engineering proceedings

MDPI

Proceeding Paper

OSIRIS—Generation of System-Specific Failure Cases Using Artificial Intelligence Based on Information from Abstract System Models †

Durga Sri Sharan Katabathula *, Marcel Mischke, Sebastian Stoppa and Robin Frank

German Aerospace Center (DLR), Institute of Electrified Aero Engines, 03046 Cottbus, Germany; marcel.mischke@dlr.de (M.M.); sebastian.stoppa@dlr.de (S.S.); robin.frank@dlr.de (R.F.)

* Correspondence: durga.katabathula@dlr.de; Tel.: +49-355-28888-116

† Presented at the 15th EASN International Conference, Madrid, Spain, 14–17 October 2025.

Abstract

The importance of system safety elevates with the introduction of novel technologies in the aviation industry. With the rise of system complexity, regular safety practices include iterative workflows and heavy reliance on expert knowledge. For the development of modern, efficient aircraft systems, there is a need for innovative approaches. This paper presents a tool, OSIRIS (operational safety and integrated risk analysis), that supports safety and risk analyses utilizing artificial intelligence (AI) concepts. Developed as a key safety feature within the HADES modeling framework, OSIRIS aligns with an architecture-based design approach for abstract system modeling, adhering to model-based systems engineering (MBSE) principles and standards. It currently aids safety engineers in formulating system failure cases consistent with functional hazard assessments (FHA), representing model-based safety assessment (MBSA) in compliance with SAE ARP4761A. The methodological concepts and their implementation in OSIRIS are demonstrated considering an abstract system model from aeronautical applications. The generated results were evaluated against the system context to confirm compliance with the FHA process required for certification. Further, the future work will explore refining OSIRIS's capabilities and its application cases.

Keywords: aircraft system development; MBSE; safety assessments; MBSA; ARP4754B; ARP4761A; FHA; artificial intelligence (AI); large language model (LLM)

1. Introduction

The modern aviation industry is actively focused on developing sustainable and climate-friendly aircraft systems, leading to the exploration of emerging technologies such as hybrid-electric and hydrogen propulsion systems [1]. As regulatory standards evolve to the latest developments, they pose challenges in ensuring safety, reliability and adaptability to justify design considerations [2]. Traditional approaches, even when employing iterative cycles, often fall short in addressing the complex interdependencies and dynamic interactions present within modern aircraft systems [3]. Additionally, they exhibit significant limitations in ensuring seamless data transfer across the diverse disciplines involved in aircraft development [4]. These shortcomings can lead to inefficiencies in safety assessments and hinder effective decision-making processes.

Current safety assessments are highly dependent on manual labor, human expertise, and documentation provided by the regulatory authorities [5]. According to SAE ARP4754B [6], in aircraft system development, the determination of failure cases for the

Check for updates

Academic Editors: Spiros Pantelakis, Andreas Strohmayer and Gustavo Alonso

Published: 13 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

system under development starts with functional hazard analysis (FHA) at the aircraft level, following the guidelines specified in SAE ARP4761A [7]. The results of this FHA inform safety objectives that, in turn, influence system design. However, during the system development process, conflicting system architecture can trigger the formulation of further safety requirements. This interdependence creates an iterative relationship between system design and safety assessment, emphasizing the need for a quicker decision-making process to cope with the growing amount of information.

In the aviation domain, model-based methodologies have been an effective solution for developing different artifacts of systems and safety engineering. Model-based systems engineering (MBSE) has been a key concept to abstractly handle prominent systems engineering challenges, with functional and conceptual representation of modern aircraft systems through abstract models [8]. Considering safety engineering, model-based safety assessment (MBSA) has gained significant prominence due to its connection of safety artifacts to model elements in terms of visualization inside the model-based development process [9]. This synergy enables designers to actively link abstract system elements with essential safety objectives, leading to transformative changes in system design. In the current landscape, numerous MBSE-MBSA frameworks have been identified, each employing distinct approaches [10]. However, the successful implementation of these frameworks often relies heavily on human expertise and knowledge, underscoring the need for intuitive approaches following current standards to ensure seamless adoption [10].

