Aalto-yliopisto
Insinööritieteiden
korkeakoulu

Sunil Basnet

Review and comparison of the modeling approaches and risk analysis methods for complex ship system.

Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Technology

Espoo 07.09.2018
Supervisor: Professor Pentti Kujala
Advisor: D.Sc. Osiris Valdez Banda

Aalto University
School of Engineering

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Sunil Basnet

Title of thesis Review and comparison of the modeling approaches and risk analysis methods for complex ship system.

Degree programme Mechanical Engineering

Major/minor Marine Engineering

Thesis supervisor Prof. Pentti Kujala

Thesis advisor(s) D.Sc. Osiris Valdez Banda

Date 07.09.2018 **Number of pages** 61 **Language** English

Abstract

Marine industry is leaning towards autonomous vessels with companies such as Rolls-Royce and Kongsberg leading the development. However, this rapid technological change invites greater risks and responsibilities for marine professionals. Ship systems are getting more complex with time as the interactions between components are increasing and software are getting embedded. As a result, the nature of risks in modern systems can be different than in the traditional systems, where the risks were mostly limited to human errors and component failures. However, for identifying risks in modern complex systems, it is first important to understand the structural composition of the system, and the component's behavior, functions and interactions. Although, modern systems are quite different than traditional systems, traditional system-safety engineering techniques developed are still widely used.

This thesis aims to review a modern modeling approach known as Systems Modeling Language (SysML) and a risk analysis method known as Systems-Theoretical Process Analysis (STPA); and compare them against widely used traditional methods known as the Tree structure method and Fault Tree Analysis (FTA). SysML, developed in 2006, is a graphical modeling language which presents structural composition, component functions, behavior, constraints and requirements of a system. SysML aims to support the analysis, specification, design, verification and validation of complex systems. STPA, developed in 2011, is a risk analysis method which aims to identify and mitigate risks in a complex system. Unlike traditional methods such as Fault Tree analysis (FTA), STPA focuses on risks due to the unsafe control actions and component interactions. Furthermore, STPA can be also used during the early phases of the system development process to generate safety constraints and requirements for a safer design of the system.

This thesis also includes a workshop with Rolls-Royce where FTA, STPA, SysML and the Tree structure method were applied to a sample complex ship system. The results and feedback received from the workshop are presented and analyzed.

The results suggest that the modern methods such as SysML and STPA are more suitable than traditional methods for modeling and identifying risks in a complex ship system if the results of the method's implementation are considered. SysML presents several aspects of systems in a model which are missing in the Tree structure method, such as the requirements of a system, and behavior and interaction of components. Furthermore, it also provides a model that can be used as a tool for conducting an analysis of a system.

Similarly, STPA succeeds on identifying higher number of risks related to component interactions and human errors in comparison to FTA, as STPA analyzes all possible control actions in a system, whereas FTA only analyzes the risks that are known to the analysts. However, some drawbacks of SysML and STPA have also been identified. Although the methods are suitable for complex ship systems, the methods have higher degree of complexity and require more time for an analysis in comparison to traditional methods. Furthermore, some solutions to improve the identified drawbacks of SysML and STPA are proposed in this thesis. Finally, some viable future research topics to improve the research results are presented.

Keywords: Modeling approach, Risk analysis method, STPA, FTA, SysML Complex ship system, safety-engineering techniques, Marine risk and safety.

Acknowledgement

It is my great pleasure to thank everyone who have helped me throughout my graduating career.

First, I want to thank Aalto University for providing me with a wonderful opportunity to develop my academic career. Then, I would like to express my greatest gratitude for my supervisor, Prof. Pentti Kujala and advisor, DSc. Osiris Valdez Banda for trusting me with this research. This research wouldn't have been possible without their invaluable guidance and I am extremely honored and satisfied to be a part of their team. Next, I would like to thank DSc. Martin Bergstrom for his great support during the initial stage of this research.

Then, I specially thank Anna-Maria Blomster and Anssi Lappalainen from Rolls-Royce for their tremendous support. Thank you for realizing the necessity of this study and helping me to make the crucial decisions required to achieve the aim. Moreover, I am very grateful for all the experts from Rolls-Royce who took part in the workshop.

This research was funded by Design for Value (D4 Value) program, run by DIMECC. D4 Value program is partially funded by Tekes. I thank them for their financial support.

Furthermore, I am thankful to Svana Helen Björnsder and Christopher Brown from RM Studio team at Iceland for providing me the unreleased version of their software which helped me in this research.

Next, I would like to thank all my colleagues from the marine department of Aalto University, all my friends and relatives for their support.

Finally, I am grateful to my parents, Bal Bahadur Basnet and Sita Basnet, my brother, Sushil Basnet, and sister-in-law, Alina Joshi Basnet for always motivating me and being there whenever I needed.

Table of Contents

1 Introduction 1
1.1 Research background 1
1.2 Research objectives 2
1.3 Research limitations 2
1.4 Structure of the thesis 2
2 State of the art 3
2.1 Shipping in future: Autonomous Ships 3
2.2 Increasing complexity in the ship systems and necessity of suitable system-safety engineering techniques. 6
2.3 Past studies on implementation of modeling approaches and risk analysis methods for advanced systems 7
3 Methodology 9
4 Modeling approaches review 11
4.1 Introduction and selection of methods 11
4.2 Tree structure method 12
4.3 Systems Modeling Language (SysML) 13
4.3.1 Introduction 14
4.3.2 SysML diagrams 14
5 Risk Analysis methods review 22
5.1 Introduction and selection of methods. 22
5.2 Fault Trees analysis (FTA) 24
5.2.1 Introduction 24
5.2.2 FTA building blocks 25
5.2.3 Procedure 29
5.2.4 FTA example of a fault in an electric motor. 30
5.3 Systems-Theoretical process analysis (STPA) 33
5.3.1 Introduction 33
5.3.2 Procedure 33
6 Case Study 37
6.1 Modeling approaches 37
6.1.1 Tree structure method 37
6.1.2 Systems Modeling Language (SysML) 38
6.2 Risk analysis methods 44
6.2.1 Fault Trees Analysis 44
6.2.2 Systems-Theoretical Process Analysis (STPA) 45
6.3 Experts Evaluation and Feedback 49
6.3.1 Modeling Approaches 49
6.3.2 Risk Analysis Methods 51
7 Discussion and possible solutions 53
8 Research conclusions 57
9 Future research possibilities 60
10 Bibliography 62
APPENDIX A: The template of the code for creating block definition diagram of SysML in Graphviz software. 66
APPENDIX B: Code for creating block definition diagram for lower gear of azimuth thruster in Graphviz software. 67

List of Figures

Figure 1. An overview of advanced features considered in the MUNIN project for Autonomous vessel (MUNIN, 2014). 4
Figure 2. The current progress and future plans of Rolls-Royce for an autonomous ship. (Daffey, 2017) 5
Figure 3. A framework of this thesis. 10
Figure 4. Classification of PEASSS Nano-Satellite with the Tree structure method. 13
Figure 5. Classification of diagrams in SysML. 15
Figure 6. The block definition diagram in SysML for Air compressor context. 16
Figure 7. Layout of a block in internal block diagram. 16
Figure 8. An internal block diagram of an air compressor. 17
Figure 9. A requirement diagram for an air compressor. 18
Figure 10. An activity diagram for compressing air. 19
Figure 11. Parametric diagram for the flow analysis of Air compressor. 20
Figure 12. An example of a package diagram in SysML. 21
Figure 13. A symbol for a normal event used in FTA. 25
Figure 14. The FTA symbols for the failure events. 25
Figure 15. A Condition Event attached to an AND Gate. 26
Figure 16. A layout of an FTA diagram using an AND gate. 26
Figure 17. A layout of an FTA diagram using an OR gate. 27
Figure 18. A layout of an FTA diagram using a Priority AND gate. 27
Figure 19. A layout of an FTA diagram using an Exclusive OR Gate. 28
Figure 20. A layout of an FTA diagram using an Inhibit Gate. 28
Figure 21. Symbols used for Transfer Events. 29
Figure 22. Sample Layout of FTA with Transfer events. 29
Figure 23. First level of the Fault Tree for Motor overheating. 30
Figure 24. Second level of the Fault Tree for motor overheating. 31
Figure 25. A FT of an electric motor overheating. 32
Figure 26. Different Components in the In-Trail Procedure (ITP). 34
Figure 27. The control structure of the components in the ITP. 34
Figure 28. The control structure with the interactions among the controllers in the ITP. 35
Figure 29. A model of a lubrication unit in the azimuth thruster using the Tree structure method. 37
Figure 30. The package diagram in SysML for lower gear. 38
Figure 31. The requirement diagram of the lower gear in the azimuth thruster. 39
Figure 32. The block definition diagram of the lower gear in the azimuth thruster. 40
Figure 33. The internal block diagram of the lower gear in the azimuth thruster. 41
Figure 34. The activity diagram for lower gear lubrication in the azimuth thruster. 42
Figure 35. Parametric diagram for bending stress analysis of bevel gear in the azimuth thruster. 43
Figure 36. Fault Tree of the steering hydraulics unit failing in the azimuth thruster. 44
Figure 37. Relationship between hazards and accident scenarios. 46
Figure 38. A safety control structure of the azimuth thruster. 47
Figure 39. A comparison of the models from graphical interface with Astah SysML (Top) and with Code using Graphviz (Bottom). 55

List of Tables

Table 1. General comparison between OPM and SysML. 12
Table 2. Symbols used in Activity diagram. 18
Table 3. Comparison of FTA, FMEA and HAZOP. 23
Table 4. Identifying unsafe control actions for the hazard loss of Minimum separation for ITP (Leveson, 2015) 36
Table 5. The list of accidents related to the azimuth thruster and their description. 45
Table 6. The list of identified accidents and hazards. 45
Table 7. List of identified hazards and their description. 46
Table 8. Identified safety constraints for hazards. 46
Table 9. Identified unsafe control actions. 48
Table 10. The safety constraints for the unsafe control actions. 48
Table 11. Scale and color codes used for the evaluation of methods from experts. 49
Table 12. The expert's evaluation of the Tree structure method. 49
Table 13. The expert's evaluations of Systems Modeling Language. 50
Table 14. Comparison of the expert's evaluation between the Tree structure method and SysML. 50
Table 15. The expert's evaluations of FTA. 51
Table 16. The expert's Evaluation of STPA. 51
Table 17. Average ratings in different criteria for FTA and STPA. 52
Table 18. Scale used in conclusion tables. 57
Table 19. An overall research conclusion for the Tree structure method and SysML. 57
Table 20. An overall research conclusion for the FTA and STPA. 58

List of Abbreviations

AAWA	Advanced Autonomous Waterborne Applications
ATC	Air Traffic Control
BE	Basic Events
CE	Conditional Events
DNV-GL	DNV and Germanischer Lloyd
ETA	Event Tree Analysis
FMEA	Failure Modes and Effects analysis
FMECA	Failure Modes and Effects Criticality Analysis
FRAM	Functional Resonance Analysis Method
FSA	Formal Safety Assessment
FT	Fault Tree
FTA	Fault Tree Analysis
HAZOP	Hazard and Operability Study
IMO	International Maritime Organization
INCOSE	International Council on Systems Engineering
ITP	In-Trail Procedure
LIDAR	Light Detection and Ranging
MDUSV	Medium Displacement Unmanned Surface Vessel
MUNIN	Maritime Unmanned Navigation through Intelligence in Networks
NAPA	Naval Architectural Package (Software Package)
NE	Node Events
OMG	Object Management Group
OPM	Object-Process Methodology
RFP	Request for Proposal
ROC	Remote Operations Center
SC	Safety Constraint
STPA	System's-Theoretical Process Analysis
SysML	Systems Modeling Language
TCAS	Traffic Collision Avoidance System
TE	Transfer Events
UCA	Unsafe Control Action
UML	Unified Modeling Language
VTT	Valtion Teknologian Tutkimuskeskus

1 Introduction

1.1 Research background

Traditional methods[1] are still widely used to analyze risks in the safety-critical ship systems. These methods were effective at past, because of their ability to analyze the system by isolating and simplifying the interfaces between system components (Leveson, et al., 2010). These approaches consider that the failure of a system as primarily resulting from a fault in a mechanical component or due to an operator error. The systems in previous decades consisted of mechanical components which could be treated independent components because of low interactions among them. As a result, the traditional approaches were a simple, effective and reliable way to analyze a ship system of past. However, the scenario is changing, as the technology is advancing at a faster pace. Projects for autonomous ships have already started, thus ship systems are becoming more advanced and complex. Unlike traditional systems, components in modern systems cannot be treated independent, as the interactions among components are increasing and software are getting embedded in components and subsystems. Because of these changes, it is unsurprising that the nature of accidents is also changing. (Leveson, et al., 2010)

There have been several examples of accidents in recent decades in other fields such as aviation, chemical and nuclear industry which occurred because of increasing component interactions and software issues and wasn't only limited to operator errors and component failures. For example, Mars polar lander crashed on 03 December 1999 because the engine of the spacecraft stopped before landing on the surface. The main reason identified for this accident was due to the software misinterpreting the noise signal generated from the deployment of landing gears as the noise signal of a surface touchdown. Hence, it stopped the engines prematurely (Board, 2000). The software was designed and supposed to detect the noise signal and carryout the action to stop the engines afterwards. The software and components didn't fail as they did their designed tasks. However, the accident resulted due to an unidentified unsafe component interaction as the software misinterpreted noise from landing gears as noise from surface touchdown. Similarly, other examples of accidents that are not related to only component failures or operator errors are presented on Mukhopadhyay et al. (2014), Flugunfalluntersuchung (2004) and Leveson (2011). Hence, the issues due to component interactions and design errors must also be assessed when identifying risks in future ship systems. However, traditional methods may not identify these risks as they were developed for the relatively simpler systems of the past with less interaction among components. Furthermore, the structure of a complex system model using these approaches often appears to be disrupted. Hence, a systemic and systematic risk analysis method is required that can also identify the new emerging failures resulting from the technological changes in a complex system.

However, for identifying risks in a complex system, it is first important to understand the system itself. Since, the interactions among components are growing, understanding how the component interacts to perform activities or functions is also crucial. Furthermore, the

[1] Here, the term "Traditional Approaches" refers to the approaches that were introduced more than 30 years ago and are still widely used, such as Fault Trees and success trees.

analysts must understand how components are interconnected inside the system for risk analysis. After understanding the system better, the risk analysis methods will then be more effective. Moreover, these models can help operators to operate the system efficiently; and allows designers or analysts to understand the system for improving the future system designs. In addition, models can also be used to guide the design process through an analysis of requirements and behavior of a system. Thus, a modeling approach which is suitable for a complex system is as crucial as the risk analysis method.

1.2 Research objectives

The aim of this research is to identify a suitable modeling approach and a risk analysis method for a complex ship system. This thesis should aim to answer following research questions:

1. Which approach is suitable for modeling a complex ship system?
2. Which method is suitable at identifying risks in a complex ship system?
3. Can the risk analysis method identify emergent failures such as component interaction issues?

1.3 Research limitations

As the scope of this research is wide, following limitations have been considered:

1. Only two modeling approaches and two risk analysis methods are selected for review and comparison.
2. For limiting the scope, following simplifications were made on the methods and the sample complex ship system.
    a. A simplified version of SysML labelled as SysML-lite has been used for the review in this thesis. (more information on Chapter 4.3)
    b. Deriving cut sets for failures in FTA has not been considered in the review.
    c. In the workshop, the methods were only applied to some parts of a sample system due to time restrictions.
3. Due to the lack of data about the failure in complex ship systems, probabilistic methods are not considered.

1.4 Structure of the thesis

The state of the art is presented in Chapter 2 of this thesis which includes a summary of some ongoing advance ship projects, the necessity for finding suitable methods for the modeling and risk analysis of the complex ship systems, and previous studies on implementing the methods on complex systems. Next, Chapter 3 of this thesis presents the methodology used to achieve the aim of this research. Chapter 4 and Chapter 5 then present the review of modeling approaches and risk analysis methods respectively.

Next, in Chapter 6, a case study with Rolls-Royce is presented where these methods are implemented into a sample complex ship system. This chapter also includes the results, comparison, expert's evaluation and feedback from the workshop. Then, Chapter 7 presents the discussions related to the advantages and disadvantages of each method and the comparison. Furthermore, some possible solutions to improve the drawback of the methods are also presented in this chapter. Finally, Chapters 8 and Chapter 9 present the research conclusions and future possibilities respectively.

2 State of the art

2.1 Shipping in future: Autonomous Ships

Autonomous vehicles are already the state of the art for roadways and airways; some autonomous functions such as an auto-pilot and auto-parking are already available, even in some consumer vehicles. However, autonomous ships are still in the early phase of development. Nevertheless, in the past decade, companies such as Rolls-Royce, Kongsberg and Vigor Industrial have initiated projects to construct autonomous ships, which are set to mark the beginning of a new era of shipping in the near future. Moreover, an EU research project, MUNIN, to develop a concept for an unmanned dry bulk carrier was completed in August 2015. Autonomous vessels have attracted several marine professionals and companies, as they have the potential to improve the sustainability of the marine transport industry by reducing environmental impacts, operational expenses, and the shortage of seagoing professionals (MUNIN, 2016).

The MUNIN project report presents the concept and feasibility assessment of a dry bulk carrier, which will completely be unmanned at least for parts of the voyage. This research report presents an analysis of economic, technical, and legal feasibility of the vessel for a deep-sea voyage (MUNIN, 2016). The main aim of MUNIN is to develop advanced navigation, which has innovative features like an advanced sensor module for automated look-outs (such as object detection and weather observation), an autonomous navigation system for ship operation and decision-making (such as satisfying the COLREG's requirement on avoiding collisions and maintaining the stability of a vessel in harsh weather), and a shore control center that features the safe human supervision and control of the vessel from offshore (MUNIN, 2014). An overview of these features concept is shown in Figure 1. The report shows that the vessel can be commercially viable if newly built rather than modifying an existing ship. Innovative design changes and the reduction of human spaces should result in the reduction of fuel consumption and emissions, which will increase the profitability of shipping companies if the current challenge of autonomous heavy fuel oil operation is solved. Besides cost reduction, the research estimates that the safety of the unmanned vessel would also be improved, as human errors are considered the main cause of most maritime accidents. As mentioned in the report, the risk of collision and foundering for a MUNIN concept vessel would be around 10 times less than manned vessels. Better redundancies of the critical systems and the lack of hostages for pirate attacks would also further improve the safety of the vessel. However, the vessel must have an advanced software system to defend against digital attacks. (MUNIN, 2016)

Figure 1. An overview of advanced features considered in the MUNIN project for Autonomous vessel (MUNIN, 2014).

A renowned company, Rolls-Royce, is also aiming to build an autonomous unmanned ship for the future. As developing the new insurance policies for an unmanned vessel, changing the regulations regarding watchkeeping on bridge and changing the legislation regarding manning-on-board requires longer period of time, their plan is to build a remote-controlled local vessel by 2020, a remote-controlled and autonomous short sea vessel by 2025, a remote-controlled and an autonomous ocean-going vessel by 2030. To achieve this aim, a research project, Advanced Autonomous Waterborne Applications (AAWA), was launched in 2015 by Rolls-Royce with partner companies and institutes such as Delta Marin, Inmarsat, DNV GL, NAPA, Aalto University, VTT, the University of Turku, the Tampere University of Technology, and Åbo Akademi. According to the report of the first phase of AAWA, the technologies required to build and operate a remote-controlled and autonomous ship already exist. However, the optimum way to manufacture a reliable and cost-effective autonomous ship has yet to be found, which is the main aim of the AAWA project. In the first phase, the project explored technologies such as sensor fusion, control algorithms, communications and connectivity for safe navigation and the collision avoidance of vessels. Although the vessels developed by Rolls-Royce will eventually be fully autonomous, they will still require human input from land. Therefore, Remote Operations Center (ROC) will be built in various places for connecting, communicating with, and controlling the vessel. The AAWA project also discusses the safety, security and legality of autonomous ships, the plan being to make autonomous vessels at least as safe as traditional vessels to secure regulatory approval. The upcoming phases of this project will aim to develop the required technologies discussed in the first phase, identify the new emerging risks, explore legal challenges and propose appropriate rule changes at the IMO, and establish cost and revenue models for different autonomous ship types. (Levander, 2016) (Rolls-Royce, 2016) (AAWA, 2016)

In 2016, Rolls-Royce and Svitzer successfully completed Project SISU, which showcased the Svitzer Hermod, the world's first remotely operated commercial vessel. Several of its maneuvers were demonstrated in Copenhagen, Denmark (Daffey, 2017). This project strengthened the technical feasibility of an autonomous ship further. Furthermore, Rolls-Royce has also developed automatic crossing systems and are implementing them in some of Fjord1's ferries (Rolls-Royce, 2016). As battery-powered ferries have strict limits on energy consumption, these automatic crossing systems can ensure safe and energy-efficient controls by automatically adjusting the acceleration, deceleration, velocity and trajectory of the vessel. Furthermore, Rolls-Royce has collaborated with Mitsui O.S.K. lines for developing an intelligent awareness system. (Rolls-Royce, 2017). This feature will provide crew with better understanding of the vessel's surrounding. As a result, the crew can operate safely and more efficiently. Figure 2 shows the progress of Rolls-Royce and their future plans for an autonomous ship.

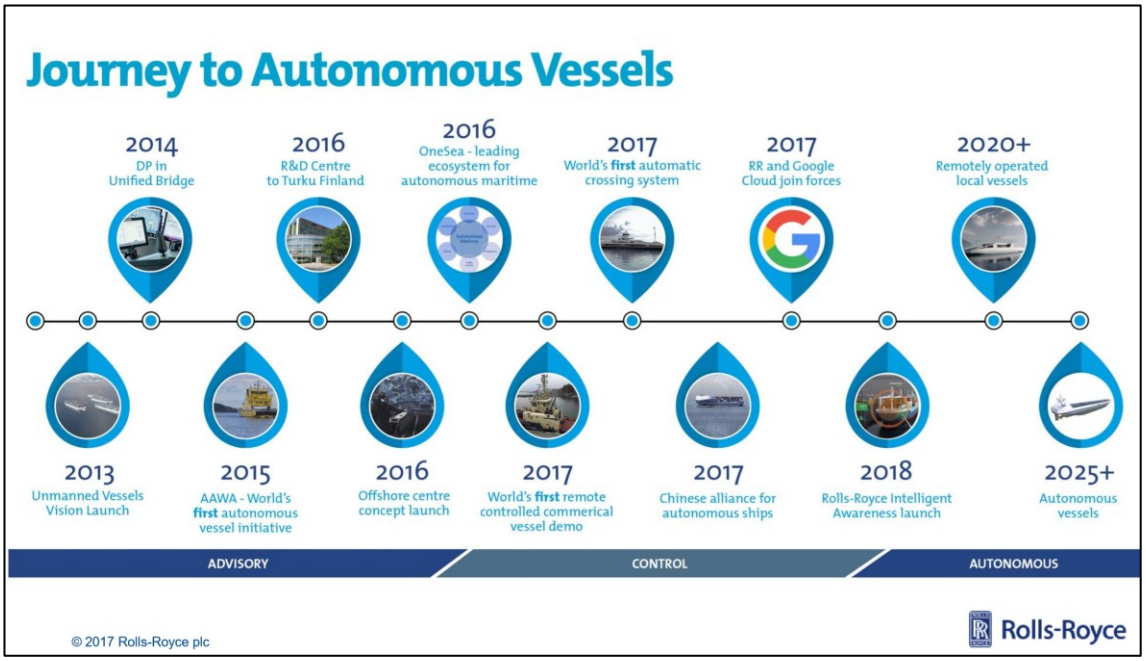


Figure 2. The current progress and future plans of Rolls-Royce for an autonomous ship. (Daffey, 2017)

Another project to build an autonomous vessel, the YARA Birkeland, by Kongsberg started in 2017. The YARA Birkeland will be the world's first fully electric autonomous container ship with zero emissions; and is set to operate between three ports, Herøya, Brevik and Larvik in southern Norway. Kongsberg has now developed the required technologies; and the vessel will begin operational tests in 2018. A small container-based bridge, which will contain a captain and a small crew, will be attached during the first year of testing. After a successful testing phase, the bridge will be removed, and then the tests for autonomous capability will be conducted in 2019. The vessel is estimated to operate completely unmanned by 2020. (Kongsberg, n.d.)

