NTNU
Norwegian University of
Science and Technology

Empirical Studies of Safety and Security Co-analysis of Autonomous Systems

Erik Nilsen Torkildson

Master of Science in Informatics
Submission date: May 2018
Supervisor: Jingyue Li, IDI

Norwegian University of Science and Technology
Department of Computer Science

Preface

This thesis is the concluding part of my 2-year Master's degree in Informatics with the field of study in software engineering. With finalization of this project in June of 2018, the author is graduating from the Department of Computer Science at University of Science and Technology (NTNU). This master thesis was written in the fall of 2017 and spring of 2018 at Gløshaugen, Trondheim.

The motivation behind this thesis is to try to find appropriate methods for security and safety co-analysis of autonomous systems. With the case study of an Autonomous Boat model with the name Revolt.

I would like to thank my main supervisor in this master thesis project, Associate Professor Jingyue Li (Bill). I would also like to thank Jon Arne Glomsrud at DNV GL for valuable knowledge, and for allowing me the opportunity to work on the Revolt model. Furthermore, I want to thank Dr. Stig Ole Johnsen at SINTEF/ NTNU for valuable input during the technical work.

Also, many thanks to Professor Frank Lindseth at NTNU for valuable knowledge, and for allowing me the opportunity to work on setting up the software for the autonomous cars.

Finally, I would like to thank my family, friends and fellow students for the support and valuable input to my work.

Trondheim, May 31th 2018 *Erik Nilsen Torkildson*

Abstract

In recent years, the development of autonomous vehicles has an increase in multiple domains e.g. autonomous cars, boats, drones or aerial vehicles for military purposes. These vehicles might solve many current problems related to logistics, environment issues and safety. However, what is common for all of these autonomous systems is that they rely heavily on communication with low to no downtime and this gives challenges related to safety and security.

There have been very few case studies on autonomous vehicles related to safety or security. In addition, no method has been proven to work well for analyzing safety and security on autonomous systems.

Therefore, in thesis I will explore three safety and security co-analysis methods: STPA plus STPA-Sec, FMVEA, and CHASSIS to see if any of these methods are applicable for autonomous systems. An autonomous boat i.e. the Revolt developed by DNV GL [1], is used as for an empirical case study, to compare applicability, efficiency, and safety and security related hazards/threats identified by the three methods.

The results of the case study revealed weaknesses and strengths of each method to analyze autonomous systems with different levels of autonomy, based on this experience, suggestions for improvements are made. In particular, I recommend using security analysis and the target assets as the starting point for a safety and security co-analysis. I investigated my recommendation through an improved STPA and STPA-sec analysis, and detected more system hazards and vulnerabilities. I have published a research paper based on the work of this project at the European Safety and Reliability Conference (ESREL) 2018 [2], which will be held by NTNU.

Key words: Autonomous systems, Cyber-physical systems, Safety and Security co-analysis methods.

Sammendrag

Det er i dag blitt utviklet autonome droner, biler, båter og tog. Men hva de alle har til felles er evnen til å ha ulike nivåer av autonom funksjonalitet. Nøkkel funksjonaliteten i et autonomt system er situasjons-klassifisering og beslutningen som følger når en uventet hendelse oppstår. Det som tidligere var menneskets ansvar, er nå i forskjellige nivå systemet sitt ansvar, og dette gjør disse type system enda mer selvstendig.

Når man nevner de to begrepene "safety" og "security", når det gjelder autonome system, kan man tro at det er en bestemt mening og hensikt for dem. Sannheten er at i forhold til hvilket fagfelt det gjelder, så kan disse begrepene være knyttet til forskjellige betydninger. Derfor kan tvetydigheter være en utfordring når man utvikler metoder for å ivareta sikkerheten på forretningskritiske systemer.

Revolt-plattformen er et autonomt båtkonsept i en liten skala, utviklet og implementert av DNV GL. Revolt er en testplattform for en autonom ferge, med det formål å transportere last med minimalt energiforbruk og kostnader forbundet med det. Revolt fungerer også som en tidlig testplattform for sensorer og styresystemer dedikert til autonome fartøy.

