA information against system models. The implementation steps are illustrated in Figure 8, which demonstrates the application of the methodology to a real-world case study. These steps are outlined as follows:

- Step 1: Conduct a case study data analysis, including FMEA spreadsheets and system model data.
- Step 2: Perform text preparation and preprocessing.
- Step 3: Populate the preprocessed data into the constructed framework.
- Step 4: Explore the populated ontology.

Step 1: Case study analysis

For the headlamp case study, the documentation provided included two key elements as below:

(1) System architecture documentation, along with descriptions of system decomposition, functional decomposition, and functional requirements. Figure 9 presents the block diagram of the case study, illustrating the structural decomposition derived from the system architecture PDF.
(2) The existing FMEA documentation for the headlamp system, in a tabular database format (Excel spreadsheet) containing multiple sheets:
    - The FMEA table, structured similar to Table 2, following the AIAG/VDA standard.
    - The system architecture matrix, including the function and behavior analysis for the headlamp:
        - A table listing the functions of the focus element.
        - A column displaying the behavior of each function, referred to as the "function criteria."

The headlamp system case study comprises three system levels, as depicted in the snapshot from the FMEA table (Fig. 10):

- **Level 1**: Represents the overall headlamp system, identified as the "Low Beam Lamp."
- **Level 2**: Includes the focus elements, such as the "LB Module" (Low Beam Module).
- **Level 3**: Represents individual components, such as the "Housing."

For the headlamp system case study analysis, the following parameters were used to populate the ontology by instantiating the FBSFM framework with data specific to the headlamp system.

1. System structure
2. System function
3. System behavior
4. Failure mode name
5. Cause of failure
6. Effects of failure
7. Detection control
8. Prevention control
9. Detection
10. Occurrence
11. Severity

Step 2: Text preparation and preprocessing

To ensure compatibility between the provided case study data and the ontological framework developed in Protégé, a preprocessing step was necessary. This involved handling limitations related to spaces, special characters, and missing values. A Python-based script was developed to automate text cleaning, space replacement, and missing data handling to prepare both the FMEA and functional requirements datasets for ontology population.

Step 3: Populate data into the ontological framework

Before populating instances from the case study into the ontological framework, it is important to note that the case study involves three

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

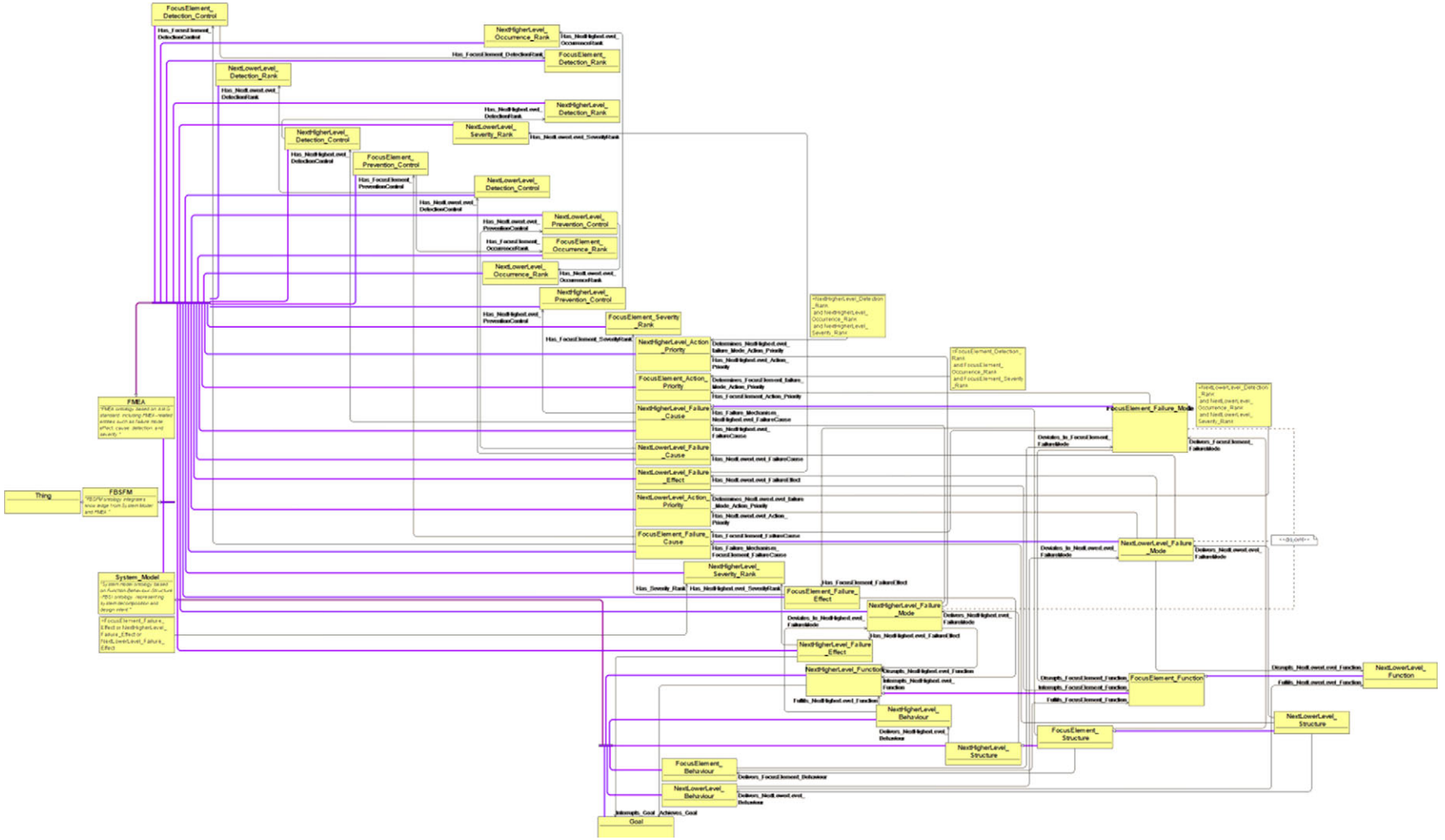


Figure 7. Knowledge graph representation of the FBSFM framework.

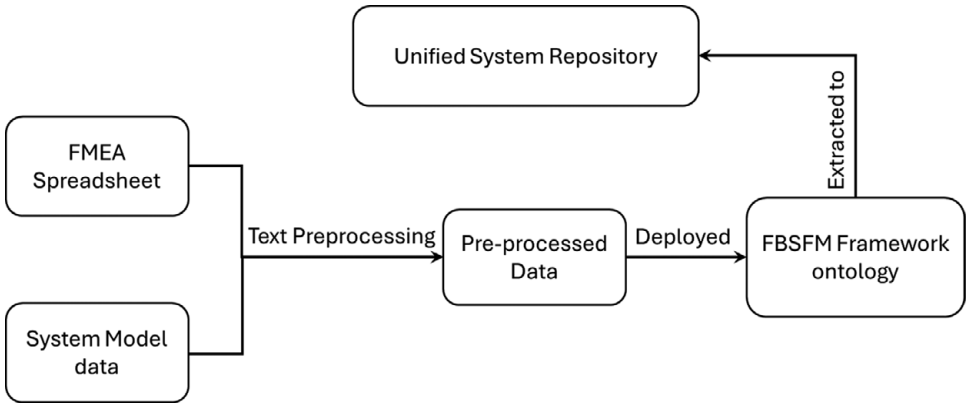


Figure 8. Implementation steps for applying the ontological framework to a real-world case study.