Several other projects such as Autonomous Ship Transport at Trondheimsfjorden (ASTAT-research project for operating small and electric unmanned ship), Milli-Ampere (Autonomous passenger ferry) have also started in this decade. Moreover, A Medium Displacement Unmanned Surface vessel (MDUSV), Sea Hunter, was built by vigor industries in 2016. It is designed to hunt submarines and will be operated by the U.S.

Navy. After passing initial testing in January 2016 and sea trials in January 2017, the MDUSV is now set to begin operating soon (Barton, et al., 2017). Furthermore, some autonomous underwater research and military vessels are already in use.

2.2 Increasing complexity in the ship systems and necessity of suitable system-safety engineering techniques.

A system is a complex combination of resources (such as materials, hardware, software, information and human) integrated to perform a specific task or series of tasks (Blanchard & Blyer, 2016). New discoveries, inventions and technological changes are further increasing the complexity of systems. This heightened complexity may also possess and create new risks and threats.

Perrow (1999), for example, argues that in complex systems, accidents are inevitable or even "normal." Complex systems have a higher accident rate, as there are always new emergent risks and potential interactions that may not be planned, understood, anticipated and avoided (Leveson, et al., 2009). In addition to their complexity, such systems also contain a higher risk of failure, because increasing market competition has led to limited resources and decreasing safety margins. Moreover, the environment where these systems operate changes with time. These changes further lead to small modifications or compromises which might be ignored, as each change can be considered a small deviation from the previously accepted condition. As complex systems are highly dependent on their initial conditions or designed conditions, these small modifications or decisions can have a catastrophic effect later. (Dekker, 2011). Unlike road transport, a marine accident usually involves significantly higher losses. Apart from the manufacturing cost of the vessel, a marine accident generally includes the loss of lives and cargo. For example, more than 4000 people died when the vessel MV Dona Paz sank in 1987 (Perez, et al., 2011). Hence, risk analysis must always be prioritized in the marine industry.

The main factors attributed to the cause of marine accidents are human errors, component failures and unsafe component interactions. Although human operators are blamed for most accidents, the majority of these errors tend to occur due to the implementation of technologies, working environment, management and organizational factors. The main aim of increasing automation in ship systems is to reduce human errors. Although, it may be successful at reducing operator errors, there is a chance of increased design errors as the systems are more advanced and complex. In addition to design errors, it can also lead to new emergent errors due to software and component interaction. It is necessary to find a suitable modeling approach for marine systems that can identify majority of the possible risks for system safety. (Rothblum, 2000) (Leveson, et al., 2010)

With the further advancement of systems, changes in safety engineering techniques are also essential. However, system-safety engineering techniques lag far behind rapid technological changes. Thus, traditional methods are still widely used to model and identify risks in modern systems. Although several attempts have been made to modify and adapt these approaches for modern systems, they are still limited due to their method foundation. As there are large dissimilarities between traditional systems and modern systems, designing a better system by learning from past accidents might no longer be effective. Furthermore, the trial and testing period of systems is decreasing, because of the competitive market. As a result, it is no longer a possibility to understand all the potential risks of a system during the testing period. Moreover, interactions between the

components in a system are increasing; and there is insufficient communication between the machines and operator. Hence, current and future marine industry possess increased risks, unless suitable safety engineering techniques for ship systems can be identified and are applied. (Leveson, 2011)

Safety in the maritime industry has always been a priority for International safety community. In 2002, the International Maritime Organization (IMO) approved a Formal Safety Assessment (FSA), which can be used as a tool for the rule-making process and to evaluate regulations for marine vessels. The FSA has been described as "a rational and systematic process for assessing the risks associated with shipping activity and for evaluating the costs and benefits of IMO's options for reducing these risks." The aim of the FSA is to improve maritime safety and consists of five steps: 1) Hazard identification 2) Risk assessment 3) Risk control options 4) Cost benefit assessment 5) Recommendations for decision making (Smita, 2016). Moreover, International safety management code (paragraph 1.2.2) states that "Safety management objectives of the company should…. Assess all identified risks to its ships, personnel and the environment and establish appropriate safeguards…". However, International safety management code doesn't specify the standard assessment or an analysis method to be used. Different analysis methods lead to different end-result. Moreover, a specific modeling approach to be used for marine systems is not defined either. Each of the modeling methods provides a different graphical model of a system; and this model is important to understand the structure, processes, connections between the components and overall reliability of a system. It is unlikely that a company would model or evaluate the risks of a system using multiple methods, as implementing any of the methods is highly time consuming and costly. Hence, a comparison study of modeling approaches and risks analysis methods for advanced ship systems might be helpful in future marine projects.

2.3 Past studies on implementation of modeling approaches and risk analysis methods for advanced systems

Several traditional methods such as the Tree structure method, Failure Modes and Effect Analysis (FMEA), Hazard and Operability study (HAZOP) and Fault Tree Analysis (FTA) are still widely used to model and identify risks in modern advanced systems. Some modern methods like Systems Modeling Language (SysML), Object Process Methodology (OPM), Systems Theoretical Process Analysis (STPA) has been developed in recent decades, with the aim of surpassing the limitations of traditional approaches, providing more systematic way for analysis, and identifying and mitigating more types of risks than traditional approaches. Several attempts to model or analyze risks in advanced systems using these modern approaches have been made in other industries. However, less research has been conducted to study their implementation to advanced ship systems.

In a review of FTA modeling approach, Lee et al (1985) suggested that the FMEA approach is a cost effective and easier to implement than FTA. Hence, FMEA should be used to analyze small systems when a single-point failure analysis is sufficient. However, if the systems are more complex and are safety-critical, then FTA is more appropriate. In their opinion, FTA can be the simplest analytic reliability tool for a system as it provides an effective visualization tool for evaluating the overall reliability of the system. Yet, it can also be the most detailed if required.

Similarly, A comparative critical study between the FMEA and FTA had been provided by G Cristea and DM Constantinescu (Cristea & Constantinescu, 2017). The report mentions that it is uncommon in industry to model systems in higher detail using these methods, because the methodology is much time consuming. As a result, all possible failure modes may not be identified during analysis. According to them, the FMEA method provides a spreadsheet of all possible potential failure modes and their effects on a system while FTA allows a detailed analysis of failures in a top-down approach. Applying both methods to a system can provide detailed information about systems failure than only applying one of them. Each of the methods allows an analyst for the clear observation and investigation of the processes of systems from a different point of view. Hence, they have suggested a mixed approach that combines FTA and FMEA, where FMEA is guided by FTA. They have mentioned that there are also some shortcomings of the FTA and FMEA. They only allow a safety analysis between two levels: system level and component level. Furthermore, the importance of failure analysis at a higher level of detail is often not considered because of the time which a detailed analysis requires. Also, these methods do not guide an analyst to prioritize the most critical elements of the system which will affect the overall efficiency of the analysis as similar time is spent in analyzing both the most and the least critical element of the system.

S. Ahvenjärvi (2001) analyzed the application of FMEA in the automation systems of ships. The analysis was carried out by studying experiences of experts from FMEA projects during 90's, on the ship systems in Finland. Based on their experiences, the effect of FMEA in the safety of automated systems of a ship were discussed. The author reports that FMEA can identify the weak points of the safety-critical systems. In addition to the identification of risks, FMEA analysis of a system can also help the analysts to get more knowledge about the system and its components. Furthermore, the author mentions that this method can also be applied during the preliminary stage of a system design through the information from a design documentation, which can prevent some design errors beforehand instead of mitigating these errors later. However, he also mentions that, for an integrated system with software embedded controls, the identification of possible failure modes and their consequences can be difficult with FMEA.

F. Mushtaq et al (2000) has reported by applying a systematic HAZOP procedure to the pipeless plants. A pipeless plant fulfils a basic idea to move the process vessel between fixed stations for mixing, separation and other activities. The HAZOP procedures for batch processes were further developed and used for this research. In the report, they have demonstrated HAZOP methodology which breaks down a complex system design into smaller sections to fulfil the purpose of analyzing system safety in a strict sequential way. However, they found that implementation of this methodology, like any other systematic safety method, required a long period of time. They believe that computer support tool to guide and document some procedures of HAZOP would be useful for the analyst.

O. A. Valdez Banda et al (2018) has presented a systematic hazard analysis process for an autonomous vessel using the STPA method. The study aimed to present a hazard analysis for the earliest design phase of an autonomous vessel and has covered the operational context. As a result of the analysis, the safety controls or guidelines are presented in the report, which then can be followed during the planning of ship design, materials, structures, components, systems, and the services. Following these guidelines, should eliminate the hazards or risks identified during the analysis. The analysis

identified ten accidents and fifteen hazards for an autonomous vessel; and in an attempt to mitigate these hazards, 75 safety controls are presented in the report. The report concludes that the implementation of this process can guide the initial design process of autonomous vessels which makes the vessel design more efficient and safer.

Similarly, A. Abdulkhaleq et al (2013) has reported their experience of applying STPA to software-intensive systems in the automotive domain. In this study, they applied STPA to adaptive cruise control (ACC), which is an automotive feature that controls the speed of vehicle according to the traffic environment. During this study experience, they found that STPA is more powerful, systematic and useful to evaluate safety-critical systems in an automotive domain as it identifies many accident scenarios, such as software issues, design errors, component interactions accidents and human decision errors. Furthermore, they realized some potential design improvement for an ACC system. However, they found the process of identifying the unsafe control actions much time consuming; and in addition, the process is unclear for the system that has interference between multiple controllers.

E. Herzog et al (2005) prepared an assessment of SysML. The aim of this research was to evaluate the strengths and weaknesses, and to identify the possibilities for future improvements. In their opinion through the research, SysML allows system engineers to model system design in a traditional manner. Since, this method was derived from UML which is used in software engineering, they expect that the use of this method will help to improve the interactions between software engineers and system engineers. However, they have also identified some issues against the modeling language. The report mentioned that SysML doesn't provide any support for representing produced models. Furthermore, there is no built-in mechanism for capturing the configuration of a system being verified. Although some issues have been identified against SysML, they conclude that none of them can be considered as major issue.

3 Methodology

This section describes the methods used to achieve the aim of this thesis described in Chapter 1.2. The methods used are a) study of the art research b) detailed reviews of the modeling approaches and risk analysis methods, and c) case study with Rolls-Royce. The framework of this thesis is presented in Figure 3.

This thesis can be categorized in two different sub-topics: Modeling approaches and Risk analysis methods. After reviewing literatures, careful research and expert's suggestions, two modeling approaches and two risk analysis methods are selected among several available methods for detailed review. Out of two methods, one method is selected from widely used traditional methods, while another is selected from modern methods developed within 1-2 decades. The methods selected for the review of modeling approaches are the Tree structure method and SysML; and for the risk analysis, the methods selected are FTA and STPA. These methods are presented in detail in Chapter 4 and Chapter 5.

As the review alone is insufficient for fulfilling the aim of finding a suitable modeling approach and a risk analysis method, hence a case study with Rolls-Royce was organized for assessing the advantages of implementing these methods in a sample complex ship

system. In this workshop, the methods were briefly discussed and implemented in a sample complex system. Computer tools were then used to prepare the graphical models for modeling approaches and for the risk analysis. The results were then discussed, methods were evaluated, and then the feedback were collected from the experts in the workshop.

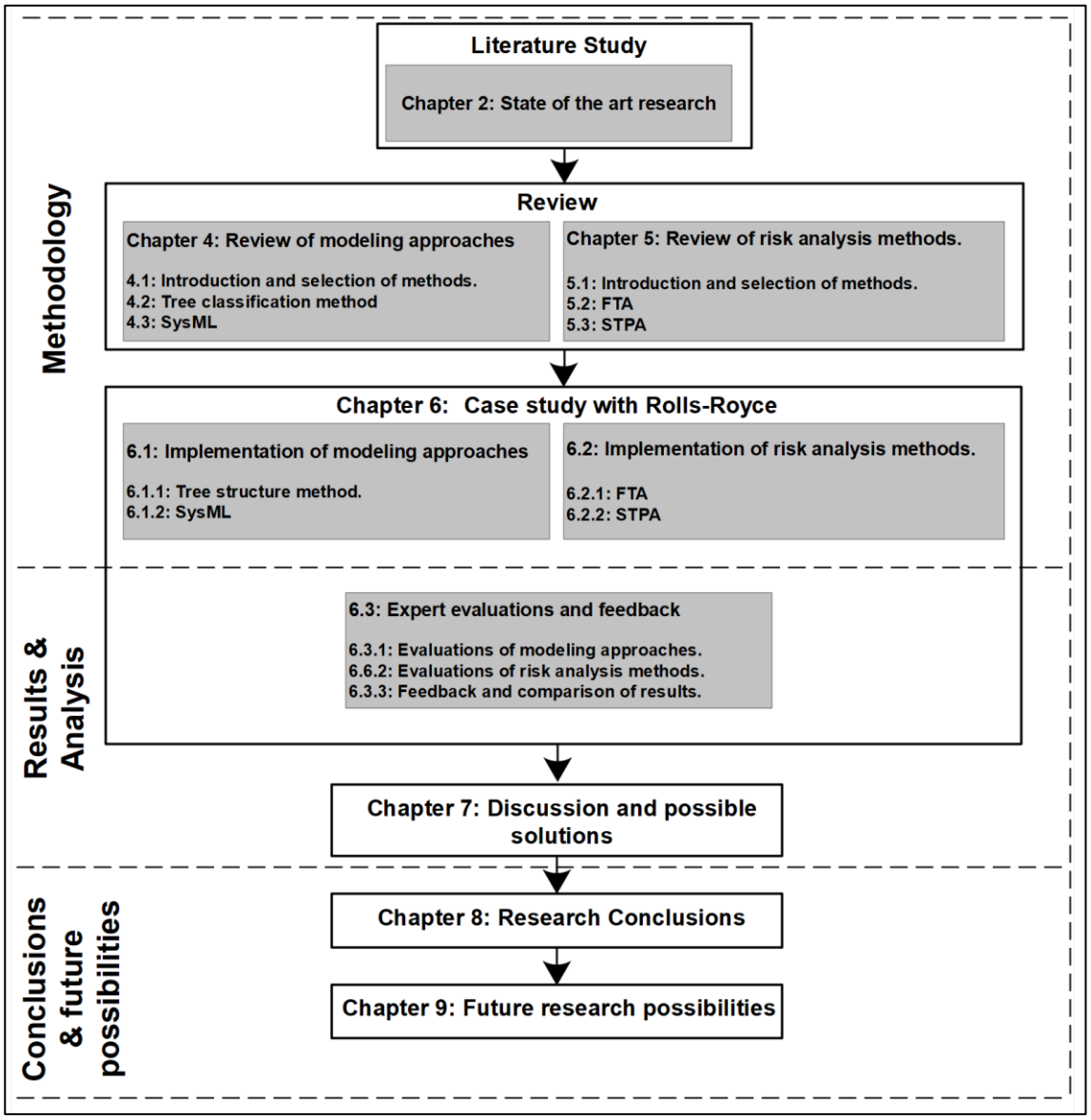


Figure 3. A framework of this thesis.

4 Modeling approaches review

4.1 Introduction and selection of methods

Modeling approaches aim to provide models or diagrams of a system that can help operators and analysts to understand the system better. They present the overview of a system which can contain the structural composition of a system, and interactions and behavior of components. Furthermore, the models of the system can be prepared earlier to guide the design process which ensures that systems requirements and functions are fulfilled. Chapter 1 of this thesis explained the importance of modeling approaches and why a better modeling approach is required for modern systems. As explained in Chapter 1, understanding the systems of past wasn't difficult as they were relatively simpler with mostly mechanical parts having very less interactions among them. Thus, modeling approaches for physical systems weren't much developed in past. A classification method that can present the composition of a system was enough to understand those systems. However, the situation is different now as software's are getting embedded in most of the components of a system and the interactions among components is growing rapidly. Hence, the importance of modeling approaches has been realized and are being developed recently.

The only traditional approach that is being widely used for this modeling purpose is the Tree structure method which only presents a structure of a system in a hierarchy. Thus, it is selected for the review as a traditional method. On the other hand, there are two modeling approaches developed recently: Systems Modeling Language (SysML) and Object-Process Methodology (OPM).

OPM is a modeling approach which aims to model complex systems in a holistic approach. It presents the structural composition, the behavioral and the functional aspects of the system in a single diagram. In addition to a graphical model, it also includes textual representations for better understanding about the system. (Grobshtein, et al., 2007)

Similarly, SysML is a general-purpose modeling language which supports the analysis, design, verification, specification and validation of complex systems. It includes nine different types of diagrams to present the structure, behavior, and requirements of the system. Furthermore, it also provides support for the engineering analysis of a system with a parametric diagram. (Friedenthal, et al., 2015)

Both, SysML and OPM were developed to model complex systems. OPM aims to present an overview of a complex system with a single diagram and texts. SysML on the other hand, present diagrams of 9 different kind. Table 1 shows the general differences between these two methods.

Table 1. General comparison between OPM and SysML.

Question	OPM	SysML
Does it model the structural composition of a system?	Yes	Yes
Does it model the behavior of a system?	Yes	Yes
Does it present the requirements of a system?	No	Yes
Does it provide any support with a tool for system analysis?	No	Yes
Structure of the model	A single diagram and texts	Diagrams of 9 different kind

Although both methods manage to present the structure and behavior of complex systems and can also be used to guide the design process, SysML presents the requirements of systems and supports analysts for performing the engineering analysis of a system which is lacking in OPM (Grobshtein, et al., 2007). Furthermore, Modern vessel usually consists of several complex systems. Thus, a single type of diagram for modeling all systems can be difficult and complex to manage. After discussions with experts and an advisor about the advantages and disadvantages of these methods, SysML was selected for the review as the second modeling approach.

4.2 Tree structure method

The Tree structure method is one of the widely used traditional modeling approaches which presents a graphical model of the composition of a system. In this model, the system is classified into subsystems and components in a hierarchy which resembles like a tree. A tree structure starts with a single source or edge and the classification is shown with branches that develop along nodes (Sage & Aarmstrong, 2000). Each element of the tree such as systems, sub-systems and components are represented as nodes and are connected with a solid line.

In the Tree structure method, the system is placed in the first level node of the tree. The system is then classified into sub systems in the second level. The sub systems are further classified into components and the level continues further as required. Figure 4 shows a tree structure modeling method utilized by PiezoElectric Assisted Smart Satellite structure (PEASSS) European space project for presenting the structure of their Nano-satellite (PEASSS, n.d.).

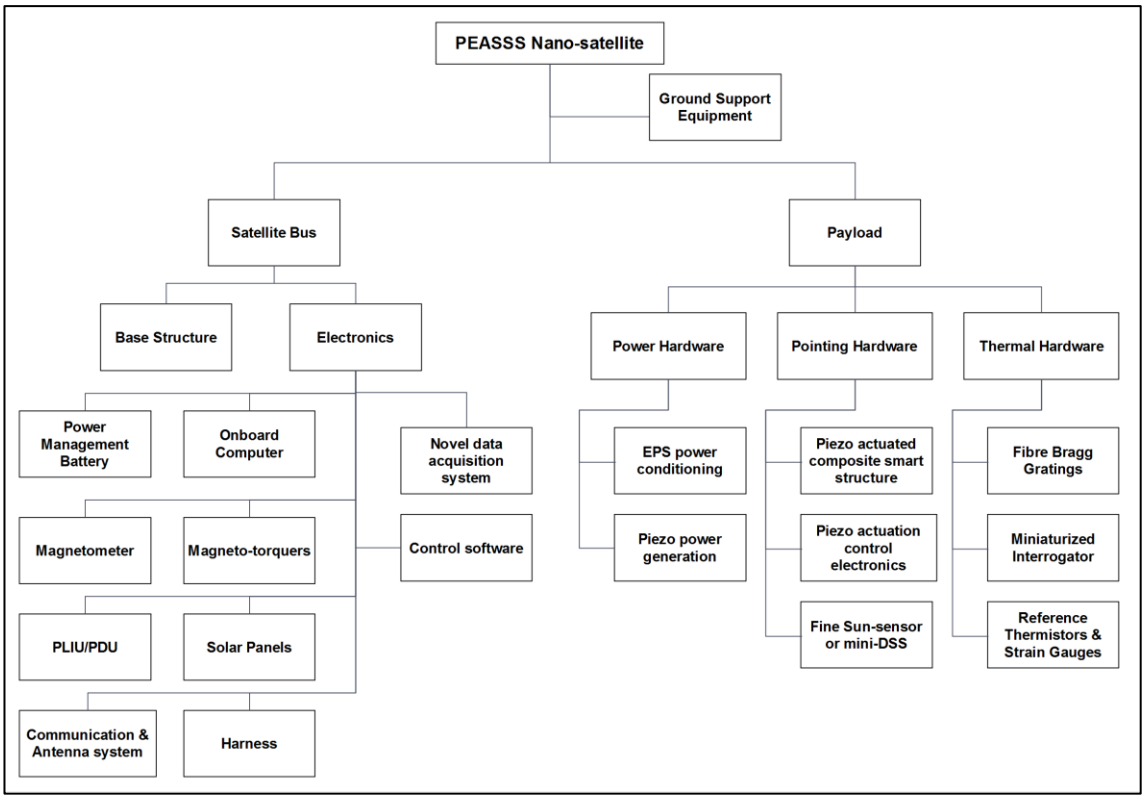


Figure 4. Classification of PEASSS Nano-Satellite with the Tree structure method.

Moreover, this approach has also been adapted to various fields. For example, FTA uses this structure to classify faults and processes by adding this structure with Boolean logic gates and different node types. Similarly, decision trees used in machine learning also utilizes tree structure with the addition of various elements to it. Hence, it is utilized widely in different fields where there is a need to show the classification of a system, event and data into further details in a simple manner.

4.3 Systems Modeling Language (SysML)

Note: This section aims to review a modeling language called Systems Modeling Language (SysML). As this language itself has wide scope, a simplified version of SysML known as SysML-lite is reviewed in this thesis. The SysML-lite is provided in a Chapter 3 of book "A Practical Guide to SysML" by Sanford Friedenthal, Alan Moore and Rick Steiner. The example which provided in the book is used to explain the method in this review. (Friedenthal, et al., 2015)

The diagrams in this review were generated by using Astah SysML (Apache, 2016) and Modelio Open Source 3.7 (Modelio, 2018).

4.3.1 Introduction

SysML is a graphical modeling language for presenting an overview of a system which includes the structural composition, behavior, constraints and requirements of a system. SysML supports the analysis, specification, design, verification, and validation of complex systems.

SysML is an extension of a subset of the Unified Modeling Language (UML) used in software engineering. In 2003, Object Management Group (OMG) issued the "UML for systems engineering request for proposal (RFP)", following a decision by International Council on Systems Engineering (INCOSE) to customize UML for systems engineering. Several specifications were developed in response to the requirements. In July 2006, these specification proposals were merged and adopted by the OMG as OMG SysML. OMG defines SysML as "a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities."
SysML models the following aspects:

1. The structural composition of a system.
2. Interconnection between systems, subsystems, and components.
3. The actions and behavior of the system and components.
4. Exchange of messages between parts of the system.
5. The behavior of the system and its components in different states and transitions.
6. The parametric relationships of the properties of the system and its components.

