Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P5055-cd

Mitigating Unsafe Control Actions in Autonomous Navigation Systems: A SysML-Based Analysis for Enhanced Safety

Raheleh Farokhi [a,b], raheleh.farokhi@aalto.fi
Sunil Basnet [a,b], sunil.basnet@aalto.fi
Osiris A. Valdez Banda [a,b], osiris.valdez.banda@aalto.fi
[a] Research group on Safe and Efficient Marine and Ship Systems, Marine and Arctic Technology, Department of Mechanical Engineering, Aalto University, Finland
[b] Kotka Maritime Research, Kotka, Finland

Abstracts- The increasing complexity of autonomous navigation systems (ANS) poses significant challenges to ensuring safety and reliability, particularly in dynamic and high-risk environments. This paper presents a SysML-based STPA methodology that enhances hazard analysis efficiency, traceability, and integration within system design. Unlike traditional approaches that rely solely on STPA control structures, this method replaces them with SysML diagrams, providing a more structured and dynamic representation of system interactions over time. Sequence Diagrams are used to explicitly depict control actions and feedback, improving the identification of unsafe control actions (UCAs) and their causal factors, such as software errors, communication failures, and human errors. Additionally, this approach explores loss scenarios, which have not been addressed in previous studies. The proposed methodology is applied to an ANS operating in winter conditions in the Baltic Sea. This integration of SysML and STPA offers a unified framework for system and safety engineering, reducing analysis time while improving scalability and applicability to complex autonomous systems.
Keywords: Systems Modeling Language, unsafe control actions, safety, risk

1. Introduction

The recent technological advancements with autonomous ships, such as the implementation of advanced navigation systems, dynamic route optimization algorithms, and real-time sensor integration, have significantly increased the complexity of maritime systems (Thombre et al., 2020). The growing number of software controllers, the shift in the role of human operators from direct control to supervisory tasks, and the increased reliance on automation have all led to higher interactions between hardware, software, and human operators, thus increasing the complexity (Veitch & Andreas Alsos, 2022). Consequently, ensuring the safety and reliability of these systems is challenging. Such systems require rigorous hazard analysis techniques to effectively identify and mitigate risks due to faults, failures, and unsafe component interactions (Abdulkhaleq et al., 2015).

Systems-Theoretic Process Analysis (STPA) has emerged as a powerful method for identifying and mitigating hazards in complex systems (Sulaman et al., 2019; Thieme et al., 2019). Instead of focusing on faults and failures like traditional hazard analysis methods, STPA focuses on component interactions and aims to identify scenarios where these interactions could be unsafe (N. Leveson and J. Thomas, 2018). Since the number of interactions is growing in complex systems, STPA has been increasingly adopted in this decade. Despite the strengths, the STPA process can be time-consuming, and building a control structure, which is a system representation of its components and interactions, is challenging for new systems (Basnet et al., 2023). Furthermore, the analysis is isolated as it is not conducted within the system engineering frameworks. Therefore, improving the integration of STPA with system engineering tools and methods can be beneficial and should be considered.

Model-based System Engineering languages like the Systems Modelling Language (SysML) have become standard tools for engineers to represent system information and have been widely used (Friedenthal et al., 2014). Like the STPA control structure, SysML diagrams represent system components, interconnections, and interactions. Block definition diagrams (BDD) show component hierarchies, activity diagrams detail component activities, and sequence diagrams capture interactions during activities. These similarities suggest that SysML diagrams can directly create or replace the STPA control structure. This integration reduces analysis time and supports STPA practitioners by streamlining

the process of building or substituting the control structure, enhancing the efficiency and effectiveness of hazard analysis.

This paper proposes a novel methodology that integrates SysML and STPA for hazard analysis by replacing the STPA control structure with relevant SysML diagrams. This approach aims to utilize the strengths of SysML for system modeling and STPA for hazard analysis, creating a unified process to integrate System engineering and Safety engineering discipline.

The rest of the paper is structured as follows: Section 2 provides background on SysML and STPA and presents the mapping of SysML diagrams to the STPA control structure. Section 3 presents related works and compares them to the study. Section 4 details the proposed methodology, describing integrating SysML diagrams within STPA analysis. Section 5 demonstrates the methodology through a case study, while Section 6 evaluates its effectiveness and discusses potential challenges. Finally, Section 7 concludes with key findings and directions for future work.