With artificial intelligence (AI) in the spotlight in recent times, most of the research is focused on investigating its utilization in respective fields. According to [11], the field of Systems Engineering (SE) has presumed its reliance on human-machine teams for various activities. Further, MBSE has emerged as a discipline for managing the complexity of modern technical systems, providing structured frameworks with SysML. However, its standalone application often falls short in addressing present-day challenges, necessitating integration with advanced technologies such as AI. Recent research highlights the promising synergy between MBSE and AI, aiming to optimize development processes and overcome hurdles [12]. According to the studies [13,14], the usage of the current state of the art in the field of AI, especially large language models (LLMs), has indicated considerable potential to perform safety assessments. However, these studies were performed with the manual inputs from the authors to ChatGPT (GPT-3 in [13] and GPT-4+ in [14]), an AI chatbot that uses an LLM to provide responses to users. To fully realize the potential of LLMs and validate their usage in different areas of system development, further exploration studies are required.

The hierarchical architecture design for engineering systems (HADES) modeling framework was developed to address the challenges with current model-based workflows for complex system development. It facilitates the development of abstract systems using system modeling language (SysML), with established linking across diverse levels and domains, including critical safety aspects [15]. To complement HADES, an operational safety and integrated risk analysis (OSIRIS) tool has been developed as an add-on to support different safety assessments. In its current form, OSIRIS is a plugin to the present version of HADES within the widely recognized MBSE tool, Cameo Systems Modeler (CSM). Further sections of this paper discuss the concepts behind the construction of the tool with effective usage of a localized open-source LLM. Additionally, a concise overview is provided on its role in the development of AI-based abstract systems with a focus on operational safety and risk analysis.

2. Methodology Behind OSIRIS

The OSIRIS tool was built upon the concept of utilizing the data from the abstract system model of the HADES modeling framework and leveraging the potential of contextual text generation of a locally hosted LLM to generate failure cases automatically. The preceding sections detail the methodological concepts that give the OSIRIS tool its core functionality.

2.1. Abstract System Data Retrieval and Request Formulation

In general, a system's deviation from its intended function is considered a failure. Within the developed HADES modeling framework [15], these system functions are considered as one of the key elements that provide an estimation for the system failure. These functions are defined by multiple elements that span various viewpoints that contribute to system design. The operational context of the system also plays a key role in deriving the functions that fulfill the goals of the involved stakeholders. The initialization of the OSIRIS tool starts with the function element within the HADES system model. A dedicated graphical user interface (GUI) was designed to retrieve user-desired data from the system model. This GUI renders detailed information about the selected function and its interconnected elements, leveraging the framework's inherent traceability onto a graphical window, as depicted in Figure 1.

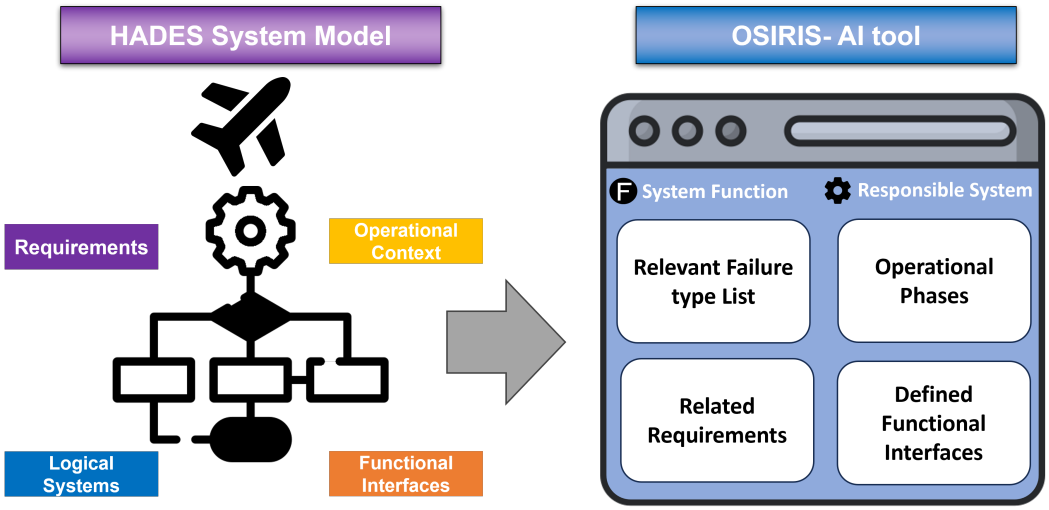


Figure 1. Retrieval of model data onto a graphical window.