4.3.2 SysML diagrams

SysML includes 9 different diagrams which are as follows:

1. Package diagram
2. Requirement diagram
3. Activity diagram
4. Sequence diagram
5. State machine diagram
6. Use case diagram
7. Block definition diagram
8. Internal block diagram
9. Parametric diagram

SysML-lite excludes the sequence diagram, the state machine diagram and the use case diagram of SysML. Furthermore, it only includes a subset of available language features. However, it still provides significant modeling capabilities. Figure 5 shows the classification of diagrams in SysML where the excluded diagrams in SysML-lite are represented with dashed outline.

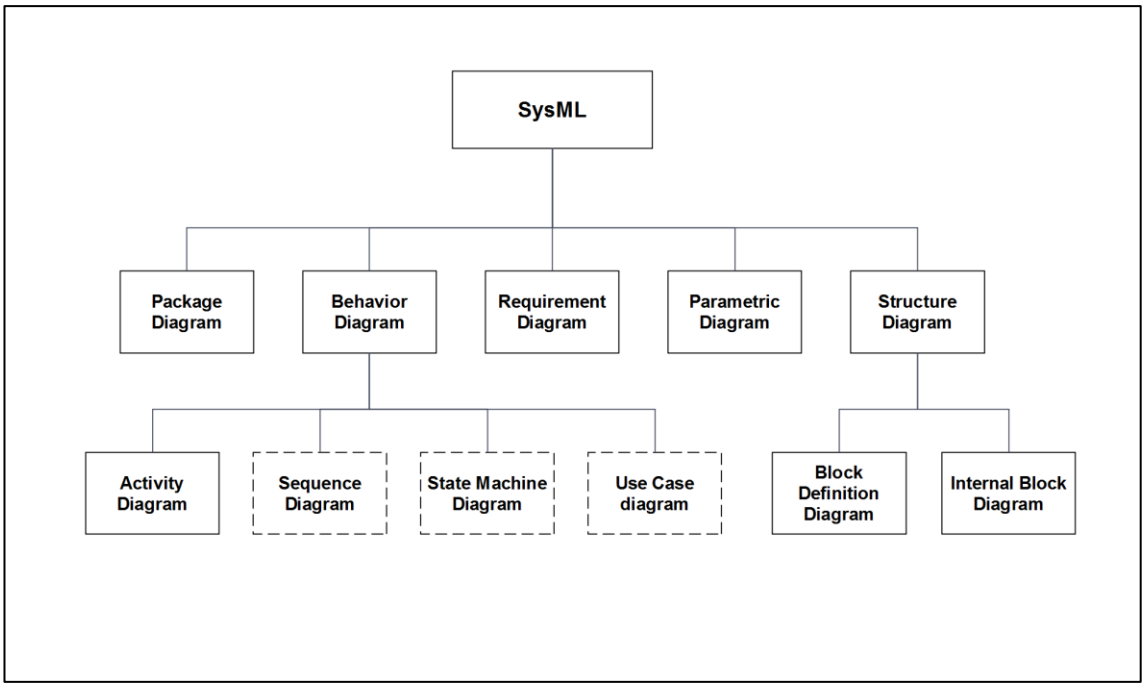


Figure 5. Classification of diagrams in SysML.

Each SysML diagram has a frame, a content area and a header. The frame limits the boundary for the diagram. Then, the header provides the information about the diagram such as SysML diagram label and user defined diagram name. Finally, the content area is the space allocated for diagram placement.

Block Definition Diagram

Blocks are the basic structural elements in SysML and are used to represent the components of a system. The component can be hardware, software, data, procedure, facility, or a person. Furthermore, a block can contain different compartments which hold block features such as properties, operations, and constraints.

The block definition diagram, labeled *bdd*, is often used to describe the structural composition of a system. It shows the sets of blocks and its characteristics in a system. An example of the block definition diagram for air compressor context is shown in Figure 6. The figure shows that the air compressor has 4 different components: a motor controller, a pump, a tank and a motor. The connector with black diamond at one end and arrow at another represents a whole-part relationship. The system is placed in the black diamond end and its components are placed at the arrow end. If a system requires components which are not owned by the system itself, then a shared association is used by replacing black diamond with a white diamond symbol. This indicates that the component is being shared by another system which owns the component. These components are known as shared or reference parts.

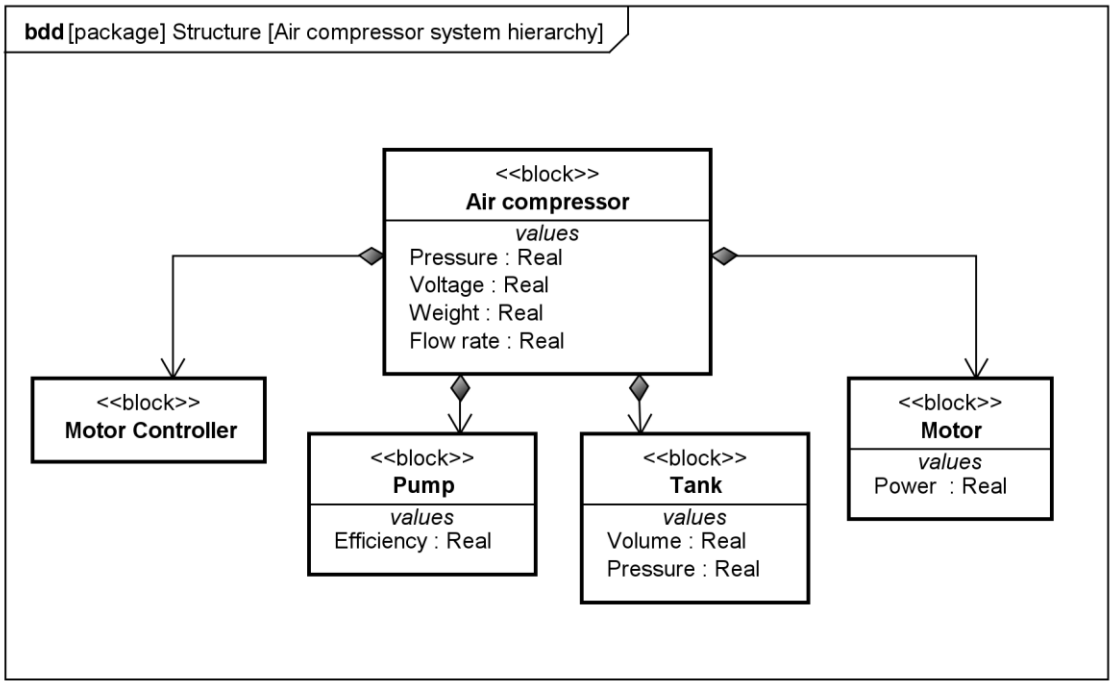


Figure 6. The block definition diagram in SysML for Air compressor context.

Internal Block Diagram

An internal block diagram, labeled *ibd*, in SysML presents the internal structure and connections of the components in a system. A layout of a motor block in internal block diagram is presented in Figure 7. The interconnections between components are shown with ports and connectors. Ports are the interaction points on a block for the connection and specify component interfaces; and a connector is a line that connects the blocks in the internal block diagram. For representing shared or reference blocks from other systems, dashed outlines are used instead of solid outlines.

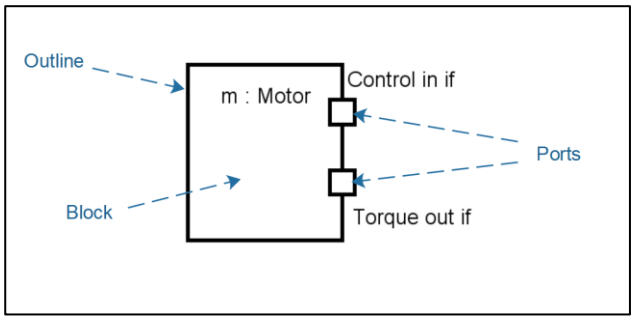


Figure 7. Layout of a block in internal block diagram.

The motor in an air compressor has two properties which are to take commands from a motor controller and to generate torque required by the pump. Hence two ports are created in the block out of which one will be connected to the pump while another will be connected to the motor controller. An internal block diagram of an air compressor is shown in Figure 8. The figure presents the interconnection and interactions between the components inside an air compressor.

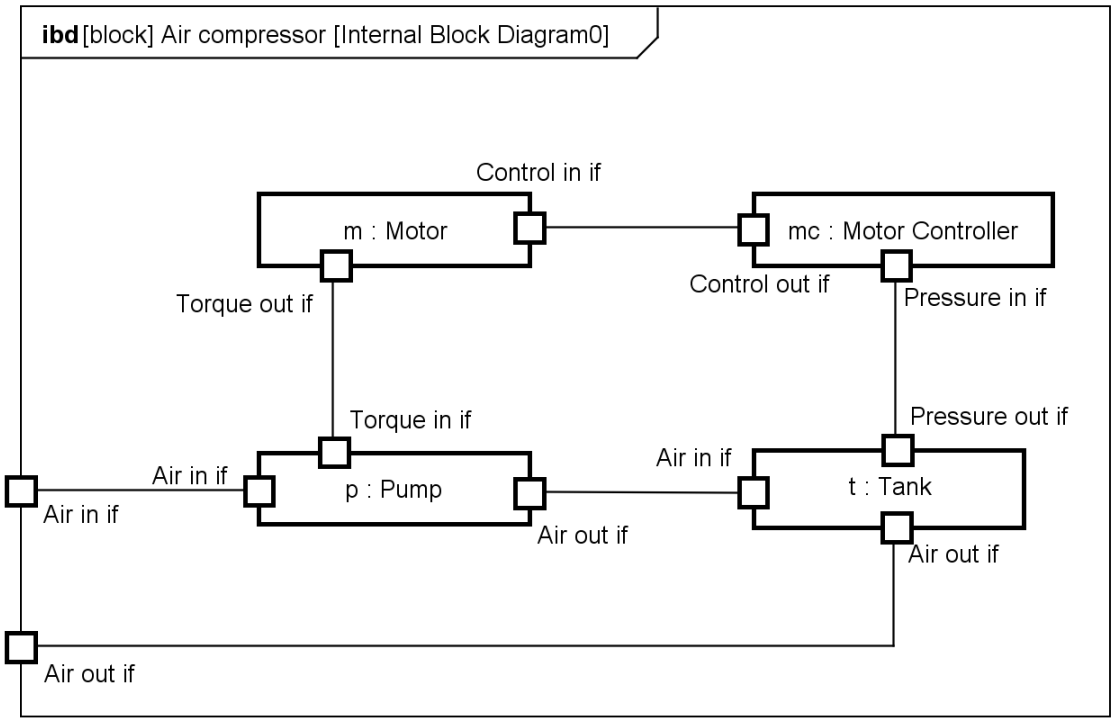


Figure 8. An internal block diagram of an air compressor.

Requirement diagram

Complex systems have a set of requirements that needs to be fulfilled for a system to function, which is contained in its specification document. A requirement diagram, labeled *req*, is used in SysML to show these sets of text-based requirements in a graphical model. Each requirement block in this diagram has one compartment that contains the title of the requirement, and another compartment that displays the id and text explaining the requirement. This diagram helps designers to create a design according to the requirements of the system being developed and it can also be used to verify the design later. In addition, it can also be used for the system analysis during an operational period, to check if the system deviates from intended design for identifying risks in a system. Figure 9 shows a requirement diagram for an air compressor. This diagram presents all identified requirements such as required power for operating, pressurized air storing capacity and maximum flow rate.

Note: The values for each requirement are not provided in the diagram and are replaced with X instead.

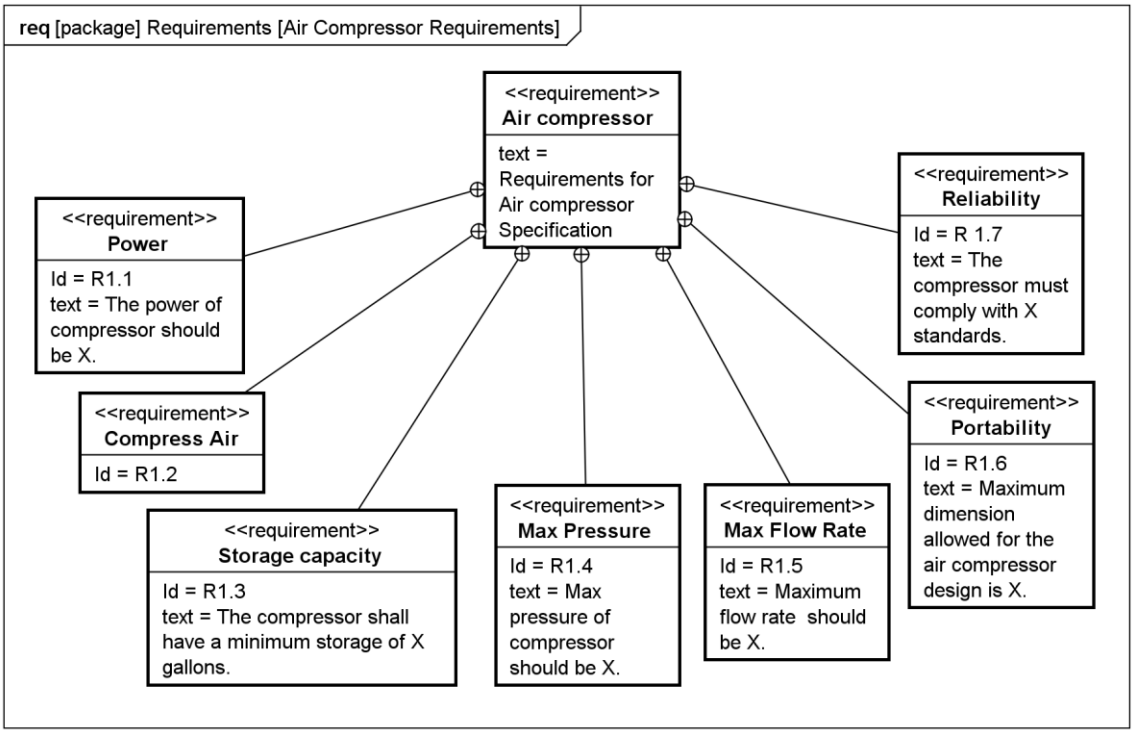


Figure 9. A requirement diagram for an air compressor.

Activity diagram

The activity diagram, labeled *act*, in SysML presents how an initialized process or activity is carried out inside a system. It shows all the components involved in the activity, the sequence of the interactions, the required inputs to the activity, and the output produced from the activity. The symbols used in activity diagram and their descriptions are presented in Table *2*.

Table 2. Symbols used in Activity diagram.

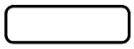
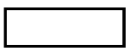
Symbol	Description
	**Initial Node**: This symbol is used to indicate the starting point of the activity.
	**Final Node:** This symbol is used to denote the ending point of the activity.
	**Fork Node:** This node is used to duplicate a flow of action into multiple parallel flows.
	**Join Node:** This node is used to join different multiple flows together into one.
	**Action Node**: This symbol is used to denote an action.
	**Object Node**: This symbol is used to denote the inputs and outputs of the activity.

Figure 10 shows the activity diagram for compressing air. At first, the total content area available for the activity diagram is partitioned depending on the number of subsystems or components which are required for the activity and are labelled. An initial node is placed to denote the start of the activity. Then the action nodes are placed in a correct sequence in their respective component partition. Furthermore, the inputs required for the process and the outputs from the process are placed in an object node and are connected to the action nodes. The control flows in the activity such as a connection between an initial node to a controller are represented with a dashed line; while the action flows and object flows are represented with a solid line. A final node is then added to the control flow which specifies that the control action is completed.

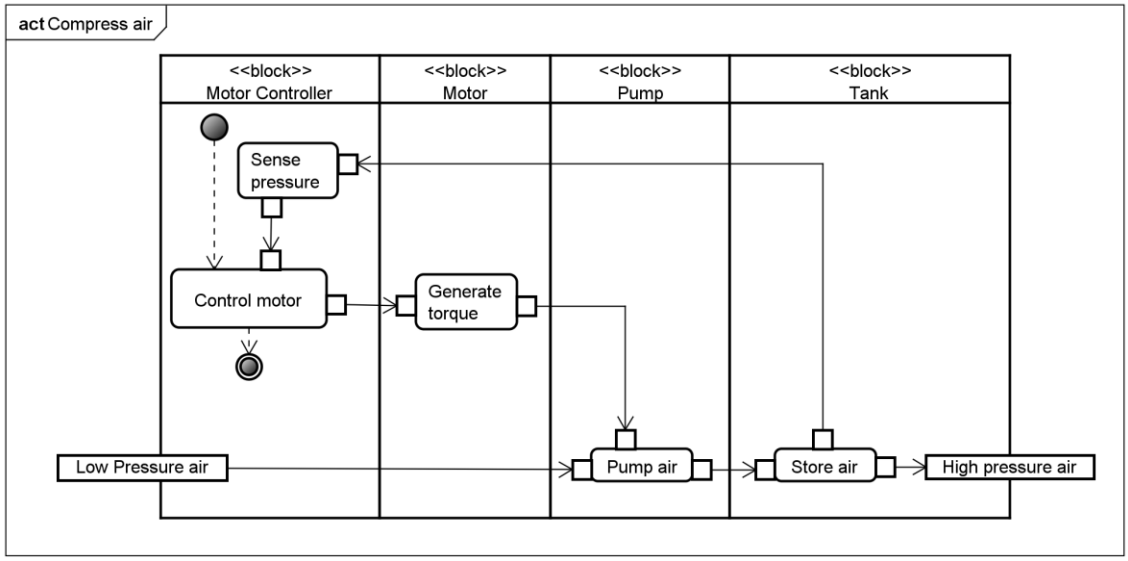


Figure 10. An activity diagram for compressing air.

In this example of compressing air activity, 4 different subsystems are required: a motor controller, a motor, a pump, and a tank. Thus, there are 4 different labelled partitions in the diagram. The actions in this activity such as generate torque, pump air and store air are placed in an action node, while the input and output of the process which are low-pressure air and high-pressure air respectively are placed in an object node. Then all the actions are connected in a correct sequence with initial node and final node.

In this example, the activity initiates by providing control action on a motor controller. The motor controller takes input from a pressure sensor inside the pump. Then if the input from the sensor satisfies the predefined condition, the motor controller starts the motor which generates the torque. The generated torque is then used by a pump for converting low pressure air to high pressure air and is stored in the tank. Hence the activity of compressing air is completed, and the motor controller stops the action which is denoted by adding a final node in the motor controller.

Parametric diagram

Parametric diagrams, labeled *par*, in SysML are used to express constraints for supporting the engineering analysis of the system such as performance and reliability. Furthermore, it also helps to identify the critical performance properties of the system for design improvements. In a parametric diagram, a constraint block is used in the model that holds an equation or set of equations for the analysis. The properties or values that are required by the equations are then imported from the blocks in the block definition diagram.

Figure 11 presents the parametric diagram for the flow rate analysis of the air compressor. A constraint block is prepared with all required constraint properties required for the flow rate analysis such as the air flow rate, volume, pressure, power and pump efficiency. The values for these constraint properties are then imported from the air compressor block and its components: tank, motor and pump. An equation for flow rate analysis is also presented in the constraint block which is not shown in the figure for simplicity.

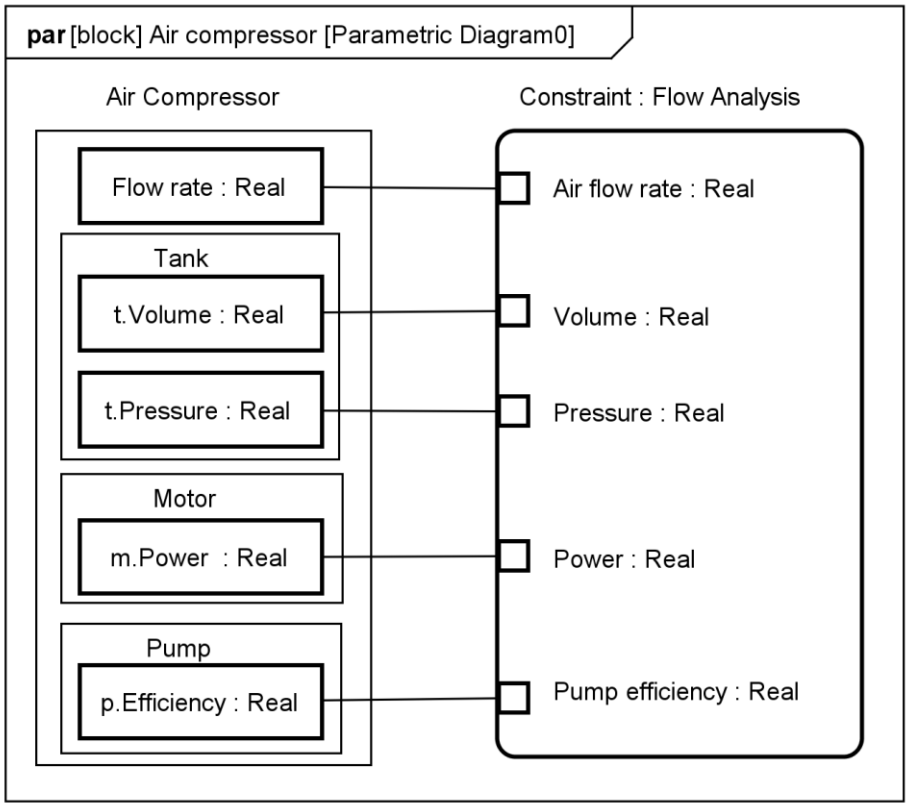


Figure 11. Parametric diagram for the flow analysis of Air compressor.

Package Diagram

The SysML diagrams contain several model elements such as blocks, requirements, constraints as discussed in previous diagrams. As the modern systems are usually comprised of several components and functions, the number of model elements in a SysML model can get large. Thus, managing these vast number of elements is necessary; and for this purpose, packages are created in SysML. A package acts as a folder and is used to group similar model elements together. (Sanford & Oster, 2016)

A package diagram, labeled *pkg*, in SysML displays all the packages within a system model. An example of a package diagram is shown in Figure 12.

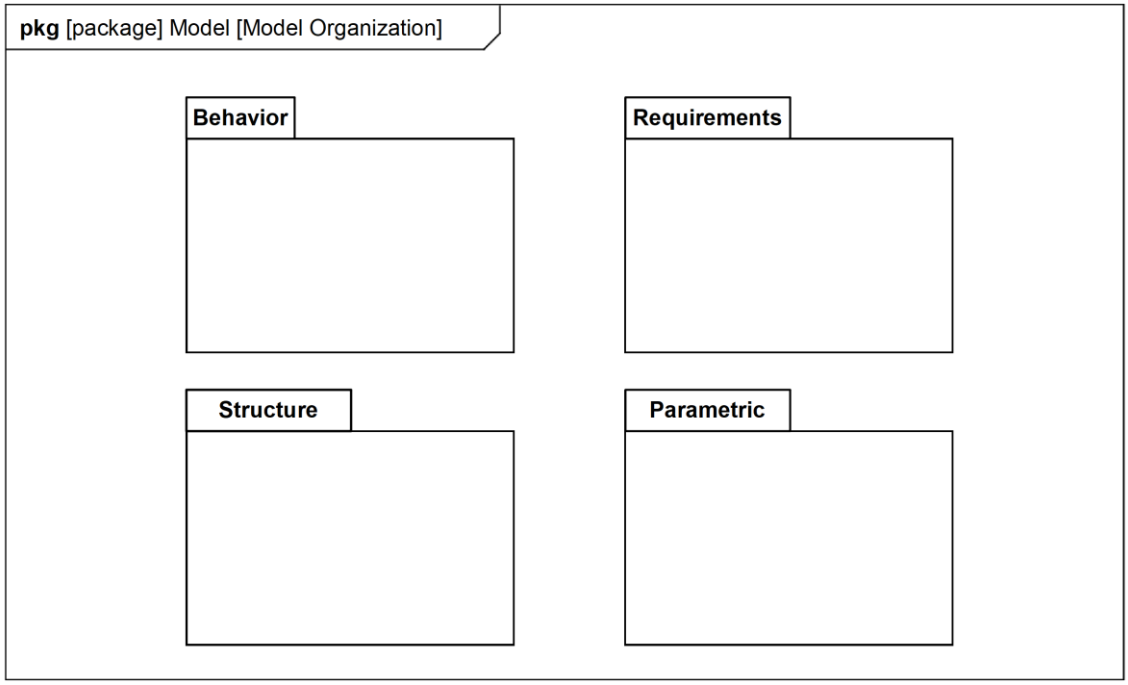


Figure 12. An example of a package diagram in SysML.