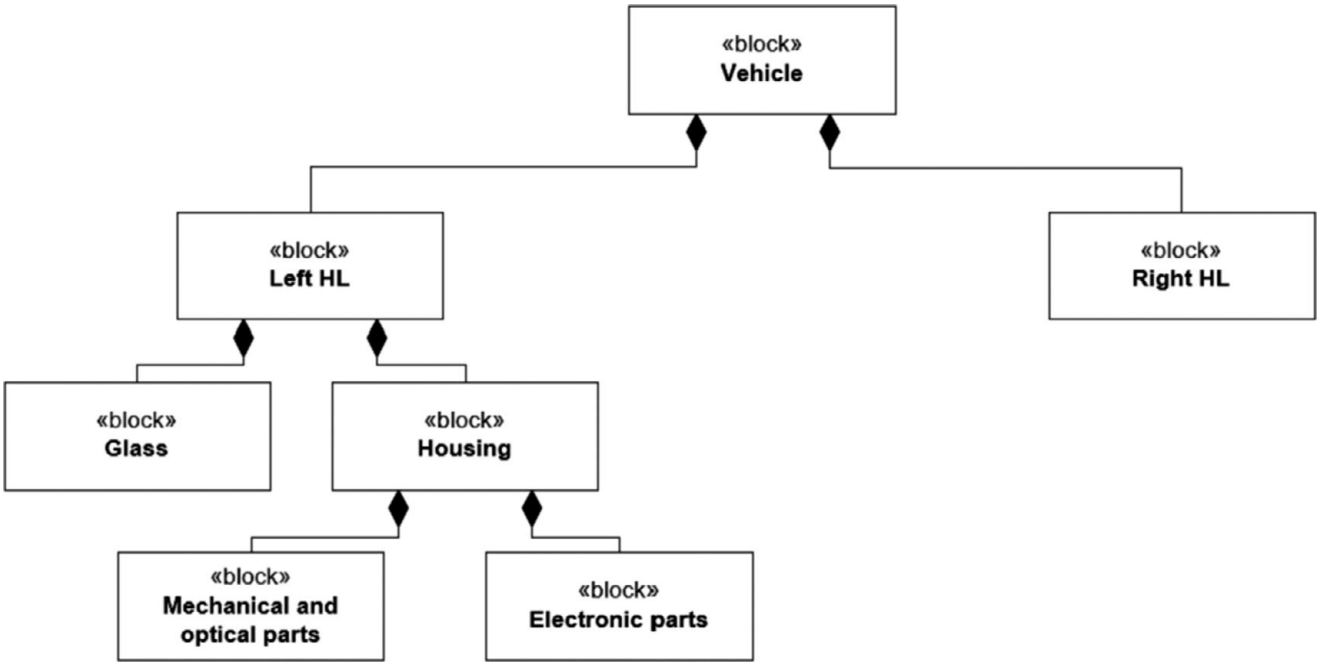


Figure 9. Block diagram illustrating the structural decomposition of the case study derived from the system architecture.

levels of decomposition and analysis. The system is decomposed into a higher level, a focused element, and a lower level.

Figure 11 provides a screenshot of the ontological framework, showcasing the three-level system decomposition represented as a KG. The snapshot illustrates classes, such as "Lower-Level Function," "Higher Level Function," "Higher Level Effect," and "Higher Level Cause," along with their associated relationships.

The FMEA sheets and function analysis sheet were imported into the ontology using Protégé. This was accomplished through the Tools menu by selecting the "Create axioms from Excel sheet" option. During this process:

- Column mapping: The Excel sheet columns were identified and mapped to their corresponding ontology classes.
- Relation assignment: The corresponding relationships between these classes were also defined.

These mappings and relationships were saved in JSON format by Protégé, enabling reuse for future case studies. This eliminates the need for the design team to recreate the ontological framework or reassign relationships for subsequent cases.

Once the instances were imported and assigned to their respective classes and relationships, the ontology was saved in Manchester Syntax format. This format facilitates further exploration and analysis of the populated ontology in the next step.

Step 4: Explore the populated ontology

The primary purpose of this exploration step is to support engineering practitioners in retrieving targeted failure knowledge linked to specific design elements, thereby enhancing traceability, reuse, and design validation. SPARQL queries were employed within the Protégé ontology editor to systematically explore the FBSFM ontology, enabling the extraction of structured knowledge from its semantic relationships. Integrated directly into the Protégé interface, this querying capability allows users to retrieve specific classes, properties, and instances that would otherwise be difficult to access manually (Satyamurty et al., 2017). For instance, as shown in Figure 12, a

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

1.Next Higher Level	2. Focus Element	3. Next Lowel Level or Characteristic Type	1-Next Higher-Level Function			2. Focus Element Function			3. Next Lower-Level Function or Characteristic		
			Next Higher-Level Function Expression	Next Higher-Level Function Criteria	Next Higher-Level Function ID	Focus Element Function Expression	Focus Element Function Criteria	Focus Element Function ID	3. Next Lower-Level Function	3. Next Lower-Level Function Criteria	Next Lower-Level Function ID
Low Beam LAMP	LB module	Housing	To Convert Regulated Electrical Power from Headlamp to projected low beam considering External Constraint	NA	NA	To Convert Regulated Electrical Power from Headlamp to projected low beam considering External Constraint	NA	NA	To Ensure a continuous mechanical & electrical link with car Harness	NA	NA
Low Beam LAMP	LB module	Programmed PCBa	To Convert Regulated Electrical Power from Headlamp to projected low beam considering External Constraint	NA	NA	To Convert Regulated Electrical Power from Headlamp to projected low beam considering External Constraint	NA	NA	To Convert Regulated Electrical Power from Headlamp to projected low beam considering External Constraint	NA	NA

Figure 10. Example from case study FMEA table showing three system levels of the headlamp system.