At its current state, OSIRIS facilitates the definition of potential failure conditions for an abstract aircraft system following principles of functional hazard assessment (FHA). It is achieved by incorporating a set of default failure types based on FAA Advisory Circular AC23.1309-1E [16] and EASA CS25.1309 [17]. Additionally, users can define custom malfunctions specific to the selected function utilizing the tool user interface to incorporate failure types derived from the system's unique operational scenarios. To further aid in the formulation of failure cases, the tool lists relevant flight phases identified by the system's operational context. By presenting these key elements within a user-friendly GUI, it facilitates users in precisely selecting the data necessary for creating accurate and comprehensive failure conditions that are further considered for the derivation of safety objectives.

The data transfer between the system model and LLM was established through a custom-built representational state transfer application programming interface (REST API). REST API is a general architectural solution used to connect different software entities, which is considered for efficient data transfer. The user-desired data from the GUI was

utilized for formulating a structured data request using a predefined logic, which initiates a text generation request with the relevant information for failure case creation. The logic further facilitates the addition of contextual data that is required for the formulation of failure conditions to the request containing system data. This contextual data includes key instructions to the LLM to provide the failure condition, considering relevant standard and regulation data that would help in making them more suitable for the FHA process.

At the current state of OSIRIS, the addition of contextual data for failure case generation is possible in two approaches:

1. Variable context: Utilization of a customized data retrieval strategy to extract a limited amount of relevant information as additional context for the LLM.
2. Fixed context: Inclusion of all the considered regulatory data was added as context for text generation in the form of structured instructions in a tuned manner.

With the described data retrieval and request formulation approach, an enriched data request is developed. It is further utilized to initiate the text generation process with a considered LLM. The text generated is then processed to reproduce a desired safety artifact in a model-based workflow. The next section illustrates the translation of generated textual data into representative failure conditions for an FHA process. Further, the validity of these failure conditions and the effectiveness of the tool are discussed in Section 3.

2.2. Structuring a Failure Case in Context of FHA

As per the traditional FHA process, failure conditions provide an estimate of failure cases with the probability of their occurrence, particular to the system and its functions. Figure 2 depicts the regular workflow that produces failure conditions according to SAE ARP4761A.



Figure 2. Typical workflow for the FHA process.

Considering the steps in Figure 2, a custom "failure condition" stereotype is defined in the HADES modeling framework. It was created using the concept of domain-specific language (DSL) customization and UML profiling of CSM to serve the purpose of representing failure conditions under a safety aspect. Further, it includes the attributes mentioned in Table 1 to support conducting FHA considering MBSA.

Table 1. Failure condition stereotype attributes in the HADES modeling framework.

Attribute	Significance
Function	Function on which the failure condition is derived
System	Logical system that is considered from an abstract system model
Failure type	Failure modes considered for derivation as per AC23.1309-1E [16]
Phase	Operational phase or flight phase considered
Description	Elaborated description of the failure condition
Effect on aircraft	Failure effect of the FC on an aircraft system
Effect on crew	Effect of the FC on crew considering the operational phase
Effect on occupants	Effect of the FC on occupants considering the operational phase
Effect on a higher level	Failure effect of the FC on a higher level system
Hazard classification	Classification of the FC based on EASA CS25.1309 [17]

In HADES, each failure condition is treated as a separate model element. These elements can be populated either manually by a user or automatically through the OSIRIS tool. The generation includes the formulation of the failure condition utilizing the predefined

stereotype along with its attributes. This automated approach ensures better representation of failure conditions with respect to SAE ARP4761A.

The concept of regular expression parsing is used to extract failure condition data from the text generated by LLM in OSIRIS. As discussed in Section 2.1, the additional context provided required data, which includes instructions to the LLM to structure its response using regular expressions. These instructions result in the formulation of each failure condition in a regular expression with all the attribute data as mentioned in Table 1.

With the generation of failure conditions, OSIRIS also defines relations with the elements that were responsible for their formulation. This automated traceability between different elements of the abstract system model was developed considering MBSE principles and safety artifacts. This makes OSIRIS a standout tool for effectively integrating safety aspects into model-based engineering workflows.

The capability of connecting elements comes from the usage of a predefined stereotype with attributes from Table 1 to represent generated failure conditions by OSIRIS. Key attributes like system, function, and phase are used to link respective abstract system elements involved in the formulation of failure conditions. Furthermore, with existing relationships in the HADES modeling framework, these failure conditions can be utilized for the definition of next-level failure conditions and derived safety requirements.

3. Results

An abstract system model representing a top-level aircraft system with a defined function structure is considered for the evaluation of OSIRIS. The evaluation process is focused on identifying the tool's capability to produce failure conditions. The performance of the tool was determined with a survey analysis of the failure conditions and observations made from the functioning of the tool.