2. Related works

Several studies have integrated STPA and SysML to enhance safety analysis by leveraging the strengths of both methods. De Souza et al. (2020), Ahlbrecht et al. (2022), Li et al. (2023), and Basnet (2024) first applied STPA to systematically identify UCAs (UCAs) and establish safety constraints before modeling the system. This approach ensures that hazard identification is completed first, as STPA provides a structured method for analyzing interactions and potential risks within complex systems. After defining the UCAs, these studies used SysML to represent the system architecture, behaviors, and interactions, employing BDDs, Use-Case Diagrams, and State Machine Diagrams to integrate safety constraints into system design. In contrast, Ahlbrecht (2021) applied MBSE first, using BDDs and IBDs to define the logical system components, control structures, and physical interconnections before applying STPA to analyze safety risks and derive safety requirements that influenced architecture selection. In our study, we enhanced STPA modeling by incorporating Sequence Diagrams to explicitly represent control actions and feedback, allowing for a clearer visualization of dynamic system interactions and response sequences. Sequence Diagrams provide a temporal perspective, capturing the order of interactions, the timing of feedback loops, and the dependencies between system components. This helps identify hazards related to delayed, missing, or incorrect responses, which might not be as apparent in static representations. Additionally, Sequence Diagrams significantly improve the understanding of system failures by mapping UCAs within real-world operational scenarios, making it easier to comprehend how failures propagate through the system over time. Additionally, a key novelty of our approach is the inclusion of cause-of-loss analysis and loss scenarios, which were not previously explored in these studies. These additions improve traceability and depth in safety analysis, providing a more comprehensive understanding of hazard propagation and mitigation strategies.

3. Methodology background

3.1. STPA

STPA (Systems-Theoretic Process Analysis) is a hazard analysis method based on the System-Theoretic Accident Model and Processes (STAMP). It approaches safety as a dynamic control problem rather than focusing on failure prevention. Unlike traditional methods, STPA recognizes that hazards can arise from component failures and unsafe interactions between non-failing components (N. Leveson and J. Thomas, 2018).

The STPA methodology consists of the following steps: (1) Define the analysis's purpose, including losses, system-level hazards, and constraints. (2) Model the control structure: Develop a hierarchical system representation illustrating the components, control actions, and feedback. (3) Identify Unsafe Control Actions (UCAs): Analyse the control actions with guidelines to identify how they can be unsafe. (4) Identify Loss Scenarios: Determine the causes of the identified unsafe control actions.

3.2. SysML

Systems Modelling Language (SysML) is a graphical modeling language based on UML (Unified Modelling Language) specifically designed for systems engineering. It provides a standardized approach for capturing and analyzing complex systems, encompassing

requirements, architecture, behavior, parameters, and other key aspects. SysML enables rigorous model-based systems engineering practices, facilitating communication, analysis, and simulation throughout the system lifecycle (Holt & Perry, 2008).

SysML includes nine diagram types: Block definition diagrams (BBDs) define system components and properties, while internal block diagrams (IBDs) detail their structure and connections. Package diagrams organize other diagrams, and activity diagrams show activity flows. Furthermore, sequence diagrams depict message exchanges and state machine diagrams capture states and transitions. Requirements diagrams trace system requirements and use case diagrams to describe user-system interactions (Hause, 2006).

3.3. Mapping of SysML diagrams to STPA control structure

At a general level, the STPA control structure includes controllers (human, software, or hardware) that issue control actions, controlled processes (human, software, or hardware) that execute these actions, control actions (commands or functions sent by controllers), and feedback, where controlled processes inform controllers of their states and conditions.

Therefore, the controllers and controlled processes are the system components, while control actions and feedback represent the system interactions. In SysML, controllers, and controlled processes are depicted as blocks in BDD, parts in IBD, swim lanes in activity diagrams, and lifelines in sequence diagrams. Control actions appear as connectors and flow in IBD, messages in sequence diagrams, and flows in activity diagrams, while feedback is represented similarly to controlled processes.

4. Methodology

Fig. 1 This paper presents the methodology for integrating SysML diagrams into STPA methodology. The process is like STPA but replaces the control structure with SysML diagrams.

The details of the methodology are as follows:

Step 1: Define the scope of the hazard analysis.

This step defines the hazard analysis scope, including the losses and hazards related to the system under assessment.