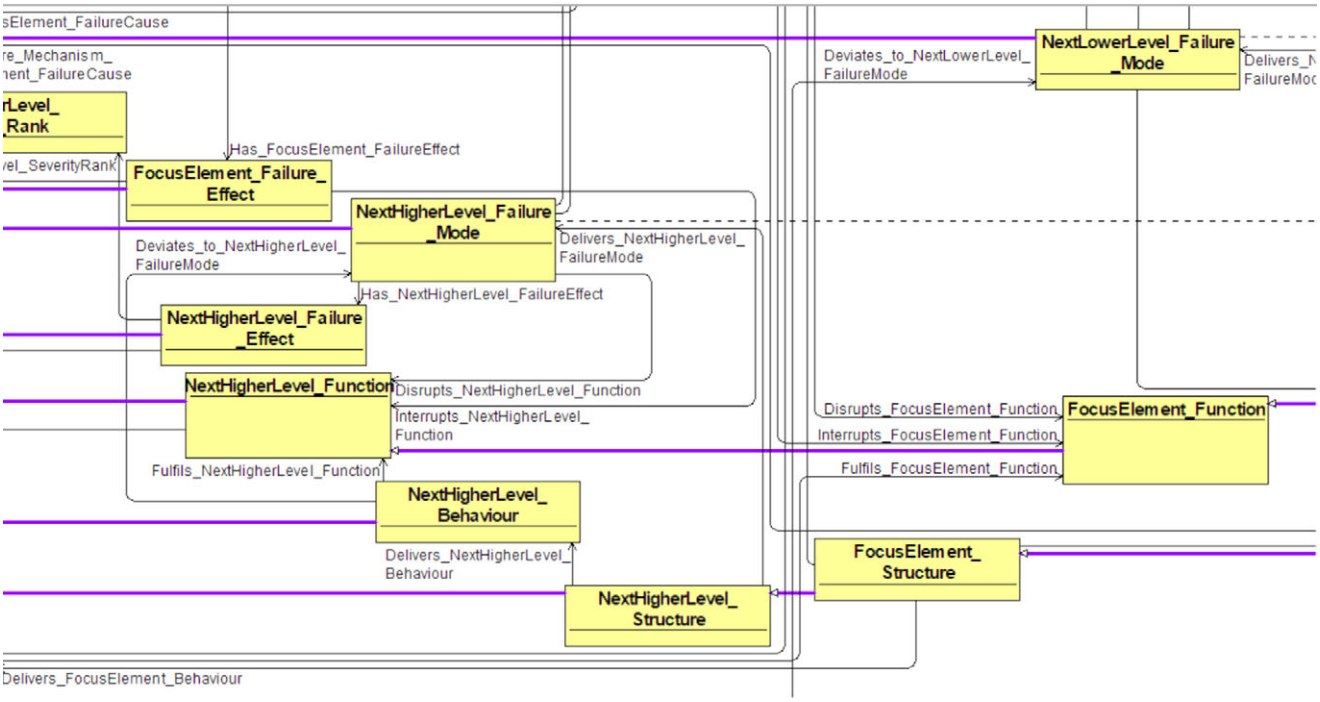


Figure 11. Extended ontological framework representing three-level system decomposition as a knowledge graph.

SPARQL query was used to retrieve all failure causes associated with the "Lens" component by filtering individuals linked through the object property Has_Failure_Mechanism_FocusElement_Failure-Cause. The resulting list presents domain-specific failure causes, such as: "Temperature_in_the_lens_is_higher_than_the_material_limit…," and "Reduced_thickness_does_allow_a_proper_material_flow," which were directly extracted from the ontology. This function supports both consistency checking and traceability, offering practical insight into the relationships among functions, structures, and failure mechanisms in complex engineering systems.

To simplify the use and retrieval of knowledge from the saved ontology, a custom Python script was developed to process the populated knowledge repository and is available at https://github.com/Haythamyounus/FBSFM-Ontology. This script also replaces underscores in entity names and cleans URLs to enhance human readability. Compared to using SPARQL queries directly in

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

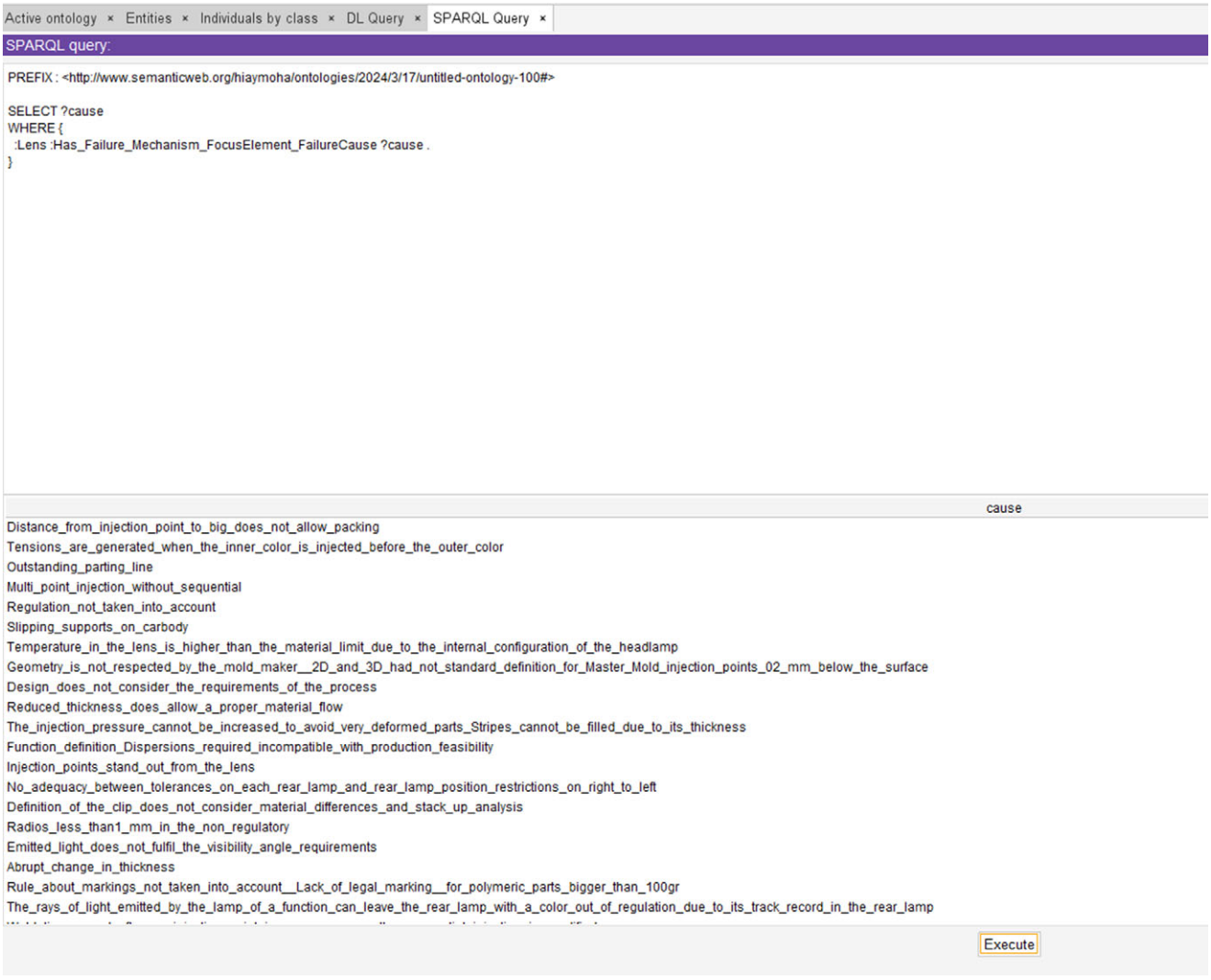


Figure 12. Example SPARQL query retrieving all failure causes related to the lens.