3.1. Evaluation of Failure Conditions

To better understand the plausibility of failure conditions provided by OSIRIS, a structured evaluation is performed with the support of system and safety experts in the aviation domain from the DLR community. The evaluation process included the data from the abstract system model and failure condition data generated by the LLM for each attribute. This data is rated with a score of agreement from strongly agree to strongly disagree.

These evaluation results were analyzed considering 3 aspects of the failure condition definition:

1. Failure condition statement;
2. Effect analysis;
3. Hazard classification (based on effects).

Figure 3 illustrates the distribution of agreement of these aspects in pie charts considering both variable and fixed contexts described in Section 2.1. Further, the following conclusions were drawn to evaluate the failure conditions generated by LLM:

- More than 50% of the information generated by the LLM was suitable for consideration in a failure condition.
- The fixed-context variant failure conditions have been identified with more valid failure condition statements than the variable-context.
- In the case of effect analysis, both versions provided similar kinds of results, whereas the fixed context had showcased a slightly better outcome.
- The hazard classification aspect has the highest disagreement percentages in both cases, with variable context implicating slightly less disagreement than the other.

https://doi.org/10.3390/engproc2026133134

The outcome of the survey analysis depicts that LLMs with text generation capabilities can be utilized for automating FHA documentation. Further, there is scope for improvements, which were further discussed in Section 4.