In this figure, the model has 4 packages which is based on diagram kinds: behavior, requirements, structure and parametric. However, it is also possible to have packages based on system hierarchy level such as enterprise, system, design and verification. For example, all the model elements of enterprise level will be inside the enterprise package.

5 Risk Analysis methods review

5.1 Introduction and selection of methods.

Risk analysis methods aim to identify risks in a system to avoid hazards and accidents. As explained in Chapter 1, the nature of risks is changing because of increasing component interactions. Most of the traditional risk analysis methods which are currently used for identifying risks in a ship system do not focus on potential issues due to component interactions. However, traditional risk analysis methods are still dominant in the risks analysis of the ship systems. Hence, this review aims to compare a widely used traditional method which was developed for simpler systems of past and a modern method that was developed for identifying risks in complex systems.

The mostly used traditional risks analysis methods are Fault Tree analysis (FTA), Failure modes and effect analysis (FMEA) and Hazard and operability study (HAZOP). Thus, these methods were considered for the selection.

Fault Trees analysis: FTA aims to identify all combinations of events that lead to a fault in a system. It is a top down approach which uses logic gates to illustrate the combinations in a graphical model.

Advantages of FTA are provided below (Ericson, 2015):

1. FTA is a structured and methodological approach.
2. The analysis process can be aided by computer as commercial software's are available.
3. It is relatively simple to learn and implement than other techniques.
4. It provides a visual model.
5. It is efficient than other traditional methods since it analyzes the combination of events which can lead to faults.
6. It uses different scientific theories and principal as foundation such as logic theory, Boolean algebra and reliability theory.
7. Although the result is better with a team of experts, the analysis can still be done by a single analyst.

Disadvantages of FTA are (Ericson, 2015):

1. It consumes more time than other traditional methods if the analysis is not conducted carefully.
2. The Analysts require more time to understand the method and its implementation process in comparison to other traditional methods.

Failure mode and Effect analysis: FMEA is a method used to identify faults and failure modes of components in a system. Furthermore, it also includes the effect and severity of faults. Unlike FTA, the result of this analysis is presented in a table. (Alverbro, et al., 2010)

The Advantages of FMEA are (Ericson, 2015):

1. FMEA is easy to understand and implement.
2. It is relatively inexpensive to implement.
3. The process is usually faster than FTA.
4. Commercial software is available to assist the process.
5. The result of FMEA includes different failure modes of systems, and the effect and severity of the faults.

The disadvantages of FMEA are (Ericson, 2015):

1. The affects due to failure mode combinations are not considered.
2. The method only identifies the hazards related to failure modes of components.
3. The focus on human errors is limited.
4. Requires a team of experts for implementation.

Hazard and Operability study: HAZOP is a method that uses different guidewords such as "no" and "less" to identify potential deviations from intended or designed function in a system. The end result of HAZOP includes the identified deviations, causes and consequences and this result is presented in a table. (Alverbro, et al., 2010)

The advantages of HAZOP are (Ericson, 2015):

1. HAZOP is easy to learn and implement.
2. Commercial software is available to assists the analysis process.
3. HAZOP is performed by diverse study team which can identify the hazards better.

The disadvantages of HAZOP are (Ericson, 2015):

1. The analysis relies on the judgements of analysts.
2. It was originally developed for chemical industry and can have some limitations to application in marine systems.
3. It requires a team for performing the analysis.

Based on the literature review, Table 3 provides a general comparison of FTA, FMEA and HAZOP.

Table 3. Comparison of FTA, FMEA and HAZOP.

Question	FTA	FMEA	HAZOP
How difficult is it to understand the method?	Moderate	Easy	Easy
How difficult is it to implement the method?	Easy	Easy	Easy
Is it possible to implement without a team?	Yes	No	No
Is there any availability of software assistance for the analysis process?	Yes	Yes	Yes
Does it analyze the combination of events for identifying fault in a system?	Yes	No	No
What is the result format?	Graphical model.	Table	Table

After considering the advantages and disadvantages of the methods, FTA was selected as one of the risk analysis method for detailed review in this research. The opinions from the experts of Rolls-Royce, literatures review and discussion with the advisor were also the major factors leading to this selection.

Since, this research is restricted to non-probabilistic risk analysis methods and there aren't many modern methods which are aimed to identify risks in complex systems, STPA was selected as the modern method for this review.

System's-Theoretical Process Analysis: STPA, based on system's theory, aims to identify risks in a complex system. Instead of identifying risks by breaking down the system into component level, it uses a holistic approach and starts by identifying accidents and risks at system level.

Advantages of STPA are:

1. It provides a systematic approach for the analysis.
2. It has a major focus on risks due to component interactions which is lacking in the traditional approaches.
3. It presents a visual control structure of a system.
4. It can be used in the early design phase.
5. Computer software is available to assist the process and for documenting the results.

Disadvantages of STPA are:

1. It requires significant amount of time for the analysis.
2. It is an iterative process; thus, it might be difficult to know when to conclude the analysis.

5.2 Fault Trees analysis (FTA)

Note: This section presents a review of a traditional risk analysis method called FTA. As mentioned in the research limitations in Chapter 1.3, cut sets and the probability of events are not considered in this review. The Fault Tree diagrams presented in this thesis were generated using Edraw Max 9.1 software (Edrawsoft, 2018).

5.2.1 Introduction

Fault Tree analysis is a traditional risk analysis method developed in 1962 by H. Watson and Allison B. Mearns of Bell Telephone Laboratories. It was developed for the U.S. Air force to evaluate Minuteman missile launch system. Later, this method was adopted and developed by a Boeing company, and since then many other industries have implemented FTA as a part of their hazard analysis process. (Ericson, 2015)

FTA is a graphical model that determines how a combination of fault processes and component failures or even a normal process can lead to an undesired event. This undesired event can be an accident or hazard for a system. In qualitative FTA, several types of events are represented with different node shapes. Moreover, the diverse combinations of these events are then presented with Boolean logic gates and symbols in a tree-like structure.

5.2.2 FTA building blocks

Different node types are connected to create an FTA diagram. Each node contains a rectangular block for texts and are interconnected by Boolean logic and symbols. There are 4 categories of node types in FTA which are Basic Events (BE), Gate Events (GE), Conditional Events (CE) and Transfer Events (TE). (Ericson, 2015)

5.2.2.1 Basic Events (BE)

This category consists of the normal events and failure events of the system which can lead to a hazard or fault.

Normal event: Normal event is described as a function or operation that occurs as intended or designed. Although the events are normal in an individual level, but when combined with other events can result in faults. Figure 13 shows the symbol used to denote a normal event.

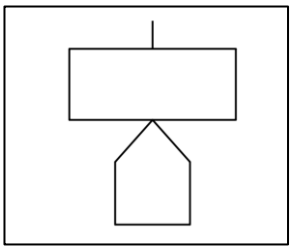


Figure 13. A symbol for a normal event used in FTA.

Failure event: Failure event is described as an event which fails to function or operate as intended or designed. This failure event is further classified into two categories: primary failure and secondary failure. Figure 14 shows the symbols used to illustrate the failure events.

Primary failure represents a basic failure event such as component failure which cannot be further developed and is illustrated with a circle symbol. While a secondary failure represents the undeveloped failures, which can be further developed in detail if required and is illustrated with a diamond symbol. (Ericson, 2015)

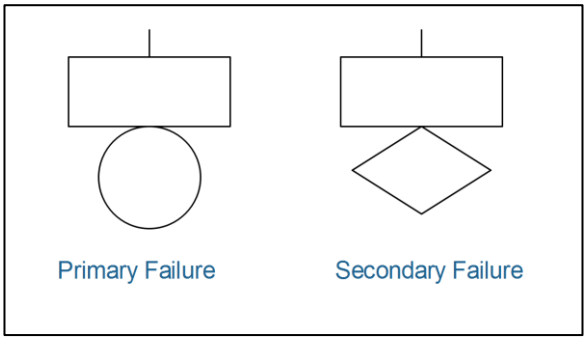


Figure 14. The FTA symbols for the failure events.

5.2.2.2 Conditional Event (CE)

This event denotes a condition that is required for some specific gate events to occur. A CE is represented by an ellipse and is attached to the gate events. A Conditional Event attached to an AND Gate is shown in Figure 15. (Ericson, 2015)

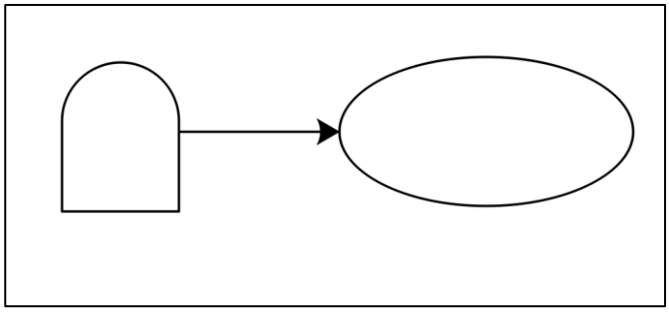


Figure 15. A Condition Event attached to an AND Gate.

5.2.2.3 Gate Events (GE)

In FTA, the events are linked with different logical operators known as gates. There are 5 different gate types in FTA and each of the gates represent a unique combination of events leading to the fault.

AND Gate: If the output event occurs only when several input events occur together, then And Gate is used. Figure 16 shows a layout of FTA diagram that consists of AND Gate and is labelled with dashed lines. In this figure, if primary failure events, A and B, occur together, then it will lead to output event G. (Ericson, 2015)

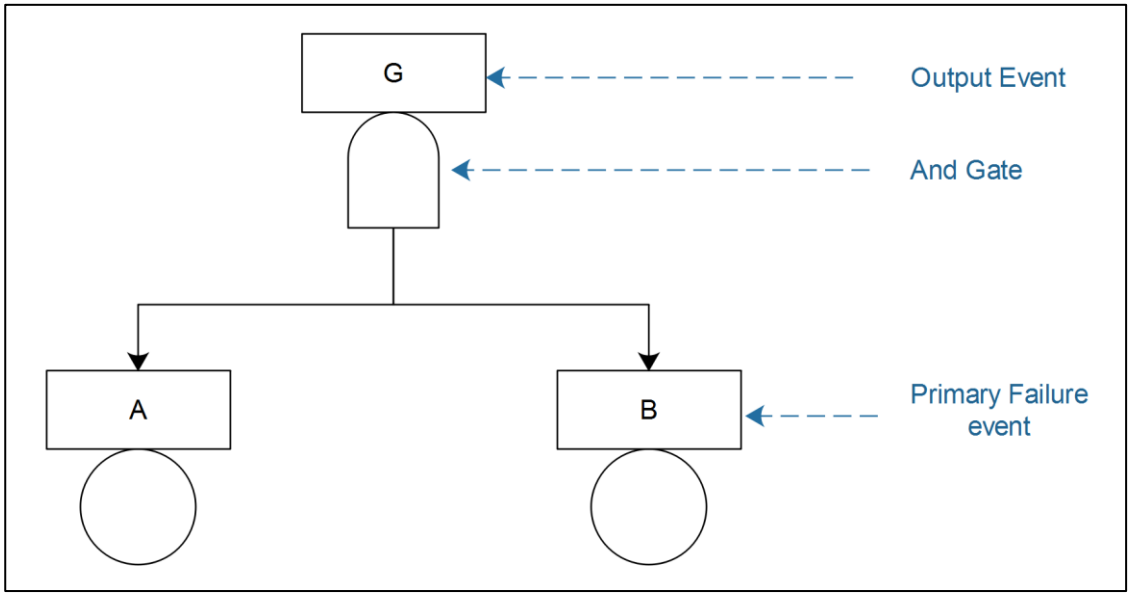


Figure 16. A layout of an FTA diagram using an AND gate.

OR Gate: If one of the several input events is enough to trigger the output event, then OR gate is used. Figure 17 shows a layout of FTA diagram that contains an OR gate. In this figure, if any of the primary failure event A or B occurs, then it will lead to the output event G. (Ericson, 2015)

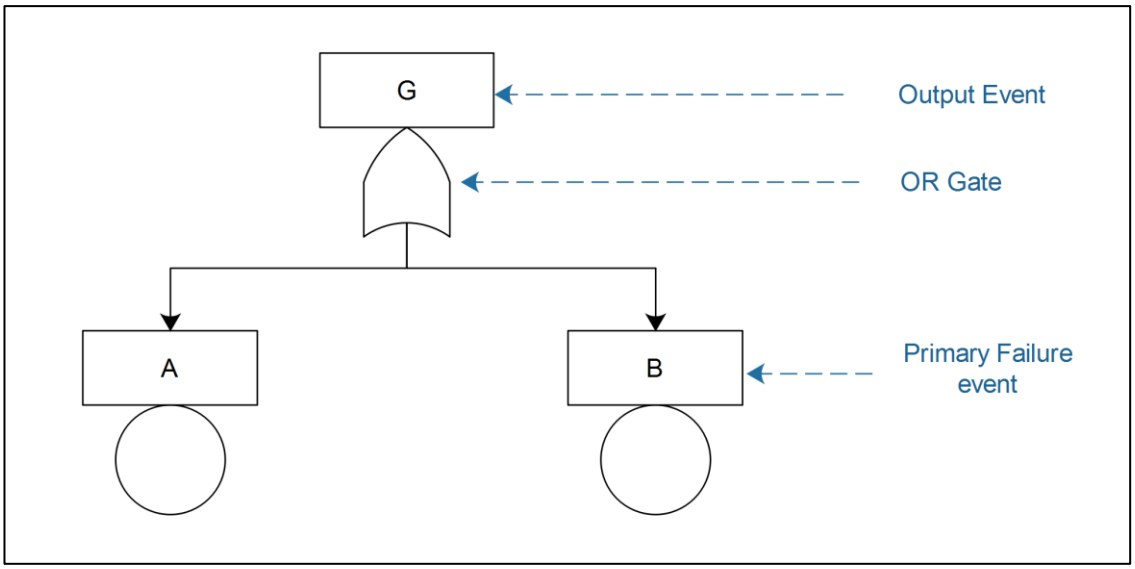


Figure 17. A layout of an FTA diagram using an OR gate.

Priority AND Gate: The Priority AND Gate is used if the output event occurs when all the input events occur together but in a specific order or priority. A condition event is added to the AND gate where the priority statement is written. Figure 18 shows a layout of FTA diagram with Priority AND Gate and is labelled with dashed lines. In this figure output event G occurs if the primary failure event A and B occurs but in a specific order. The priority statement will be mentioned inside the conditional event as shown in the figure. (Ericson, 2015)

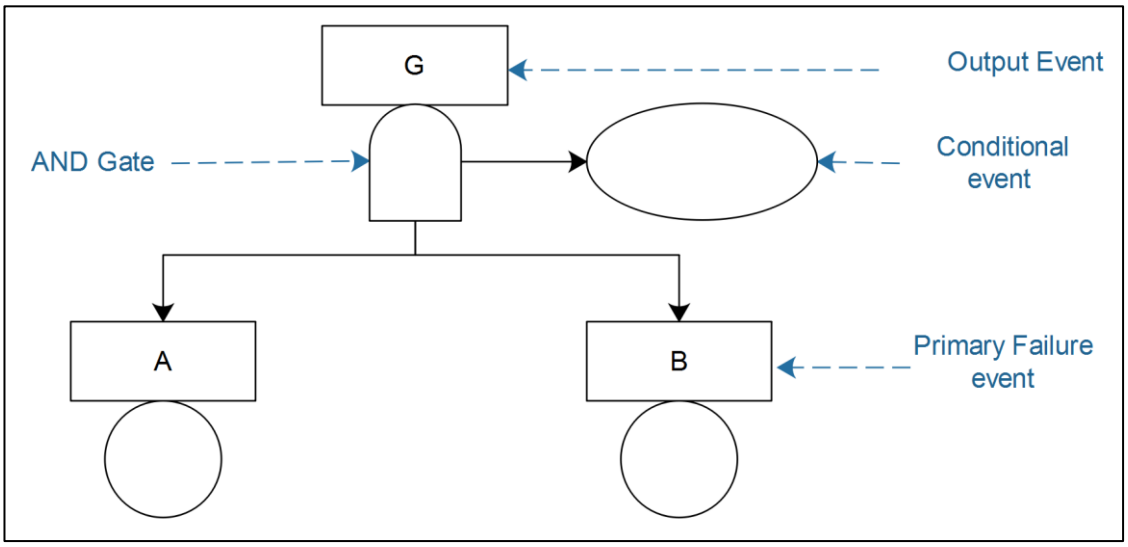


Figure 18. A layout of an FTA diagram using a Priority AND gate.

Exclusive OR Gate: The Exclusive OR Gate is used when the output occurs if either of the inputs occurs, but not both. The conditional event attached to the OR Gate contains this exclusivity statement. Figure 19 shows a layout of an FTA diagram using an Exclusive OR Gate and is labelled with dashed lines. In this figure, the output event G occurs only when either of the primary failure events A and B occur. (Ericson, 2015)

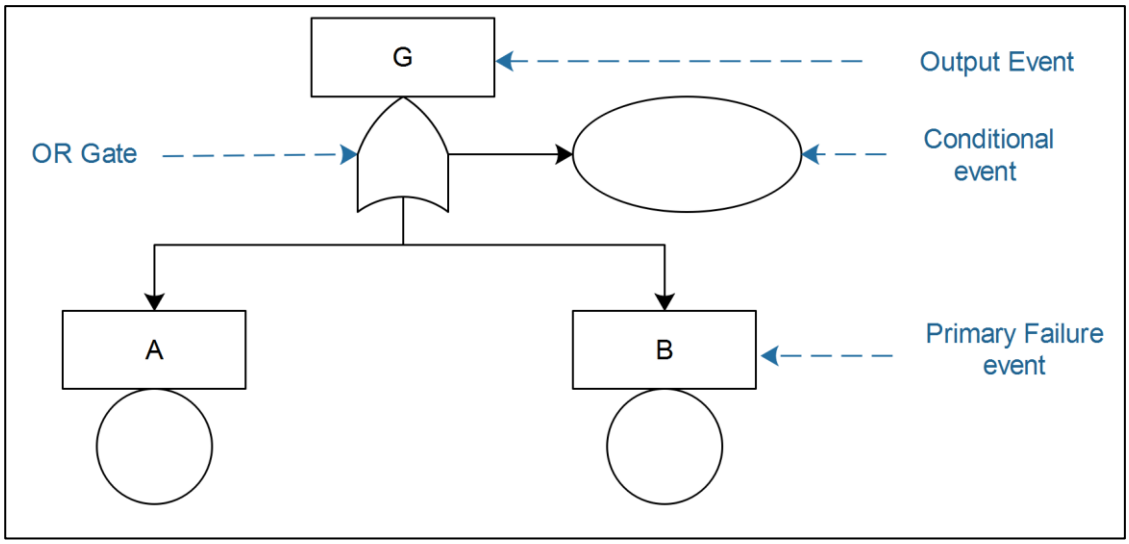


Figure 19. A layout of an FTA diagram using an Exclusive OR Gate.

Inhibit Gate: If the output occurs when the input event occurs, and a specific condition is satisfied, then the Inhibit Gate is used to represent this combination in FTA. A conditional event is added which contains the condition that needs to be satisfied. Figure 20 shows a layout of FTA using Inhibit Gate and is labelled with dashed lines. In this figure, the output event G occurs only when the primary failure event A occur, and the attached condition Y is satisfied. (Ericson, 2015)

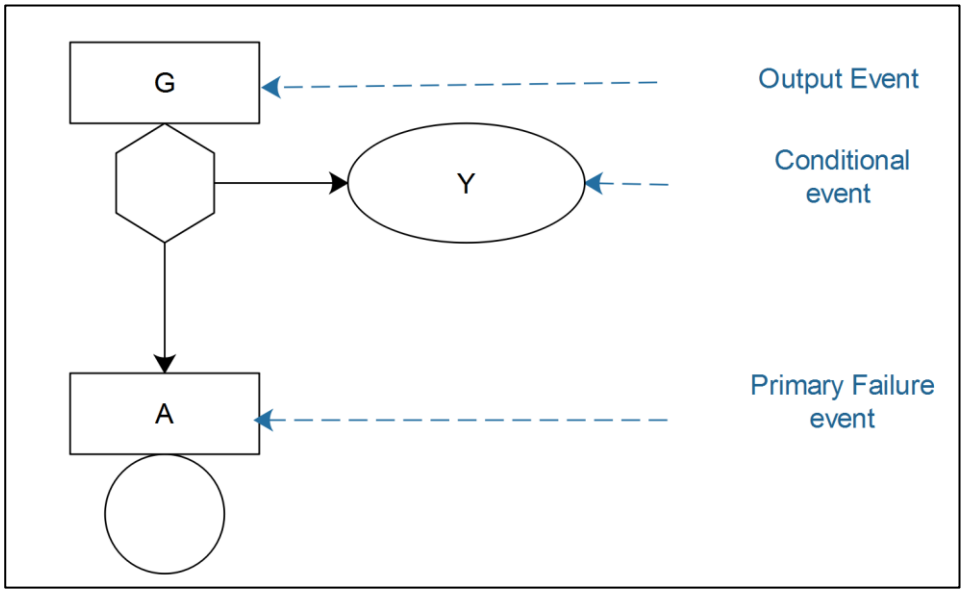


Figure 20. A layout of an FTA diagram using an Inhibit Gate.

5.2.2.4 Transfer Events (TE)

Transfer event is used to indicate a subtree branch which is used elsewhere in the tree. A triangle symbol shown in Figure 21 is used for this combination. A Transfer In symbol is used in the place where the branch is getting imported. A Transfer Out symbol is then connected to a branch that is getting transferred indicating that the branch is used by a tree somewhere else in the diagram. A sample layout using a transfer events is shown in Figure 22. (Ericson, 2015)

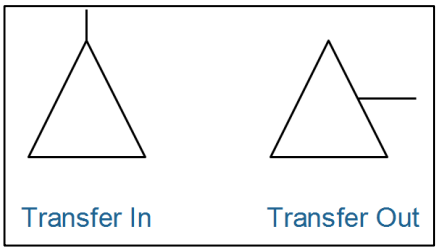


Figure 21. Symbols used for Transfer Events.