Protégé, the script provides a more accessible and user-friendly way to explore the ontology content.

In the following examples, highlighted colors are used to distinguish ontology syntax:

- Green: Represents entity categories.
- Grey: Denotes related properties for each category.
- Yellow: Highlights data derived from the case study.

Example 1: High-level structure

In the following example from case study ontology in Manchester Syntax, the example demonstrates one of the entities within the ontology: "Low Beam LAMP."

Each entity in the ontology is defined as an "Individual" with the following key attributes:

- Type: Indicates the individual's class name.
- Facts: Displays the relationships of the individual with other classes.

Figure 13 illustrates the details for the individual "Low Beam LAMP."

The ontological individual "Low Beam Lamp" is categorized as a "Higher Level Structure" and delivers two higher-level behaviors. However, the related failure mode for this system structure is not provided in the case study. This highlights the ontology's capability to identify data gaps where important risk information is missing, providing actionable insights for further investigation. Additionally, the example demonstrates the semantic bridging from system structure to Structure behavior and Unintended behavior (failure modes), even where explicit risk elements are absent, reinforcing the framework's support for linking design intent to potential risk analysis across system levels.

To validate the functionality of the developed script, a SPARQL query was executed to retrieve information about "LowBeam_LAMP." The result was identical to that produced by the script, confirming its correctness, as illustrated in Figure 14.

Example 2: Focused element structure

In the selected example shown in Figure 14, the individual is identified as "LB module," which is classified in the case study as a "Focused Element Structure." The facts presented in Figure 10 illustrate other entities related to this individual. Specifically, the "LB module" delivers two distinct focused-level behaviors and is associated with three different failure modes. Additionally, the "LB module" is part of a higher-level structure, "Low Beam LAMP."

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

Individual: LowBeam LAMP

Types:

Higher Level Structure

Facts:

Delivers Higher Level Behaviour Pproduct IMDS list in regards of Regulation lows Customer request JDM H30 Allowed material,

Delivers Higher Level Behaviour 1 Photometric value According to Regulation & Customer2 Vertical cut off adjustment According to Regulation & Customer3 Vertical position of cut off lin According to Regulation & Customer4 Cut off stability Change in LB cut off less than 015 Beam color According to Regulation & Customer6 Beam width Beam width = +/ XX7 Beam height Beam height bigger than X8 Homogeneity According to Customer9 Beam Flux Beam Flux = XXX Lm10 Emax Emax bigger than XX Lx11 Emax position Luminance level at point 00 bigger than XX Emax,

Delivers Higher Level Failure Mode Not Avaliable Higher level Failure Mode

Figure 13. Detailed ontological representation of the individual "Low Beam LAMP" highlighting class type and associated relationships.

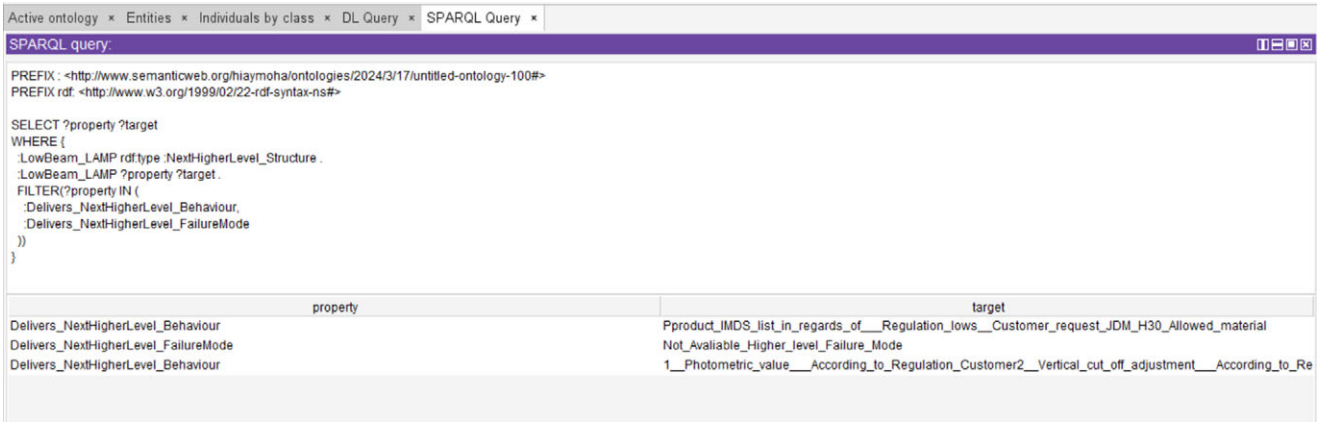


Figure 14. SPARQL query result for the individual "LowBeam_LAMP."

This example shows the ontology's ability to represent multiple failure modes associated with a system element, providing a detailed and structured risk view. It also reveals a missing link: no failure mechanism has been captured to connect this component back to the failure causes at the higher level, indicating a potential gap in the case study data due to incomplete risk analysis.

Importantly, this example demonstrates semantic bridging from structure to behavior and failure modes, offering a machine-readable traceability from system design elements through to risk elements.

Example 3: Lower-level structure

As illustrated in examples 1 and 2, example 3 (shown in Figure 15) demonstrates the knowledge repository of the case study ontology by exploring an individual called "Housing." This individual is classified in the ontology as a "Lower-Level Structure" which delivers one lower-level behavior (Figure 16). However, no direct failure mode is available for this structure, as the case study focuses on failure modes at the focused level.

While the ontology does not link a direct failure mode to the "Housing" structure, it captures a failure mechanism: "electrical connection not ensured." This provides insight into potential risks even at lower decomposition levels, demonstrating the ontology's ability to systematically trace failure mechanisms. Furthermore, this example confirms the semantic bridging from lower-level structure to behavior and corresponding failure mechanisms, reinforcing traceability across decomposition levels in a machine-readable format.

Following the presentation of the three illustrative examples, it is important to clarify the scope and purpose of the ontology exploration. The selected examples were chosen to demonstrate different system levels within the case study and to highlight the ontology's ability to support structured querying, cascading decomposition, and traceability across functions, structures, and failure modes.

While the publicly available version of the ontology contains only these three instances for confidentiality reasons, the framework itself was tested on a broader set of data internally. A dedicated Python script was developed to interface with the ontology, allowing users to specify search terms and retrieve associated knowledge entities and relationships automatically. This tool was provided to practicing engineers and domain experts within the industrial partner's team, who were invited to explore and validate the ontology independently. Their interaction with the system demonstrated its usability and flexibility for navigating complex engineering knowledge, and their informal feedback indicated that the structured repository and search mechanism effectively supported traceability and design analysis tasks.