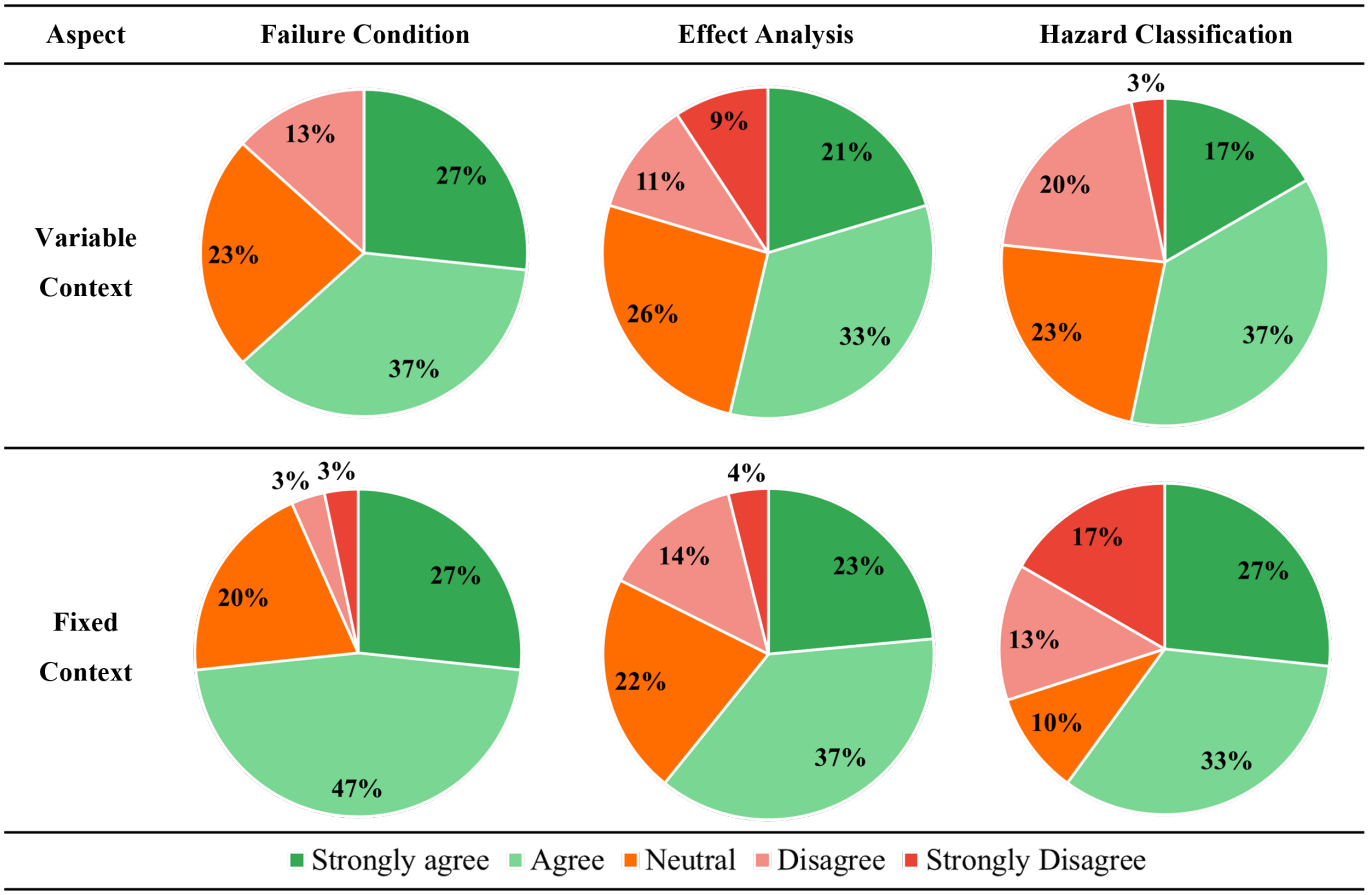


Figure 3. Overview of survey analysis of failure conditions, considering the failure condition, effect analysis and hazard classification.

3.2. Observations on OSIRIS Functionality

During the development of OSIRIS, a few key observations were made considering the interface establishment with the LLM and the quality of the failure conditions. Firstly, the developed interface for utilizing the LLM has the following insights that replicate the capability of the tool:

- The variable context addition, considering a customized data retrieval, has provided flexibility to include relevant key data, which is not a possibility in the fixed context addition option.
- The increase in size of contextual data has a negligible difference in the time required to generate failure conditions.
- With the increase in the number of failure conditions, the time taken by the model with limited computing power has increased proportionally.

Apart from these insights, a working demonstration was given to the members who took part in the survey mentioned in Section 3.1, followed by a review session. The feedback from the review session is further structured into pros and cons as illustrated in Table 2, to depict the support of OSIRIS for the FHA process in aircraft system development.

Table 2. Pros and cons of OSIRIS considering FHA.

Pros	Cons
• Provides initial reference to the failure condition formulation in case of FHA for aircraft systems. • A faster alternative to perform the FHA process considering the abstract system model. • Establishes a better and customized traceability between safety artifacts and the HADES system model. • OSIRIS in the HADES modeling framework reduces manual efforts required for safety assessments.	• Certain failure cases were not formulated for scenarios required by the user, due to limited data availability. • Around 30% of statements include indecisiveness because of the usage of a quantified LLM. • Nearly 50% of the hazard classifications with a text generation model did not meet the regulation criteria. • Even with some standard data, the LLM struggles to apply standard terminology to failure cases.

4. Future Work

The current version of OSIRIS has a limited data retrieval capability that restricts the contextual information provided to the LLM. An extended version with better contextual data retrieval and request formulation could improve the quality of generated failure conditions. Although most of the failure conditions formulated were compliant with the principles of the FHA, there were significant setbacks that were identified, as mentioned in Table 2. To address these limitations and to improve the tool's competency, targeted solutions with better software interfaces are to be explored.

The reliance on a quantized lower-parameter model with basic text generation capabilities has been a major constraint in terms of resource utilization and text formulation capabilities of the tool. Developing an efficient approach for the utilization of resources combined with aviation-specific, fine-tuned LLMs is expected to increase the validity of failure conditions and yield better outcomes.

Currently, OSIRIS supports the FHA process for aircraft system development. However, according to SAE ARP4761A, multiple safety assessments, including fault tree analysis (FTA), failure mode and effects analysis (FMEA), etc., are mandatory at different stages of aircraft system development. Therefore, future plans involve expanding OSIRIS' functionalities to accommodate these additional safety assessments, ensuring a more comprehensive approach to safety evaluation.

5. Conclusions

OSIRIS depicts a practical application of LLMs to automate safety assessments, with a focus on the failure case definition. Its current capability is limited to the derivation of failure conditions for the FHA process, according to SAE ARP4754B and ARP4761A. This work provides an overview of the employed methodology, as well as the results obtained through practical application. Though the discussed outcomes represent a promising prospect in the application of OSIRIS, there were a few necessary aspects that require attention. With proposed plans, the scalability and effectiveness of the tool are expected to be increased in supporting the system and safety engineers to develop efficient aircraft systems. Finally, this paper represents one of the existing possibilities of utilizing AI concepts in the system development domain, considering safety assessment in the evolving aviation industry.