At first, the high-level consequences, i.e., losses, are defined. The STPA handbook defines losses as “A loss involves something of value to stakeholders. Losses may include a loss of human life or human injury, property damage, environmental pollution, loss of mission, loss of reputation, loss or leak of sensitive information, or any other loss unacceptable to the stakeholders.” The losses are defined depending on the stakeholders and the system.

Next, each loss is identified with the hazards that can lead to losses in worse-case scenarios. The STPA handbook defines hazards as “a system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to a loss.”

Next, each loss is identified with the hazards that can lead to losses in worse-case scenarios. The STPA handbook defines hazards as “a system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to a loss.”

Depending on the scope and defined losses, the System-Level hazards are identified. Generally, it is advised to have less than 10 system-level hazards, and these should not include the causes, i.e., hardware failures or human errors. The hazard statement should include the system, unsafe condition, and their link to the losses.

The last step of scope definition defines constraints for each system-level hazard. STPA handbook defines a system-level constraint as “A system-level constraint specifies system conditions or behaviors that need to be satisfied to prevent hazards (and ultimately prevent losses).” The statement of the system-level constraint should include the system, the condition to enforce, and the link to the hazards.

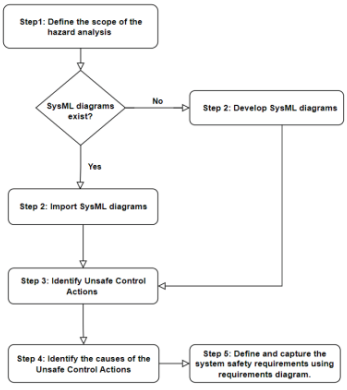

Fig. 1. The overall methodology to SysML-based STPA hazard analysis.

Step 2: Develop/Import SysML diagrams of the system

Depending on availability, relevant SysML diagrams should be developed or imported.

If sequence diagrams exist, they can be directly imported and assessed in the next step. If not, use-case diagrams representing high-level system functions should be developed first. Then, sequence diagrams showing interactions between components during activities should be created. BDD diagrams must be developed to represent component information. While these are core to the methodology, other diagrams like IBD, Parametric, and Requirements diagrams can also be integrated, provided they are available.

Step 3: Identify Unsafe Control Actions

The interactions in the sequence diagram representing a certain activity are then assessed to identify the unsafe control actions. As explained in section 2.3, the interactions moving to the right are equivalent to the control actions in STPA, and the interactions returning from the right are equivalent to the feedback in STPA. Therefore, each control action in the sequence diagram must be assessed with STPA guidewords. STPA consists of the following guidewords:

Not providing causing hazards: This guideword includes unsafe situations that could occur when the control action is not provided at all.

Providing causes hazards: This guideword considers unsafe situations that could occur when the control actions are provided in the wrong situations or are provided incorrectly.

Too early, too late, out of order: This guideword considers unsafe situations that could occur due to the timing of the control action, which includes the action provided early, late, or in the wrong order.

Stopped too soon, applied too long: Like the preview guideword, this refers to control actions provided for a duration shorter or longer than required, impacting the system's performance. Unlike previous guidewords focused on qualitative actions, this guideword addresses timing quantitatively, such as braking applied for "n" seconds. Parametric diagrams can be used to validate timing assumptions through calculations. After applying guidewords and identifying UCAs, these UCAs must be linked to related hazards from earlier steps to ensure traceability and maintain a clear connection between identified risks and their potential impacts.

Step 4: Identify the causes of the Unsafe Control Actions

Next, the UCAs should be accessed to identify their causes. The causes can be grouped into 4 categories:

Causes related to controller behavior: This step includes human errors, software bugs, and design issues. For this purpose, several SysML diagrams can be utilized. For example, an IBD can show how the controller is connected to other system components to see if there are issues or conflicts with other components, and all the activity diagrams involving the controller can be fetched to see if there is a conflict due to other activities of the same controller.

Causes due to unsafe feedback path: This category involves issues with sensors or transmission lines that transmit the sensor's information to the controller. Like previous categories, the diagrams that present the sensor's information and interactions, such as IBD, activity diagram, and state-transition diagram, can be used.

Causes related to the unsafe control path: This category involves issues with the actuator and the related transmission lines. If the actuators are physical components, then the issues such as component failures and design issues should be considered. For the transmission of the control, the transmission line, such as wires, wireless connectivity, etc, shall be considered.

Causes related to controlled process behavior: This category includes issues related to the controlled process, such as human errors and hardware issues, depending on the type of controlled process. Similar diagrams can be used here as specified in the first category. However, the diagrams should focus on controlled processes instead of controllers.