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

Individual: LB module

Types:

Focused Element Structure

Facts:

Delivers Focused Level Behaviour Pproduct IMDS list in regards of Regulation lows Customer request JDM H30 Allowed material,

Delivers Focused Level Behaviour 1 Photometric value According to Regulation & Customer2 Vertical cut off adjustment According to Regulation & Customer3 Vertical position of cut off lin According to Regulation & Customer4 Cut off stability Change in LB cut off less than 015 Beam color According to Regulation & Customer6 Beam width Beam width = +/ XX7 Beam height Beam height bigger than X8 Homogeneity According to Customer9 Beam Flux Beam Flux = XXX Lm10 Emax Emax bigger than XX Lx11 Emax position Luminance level at point 00 bigger than XX Emax,

Delivers Focused Level Failure Mode Loss of light projected,

Delivers Focused Level Failure Mode Lost totally the function, no more work,

Delivers Focused Level Failure Mode module material not conform to current regulations / cutomer request,

Has Higher Level Failure Mechanism Not Avaliable Higher level Cause,

Is a Part of Higher Level Structure LowBeam LAMP

Figure 15. Representation of the "LB module" as a focused element structure.

Individual: Housing

Types:

Lower Level Structure

Facts:

Delivers Lower Level Behaviour Dimmensional characteristics of parts in regard of Vehicle connector Ref Vibration requirement JDQ xxx Power Interruptions JDQ202,

Delivers Lower Level Failure Mode Not Avaliable Lower level Failure Mode,

Has Focused Level Failure Mechanism electrical connection not ensured,

Is a Part of Focused Level Structure LB module

Figure 16. Ontological representation of the individual "Housing" as a lower-level structure.

Additional instances can be generated and imported into the ontology framework by end users, depending on the focus of their analysis or the system under consideration. The GitHub repository (https://github.com/Haythamyounus/FBSFM-Ontology) contains only the three public examples discussed above, but the ontology structure is fully extensible and designed to support future population and reuse across case studies.

Discussion

The primary motivation for this research stemmed from the persistent challenge of integrating system design knowledge, often represented through models like FBS, with risk analysis activities such as FMEA. This integration gap hinders traceability and consistency, potentially leading to incomplete risk assessments during the design phase. This article presents the development and validation of the FBSFM ontological framework, designed to bridge this gap by providing a unified, semantically rich structure. The case study demonstrated the framework's capability to represent and link information from both system design (FBS) and risk analysis (FMEA) domains within a single knowledge repository, thereby enhancing the potential for structured knowledge representation and traceability compared to traditional, disconnected approaches.

The choice of an ontology-based approach, specifically extending the FBS model, was driven by the need for formal semantics and computational reasoning capabilities to manage the complex relationships between design and failure information. Ontologies

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

provide a structured, unambiguous representation that facilitates consistency checking and the potential for automated analysis, which may be less achievable with direct MBSE-FMEA tool links or less formal graph database approaches that lack explicit semantics. The FBS model was chosen as a foundation due to its established role in representing design rationale through function, behavior, and structure. However, this approach is not without drawbacks. Developing robust ontologies requires significant domain expertise and effort, and ensuring scalability for extremely large, complex systems remains an ongoing challenge in ontology engineering. While alternatives might offer simpler integration paths, they may sacrifice the semantic depth and reasoning potential offered by the FBSFM ontology.

The development process draws upon established ontology engineering practices, aligning with the NeOn methodology's scenario presented in Suárez et al. (2011) for "ontology reengineering and merging" as it involved extending the existing FBS concepts and merging them with concepts derived from FMEA standards and practices. This structured methodology guided the knowledge acquisition, conceptualization (including the crucial extension of FBS with unintended behavior), formalization in OWL, and subsequent validation phases, aiming to ensure the robustness and relevance of the resulting ontology.

The principal contribution of this research lies in the development of a novel ontological framework for enhanced representation and integration of system knowledge. As a theoretical development, the FBSFM top-level ontology provides a structured and detailed method for capturing and linking system design information in terms of functions, intended behaviors, and structure, drawing from the FBS ontology, with an ontology schema for the representation of the actual behavior as function failure modes, enriched with linkages to causes and effects across multiple levels of system abstraction. This integrated representation improves design risk analysis by facilitating the traceability between design decisions and potential failure scenarios compared to traditional documentation or separate models, as demonstrated in the headlamp case study.

While several foundational ontologies, such as BFO and DOLCE, provide rigorous philosophical treatments of entities like functions, dispositions, and behaviors, they do not offer an integrated view that semantically links system design constructs with structured FA. Specifically, those ontologies lack the representational scope to formally align system functions, behaviors, and structures with FMEA artifacts, such as failure modes, causes, and effects. It is important to note that some FMEA standards, including ISO 60,812, still follow a bottom-up approach focused on component-level failure identification. In contrast, the proposed framework is explicitly integrated with the AIAG and VDA standard, which adopts a top-down, systems engineering perspective and is designed to promote early alignment between system design models and risk analysis activities. The FBSFM framework addresses this gap by explicitly bridging these domains, enabling a multilayered ontological representation that reflects both the intended design and potential failure scenarios. This dual representation supports not only system modeling and behavioral analysis but also facilitates the reuse of failure knowledge and semantic querying, making it highly suitable for engineering applications where traceability and diagnostic reasoning are critical. Unlike DOLCE and BFO, which remain abstract and often detached from implementation-level semantics, FBSFM is developed with practical deployment in mind, as demonstrated in the case study with a global automotive manufacturer.

Reflecting on the validation case study presented, the ontological framework effectively demonstrates its capacity to integrate data from two different activities during the design phase: reasoning about the system model and FMEA. Furthermore, the implementation within the case study illustrates the ontology's proficiency in consolidating this data into a unified knowledge repository. This repository, as evidenced in the examples, underscores the ontology's ability to logically track system structures across various levels, as well as the logical tracking for the failure modes and their causes and effects, with their corresponding entities from both the system model, which allows failure propagation tracking across different system levels.

The utilization of Manchester syntax for exploring and analyzing the stored ontology provided a significant advantage in terms of human readability. This, however, is not the sole benefit. The ontology also enables reasoning relations between different entities in the design phase, which are machine-readable, thereby facilitating advanced analysis and exploration powered by machine processing. Ultimately, this promotes the efficient sharing and reuse of knowledge among different teams throughout the design phase, while simultaneously reducing time consumption.

The case study implementation and validation were carried out in conjunction with engineers and domain experts from a global automotive supplier and took place in the context of a broader review of internal processes, including the alignment of the Company's FMEA practice with the systems engineering approach, within the AIAG/VDA standard framework. Their feedback and direction played an important role in shaping the development and implementation, and in particular, the development of the system knowledge querying facility. The main benefit demonstrated was the potential to improve the consistency and completeness of FMEA by ensuring it aligns with the system model represented within the ontology. By explicitly linking failure modes to unintended behaviors and tracing causes back to structural elements, the framework facilitated a more systematic review process, which was the main aim set by practitioners for the case study. The automotive case study of an electromechanical system provided a good empirical validation for the proposed FBSFM ontological framework. While the approach, as a top-level ontology, is theoretically generalizable, further testing on more complex multidisciplinary systems, including control and safety-related features, is necessary to confirm scalability and adaptability.