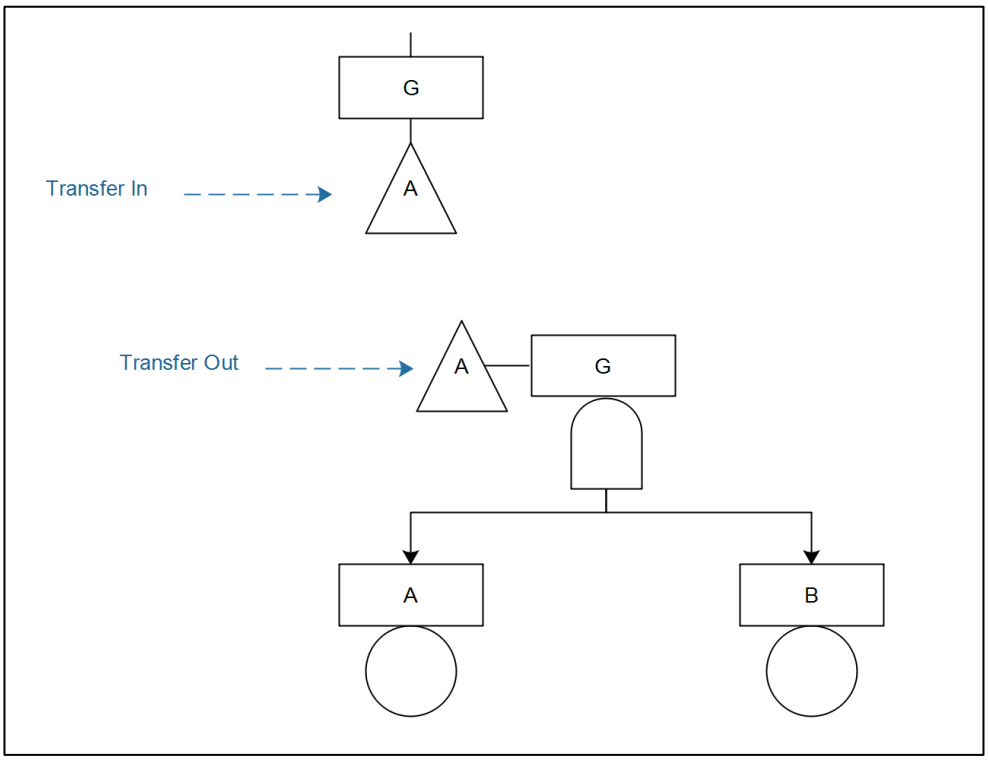


Figure 22. Sample Layout of FTA with Transfer events.

5.2.3 Procedure

A Fault Tree is developed at different levels with branches. The basic steps for constructing a Fault Tree are as following (Ericson, 2015):

1. Review and understand the fault event under investigation.
2. Identify all the probable causes of this fault event and develop further if it is required.
3. Identify the relationship or logic of the Cause-Effect events.
4. Structure the tree with appropriate gate events for identified input events.
5. Review for a possible repetition of events.
6. Move to the next fault and repeat the process.

5.2.4 FTA example of a fault in an electric motor.

An example FTA diagram of an electric motor for the fault, “motor overheats” from is provided in this section (Sundararajan, 2012).

As described in the procedure, the first step for Fault Tree Analysis is to review and understand the fault event under investigation. In this example, the fault is motor overheating, and thus it will be placed at the top of the Fault Tree. After reviewing and understanding the fault, all probable causes leading to it are identified. The causes for motor overheating are due to i) an internal malfunction of the motor itself (A1) or ii) excessive current supplied to the motor (A2). As either of these causes can lead to the fault, OR Gate is selected. The first cause is a component failure and it cannot be further developed. However, second cause will be further developed at second level of the Fault Tree. Hence the tree is created using the correct symbol for events and gate which is shown in Figure 23.

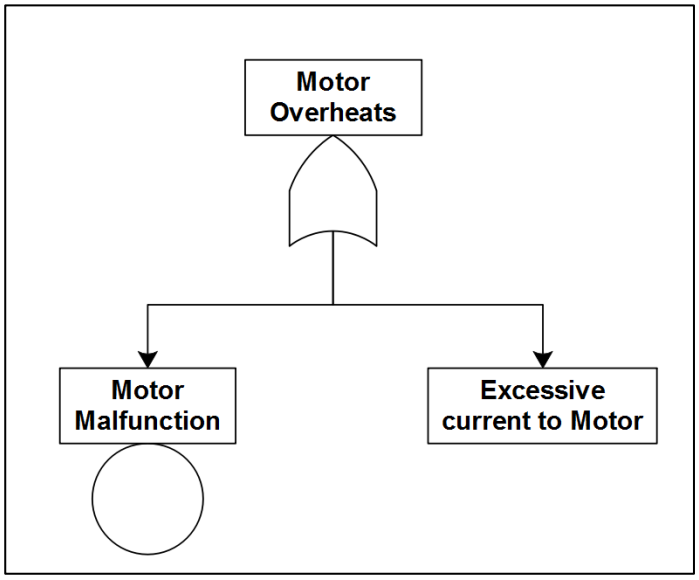


Figure 23. First level of the Fault Tree for Motor overheating.

The second event A2 is analyzed and probable causes leading to this event are identified Event A2 which is “Excessive current to motor” can be caused if there is i) Excessive current in the circuit (B1) and ii) The fuse fails to operate (B2). Since, the event A2 will only be triggered if both B1 and B2 happen together, AND Gate is used to denote this combination. Similar to event A1, a primary failure symbol will be used to denote event B2 as it is a component failure. Event B1 will be further developed at third level of the Fault Tree. Figure 24 shows the second level of the Fault Tree for motor overheating example.

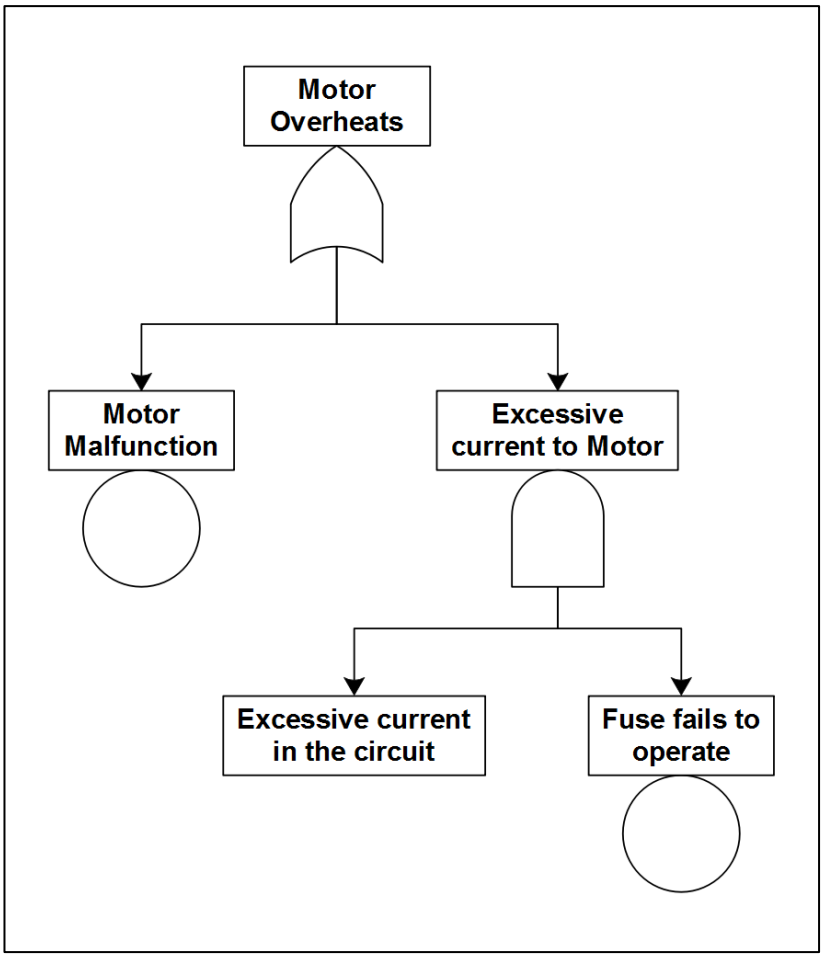


Figure 24. Second level of the Fault Tree for motor overheating.

Again, for the third level or final level of this example, the probable causes of Event B1 are identified. Event B1 can happen if there is i) short circuit in wiring (C1) or ii) if there is a power surge (C2). Or Gate will be used to denote this combination, event C1 will be denoted with primary failure symbol and event C2 will be denoted with secondary failure as it is an undeveloped event in this Fault Tree and can be further developed later if required. Hence a Fault Tree for "motor overheating" is completed and is shown in Figure 25.

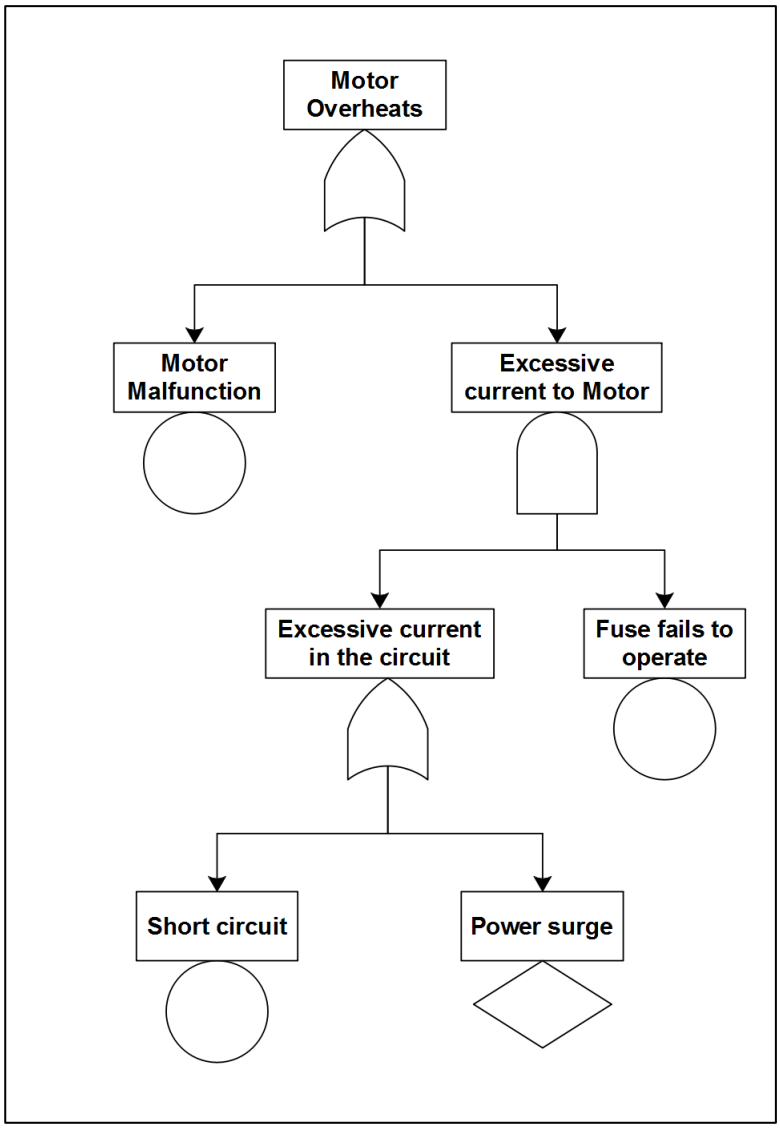


Figure 25. A FT of an electric motor overheating.

After completing FT diagram, it must be reviewed to ensure that correct symbols have been applied and identified events are not repeated. Finally, the analyst can move to another fault event of the system.

5.3 Systems-Theoretical process analysis (STPA)

5.3.1 Introduction

STPA is a new hazard analysis technique developed in 2011 by Nancy Leveson (Professor of Aeronautics and Astronautics in MIT). Similar to other hazard analysis methods, it aims to identify the hazards and risks of the system to mitigate the risks in a system. As traditional methods focus on identifying risks related to component failures and human errors, STPA also aims to identify other possible failures such as unsafe interactions among non-failing components which can be caused from design flaws. (Leveson, 2015)

STPA is based on control and system theory which deals the emergent properties of systems that arise due to the interaction of components within a larger environment. Furthermore, STPA establishes a set of safety constraints enforcing the safety of a system. When these safety constraints are violated or not imposed, then the accidents occur. Hence the method focuses on enforcing safety constraints to each controller in the system such as operators, designers, management, government and control units. Moreover, STPA can be used during the early phase of a system development process to generate safety constraints and requirements for the system. These constraints will guide designers in the early phase on building a safer design of the system. As a result, many design errors will be mitigated during the design phase which is important as the cost of rework due to design errors at later stages is enormous. STPA analysis is an iterative process and includes the following steps (Leveson, 2015):

1. Establish the foundation for the analysis
2. Identify potentially unsafe control actions.
3. Create safety constraints and requirements for unsafe control action.
4. Determine how the unsafe control action could occur.

5.3.2 Procedure

Step 1: Establishing the foundation for the analysis.
STPA analysis starts by defining the accidents of the system and identifying hazards that can lead to such accidents. Accidents can be defined as events that involve the loss or injury of humans in the system or loss of the system itself. Nancy Leveson has defined an accident as following:

"An accident is an undesired and unplanned event that results in a loss, including a loss of human life or human injury, property damage, environmental pollution, mission loss, financial loss, etc."

After defining all the accidents of the system, hazards leading to each accident are identified. Nancy Leveson defines a hazard as following for STPA analysis:

"*Hazard is a system state or set of conditions that together with a worst-case set of environmental conditions, will lead to an accident."*

Identified accidents and hazards are reported in a table format. All hazards are then analyzed in detail; and safety constraints for eliminating or controlling them are created.

Next, all possible controllers of the system and their controlling actions are analyzed, by preparing a control structure of the system. As identifying accidents, hazards and safety

constraints are common features of most of the hazard analysis techniques, this step makes STPA analysis unique from the rest. The control structure provides a graphical illustration of interactions among components and controllers. These interactions or control actions are then analyzed with keywords to identify the unsafe control actions. After a successful identification of the unsafe control actions, safety constraints to mitigate these unsafe actions are prepared and implemented in the system.

The first step of making a control structure is to identify the main components of the system. For example: The components involved in the In-Trail procedure (ITP) for allowing two aircrafts to pass one another are shown in Figure 26 (Leveson, 2015).

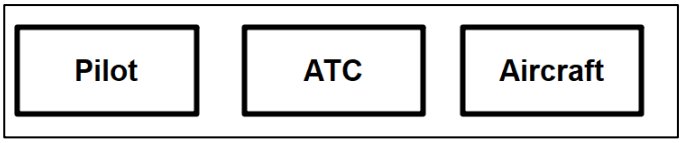


Figure 26. Different Components in the In-Trail Procedure (ITP).

After determining the components, controllers and controlled components are identified among them. Then, a control structure that shows the interactions between the components is created as shown in Figure 27 (Leveson, 2015):

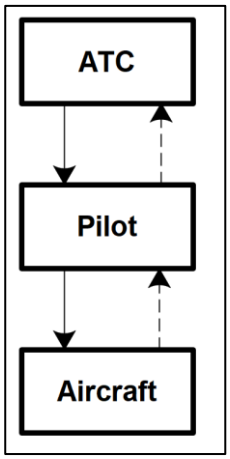


Figure 27. The control structure of the components in the ITP.

The control structure shows that the ATC (Air traffic controllers) gives instructions to the pilot and using these instructions pilot controls the aircraft. The action of each controller controlling other parts of the system is represented with a solid line in the figure. Then, a dashed line in used to represent the feedback received by the controllers. Next, the type of control actions and feedback received by each part of the system are also included in the control structure. A control structure with the interactions (control actions and received feedback) are shown in Figure 28 (Leveson, 2015):

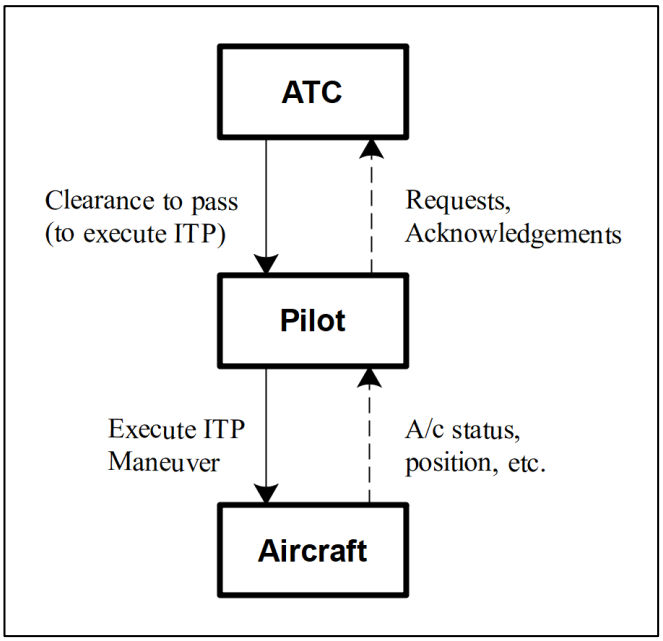


Figure 28. The control structure with the interactions among the controllers in the ITP.

The control structure at Figure 28 shows the control actions of controllers ATC and pilot. For example, the figure illustrates that the ATC receives requests and acknowledgements from pilots for ITP procedure and ATC gives clearance to pass to the pilot for executing ITP. Then, the pilot receives A/c status, positions, etc. from aircraft and executes the ITP maneuver. This example shows a simple control structure. However, the control structure can be prepared in detail according to the requirement. (Leveson, 2015)

Step 2: Identifying the potential unsafe control actions.
After determining all possible control actions within the system, the unsafe control actions are identified using the guidewords. Standard guidewords used in STPA analysis are (Leveson, 2015):

- Not providing the control action.
- Providing the control action.
- Providing the control action too early or too late.
- Stopping the control action too soon or applying for too long.

All control actions are analyzed using the guidewords. For example, the analyst will check if not providing the control action can cause any hazard in a system or if providing the control action in some conditions may cause any hazard.

Guidewords can be modified or added according to the type of the system being analyzed or based on the depth of analysis required for the system. The results of analyzed control actions with these keywords are documented on the table. Table *4* shows the table for identifying the unsafe control actions for the hazard loss of minimum separation for ITP. (Leveson, 2015)

Table 4. Identifying unsafe control actions for the hazard loss of Minimum separation for ITP (Leveson, 2015).

Control action	Not providing the action causes hazard	Providing the action causes hazard.	Wrong timing or order of actions causes hazard	Action stopped too soon/applied too soon
Execute ITP		ITP executed when not approved. ITP executed when ITP criteria are not satisfied. ITP executed with incorrect climb rate, final altitude, etc.	ITP executed too soon before approval. ITP executed too late after the reassessment.	ITP aircrafts levels off above requested FL. ITP aircrafts levels off below requested FL.
Abnormal Termination of ITP	FC continues with maneuver in dangerous situation.	FC aborts unnecessarily. FC does not follow regional contingency procedures while aborting.		

Step 3: Create safety constraints and requirements for unsafe control action.
After identifying all unsafe control actions of the system, safety constraints are implemented to control the hazard. For example, the safety constraints for unsafe control action "ITP executed when not approved" can be that "the flight crew must not execute the ITP until approved by ATC" (Leveson, 2015). This step is completed once safety constraints for all identified unsafe control actions are provided.

Step 4: Determine how the unsafe control actions could occur.
As only identifying unsafe control action in a system is insufficient for mitigating the risks in a system, how the unsafe control actions can occur in the system and what are its effects are determined in the final step of STPA analysis process. After knowing the causes of these unsafe control actions, again safety constraints will be established and enforced in a system to mitigate the risks. However, this step requires an input from experienced experts as they are the one who can know how these unsafe control actions can happen in a system.

For example, an unsafe control action "Crew not providing manual braking in an aircraft when required" can result due to the following reason (Leveson, 2015):
"The crew incorrectly believes that the autobrake feature is armed and will be engaged. This can happen due to a mistake of the crew or system design error such as multiple and conflicting messages, and alarm fatigue".

Hence, the safety constraints to control or eliminate the cause are created and enforced which completes the STPA analysis. (Leveson, 2015)

6 Case Study

This section presents a case study with Rolls-Royce where the methods: The Tree structure method, SysML, FTA and STPA were implemented to model a part of the azimuth thruster. This study was prepared in a workshop organized in the Rolls-Royce Research and development center located in Turku, on the 28th of May. The workshop was attended by 4-5 experts of Rolls-Royce with diverse backgrounds.

Due to the time restrictions, only some parts of the system were modeled. Furthermore, the aim of this workshop was to understand the method's complexity for implementation, time consumption and method's functionality.

6.1 Modeling approaches

6.1.1 Tree structure method

The graphical model of a Lubrication unit in the azimuth thruster was prepared using the Tree structure method with Edraw max 9.1 software (Edrawsoft, 2018). The prepared model is presented in Figure 29.

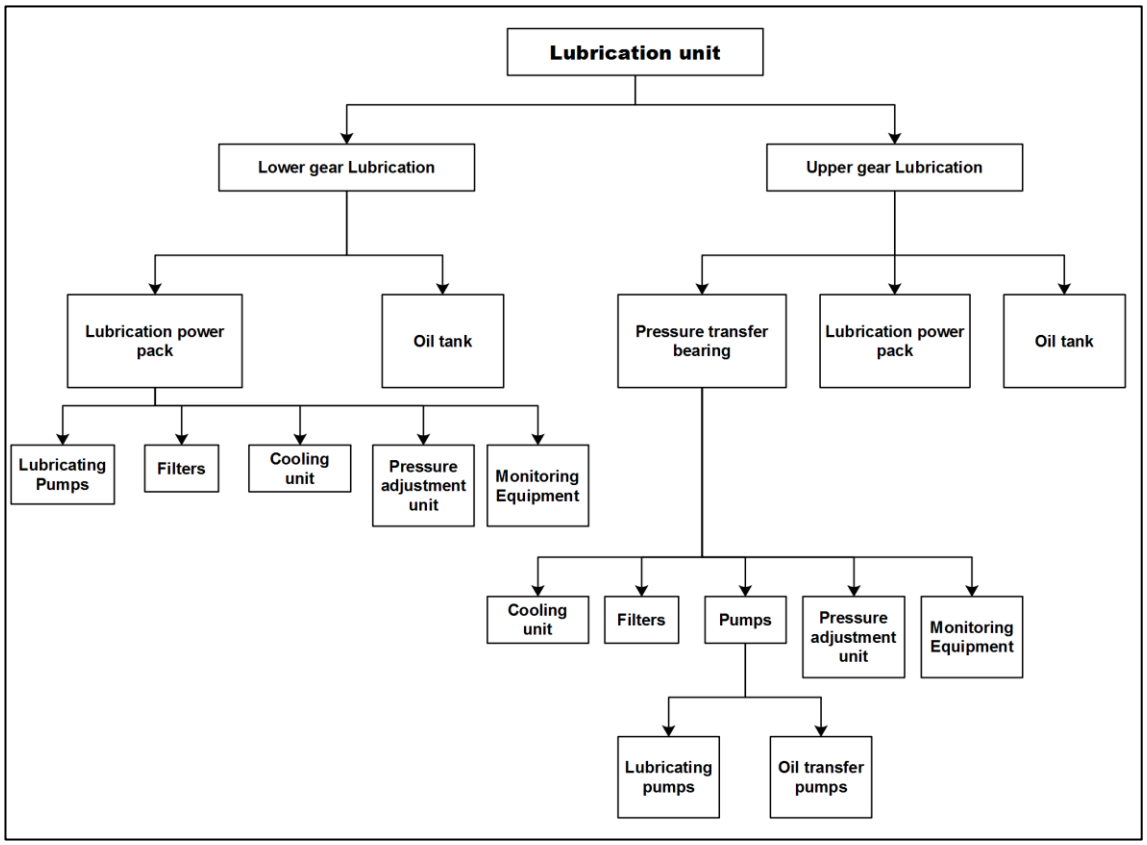


Figure 29. A model of a lubrication unit in the azimuth thruster using the Tree structure method.