Step 5: Define and capture the system safety requirements using a requirement diagram.

In the final step, the safety requirements for the results of STPA analysis should be recorded using the requirements diagram. The diagram can include requirements for the different steps of STPA analysis, such as a diagram to mitigate or avoid the cause of UCAs. requirements to avoid or mitigate causes, requirements to avoid or mitigate System-Level Hazards, and henceforth, which then completes the SysML-STPA hazard analysis.

5. Results

This section presents the application of the proposed SysML-STPA methodology to analyze the safety of an Autonomous Navigation System (ANS) operating in the challenging winter conditions of the Baltic Sea. This function was chosen as it represents the most challenging aspect of a ship's transition to autonomy, given its strong dependence on human senses and decision-making (Chaal et al., 2020).

The ANS serves as the primary control system for navigation, encompassing situational awareness, collision and grounding avoidance, dynamic positioning, route planning, weather monitoring, and communication. By interacting with these subsystems, it manages the ship's speed and direction through the propulsion and motion control system (EMSA, 2022).

Step 1: Scope Definition

The potential losses identified for the ANS are shown in Table 1.

Table 1 The losses related to ANS

ID	Losses	ID	Losses
L1	Loss of life	L5	Loss of mission
L2	Injury to people	L6	Loss of cargo
L3	Loss of ship		
L4	Damage to ship		

In the next layer, Table 2 presents the system-level hazards that could lead to these losses in worst-case scenarios.

Table 2 Lists of System-level hazards leading to losses

ID	System-level hazards	Related losses
H1	Ship fails to detect and respond to environmental obstacles in time.	L1, L2, L3, L4, L5, L6
H2	Ship is unable to adapt or perform accurate route adjustments	L3, L4, L5, L6

To prevent these hazards and associated losses, the following system-level constraints are defined (Table 3).

Table 3 Lists of safety constraints for preventing system-level hazards

ID	Safety constraints	Related hazards
SC1	The ship must ensure continuous detection and timely response to environmental obstacles.	H1
SC2	The system must provide accurate and real-time route adjustments.	H2

Step 2: Develop/Import SysML diagrams of the system

The use case diagram, Fig. 2 illustrates the key relationships between the ANS and its external actors, including the port authority, remote operators or controllers, weather forecast systems, icebreaker vessels, and other nearby vessels.

In the next step, a sequence diagram, Fig. 3 is developed to show the interaction between ANS and the external actors during navigation planning and execution.

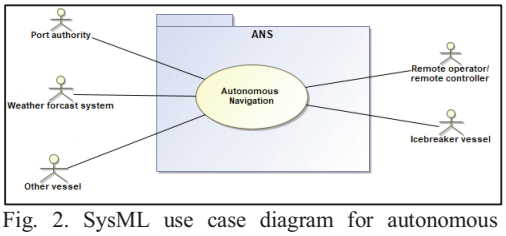

Fig. 2. SysML use case diagram for autonomous navigation in ice waters

As seen in the Fig. 2, the “Port Authority” provides navigation clearance and route

restrictions to the ANS. The system then requests critical weather and ice data from the "Weather Forecast System," which is provided in response. Simultaneously, the ANS initiates a scan of the surroundings using its "Sensors," receiving data on environmental obstacles. If required, the ANS requests "Icebreaking Assistance," and confirmation is received from the Icebreaker Vessel. Finally, the route is approved by the "Remote Operator/Controller", completing the interaction sequence. This diagram highlights the coordinated efforts between the ANS and external systems to ensure reliable and safe autonomous ship navigation, particularly under challenging conditions such as ice-infested waters in the Baltic Sea.

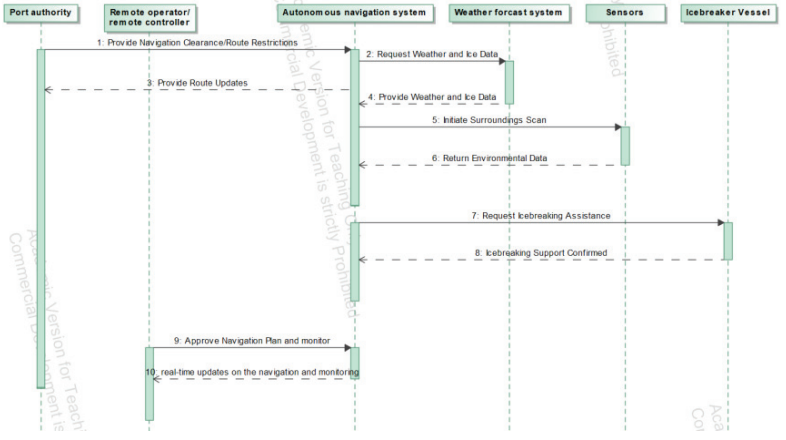

Fig. 3 SysML sequence diagram to show control actions and feedback in STPA Table 4 UCAs and