To formally position the FBSFM framework within the broader ontological landscape, future work will consider mapping it to foundational upper ontologies such as BFO or Industrial Ontologies Foundry. This would support broader interoperability and allow integration with domain-independent reasoning frameworks. However, such an alignment requires a more extensive evaluation process, including benchmarking against existing ontologies, assessing semantic coverage, and verifying the structural consistency of the ontology.

In this study, the FBSFM ontology was developed and applied in the context of a specific engineering case study to represent and integrate function, behavior, structure, and failure knowledge. The evaluation focused on demonstrating its practical implementation and traceability support within that defined context. A more comprehensive structural assessment against formal ontology engineering criteria and a comparative benchmarking exercise against existing ontologies were beyond the current scope. We do recognize that further work is needed on aligning and merging the FBSFM ontology with foundational ontologies such as BFO to enhance interoperability, philosophical rigor, and integration with broader semantic infrastructures.

While the case study illustrated the FBSFM effectiveness for a typical electromechanical automotive system, further validation case studies are needed for multidisciplinary systems, incorporating

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

embedded software and control components, to prove the scalability of the proposed approach. A key lesson learnt from the real-world case study was that developing ontologies that capture detailed domain-specific knowledge requires substantial investment of time and input from experienced engineers, which may act as a barrier to adoption in practice.

Conclusion

This article presented the development and validation of the FBSFM ontological framework to address the integration gap between system design models and risk analysis activities during the early design phase. By extending the established FBS model and formally integrating it with FMEA concepts, the proposed framework enables a unified, semantically rich representation of design knowledge and failure information.

Through the real-world automotive case study, in conjunction with practicing engineers and domain experts, the FBSFM ontology demonstrated its ability to consolidate multisource engineering knowledge into a coherent, machine-readable repository, supporting enhanced traceability from user goals to potential system failures. The use of ontological reasoning and structured querying facilitates a more systematic review and validation of FMEA information against system models, with a positive impact on product development practice.

Data availability statement. Data cannot be made available due to the commercially sensitive nature of the technical case study. Any enquiries about data should be made to Pascal Bonnaud at pascal.bonnaud@valeo.com. However, for reproducibility, a version of the developed ontology containing the three instances explored in this article is published online at GitHub: https://github.com/Haythamyounus/FBSFM-Ontology.

Funding statement. The authors acknowledge funding support for the research presented in this article through the France2030 OTPaaS project funded by BPI-France.

Competing interests. The authors declare none.

References

AIAG & VDA (2019) *Failure Mode and Effects Analysis-FMEA Handbook: Design FMEA, Process FMEA, Supplemental FMEA for Monitoring et System Response*. Southfield, MI: Automotive Industry Action Group.

AIAG-VDA. (2019). AIAG-VDA FMEA Handbook *(first ed.)*. AIAG-VDA. https://www.aiag.org/quality/automotive-core-tools/fmea

Alkhariji L, **De S**, **Rana O and Perera C** (2023) Semantics-based privacy by design for internet of things applications. *Future Generation Computer Systems* **138**, 280–295.

Anugerah AR, **Ahmad SA**, **Samin R**, **Samdin Z and Kamaruddin N** (2022) Modified failure mode and effect analysis to mitigate sustainable related risk in the palm oil supply chain. *Advances in Materials and Processing Technologies* **8**(2), 2229–2243.

Avizienis A, **Laprie J-C**, **Randell B and Landwehr C** (2004) Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing* **1**(1), 11–33.

Baklouti A, **Nguyen N**, **Mhenni F**, **Choley J-Y and Mlika A** (2019) Improved safety analysis integration in a systems engineering approach. *Applied Sciences* **9**(6), 1246.

Belu N, **Ionescu L**, **Rachieru N and Mazare A** (2022) A model for monitoring of the 8D and FMEA tools interdependence in the era of industry 4.0. *International Journal of Modern Manufacturing Technologies* **14**, 86–91. https://doi.org/10.54684/ijmmt.2022.14.3.86.

Borgo S, **Carrara M**, **Garbacz P and Vermaas PE** (2009) A formal ontological perspective on the behaviors and functions of technical artifacts. *AI EDAM* **23**(1), 3–21.

Borgo S, **Ferrario R**, **Gangemi A**, **Guarino N**, **Masolo C**, **Porello D**, **Sanfilippo EM and Vieu L** (2022) DOLCE: A descriptive ontology for linguistic and cognitive engineering. *Applied Ontology* **17**(1), 45–69.

Cabeza-Pullés D, **Fernández-Pérez V and Roldán-Bravo MI** (2020) Internal networking and innovation ambidexterity: The mediating role of knowledge management processes in university research. *European Management Journal* **38**(3), 450–461.

Campean IF and Henshall EJ (2012) The functional basis for failure mode avoidance in automotive systems engineering design. In *DS 72: Modelling and Management of Engineering Processes-Concepts, Tools and Case Studies*, pp. 45–60.

Campean IF, **Henshall E and Rutter B** (2013) Systems engineering excellence through design: An integrated approach based on failure mode avoidance. *SAE International Journal of Materials and Manufacturing* **6**(3), 389–401.

Campean F, **Henshall E and Rutter B** (2014) A systems approach to the development and use of FMEA in complex automotive applications. *SAE International Journal of Materials and Manufacturing* **7**, 280–290. https://doi.org/10.4271/2014-01-0740.

Cebrian-Tarrason D, **Lopez-Montero J and Vidal R** (2008) OntoFaBeS: Ontology design based in FBS framework. *Structure* **11**, 12.

Compagno F (2025) *Modelling of Functions and Malfunctions in Industrial Processes: An Applied Ontology Approach*. PhD Thesis, University of Trento, pp. 1–230.

Compagno F and Borgo S (2024) Towards a formal ontology of engineering functions, behaviours, and capabilities. *Semantic Web* **15**(2), 285–318.

da Silva TR and Pereira CE (2014) Building an ontology for intelligent maintenance systems and spare parts supply chain integration. *IFAC Proceedings* **47**(3), 7843–7848.

Day J, **Donahue K**, **Ingham M**, **Kadesch A**, **Kennedy A and Post E** (2012) *Modeling off-nominal behavior in sysml. Infotech@ Aerospace 2012*, pp. 2576.

De Bem Machado A, **Secinaro S**, **Calandra D and Lanzalonga F** (2022) Knowledge management and digital transformation for industry 4.0: A structured literature review. *Knowledge Management Research & Practice* **20**(2), 320–338.