The presented model shows the structural composition of the lubrication unit. The lubrication unit is comprised of two separate units with specific function. One unit is assigned to lubricate the lower gear, while the other unit is assigned to the upper gear. While both units contain lubrication power pack and oil tank, the upper gear lubrication unit also contains a pressure transfer bearing. The lubrication power packs can be further classified into pumps, filters, cooling unit, pressure adjustment unit and monitoring

equipment's. The lower gear lubrication power pack consists of the lubricating pumps while the upper gear lubrication power pack consists of the lubricating pumps and the oil transfer pumps. Thus, 5 different levels of structural composition in the lubrication unit in the azimuth thruster are illustrated using this method.

6.1.2 Systems Modeling Language (SysML)

Similarly, a graphical model of lower gear in the azimuth thruster was prepared using SysML-lite in the workshop. All diagrams of SysML-lite, presented at Chapter 4.3 in this thesis, were prepared in the workshop using Astah SysML (Apache, 2016) and Modelio Open Source 3.7 software (Modelio, 2018).

Package diagram

In order to manage all the elements of SysML diagrams properly, the modeling started with the package diagram. The package diagram for lower gear is shown in Figure 30.

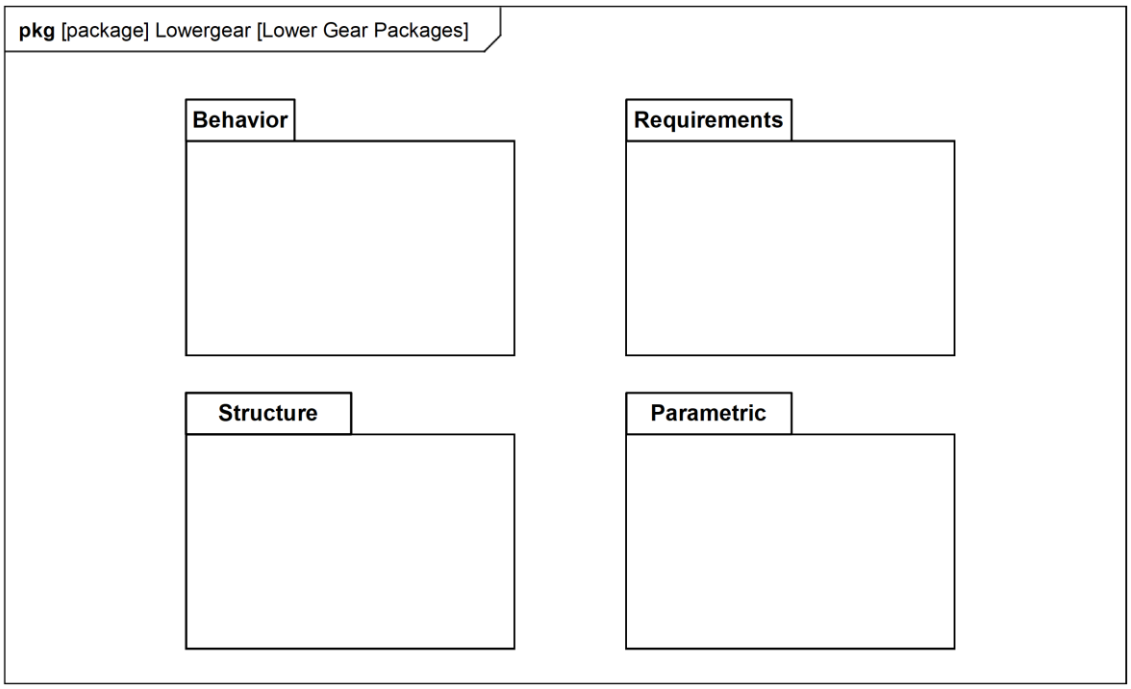


Figure 30. The package diagram in SysML for lower gear.

As shown in the diagram, four different packages: behavior, structure, requirements, parametric, were created in the tool. Elements that present the behavior of the system such as an activity diagram will be placed in the behavior package. The requirements package will contain the requirements of lower gear and the parametric package will hold all the constraints for engineering analysis for lower gear. The structure package will then contain all the elements that present the structure of the lower gear such as blocks, block definition diagram and internal body diagram.

Requirement diagram

The requirement diagram which presents all the requirements for lower gear is presented in Figure 31.

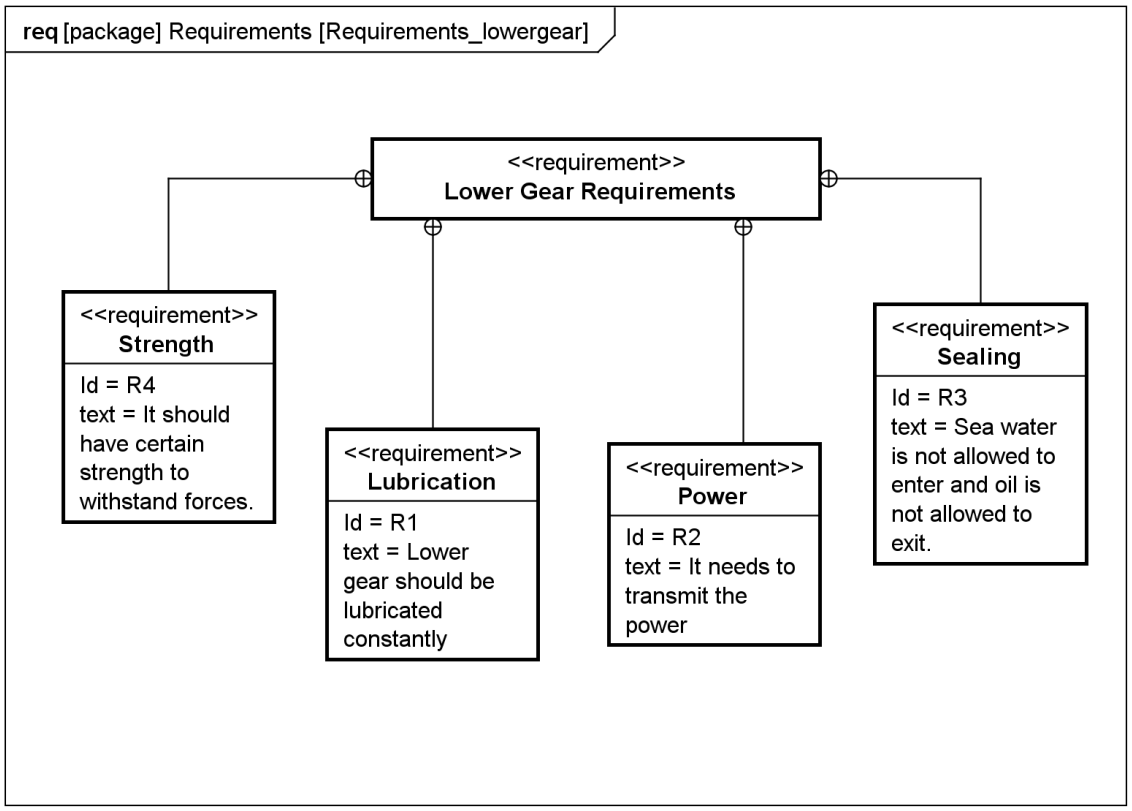


Figure 31. The requirement diagram of the lower gear in the azimuth thruster.

There were 4 different requirements identified in the workshop for the lower gear which are: strength, lubrication, power and sealing. The text in the bottom compartment of each block explains the requirements in detail. Again, specific values for requirements such as strength and power are not added as they are not required to achieve the aim of this workshop.

Block Definition Diagram

The block definition diagram for lower gear prepared in the workshop is shown in Figure 32.

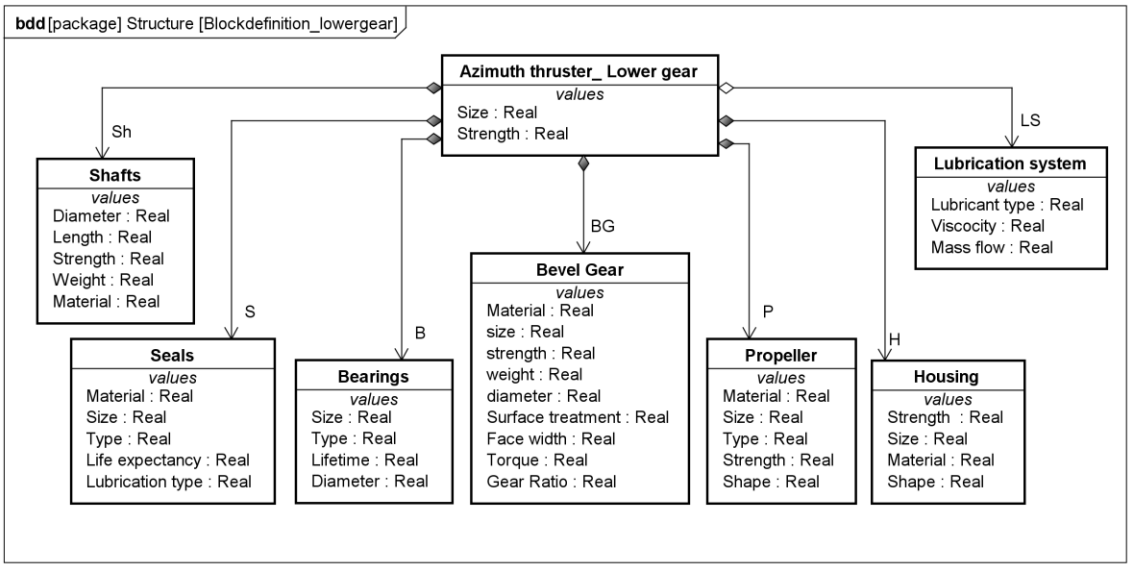


Figure 32. The block definition diagram of the lower gear in the azimuth thruster.

The sub-systems and components of Lower gear are presented in the block definition diagram. Furthermore, all relevant properties of these components are added in the values compartments as shown in the above figure. The exact values of these component properties were not added as it was not required to achieve the aim of this workshop. The diagram shows that the lower gear is composed of shafts, bevel gear, bearings, seals, propeller, housing and the lubrication system. The lubricating system is not owned by the lower gear, but it is used for the lubrication of the lower gear. Thus, it is regarded as the shared component and is represented by the white diamond pointer instead of black diamond.

Internal block diagram

The internal block diagram of the lower gear was then prepared which is presented in Figure 33.

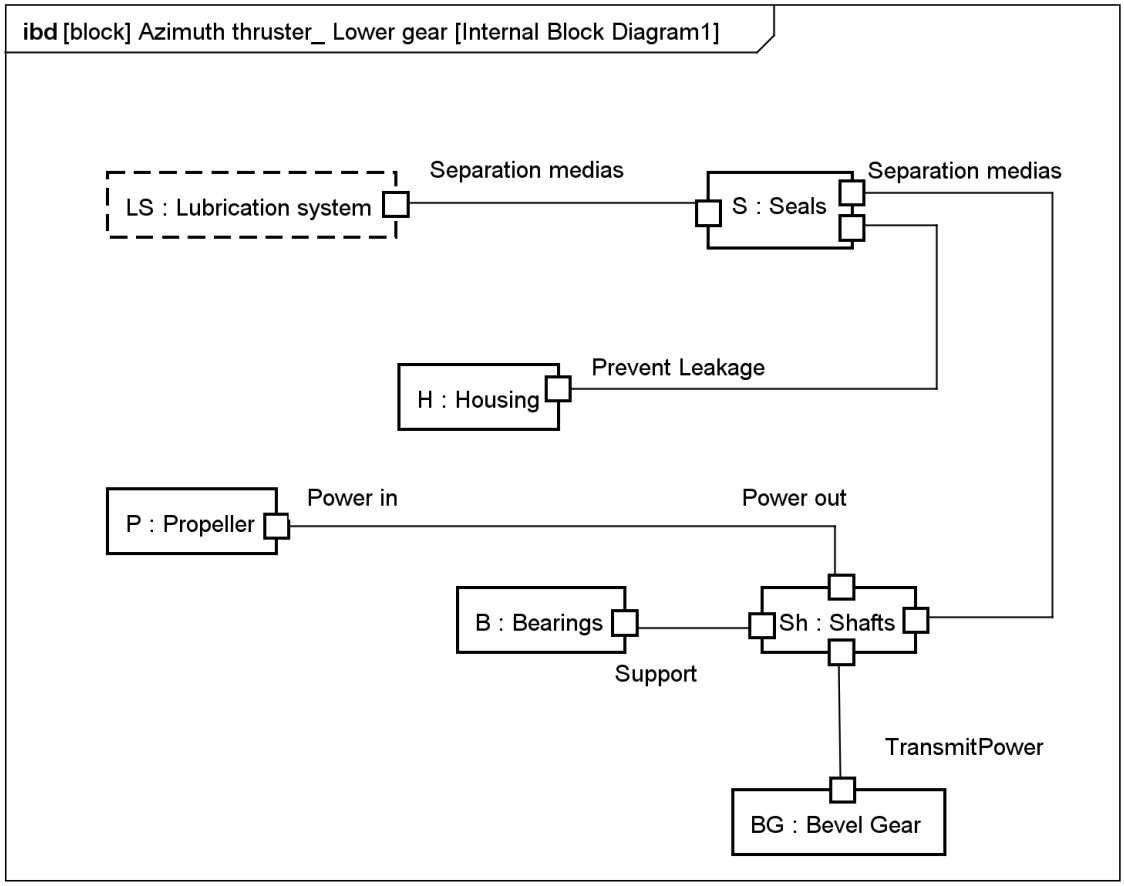


Figure 33. The internal block diagram of the lower gear in the azimuth thruster.

This diagram shows the interconnection between the components of lower gear. The layout of the structure from internal perspective and the connections between components are shown in the diagram. The lubrication system is regarded as a shared component since the lower gear doesn't own the lubrication unit, thus a dashed outline is used for the lubrication system.

Activity diagram

The activity diagram of the lubrication in lower gear was prepared and is shown in Figure 34.

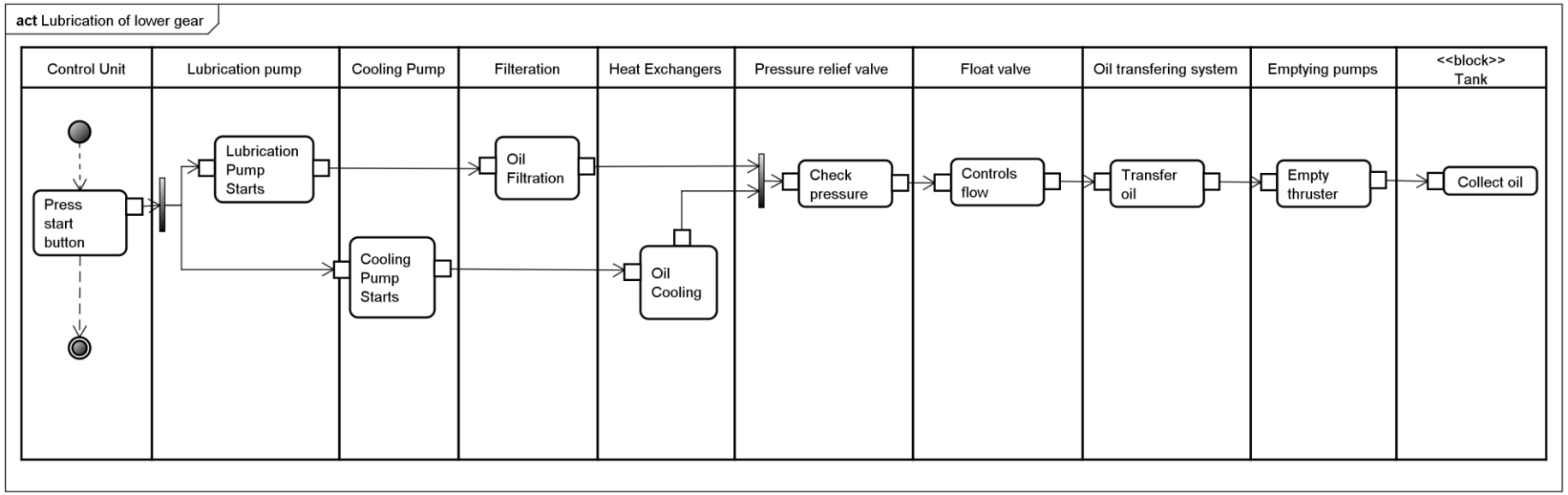


Figure 34. The activity diagram for lower gear lubrication in the azimuth thruster.

The activity diagram presents how the process or activity of lubrication in lower gear is carried out. It shows all the components that are involved in this process and also present the sequence in which they are used. The process starts by pressing a button on a control unit. This action activates two processes: it starts the lubricating pump and the cooling pump. The oil from the lubricating pump is then filtered and supplied while the cooling unit reduces the temperature of oil. Then, the pressure relief valve checks the pressure of the oil and is passed to the floating valve which controls the flow of oil. If the oil is not critical in pressure and temperature, then the oil transferring system starts transferring the oil in the lower gear for lubrication. After lubrication, the emptying pumps are used to transfer the lubrication oil back to the tank which completes the lubricating activity.

Parametric diagram

A parametric diagram for analyzing bending stress of a bevel gear was prepared in the workshop and is presented in Figure 35.

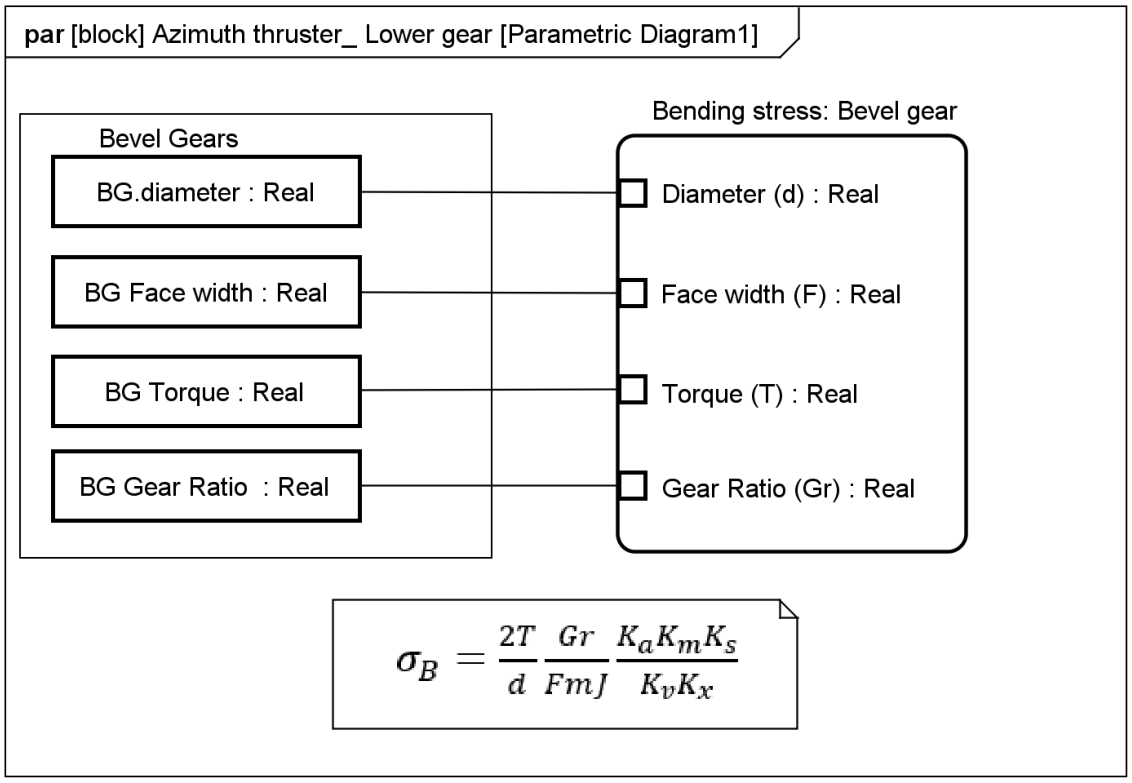


Figure 35. Parametric diagram for bending stress analysis of bevel gear in the azimuth thruster.

A constraint block for the bending stress of bevel gear is shown on the right side of the figure. The equation for the analysis is also presented in the diagram. For the analysis, various property values of components such as the gear ratio, torque, diameter and face width of the bevel gear are required. These required properties values are then imported from the bevel gear block from the block definition diagram of lower gear. More constraint blocks and properties for other analysis can be added to the diagram if its needed. Thus, the parametric diagram provides most of the necessary information for an analyst to perform the engineering analysis of the system.

6.2 Risk analysis methods

6.2.1 Fault Trees Analysis

A Fault Tree of the steering hydraulics unit failing in the azimuth thruster was prepared in the workshop using Edraw 9.1 software (Edrawsoft, 2018). The diagram is presented in Figure 36.

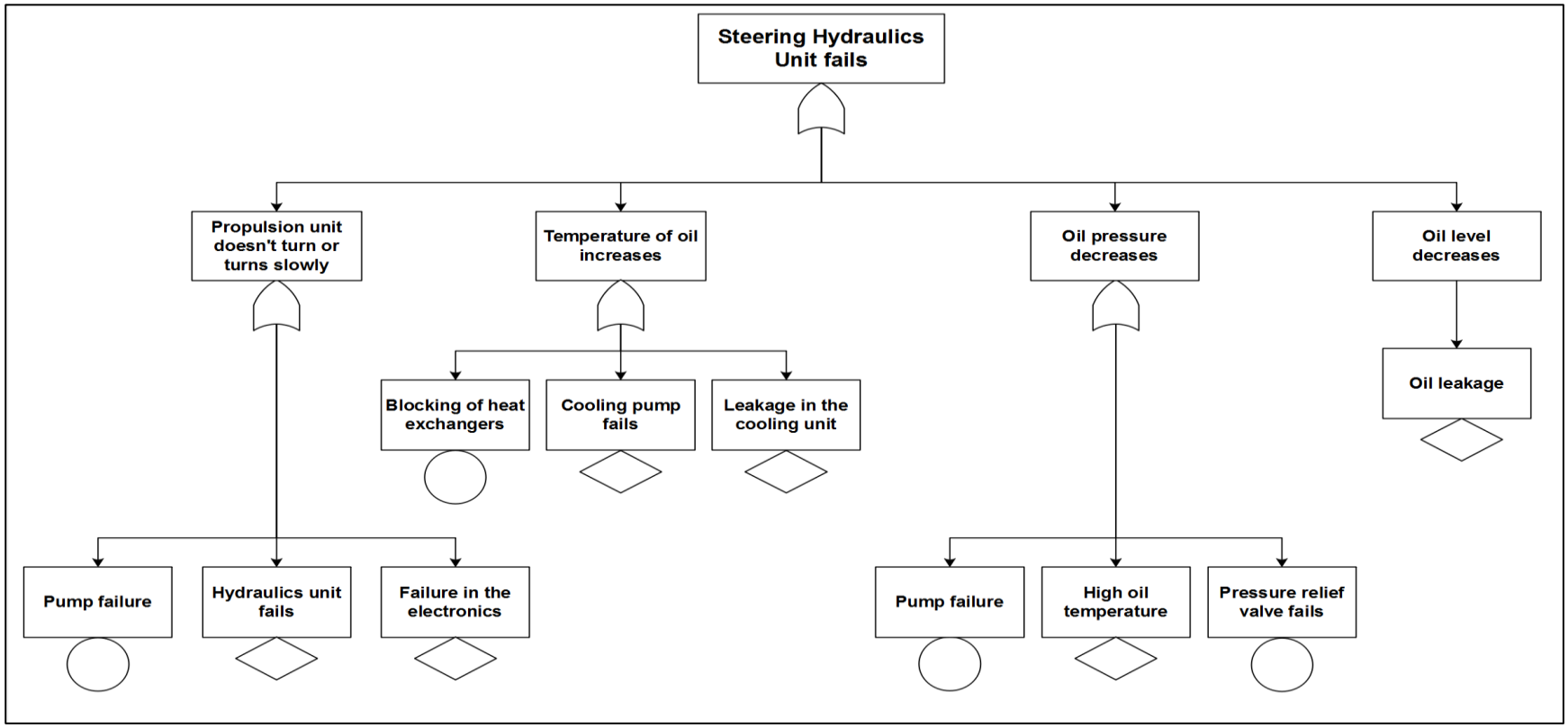


Figure 36. Fault Tree of the steering hydraulics unit failing in the azimuth thruster.