Step 3: Identifying Unsafe Control Actions

Table 4 presents unsafe control actions (UCAs) and related consequences.

Next, the scenarios (SC) leading to each UCA were identified. For example, Table 6 presents the scenarios leading to UCA-1.

Table 5 List of UCAs

Controller	ANS	RO
Control actions	Provide environmental data	Approve navigation plan and monitor
UCAs		
Not providing	UCA-1: ANS fails to request for the sensors to scan for data	UCA-5: RO fails to approve the navigation plan or monitor the situation
Providing causing hazard	UCA-2:ANS requests the sensors to scan for data during inappropriate conditions	UCA-6: RO approves or monitors wrong, inaccurate, or unclear navigation plan
Providing too early, late, or out of order	UCA-3: ANS requests the sensors to scan for data too late or in the wrong order	UCA-7: RO approves and monitors the navigation plan too early, too late, or out of order
Stopped too soon, applied too long	UCA-4: NA	UCA-8: RO approves and monitors too-long

Table 6 List of scenarios leading to UCA-1

Scenario ID	**Scenarios leading to UCA-1**
SC1	ANS fails to request for the sensors to scan for data due to software errors.
SC2	ANS fails to request for the sensors to scan for data due to control logic errors.
SC3	ANS fails to request to scan for data due to a Power supply failure in the sensors.
SC4	ANS fails to request for the sensors to scan for data due to Missing input or trigger.
SC5	ANS fails to request for the sensors to scan for data due to human errors.
SC6	ANS fails to request for the sensors to

	scan for data due to a lack of routine maintenance.
SC7	ANS fails to request for the sensors to scan for data due to communication failure.

Step 4: Identifying the causes (i.e. loss scenarios) of the UCAs

In this step, the causal factors leading to the UCAs are identified.

Table 7 illustrates the specific causes of UCA-1 and their category.

Table 7 The list of causal factors and corresponding category

Scenario ID	Causal factors	Category
SC1	Software errors	G-1
SC2	Control logic errors	G-1
SC3	Power supply failure	G-3
SC4	Missing input or trigger	G-1
SC5	Human errors	G-1
SC6	Lack of routine maintenance	G-4
SC7	Communication failure	G-2

Step 5: Define the system safety requirements

To address and mitigate, for example, UCA-1 to UCA-4 (ANS: Provide Environmental Data), the safety requirements are outlined in Table 8.

6. Discussion

Using SysML diagrams in the STPA process provided various advantages compared to the control structure. Firstly, instead of one diagram for all approaches with STPA control structure, SysML diagrams provide the same system information with different diagrams with unique properties.

Table 8 Safety Requirements to Mitigate UCA-1 to UCA-4

ID	Safety requirements
SR1	ANS must verify that sensors are operational before requesting scans.
SR2	ANS must request scans only under suitable environmental conditions.
SR3	ANS must trigger scans in the correct sequence and timing.
SR4	ANS must adjust scanning based on real-time feedback.

This feature can benefit complex systems with numerous components and interactions. For example, BDDs excel at defining system components, their attributes, and their interrelationships, providing a comprehensive overview of the system's static structure. On the other hand, requirements diagrams, a key feature of SysML, are invaluable for capturing and tracing system requirements, as well as design and implementation to align seamlessly with stakeholder expectations. Furthermore, the sequence diagram provides the interactions in a time sequence, further strengthening the understanding of how the system operates. The sequence of how different components interact cannot be understood from the STPA control structure. In STPA, unsafe control actions are analyzed using guidewords that directly correspond to the interactions visualized in a sequence diagram, making identifying and tracing unsafe scenarios easier. For example, a sequence diagram showing the timing of an environmental data request can reveal whether the control action is executed too late or skipped entirely. This direct mapping between the diagram and the STPA framework ensures a more comprehensive hazard analysis process.