Dermeval D, **Vilela J**, **Bittencourt II**, **Castro J**, **Isotani S**, **Brito P and Silva A** (2016) Applications of ontologies in requirements engineering: A systematic review of the literature. *Requirements Engineering* **21**, 405–437.

Ebrahimipour V, **Rezaie K and Shokravi S** (2010) An ontology approach to support FMEA studies. *Expert Systems with Applications* **37**(1), 671–677.

Eckhardt J, **Vogelsang A and Fernández DM** (2016) Are "non-functional" requirements really non-functional? An investigation of non-functional requirements in practice. In *Proceedings of the 38th International Conference on Software Engineering*, pp. 832–842.

Eisenbart B and Gericke K (2020) Function in engineering. In Diane P. Michelfelder, Neelke Doorn (eds.) *The Routledge Handbook of the Philosophy of Engineering*. Taylor & Francis, pp. 245–262.

Filz M-A, **Langner JEB**, **Herrmann C and Thiede S** (2021) Data-driven failure mode and effect analysis (FMEA) to enhance maintenance planning. *Computers in Industry* **129**, 103451. https://doi.org/10.1016/j.compind.2021.103451.

Galle P (2009) The ontology of Gero's FBS model of designing. *Design Studies* **30**(4), 321–339.

Gavanas N (2019) Autonomous road vehicles: Challenges for urban planning in European cities. *Urban Science* **3**(2), 61.

Gero JS (1990) Design prototypes: A knowledge representation schema for design. *AI Magazine* **11**(4), 26–26.

Gero JS and Kannengiesser U (2007a) A function–behavior–structure ontology of processes. *AI EDAM* **21**(4), 379–391.

Gero JS and Kannengiesser U (2007b) An ontology of situated design teams. *AI EDAM* **21**(3), 295–308.

Girard G, **Baeriswyl I**, **Hendriks JJ**, **Scherwey R**, **Müller C**, **Hönig P and Lunde R** (2020) Model based safety analysis using SysML with automatic generation of FTA and FMEA artifacts. Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference (Esrel 2020 PSAM 15), 1–5 November 2020, Venice, Italy.

Godina R, **Silva BGR and Espadinha-Cruz P** (2021) A DMAIC integrated fuzzy FMEA model: A case study in the automotive industry. *Applied Sciences* **11**(8), 3726.

Goktas Y, **Hu Y and Yellamati DD** (2024) P-diagram driven robust PFMEA development. In **2024** *Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–7.

Hamraz B, **Caldwell NH**, **Ridgman TW and Clarkson PJ** (2015) FBS linkage ontology and technique to support engineering change management. *Research in Engineering Design* **26**, 3–35.

Haytham Y, **Doikin A**, **Campean F**, **Kabir S**, **Abdullatif A**, **Delaux D and Bonnaud P** (2024) An extended function-behaviour-structure ontology to support FMEA within a system engineering context. *Procedia CIRP* **128**, 644–649. https://doi.org/10.1016/j.procir.2024.06.034.

Hecht M, **Dimpfl E and Pinchak J** (2014) Automated generation of failure modes and effects analysis from SysML models. In 2014 *IEEE International Symposium on Software Reliability Engineering Workshops*, pp. 62–65.

Herre H (2010) General formal ontology (GFO): A foundational ontology for conceptual modelling. In Poli R, Healy M, Kameas A (eds) *Theory and Applications of Ontology: Computer Applications*. Dordrecht: Springer. https://doi.org/10.1007/978-90-481-8847-5_14.

Hezla L, **Gurina R**, **Hezla M**, **Rezaeian N**, **Nohurov M and Aouati S** (2023) The role of artificial intelligence in improving failure mode and effects analysis (FMEA) efficiency in construction safety management. In *International Conference on Artificial Intelligence and Virtual Reality*, pp. 397–411.

Hodkiewicz M, **Klüwer JW**, **Woods C**, **Smoker T and Low E** (2021) An ontology for reasoning over engineering textual data stored in FMEA spreadsheet tables. *Computers in Industry* **131**, 103496.

Hogan A, **Blomqvist E**, **Cochez M**, **D'amato C**, **Melo GD**, **Gutierrez C**, **Kirrane S**, **Gayo JEL**, **Navigli R**, **Neumaier S**, **Ngomo A-CN**, **Polleres A**, **Rashid SM**, **Rula A**, **Schmelzeisen L**, **Sequeda J**, **Staab S and Zimmermann A** (2021) Knowledge graphs. *ACM Computing Surveys*, **54**(4), Article 71. https://doi.org/10.1145/3447772

Huang Z, **Swalgen S**, **Davidz H and Murray J** (2017) MBSE-assisted FMEA approach—Challenges and opportunities. In 2017 *Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–8.

Huang Z, **Hansen R and Huang Z** (2018) Toward FMEA and MBSE integration. In 2018 *Annual Reliability and Maintainability Symposium (RAMS)*, pp. 1–7.

Husung S, **Weber C**, **Mahboob A and Kleiner S** (2021) Using model-based systems engineering for need-based and consistent support of the design process. *Proceedings of the Design Society* **1**, 3369–3378.

IEC (2018) Failure modes and effects analysis (FMEA and FMECA), Standard, IEC 60812: 2018. Geneva: International Electrotechnical Commission.

Jiménez López E, **Cuenca Jiménez F**, **Luna Sandoval G**, **Ochoa Estrella FJ**, **Maciel Monteón MA**, **Muñoz F and Limón Leyva PA** (2022) Technical considerations for the conformation of specific competences in mechatronic engineers in the context of industry 4.0 and 5.0. *PRO* **10**(8), 1445.

Kitamura Y, **Takafuji S and Mizoguchi R** (2007) Towards a reference ontology for functional knowledge interoperability. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. **48078**, pp. 111–120.

Kommineni VK, **König-Ries B and Samuel S** (2024) *From human experts to machines: An LLM supported approach to ontology and knowledge graph construction. arXiv preprint arXiv:2403.08345.*

Korsunovs A, **Doikin A**, **Campean F**, **Kabir S**, **Hernandez E**, **Taggart D**, **Parker S and Mills G** (2022) Towards a model-based systems engineering approach for robotic manufacturing process modelling with automatic FMEA generation. *Proceedings of the Design Society* **2**, 1905–1914.

Kumar PP (2008) *Design Process Modeling: Towards an Ontology of Engineering Design Activities*. Clemson, SC: Clemson University.

Lai K, **Robert T**, **Shindman D and Olechowski A** (2021) Integrating safety analysis into model-based systems engineering for aircraft systems: A literature review and methodology proposal. In *INCOSE International Symposium*, vol. **31**, no. 1, pp. 988–1003.

Liou JJ, **Liu PC**, **Luo S-S**, **Lo H-W and Wu Y-Z** (2022) A hybrid model integrating FMEA and HFACS to assess the risk of inter-city bus accidents. *Complex & Intelligent Systems* **8**(3), 2451–2470.

Liu HC (2019) *Improved FMEA methods for proactive healthcare risk analysis*. Singapore: Springer, pp. 73–96.