The primary level faults identified for the failure of the steering hydraulics unit were: propulsion unit not turning or turning slowly, the oil temperature increases, the oil pressure decreases and the oil level decreases. As either of these faults can result in the failure of the steering hydraulic unit, thus the OR gate symbol was used in the tree. These faults were then further developed as shown in the figure. The faults identified for the propulsion unit not turning or turning slowly were the failure of the pump or the failure of the hydraulic unit, or the electronics failure. The faults identified leading to the increase in oil temperature were the blocking of the heat exchangers or the failure of the cooling pump or leakage in the cooling unit. Furthermore, the faults identified for the decrease of oil pressure were the pump failure or high oil temperature or the failure of the pressure relief valve. Finally, the fault that leads to the decrease of oil pressure was identified as the oil leakage. The circle symbol in the diagram represents a basic event such as component failure, while the diamond symbol in the diagram represents undeveloped events which can be further developed into details if required.

6.2.2 Systems-Theoretical Process Analysis (STPA)

A short STPA analysis of the azimuth thrusters was conducted in the workshop. The procedure mentioned in the review section of this thesis was followed using RM studio beta software (Studio, unreleased).

Step 1: Establishing the foundation for the analysis.

The accidents related to the azimuth thruster were first identified. The identified accidents and their descriptions are listed in Table 5.

Table 5. The list of accidents related to the azimuth thruster and their description.

ID	Accident	Description
A1	Loss of the azimuth thruster.	The azimuth thruster gets detached from the vessel.
A2	Failure of the azimuth thruster.	The azimuth thruster fails.

Then all possible hazards that lead to these accidents were identified. The identified hazards are presented in Table 6. Furthermore, the description of hazards is presented in Table 7 and the graphical model of the relationship between accidents and hazards is presented in Figure 37.

Table 6. The list of identified accidents and hazards.

Accident	Hazards
A1. Loss of the azimuth thruster.	H1. Vessel grounding. H2. Vessel collision.
A2. Failure of the azimuth thruster.	H1. Vessel grounding. H2. Vessel collision. H3. Bearing failure. H4. Hydraulics failure.

Table 7. List of identified hazards and their description.

ID	Hazard	Description	Relationship
H1	Vessel grounding.	Grounding of vessel occurs.	A1, A2
H2	Vessel collision	Vessel collides with an object.	A1, A2
H3	Bearing failure	The bearing inside the azimuth thruster fails.	A2
H4	Hydraulics failure	The hydraulics unit of the azimuth thruster fails.	A2

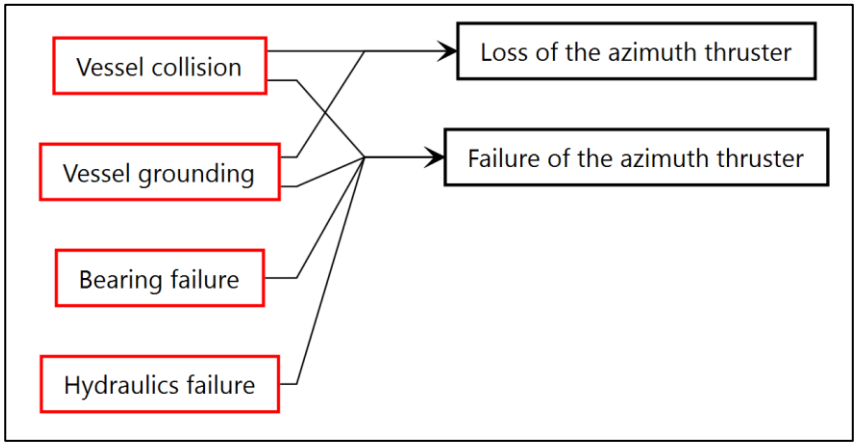


Figure 37. Relationship between hazards and accident scenarios.

Safety constraints that can prevent these hazards were then discussed and reported. The safety constraints for the hazards are presented in Table 8.

Table 8. Identified safety constraints for hazards.

Hazards	Safety Constraints
H1. Vessel grounding.	SC1. The vessel should be designed to withstand harsh weather conditions. SC2. Proper routing and constant weather and sea state monitoring. SC3. Hardware and software redundancy. SC4. Intensive software testing.
H2. Vessel collision.	SC1. The vessel should be designed to withstand harsh weather conditions. SC2. Object detection sensor and radars redundancy. SC3. Intensive testing of sensors and radars.
H3. Bearing failure.	SC1. Redundancy of components or units. SC2. Frequent maintenance. SC3. Proper Lubrication.
H4. Hydraulics failure.	SC1. Redundancy of components or units. SC2. Frequent monitoring and maintenance. SC3. Oil quality should always be monitored and maintained properly.

After listing safety constraints for hazards, the next step is to create a control structure of the system to identify all unsafe control actions in a system. The control structure of the azimuth thruster was prepared and is presented in Figure 38.

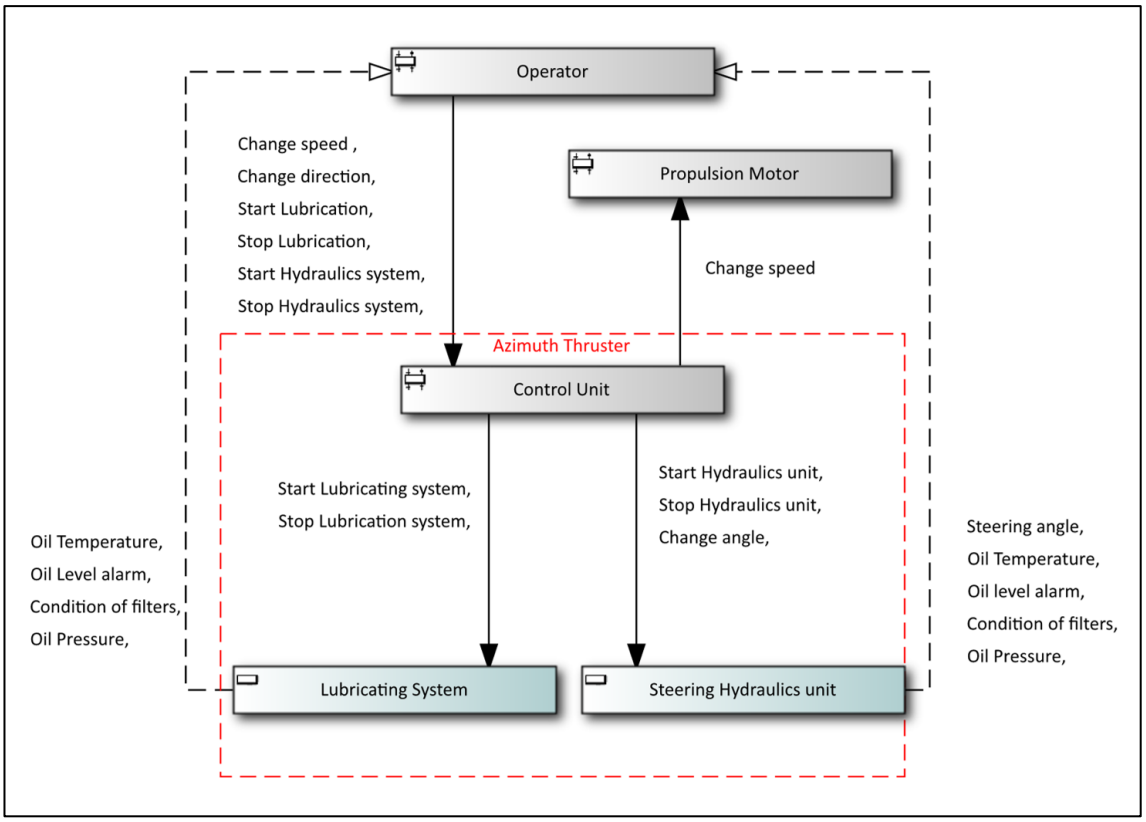


Figure 38. A safety control structure of the azimuth thruster.

The control structure models all sub-systems, units and controllers of the system and presents all possible interactions between the controllers and controlled units. The red dashed line in the structure denotes the boundary of the azimuth thruster; and the units that are owned by the azimuth thruster are placed inside this boundary. All the possible actions of the controllers such as operator and control unit, are presented in the control structure. Furthermore, the feedback that the controllers receive from the controlled units are also shown in the figure. For example, the control actions of an operator in the system are changing the speed, changing the direction, and turning on/off the lubrication system and the hydraulics unit. Moreover, the operator receives feedback from the lubricating system about the temperature of the oil, the condition of the filter, and the pressure and the level of the oil. Similarly, the operator receives feedback from steering hydraulics unit about the steering angle, the temperature of the oil, the condition of the filter, and the level and the pressure of the oil. Thus, all possible control actions of the controllers in a system are identified and are presented in the structure. The control actions in the figure are represented with solid line while the feedback is represented with dashed line.

Step 2: Identifying unsafe control actions.

After knowing the control actions of controllers in a system, the next step is then to identify the unsafe control actions among them. However due to time constraints, few control actions were considered in this case study. Table 9 presents the guidewords used in the workshop and identified unsafe control actions.

Table 9. Identified unsafe control actions.

Control action	Not providing the action causes a hazard	Providing the action causes a hazard.	Providing the action too late causes a hazard	Provided the action too early causes a hazard
Operator: Change speed	Not reducing the speed when needed.	Increasing speed without vessel's surrounding awareness.	Reducing speed too late.	
Operator: Change direction	Not changing direction when needed.	Changing direction without vessel's surrounding awareness.	Changing direction too late.	
Operator: Start Lubrication	Not lubricating the bearings and components			

Step 3: Create safety constraints and requirements for unsafe control action.

After identifying the unsafe control actions, safety requirements controlling the hazards were created. The safety constraints for these unsafe control actions are presented in Table 10.

Table 10. The safety constraints for the unsafe control actions.

Controller	Control action	Safety Constraints
Operator	Change speed	SC-FC1: The operator must reduce the speed when required. SC-FC2: The operator must change the speed with proper vessel's surrounding awareness.
Operator	Change direction	SC-FC3: The operator must change the direction when required. SC-FC4: The operator must change the direction with proper vessel's surrounding awareness.
Operator	Start lubrication	SC-FC5: The operator must lubricate the components when it is required.

Step 4: Determine how the unsafe control action could occur.

This step was skipped in the workshop due to time limitations. However, the procedure and its importance were discussed.

6.3 Experts Evaluation and Feedback

After the modeling session in the workshop, ratings and feedback for modeling approaches and risk analysis methods were collected. Experts were asked to evaluate the methods based on different criteria. The scale used for the ratings is presented in Table 11.

Table 11. Scale and color codes used for the evaluation of methods from experts.

0 – 1.0	Poor
1.1 – 2.0	Satisfactory
2.1 – 3.0	Average
3.1 – 4.0	Good
4.1 – 5.0	Excellent

6.3.1 Modeling Approaches

Tree structure method

The expert's evaluation for the Tree structure method is presented in Table 12.

Table 12. The expert's evaluation of the Tree structure method.

Tree structure method						
Criteria's	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	**Average rating**
Method complexity	5	4	4	4	4	**4.2**
Modeling time	4	4	3	3	4	**3.6**
System's structural modeling	2	3	3	2	4	**2.8**
System's behavior modeling	2	1	2	1	2	**1.6**
Functionality	4	1	3	2	3	**2.6**

As shown in the table above, this method received an excellent rating of 4.2 for complexity of the method. Furthermore, it received a rating of 3.6 for modeling time. However, the experts gave an average rating of 2.8 for the structural modeling and 2.6 for the functionality. Also, the method received a rating of 1.6 for modeling the behavior of the system.

Systems Modeling Language

The expert's evaluations for SysML are presented in Table 13.

Table 13. The expert's evaluations of Systems Modeling Language.

Systems Modeling Language						
Criteria's	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Average rating
Method complexity	2	2	3	3	2	**2.4**
Modeling time	1	2	3	2	2	**2.0**
System's structural modeling	5	4	3	4	5	**4.2**
System's behavior modeling	4	5	4	4	3	**4.0**
Functionality	3	4	4	4	3	**3.6**

As shown in the table, SysML received a rating of 2.4 for the method complexity and 2 for the modeling time. However, the experts gave a good rating of 3.6 for the functionality of the method. Furthermore, it received an excellent rating of 4.2 for the structural modeling and 4 for modeling the behavior of the system.

Comparison and feedback

The average ratings received by the Tree structure method and Systems Modeling Language are listed in Table 14.

Table 14. Comparison of the expert's evaluation between the Tree structure method and SysML.

Criteria's	Average Rating	
	Tree structure method	Systems Modeling Language
Method complexity	4.2	2.4
Modeling time	3.6	2.0
System's structural modeling	2.8	4.2
System's behavior modeling	1.6	4.0
Functionality	2.6	3.6

Most of the experts think that the Tree structure method is much easier to understand and to implement than SysML. Furthermore, they consider the Tree structure method to be more effective for simple systems in comparison to SysML. However, for complex ship systems, the Tree structure method lacks in providing details of required information such as components interaction and systems behavior. Moreover, the number of systems in ships are very high, thus linking together different tree structures of systems can be very difficult.

Considering SysML, the experts think that the results it produces are good, comprehensive and informative for complex ship systems. It manages to present behavior of the system and interactions between components in an understandable manner. Also, in their opinion the results of SysML are much useful than the Tree structure method as it can help operators, analysts and designers to understand the system better which can be further utilized to minimize risks and improve the system.

6.3.2 Risk Analysis Methods

Similarly, after the implementation of the risk analysis methods in the workshop, the ratings and opinions from experts were collected.

Fault Trees Analysis

The expert's evaluations for FTA are presented in Table 15.

Table 15. The expert's evaluations of FTA.

Fault Trees Analysis					
Criteria's	**Expert 1**	**Expert 2**	**Expert 3**	**Expert 4**	**Average rating**
Method complexity	4	4	4	5	**4.3**
Analysis time	4	3	5	5	**4.3**
Risk Identification	2	3	5	3	**3.3**
Risk Mitigation	1	3	1	3	**2.0**
Structure of the model	3	4	4	5	**4.0**

As shown in the table, FTA received an excellent rating of 4.3 for method complexity and analysis time. Furthermore, it received a rating of 4 for presenting the structural of the model. However, the method received an average rating of 3.3 for risk identification and 2.0 for risk mitigation.

Systems-Theoretical Process Analysis

The expert's evaluations of STPA are presented in Table 16.

Table 16. The expert's Evaluation of STPA.

Systems-Theoretical Process Analysis					
Criteria's	**Expert1**	**Expert 2**	**Expert 3**	**Expert 4**	**Average rating**
Method complexity	3	2	2	3	**2.5**
Analysis time	3	3	2	3	**2.8**
Risk Identification	4	3	3	4	**3.5**
Risk Mitigation	4	3	3	3	**3.3**
Structure of the model	4	2	4	3	**3.3**

STPA received a good rating of 3.5 for risk identification and 3.3 for risk mitigation and presenting the structure of the model. Furthermore, the experts gave an average rating of 2.8 for the analysis time and 2.5 for the method complexity.

Comparison and feedback
The average ratings of FTA and STPA are listed in Table 17.

Table 17. Average ratings in different criteria for FTA and STPA.

Criteria's	Average Rating	
	FTA	STPA
Method complexity	4.3	2.5
Analysis time	4.3	2.8
Risk Identification	3.3	3.5
Risk Mitigation	2.0	3.3
Structure of the model	4.0	3.3

The table shows that FTA received better rating than STPA in method complexity, model structure and analysis time. However, the experts rated STPA higher in risk identification and risk mitigation.

In expert's opinion, FTA was easier to understand and implement than STPA. Since, STPA consists of multiple modeling phases i.e. the analysis of hazards and analysis of all possible actions of controllers in a system, it requires significant amount of time than FTA. Moreover, FTA provides a good model when the focus is on component failures and combination of events in a system. However, they think that FTA lacks information in detail about component interactions issues and human errors. Although the experts rated FTA higher than STPA in the structure of the model, they mentioned that FTA diagrams for a complete vessel would be difficult to manage due to large number of complex systems.

7 Discussion and possible solutions

The case study results show that the traditional methods are easier to understand and implement than modern methods. However, it must be considered that the experts are familiar with traditional approaches, while modern methods are new to them. Furthermore, modeling or analyzing risks in a complete vessel can be different than on a single system as the model using traditional method is mostly huge in comparison to modern methods. Thus, the models for complete vessel using traditional method can be difficult to manage.

The Tree structure method is still widely used for presenting the graphical structural model of ship systems as it is simple and less time consuming, but it fails to provide behavior or requirements of a system. Furthermore, only understanding the structural composition is insufficient for understanding complex ship systems. The case study shows that SysML is better than the Tree structure method for modeling complex ship systems. However, handling the complexity of systems increases the complexity of the method itself and increases the modeling time significantly. SysML presents diagrams that can assist the design process of complex systems and validate the design afterwards. Moreover, these diagrams also help to understand the composition of the system, interactions among components, and their behavior; which can be useful in various aspects. For example, understanding the system better allows the operators to handle systems efficiently and helps analysts to identify risks in the system. Furthermore, it also provides support for performing the engineering analysis of systems which helps in the maintenance of the system and further design improvements.

Similarly, traditional risk analysis methods such as FTA are still dominant in marine industry. The review and the case study show that the FTA is a simple and effective method for systems when the focus is on component failures or human errors. However, the concern with FTA is that it doesn't consider all type of risks in the system but only covers the major risks that are known to the analysts. Moreover, they lack in analyzing all possible risks due to component interactions which is growing with time in complex ship systems and shouldn't be neglected. However, STPA analyzes higher number of possible risks due to control actions i.e. component interactions and human errors. Furthermore, STPA analysis provides a systematic approach for identifying and mitigating risks. The STPA can be also applied during the early phase of a system design that will help to reduce design errors.

As the overall functionality of SysML and STPA seems to be better than traditional methods, improving the drawbacks of these methods are important. The drawbacks of SysML and STPA identified in the review were the higher complexity of method and implementation time. The complexity of method can get better with several practices. Furthermore, the software tools for aiding the implementation process have just been developed, thus the tools will be further improved in the future. However, the analysis time consumption is also an important criterion for industries as they mostly have limited time resources because of increasing competition in the market. Some viable solutions to improve these drawbacks of SysML and STPA are presented below:

Finding links between a modeling approach and a risk assessment method.
The figures from case study show some similarities in SysML and STPA. Some of the features of SysML can be modified and used as input to STPA or vice versa. For example, the control structure of STPA shows the structure of the system and interactions (control actions and feedback) between the sub-systems and components. Similarly, SysML also includes a block definition diagram to present the structure of a system, and an activity diagram and internal block diagram to present the component interactions. It should be possible to use the diagram/model of one method as input in another method or even combining the methods. Hence, the links between a modeling approach and a risk assessment method to reduce the analysis time should be further studied.

Possibility of creating a database in a STPA and SysML software tools.
A feature of creating a database is widely used by modern software's in different fields. For example, in 3D modeling, a database is created which stores all the models created and uploaded in the software; which then allows other users to download it afterwards instead of creating the same model again. Hence in SysML a database containing all modeling elements such as blocks, requirements and constraints can be created which can be utilized to model another similar vessel or system. Similarly, in STPA, some of the hazards can be very similar between vessels or systems. Hence a database that stores the elements such as control structure can also be created.

A database has potential to reduce the modeling and risk analysis time consumption. Furthermore, downloading complex elements instead of creating them can reduce the complexity of method. Thus, a research on this topic should be done in the future.

Possibility of creating codes to generate diagrams.
Instead of using a graphical interface to generate diagrams, writing codes to generate diagrams can be easier and faster. A template for generating the diagrams can be created where the users can use coding to provide the contents for the diagrams. As an example, a code was created to replicate the model of the block definition diagram of the lower gear in the azimuth thruster which was prepared in the workshop. The code was compiled using Graphviz software and model was generated. The comparison of these two models from the workshop and the code is shown in Figure 39.

As shown in the figure, the code successfully replicates the block definition diagram of SysML created with Astah SysML software (Apache, 2016). Creating the code required a longer time than using the graphical interface. However, a template was then created out of the code; and was made in such a way that it can be edited and used for creating block definition diagram for any complex systems. By using the template, the model generation time was reduced almost by half in comparison to Astah SysML. The template of the codes for block definition diagram of complex systems and codes for generating block definition diagram for the lower gear of the azimuth thruster are presented in Appendix A and Appendix B respectively.

Similarly, the control structure of STPA can also be created using the codes. Furthermore, Coding languages such as C# uses a feature called "Intellisense" that displays all possible syntax automatically if a coder writes the starting letters of the syntax. Features likes this, makes it even much easier and faster to write codes. Hence, using code templates instead of a graphical tool has a good potential of reducing the complexity of method and modeling time for complex system.

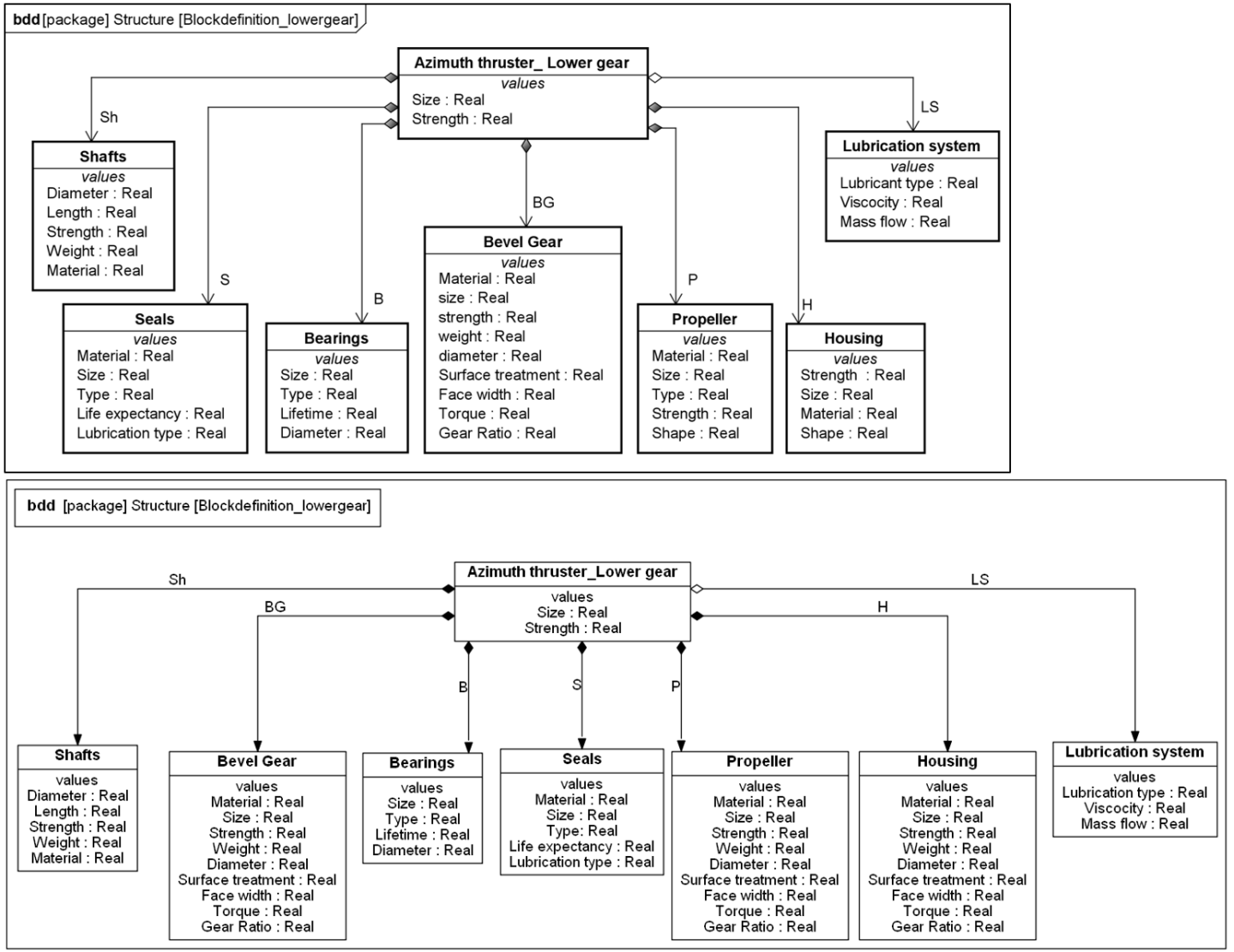


Figure 39. A comparison of the models from graphical interface with Astah SysML (Top) and with Code using Graphviz (Bottom).