In autonomous navigation, for instance, sequence diagrams can clearly show how the ANS interacts with external systems like sensors, remote operators, and weather forecast systems. This clarity helps identify specific points in the interaction sequence where errors may occur, enabling targeted safety interventions. Sequence diagrams make the analysis more intuitive and actionable, systematically addressing all potential unsafe control actions.

7. Conclusion

This study presents a methodology integrating SysML and STPA to enhance hazard analysis for complex systems, specifically focusing on autonomous navigation systems (ANS). The methodology streamlines the analysis process while maintaining rigor by replacing the traditional STPA control structure with SysML diagrams, such as sequence and activity diagrams. The dynamic representation offered by sequence diagrams proves particularly effective in identifying unsafe control actions,

highlighting their timing, sequence, and interactions. This integration unifies system and safety engineering, enhancing traceability, reducing analysis time, and ensuring comprehensive hazard identification. Its application in autonomous navigation in icy waters showcases its value as a robust framework for improving the safety and reliability of complex automated systems.

Acknowledgment

The authors gratefully acknowledge the financial provided by the Academy of Finland through the GYROSCOPE project and Merenkulun säätiö.

References

Abdulkhaleq, A., Wagner, S., & Leveson, N. (2015). A Comprehensive Safety Engineering Approach for Software-Intensive Systems Based on STPA. *Procedia Engineering*, *128*, 2-11. https://doi.org/https://doi.org/10.1016/j.proeng.2015.11.498

Ahlbrecht, A., Zaeske, W., & Durak, U. (2022). Model-based STPA: towards agile safety-guided design with formalization. 2022 IEEE International Symposium on Systems Engineering (ISSE),

Basnet, S., BahooToroody, A., Chaal, M., Lahtinen, J., Bolbot, V., & Valdez Banda, O. A. (2023). Risk analysis methodology using STPA-based Bayesian network- applied to remote pilotage operation. *Ocean Engineering*, *270*, 113569. https://doi.org/https://doi.org/10.1016/j.oceaneng.2022.113569

Chaal, M., Valdez Banda, O. A., Glomsrud, J. A., Basnet, S., Hirdaris, S., & Kujala, P. (2020). A framework to model the STPA hierarchical control structure of an autonomous ship. *Safety Science*, *132*, 104939. https://doi.org/https://doi.org/10.1016/j.ssci.2020.104939

De Souza, F. G. R., de Melo Bezerra, J., Hirata, C. M., de Saqui-Sannes, P., & Apvrille, L. (2020). Combining STPA with SysML modeling. 2020 IEEE international systems conference (sysCon),

Risk based assessment tools, European Maritime Safety Agency, PART 4, (2022). https://emsa.europa.eu/mass/rbat/download/7547/4908/23.html

Friedenthal, S., Moore, A., & Steiner, R. (2014). *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann.

Hause, M. (2006). The SysML modelling language. Fifteenth European systems engineering conference,

Holt, J., & Perry, S. (2008). *SysML for systems engineering* (Vol. 7). IET.

Li, Y., Zhao, Q., & Li, H. (2023). STPA-SysML Based Identification of Expected Functional Safety Hazards. 2023 2nd International Conference on Automation, Robotics and Computer Engineering (ICARCE),

N. Leveson and J. Thomas. (2018). STPA Handbook (no. Book, Whole). (in en), .

Sulaman, S. M., Beer, A., Felderer, M., & Höst, M. (2019). Comparison of the FMEA and STPA safety analysis methods–a case study. *Software quality journal*, *27*, 349-387.

Thieme, C. A., Guo, C., Utne, I. B., & Haugen, S. (2019). Preliminary hazard analysis of a small harbor passenger ferry–results, challenges and further work. Journal of Physics: Conference Series,

Thombre, S., Zhao, Z., Ramm-Schmidt, H., García, J. M. V., Malkamäki, T., Nikolskiy, S., Hammarberg, T., Nuortie, H., Bhuiyan, M. Z. H., & Särkkä, S. (2020). Sensors and AI techniques for situational awareness in autonomous ships: A review. *IEEE transactions on intelligent transportation systems*, *23*(1), 64-83.

Veitch, E., & Andreas Alsos, O. (2022). A systematic review of human-AI interaction in autonomous ship systems. *Safety Science*, *152*, 105778. https://doi.org/https://doi.org/10.1016/j.ssci.2022.105778