Loebe F, **Burek P and Herre H** (2022) GFO: The general formal ontology. *Applied Ontology* **17**(1), 71–106.

Mansoor A, **Diao X and Smidts C** (2023) A method for backward failure propagation in conceptual system design. *Nuclear Science and Engineering* **197**(11), 2751–2777.

Mbaye S, **Walsh HS**, **Davies M**, **Infeld SI and Jones G** (2024) From BERTopic to SysML: Informing model-based failure analysis with natural language processing for complex aerospace systems. In *AIAA SCITECH 2024 Forum*, pp. 1–13.

Mikos WL, **Ferreira JCE**, **Botura PEA and Freitas LS** (2011) A system for distributed sharing and reuse of design and manufacturing knowledge in the PFMEA domain using a description logics-based ontology. *Journal of Manufacturing Systems* **30**(3), 133–143. https://doi.org/10.1016/j.jmsy.2011.06.001.

Milakis D, **Van Arem B and Van Wee B** (2017) Policy and society related implications of automated driving: A review of literature and directions for future research. *Journal of Intelligent Transportation Systems* **21**(4), 324–348.

Musen MA and **Protégé Team** (2015) The protégé project: A look back and a look forward. *AI matters* **1**(4), 4–12.

Nakajima Y (2024) *MBSE-Driven Exploratory System Architecture Analysis Framework for the Engineering Test Satellite-9 Project*. AIAA Aviation Forum and Ascend, Vol. **2024**.

Nikitas A, **Vitel A-E and Cotet C** (2021) Autonomous vehicles and employment: An urban futures revolution or catastrophe? *Cities* **114**, 103203.

Otte JN, **Beverley J and Ruttenberg A** (2022) BFO: Basic formal ontology. *Applied Ontology* **17**(1), 17–43.

Pan JZ, **Razniewski S**, **Kalo J-C**, **Singhania S**, **Chen J**, **Dietze S**, **Jabeen H**, **Omeliyanenko J**, **Zhang W and Lissandrini M** (2023) *Large language models and knowledge graphs: Opportunities and challenges. arXiv preprint arXiv:2308.06374.*

Pinto HS, **Staab S and Tempich C** (2004) *DILIGENT: Towards a Fine-Grained Methodology for DIstributed, Loosely-Controlled and evolvInG Engineering of oNTologies*. ECAI.

Poveda-Villalón M, **Asunción G-P and Suárez-Figueroa MC** (2014) OOPS! (ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, **10**(2), 7–34.

Poveda-Villalón M, **Fernández-Izquierdo A**, **Fernández-López M and García-Castro R** (2022) LOT: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence* **111**, 104755.

Qian L and Gero JS (1996) Function–behavior–structure paths and their role in analogy-based design. *AI EDAM* **10**(4), 289–312.

Razouk H, **Liu XL and Kern R** (2023) Improving FMEA comprehensibility via common-sense knowledge graph completion techniques. *IEEE Access* **11**, 127974–127986.

Rehman Z and Kifor CV (2016) An ontology to support semantic management of FMEA knowledge. *International Journal of Computers Communications & Control* **11**(4), 507–521.

Russomanno DJ, **Bonnell RD and Bowles JB** (1993) Functional reasoning in a failure modes and effects analysis (FMEA) expert system. *Annual Reliability and Maintainability Symposium 1993 Proceedings*.

SAE-J1739 (2009) *Potential Failure Mode and Effects Analysis (FMEA), Design FMEA, and Process FMEA Including FMEA–MSR (Monitoring and System Response)*. SAE Standard J1739_200901. Warrendale, PA: SAE International. https://doi.org/10.4271/J1739_200901.

Safont-Andreu A, **Burmer C and Schekotihin K** (2021) Using ontologies in failure analysis. In *International Symposium for Testing and Failure Analysis*, vol. 84215, pp. 23–28.

Satyamurty CV, **Murthy JVR and Raghava M** (2017) Developing higher education ontology using protégé tool: Reasoning. In *Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016*, vol. **1**, pp. 233–241.

Schindel WD (2024) Failure analysis: Insights from model-based systems engineering. *Insight* **27**(5), 44–49.

Sierla S, **Tumer I**, **Papakonstantinou N**, **Koskinen K and Jensen D** (2012) Early integration of safety to the mechatronic system design process by the functional failure identification and propagation framework. *Mechatronics* **22**(2), 137–151.

Spear AD, **Ceusters W and Smith B** (2016) Functions in basic formal ontology. *Applied Ontology* **11**(2), 103–128.

Stone RB, **Tumer IY and Van Wie M** (2005) The function-failure design method. *ASME Journal of Mechanical Design* MD03-1107, 397–407.

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press

Su C-T, **Lin H-C**, **Teng P-W and Yang T** (2014) Improving the reliability of electronic paper display using FMEA and Taguchi methods: A case study. *Microelectronics Reliability* **54**(6–7), 1369–1377.

Suárez-Figueroa MC, **Gómez-Pérez A and Fernández-López M** (2011) The NeOn methodology for ontology engineering. In Suarez-Figueroa M, Gomez-Perez A, Motta E, Gangemi A. (eds), *Ontology Engineering in a Networked World*. Springer, pp. 9–34.

Tomiyama T, **Van Beek TJ**, **Cabrera AAA**, **Komoto H and D'Amelio V** (2013) Making function modeling practically usable. *AI EDAM* **27**(3), 301–309.

Tumer IY and Stone RB (2003) Mapping function to failure mode during component development. *Research in Engineering Design* **14**, 25–33.

Uludağ Y, **Evin E and Gürbüz NG** (2023) Integration of systems design and risk management through model-based systems development. *Systems Engineering* **26**(1), 48–70.

Vermaas PE (2013) The coexistence of engineering meanings of function: Four responses and their methodological implications. *AI EDAM* **27**(3), 191–202.

Wang J and Wang H (2014) Product design ontology system based on FBS-API. *J.* Compute **9**(3), 543–550.

Winton DB and Huang ZC (2021) MBSE and FMEA integration using GENESYS. *Annual Reliability and Maintainability Symposium (RAMS)* **2021**.

Wu Z, **Liu W and Nie W** (2021) Literature review and prospect of the development and application of FMEA in manufacturing industry. *The International Journal of Advanced Manufacturing Technology* **112**(5), 1409–1436. https://doi.org/10.1007/s00170-020-06425-0.

Yang L, **Chen H**, **Li Z**, **Ding X and Wu X** (2023) *ChatGPT is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling. arXiv preprint arXiv:2306.11489*.

Yazdi M (2023) Enhancing system safety and reliability through integrated FMEA and game theory: A multi-factor approach. *Safety* **10**(1), 4.

Ye J, **Stevenson G and Dobson S** (2011) A top-level ontology for smart environments. *Pervasive and Mobile Computing* **7**(3), 359–378.

https://doi.org/10.1017/S0890060425100206 Published online by Cambridge University Press