Schematic layout of the system under assessment in a modeling tool.
One of the limitations identified during this research was the unavailability of the layout of the system being analyzed inside the tool itself. The analyst must check the layout of the system constantly for generating SysML diagrams. It will be easier if the tool contains a separate window which can be toggled on/off to display the schematic layout of the system. The analyst can then toggle on the window whenever he requires during the modeling process.

8 Research conclusions

The complexity in ship systems is further increasing. Traditional methods were not designed to handle complex ship systems since they were designed for the relatively simpler systems of the past. Although they have been modified to adapt to the increasing complexity, they still lack at some extent in overall functionality when implemented to a complex ship system. It is very unlikely to have an effective method for a complex system that is simple and less time consuming than traditional methods.

Based on the literature review, case study, and evaluations of experts, Table 19 and
As presented in the Table 19, the Tree structure method is a simple and less time consuming than SysML for modeling a complex ship system. Also, it produces similar level of detail as SysML for presenting structural composition of a system. However, it doesn't present the behavior of the system components and interactions between them, which is provided in detail with Activity diagram and Internal block diagram in SysML. Furthermore, unlike SysML, the Tree structure method doesn't include any model to present the requirements of a system. Moreover, the level of support for conducting an engineering analysis using the Tree structure method is low since it only provides the structural composition of the system. On the other hand, SysML provides high support for engineering analysis through parametric diagram.

Considering the drawbacks and functionality of each method, the research concludes that SysML is highly suitable for modeling complex ship system than Tree structure method. Although, the Tree structure method is simple and less time consuming than SysML, the functionality SysML offers is very high in comparison to the Tree structure method.

Table *20* present the main research conclusions of modeling approaches and risk analysis methods respectively. The scale used in conclusion tables is presented in Table 18.

Table 18. Scale used in conclusion tables.

None
Low
Average
High
Very high

Table 19. An overall research conclusion for the Tree structure method and SysML.

Question	Tree structure method	SysML
What is the complexity of the method?	Low	High
How much time is required for the method implementation?	Low	Very high
What is the level of detail for presenting structural composition of a system?	High	High
What is the level of detail for presenting behavior of system components and interactions between them?	None	Very high

What is the level of detail for presenting requirements of a system?	None	High
What is the level of support for conducting an engineering analysis?	Low	High
How suitable is the method for modeling complex ship system?	Low	High

As presented in the Table 19, the Tree structure method is a simple and less time consuming than SysML for modeling a complex ship system. Also, it produces similar level of detail as SysML for presenting structural composition of a system. However, it doesn't present the behavior of the system components and interactions between them, which is provided in detail with Activity diagram and Internal block diagram in SysML. Furthermore, unlike SysML, the Tree structure method doesn't include any model to present the requirements of a system. Moreover, the level of support for conducting an engineering analysis using the Tree structure method is low since it only provides the structural composition of the system. On the other hand, SysML provides high support for engineering analysis through parametric diagram.

Considering the drawbacks and functionality of each method, the research concludes that SysML is highly suitable for modeling complex ship system than Tree structure method. Although, the Tree structure method is simple and less time consuming than SysML, the functionality SysML offers is very high in comparison to the Tree structure method.

Table 20. An overall research conclusion for the FTA and STPA.

Question	FTA	STPA
What is the complexity of the method?	Average	High
How much time is required for the method implementation?	Average	Very high
What is the level of effectiveness for identifying risks due to the component failures?	High	High
What is the level of effectiveness for identifying risks due to the component interactions?	Average	Very high
What is the level of effectiveness for identifying risks due to human errors?	None	High
What is the level of support for a systemic and systematic analysis?	Average	Very high
How suitable is the method for analyzing risks in a complex ship system when risk identification is prioritized?	Average	High
How suitable is the method for analyzing risks in a complex ship system when available resources are critical?	High	Average

As presented in Table 20, the research concludes that the FTA is easier and less time consuming than STPA. It is because of its foundations and less detailed analysis, in comparison to STPA. Furthermore, the identification of risks due to the component

failures in a system are similar with both methods. Although both methods can identify risks due to the component interactions, STPA has a potential to identify most of the component interaction issues as the analyst has to analyze all possible component interactions in a system; whereas in FTA the identification is limited to the knowledge of the analyst and to the system level faults being analyzed. Thus, STPA identifies more component interaction issues than FTA. Moreover, STPA identifies possible human errors in a system such as unsafe control actions from operators or any controllers in a system and unsafe design issues, which is lacking in FTA. Also, the mentioned procedure of STPA ensures that the analysis is systemic and systematic. Whereas, in FTA, the analysis process is systematic if conducted properly; but conducting a systemic FTA for modern complex systems, with several sub systems and components, will create large diagrams which will be difficult to manage.

Since the interactions between components is increasing in a system and also design errors may increase due to autonomous functions being implemented, the research concludes that STPA is better suited for analyzing risks in a complex ship system than FTA when the end results of the methods are prioritized. However, it also must be considered that the available resources for risks analysis of ship systems will vary between companies. Thus, if the available resources for the analysis i.e. time resources, human resources and financial resources are critical, then FTA is better suited than STPA. Hence, the selection between FTA and STPA for a risk analysis of a complex ship system depends upon the availability of the resources for the analysis and the results required or expected with the analysis.

9 Future research possibilities

This section presents some topics that should be studied in future to improve the research results.

Detailed comparison considering the effectiveness of methods at initial design stages.
The effectiveness of methods at the initial design stages of system is different for each method. If a method can be implemented during the initial stage and is effective, then it can help designers to achieve a better and safer design of a system from the beginning. Since, the resources required for the design changes such as human resources, time resources and financial resources can be high, it is better to consider the behavior, functions and risks in a system from the initial design stage. Furthermore, due to the higher resources required for the design changes, some designers and companies may defend their design and compromise between functionality of a system and the risks in a system. Thus, a review and comparison considering the effectiveness of the methods at the initial design changes is necessary.

Research on method's adaptation to the design changes.
A ship system's design has been evolving in the past, and with ongoing autonomous ships projects, systems will keep evolving in the future. As the modeling and risks analysis of complex ship systems are lengthy processes, implementing the methods again from scratch after every design change can be tedious and costly. It would be beneficial for the companies and analysts if the earlier models could be modified according to the design changes. In that case, a version control system that keeps track and can manage earlier versions is also required. Hence, a study about the possibility of method's adapting to the design changes should be researched in future.

Review of methods with a proper weight factor assigned to each criterion.
It was realized during this research that a detailed review of these methods providing a proper weight factor for comparison criteria is necessary. The comparison of methods in this research was done with equal weight for each criterion. For example, all criteria used for the comparison, while evaluating by experts, such as risk analysis, risk mitigation and method complexity, were given equal priority when comparing the STPA and FTA method. This assumption that all criteria provided in the workshop are equally important for evaluating a method is improper and was made due to time restrictions for this research work.

If the weight factor for each criterion is assessed properly before evaluation, then more accurate results will be obtained for method's evaluation. Furthermore, some important criteria such as the cost of implementation and available resources for analysis were not considered in this research either. Hence, a research with the addition of these mentioned criteria and including proper weight factor should be conducted in future.

Review of methods with the consideration of probability.
As mentioned in the limitations to this thesis, probabilistic methods were not considered for the review in this research due to the lack of data about the failure of ship systems. However, the feature of assessing the probability of occurrence for risks in ship systems is very important as it helps to have more focus on critical risks. As a result, it will increase the effectiveness of the risk analysis method and modeling approach as more resources can be allocated for critical risks or elements in the system. Hence, a review of methods

including the probabilistic methods can be viable if data about ship systems can be accessed.

10 Bibliography

AAWA, 2016. Remote and Autonomous Ships - The next steps. [Online] Available at: http://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/aawa-whitepaper-210616.pdf
[Accessed 27 March 2018].

Abdukhaleq, A. & Wagner, S., 2013. *Experiences with Applying STPA to Software-Intensive Systems in the Automotive Domain,* Stuttgart: s.n.

Ahvenjärvi, S., 2001. Failure mode and effect analysis of automation system of ships. *Transations on the Built Environment,* Volume 53, pp. 401-407.

Alverbro, K., Nevhage, B. & Erdeniz, R., 2010. *Methods for Risk Analysis,* Stockholm: US AB.

Apache Software Foundation, 2016. *Astah SysML 1.4.* [Online] Available at: http://astah.net/editions/sysml
[Accessed 22 April 2018]

Banda, O. A. V. & Kannos, S., 2018. *Hazard Analysis Process For Autonomous Vessels,* s.l.: s.n. [Online] Available at: http://www.aboamare.fi/_media_25402_/R&D%20kuvat/ÄlyVESI/Tulokset%20201805/Hazard-Analysis-Process-for-Autonomous-Vessels.pdf
[Accessed 25 July 2018].

Barton, T., Fralick, C. & Touchton, R., 2017. *Sea Hunter and Maritime Autonomy.* [Online] Available at: http://www.ukmarinealliance.co.uk/sites/default/files/MASRWG2017/13_Tim%20Barton%20-%20Sea%20Hunter%20and%20Maritime%20Autonomy%20-%20DISTAR%20Case%2027536%20-%20FINAL.pdf
[Accessed 28 March 2018].

Blanchard, B. S. & Blyer, J. E., 2016. *System Engineering Management.* 5th ed. New Jersey: John Wiley & Sons.

Board, J. S. R., 2000. *Report on the loss of the Mars Polar Lander and Deep Space 2 Missions,* California: California Institute of Technology.

Cristea, G. & Constantinescu, D., 2017. *A comparative critical study between FMEA and FTA risk analysis methods.* s.l., s.n.

Daffey, K., 2017. *Project SISU and the future for Autonomous Ships.* [Online] Available at: https://www.dma.dk/Vaekst/autonomeskibe/Documents/Kevin%20Daffey,%20Rolls-Royce,%20presentation%20of%20the%20Project%20SISU%20and%20the%20future%20of%20Autonomous%20Ships.pdf
[Accessed 28 March 2018].

Dekker, S., 2011. *Drift into Failure : From Hunting Broken Components to Understanding Complex Systems.* Surrey ; Burlington: Ashgate Publishing Limited.

Edrawsoft, 2018. *Edraw Max 9.1.* [Online] Available at: https://www.edrawsoft.com/edraw-max.php
[Accessed 27 April 2018]

Ericson, C. A., 2015. *Hazard analysis techniques for system safety.* 2nd ed. New Jersey: John Wiley & Sons.

Flugunfalluntersuchung, B. f., 2004. *Investigation Report,* Braunschweig: s.n.

Friedenthal, S., Moore, A. & Steiner, R., 2015. Getting Started with SysML. In: *Practical Guide to SysML - The systems Modeling Language* . s.l.:s.n., pp. 31-51.

Grobshtein, Y., Perelman, V., Safra, E. & Dori, D., 2007. *Systems Modeling Languages: OPM Versus SysML.* s.l., International Conference on Systems Engineering and Modeling, pp. 102-109.

Herzog, E. & Pandikow, A., 2005. *SysML-an Assessment,* Stockholm: INCOSE.
Kongsberg, n.d. *Autonomous ship project, key facts about Yara Birkeland.*
[Online] Available at:
https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/4B8113B707A50A4FC125811D00407045?OpenDocument
[Accessed 2 April 2018].

Lee, W. S., Grosh, D. L., Tillman, F. A. & Lie, C. H., 1985. Fault Tree Analysis, Methods, and Applications - A Review. *IEEE TRANSACTIONS ON RELIABILITY,* R-34(3), pp. 194-200.

Levander, O., 2016. *Ship Intelligence - A new era in shipping.* [Online] Available at: http://www.fathomshippingevents.com/uploads/2/5/3/9/25399626/ship_intelligence_-_a_new_era_in_shipping__2016-03-08_.pdf
[Accessed 26 March 2018].

Leveson, N., 2015. *An STPA Primer.* s.l.:s.n.

Leveson, N., Dulac, N., Marais, K. & Carroll, J., 2009. Moving Beyond Normal Accidents and High Reliability Organizations: A Systems Approach to safety in Complex Systems. *SAGE,* 30(2-3), pp. 227-249.

Leveson, N. G., 2011. *Engineering a Safer World- Systems Thinking Applied to Safety.* London: MIT Press.

Leveson, N. G., Stringfellow, M. V. & Owens, B. D., 2010. *" Safety-Driven Design for Software-Intensive Aerospace and Automotive Systems",* s.l.: Institute of Electrical and Electronics Engineers.

Modelio, c., 2018. *Modelio 3.7.* [Online] Available at: https://www.modelio.org/
[Accessed 13 May 2018]

Mukhopadhyay, S., Hastak, M., & Halligan, J. 2014. *Compare and contrast major nuclear power plant disasters: lessons learned from the past.* [Online] Available at: https://pdfs.semanticscholar.org/f839/f268589033e91843926fe6a458734225bcbd.pdf [Accessed 12 March 2018].

MUNIN, 2014. *Conducting look-out on an unmanned vessel: Introduction to the advanced sensor module for MUNIN's autonomous dry bulk carrier..* [Online] Available at: http://www.unmanned-ship.org/munin/wp-content/uploads/2014/09/MUNIN-ISIS-final-online.pdf [Accessed 23 March 2018].

MUNIN, 2016. *Research in Maritime Autonomous Systems Project Results and Technology Potentials.* [Online] Available at: http://www.unmanned-ship.org/munin/wp-content/uploads/2016/02/MUNIN-final-brochure.pdf [Accessed 22 03 2018].

Mushtaq, F. & Chung, P., 2000. A systematic HAZOP procedure for batch processes, and its application to pipeless plants. *Journal of Loss Prevention in the Process Industries,* 13(1), pp. 41-48.

PEASSS, n.d. *PEASSS Product tree.* [Online] Available at: http://www.peasss.eu/index.php/project [Accessed 16 July 2018].

Perez, A. R., Antonio, C. A. T. & Consunji, R. J., 2011. The sinking of the MV Doña Paz - A Critique on Maritime Disaster. *ACTA MEDICA PHILIPPINA,* 45(3).

Perrow, C., 1999. Introduction. In: *Normal Accidents.* New Jersey: Princeton University Press, p. 4.

Rolls-Royce, 2016. *Autonomous ships - The next step.* [Online] Available at: http://www.rolls-royce.com/~/media/Files/R/Rolls-Royce/documents/customers/marine/ship-intel/rr-ship-intel-aawa-8pg.pdf [Accessed 22 March 2018].

Rolls-Royce, 2016. *Rolls-Royce to supply first automatic crossing system to Norwegian ferry company Fjord1.* [Online] Available at: https://www.rolls-royce.com/media/press-releases/yr-2016/18-10-2016-rr-to-supply-first-automatic-crossing-system-to-norwegian-ferry-company-fjord1.aspx [Accessed 23 March 2018].

Rolls-Royce, 2017. *Rolls-Royce and Mitsui O.S.K. lines to work together to develop intelligent awareness for ships.* [Online] Available at: https://www.rolls-royce.com/media/our-stories/press-releases/2017/21-12-2017-and-mitsui-osk-lines-to-work-together-to-develop-intelligent-awareness-for-ships.aspx [Accessed 26 March 2018]

Rothblum, A. M., 2000. *Human error and marine safety,* Orlando: In National Safety Council Congress and Expo.

Sage, A. P. & Aarmstrong, J. J. E., 2000. Tree structure. In: *Introduction to Systems Engineering.* s.l.:John Wiley & Sons, p. 214.

Sanford, F. & Oster, C., 2016. Applying SysML and a Model-Based Systems Engineering Approach to a Small Satellite Design. In: J. Hsu & R. Curran, eds. *Advances in Systems Engineering.* s.l.:American Institute of Aeronautics and Astronautics.

Smita, 2016. *What is Formal Safety Assessment in Shipping?.* [Online] Available at: https://www.marineinsight.com/marine-safety/what-is-formal-safety-assessment-in-shipping/
[Accessed 5 April 2018].

Stiki e.h.f., unreleased. *RM studio beta.* [Online] Available on request at: https://www.riskmanagementstudio.com/features11/
[Accessed 18 May 2018].

Sundararajan, C., 2012. *cedengineering- Fault Tree Construction in Reliability Engineering.* [Online] Available at: https://www.cedengineering.com/userfiles/Construction%20in%20Reliability%20Engineering.pdf
[Accessed 5 July 2018].

APPENDIX A: The template of the code for creating block definition diagram of SysML in Graphviz software.

```
digraph G{
    splines = ortho;
    overlap = false;
    graph[fontname=arial]
    node [shape=record style=filled fillcolor=white    fontname=arial];
    edge [fontname=arial arrowhead=normal, arrowtail=diamond, dir=both];
        subgraph cluster_1{
        // use struct 0 for creating the diagram header, struct 1 for creating the system block,
        //and others for components block
        struct0 [label= < <B> bdd </B> Diagram header > fontsize="14" pos="-1,1.9!"];

        struct1 [label=<{<B> The system </B>|values<BR/>Property : Real<BR/>
                    Property : Real<BR/>Property : Real<BR/>Property : Real}> pos="0,1.65!"];

        struct2 [label=<{<B> Component1</B>|values<BR/>Property : Real<BR/>
                    Property : Real<BR/>Property : Real<BR/>Property : Real}> pos="-1.32,1.1!"];

        struct3 [label=<{<B> Component2 </B>|values<BR/>Property : Real<BR/>
                    Property : Real<BR/>Property : Real<BR/>Property : Real}> pos="-0.84,1!"];

        struct4 [label=<{<B> Component3 </B>|values<BR/>Property : Real<BR/>
                    Property : Real<BR/>Property : Real<BR/>Property : Real}> pos="-0.4,1.1!"];

        struct5 [label=<{<B> Component4 </B>|values<BR/>Property : Real<BR/>
                    Property : Real<BR/>Property : Real<BR/>Property : Real}> pos="0.025,1.09!"];

        struct6 [label=<{<B> Component5 </B>|values<BR/>Property : Real<BR/>
                    Property : Real<BR/>Property : Real<BR/>Property : Real}> pos="0.5,1!"];

        struct7 [label=<{<B> Component6 </B>|values<BR/>Property : Real<BR/>
                    Property : Real<BR/>Property : Real<BR/>Property : Real}> pos="1,1!"];

        struct8 [label=<{<B> Component7 </B>|values<BR/>Property : Real<BR/>
                    Property : Real<BR/>Property : Real<BR/>Property : Real}> pos="1.5,1.15!"];}
        //rest of the codes are for connecting the blocks //
        struct1->struct2 [xlabel="label for component1"];
        struct1->struct3 [xlabel="label for component2"];
        struct1->struct4 [xlabel="label for component3" arrowhead=diamond, arrowtail=normal, dir=both];
        struct1->struct5 [xlabel="label for component4" arrowhead=diamond, arrowtail=normal, dir=both];
        struct1->struct6 [xlabel="label for component5" arrowhead=diamond, arrowtail=normal, dir=both];
        struct1->struct7 [xlabel="label for component6"];
        struct1->struct8 [xlabel="label for component7" arrowhead=normal, arrowtail=odiamond, dir=both];
}

```

APPENDIX B: Code for creating block definition diagram for lower gear of azimuth thruster in Graphviz software.

```
digraph G{
    splines = ortho;
    overlap = false;
    graph[fontname=arial]
    node [shape=record style=filled fillcolor=white fontname=arial];
    edge [fontname=arial arrowhead=normal, arrowtail=diamond, dir=both];
        subgraph cluster_1{

        struct0 [label= < <B> bdd </B> [package] Structure [Blockdefinition_lowergear] >
                    fontsize="14" pos="-1,1.9!"];
        struct1 [label=<{<B> Azimuth thruster_Lower gear </B>|values<BR/>Size : Real<BR/>
                    Strength : Real}> pos="0,1.65!"];
        struct2 [label=<{<B> Shafts </B>|values<BR/>Diameter : Real<BR/>Length : Real<BR/>
                    Strength : Real<BR/>Weight : Real<BR/>Material : Real}> pos="-1.32,1.1!"];
        struct3 [label=<{<B> Bevel Gear </B>|values<BR/>Material : Real<BR/>Size : Real<BR/>
                    Strength : Real<BR/>Weight : Real<BR/>Diameter : Real<BR/>Surface treatment : Real
                    <BR/>Face width : Real<BR/>Torque : Real<BR/>Gear Ratio : Real}> pos="-0.84,1!"];
        struct4 [label=<{<B> Bearings </B>|values<BR/>Size : Real<BR/>Type : Real<BR/>Lifetime : Real
                    <BR/>Diameter : Real}> pos="-0.4,1.1!"];
        struct5 [label=<{<B> Seals </B>|values<BR/>Material : Real<BR/>Size : Real<BR/>Type: Real
                    <BR/>Life expectancy : Real<BR/>Lubrication type : Real}> pos="0.025,1.09!"];
        struct6 [label=<{<B> Propeller </B>|values<BR/>Material : Real<BR/>Size : Real<BR/>Strength : Real
                    <BR/>Weight : Real<BR/>Diameter : Real<BR/>Surface treatment : Real
                    <BR/>Face width : Real<BR/>Torque : Real<BR/>Gear Ratio : Real}> pos="0.5,1!"];
        struct7 [label=<{<B> Housing </B>|values<BR/>Material : Real<BR/>Size : Real<BR/>Strength : Real
                    <BR/>Weight : Real<BR/>Diameter : Real<BR/>Surface treatment : Real
                    <BR/>Face width : Real<BR/>Torque : Real<BR/>Gear Ratio : Real}> pos="1,1!"];
        struct8 [label=<{<B> Lubrication system </B>|values<BR/>Lubrication type : Real<BR/>Viscocity : Real
                    <BR/>Mass flow : Real}> pos="1.5,1.15!"];}
        struct1->struct2 [xlabel="Sh"];
        struct1->struct3 [xlabel="BG"];
        struct1->struct4 [xlabel="B" arrowhead=diamond, arrowtail=normal, dir=both];
        struct1->struct5 [xlabel="S" arrowhead=diamond, arrowtail=normal, dir=both];
        struct1->struct6 [xlabel="P" arrowhead=diamond, arrowtail=normal, dir=both];
        struct1->struct7 [xlabel="H"];
        struct1->struct8 [xlabel="LS" arrowhead=normal, arrowtail=odiamond, dir=both];
}

```