https://doi.org/10.3390/engproc2026133134

Author Contributions: Conceptualization, D.S.S.K. and M.M.; methodology, D.S.S.K.; software, D.S.S.K.; validation, D.S.S.K. and M.M.; formal analysis, D.S.S.K.; investigation, D.S.S.K.; writing—original draft preparation, D.S.S.K.; writing—review and editing, D.S.S.K., M.M., S.S. and R.F.; visualization, D.S.S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Soleymani, M.; Mostafavi, V.; Hebert, M.; Kelouwani, S.; Boulon, L. Hydrogen propulsion systems for aircraft, a review on recent advances and ongoing challenges. *Int. J. Hydrogen Energy* **2024**, *91*, 137–171. [CrossRef]
2. European Union Aviation Safety Agency. *Proposed CM Ref. CM-21. A-004 Issue 01 on "Acceptable Approaches for the Certification of Electric/Hybrid Propulsion Systems"*; EASA: Cologne, Germany, 2024.
3. DeSalvo, P.; Fogarty, D. *Safety Issues and Shortcomings With Requirements Definition, Validation, and Verification Processes Final Report*; (DOT/FAA/TC-16/39); Federal Aviation Administration: Washington, DC, USA, 2016.
4. Perera, O.; Noteboom, C. Data integration with diverse data: Aerospace industry insights from a systematic literature review. *Issues Inf. Syst.* **2023**, *24*, 132–143.
5. Kritzinger, D. *Aircraft System Safety: Assessments for Initial Airworthiness Certification*; Woodhead Publishing: Cambridge, UK, 2016.
6. SAE International. *Guidelines for Development of Civil Aircraft and Systems*; (ARP4754, Rev.B); SAE International: Warrendale, PA, USA, 2023.
7. SAE International. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Aircraft, Systems and Equipment*; (ARP4761, Rev.A); SAE International: Warrendale, PA, USA, 2023.
8. Yang, M.; Wu, S.; Li, J.; Luo, C.; Hu, J. Research on an Integrated Modeling and Simulation Method for Small Satellite System. In Proceedings of the 2020 32nd Chinese Control and Decision Conference (CCDC), Hefei, China, 22–24 May 2020; pp. 5885–5890.
9. Sun, M.; Gautam, S.; Elks, C.; Fleming, C. Characterizing the Identity of Model-based Safety Assessment: A Systematic Analysis. *arXiv* **2022**, arXiv:2212.05401. [CrossRef]
10. Kuelper, N.; Jeyaraj, A.K.; Liscouët-Hanke, S.; Thielecke, F. Integration of a model-based systems engineering framework with safety assessment for early design phases: A case study for hydrogen-based aircraft fuel system architecting. *Results Eng.* **2025**, *25*, 104249. [CrossRef]
11. Lawless, W.F.; Mittu, R.; Sofge, D.A.; Shortell, T.; McDermott, T.A. Introduction to "Systems Engineering and Artificial Intelligence" and the Chapters. In *Systems Engineering and Artificial Intelligence*; Lawless, W.F., Mittu, R., Sofge, D.A., Shortell, T., McDermott, T.A., Eds.; Springer: Cham, Switzerland, 2021; pp. 1–22.
12. Chami, M.; Abdoun, N.; Bruel, J.-M. Artificial Intelligence Capabilities for Effective Model-Based Systems Engineering: A Vision Paper. *Incose Int. Symp.* **2022**, *32*, 1160–1174. [CrossRef]
13. Diemert, S.; Weber, J.H. Can Large Language Models assist in Hazard Analysis? *arXiv* **2023**, arXiv:2303.15473. [CrossRef]
14. Qi, Y.; Zhao, X.; Khastgir, S.; Huang, X. Safety Analysis in the Era of Large Language Models: A Case Study of STPA using ChatGPT. *Mach. Learn. Appl.* **2025**, *25*, 100622. [CrossRef]
15. Mischke, M.; Katabathula, D.S.S.; Melico, L.F.; Berres, A. HADES—A Framework for Hierarchical Architecture Design for Engineering Systems. In Proceedings of the 15th EASN International Conference on "Innovation in Aviation & Space Towards Sustainability Today & Tomorrow", Madrid, Spain, 14–17 October 2025.
16. Federal Aviation Administration. *System Safety Analysis and Assessment for Part 23 Airplanes*; (AC23.1309-1E, 2011-11); Federal Aviation Administration: Washington, DC, USA, 2011.
17. European Aviation Safety Agency. *Certification Specifications for Large Aeroplanes*; (CS-25.1309, 2023-01); European Union Aviation Safety Agency: Cologne, Germany, 2023.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.