Hvis utviklerne av autonome systemer velger å kombinere sikkerhets- og sikkerhetsanalyse, er det betydelige fordeler involvert. Både bransjer og forskere har begynt å erkjenne behovet for å fylle gapet mellom disse to områdene. Det har derfor blitt utviklet analysemetoder i nyere tid som er utformet for det formålet, og kan sannsynligvis brukes til å analysere sikkerhet for autonome systemer.

I dette prosjektet vil jeg utforske ulike metoder for å ivareta både “safety” og “security” på autonome systemer. Med en case-studie på et autonomt båtkonsept for frakt av varer, Revolt-plattformen, med fokus på å sammenligne om de ulike sikkerhetsmetoder er egnet for bruk på Revolt-plattformen og autonome systemer generelt. Ut fra dette arbeidet så vil jeg komme med forslag til forbedringer av eksisterende metoder, spesielt når det gjelder “security”.

Table of Contents

Preface i

Abstract ii

Sammendrag iii

List of Figures viii

List of Tables x

List of Abbreviations xiii

Chapter 1 : Introduction 1

1.1 Motivation 2

1.2 Research questions 2

1.3 Research design and tasks 3

1.4 Structure of the thesis 4

Chapter 2 : Literature Review of Cyber-physical systems and Autonomous systems 5

2.1 Cyber-physical systems 5

2.1.1 A simple example of CPS system 6

2.1.2 CPS systems in the Automotive domain 7

2.2 Autonomous systems 8

2.2.1 History 9

2.2.2 Background 9

2.2.3 Hardware of Autonomous system 10

2.2.4 Software of Autonomous system 11

2.2.5 Difference between autonomous and automatic 12

2.2.6 Area of autonomous domains 13

2.2.6.1 Ground 13

2.2.6.3 Trains 14

2.2.6.4 Maritime 15

2.2.6.5 Air 18

2.2.6.6 Comparison of Autonomous Systems from different domains 20

2.2.7 Challenges with Autonomous Systems 21

2.2.8 Autonomy related perspectives 22

Chapter 3 : Safety and Security analysis 23

3.1 Safety and Security analysis methods 23

3.1.1 Safety Analysis 23

3.1.2 Security Analysis 24

3.1.3 Safety and security co-analysis methods 24

3.1.3.1 Goals and analysis targets 24

3.1.3.2 Combining security and safety analysis 26
3.1.4 Qualitative and quantitative analysis 27
3.1.5 Risk Management 27
3.1.6 Functional safety and cyber-security ambiguities, overlap and differences 28
3.2 Safety and Security Standards specific to CPS systems 31
3.3 Safety and security of Automotive Systems 31
3.3.1 Safety in Automotive Systems 31
3.3.2 Cyber-security in Automotive Systems 32
3.4. Safety and Security of Autonomous Systems 34
3.4.1 Safety in Autonomous Systems 34
3.4.1.1 Tesla driver killed in crash with Autopilot active 35
3.4.2 Security of Autonomous Systems 36
3.4.2.1 Jeep recalled 1.4 million Cherokees 37
3.4.2.2 Security vulnerabilities on the Parrot AR.Drone 2.0 39
3.4.2.3 GPS spoofing or jamming attack – Manipulation of 20 ships in the Black Sea 40
3.4.3 Methods to combine security and safety analysis in autonomous systems 41
Chapter 4 : Research Motivation and Research Questions 42
4.1 STPA and STPA-sec 43
4.1.1 STAMP 43
4.1.2 STPA 44
4.1.3 STPA-Sec 45
4.1.4 Comparison between STPA and STPA-sec 47
4.1.5 Related work with the STPA-sec 48
4.1.6 Supported tools for STPA 48
4.2 FMVEA 49
4.3 CHASSIS 50
4.4 Research Motivation 51
4.5 Research Questions 52
Chapter 5 : Research Design and Case study 53
5.1 Introduction 53
5.1.1 Overview of the Revolt platform 54
5.1.2 Software 55
5.1.3 Component list 56
5.1.4 Communication flow diagram 57
5.1.5 Communication system 57
5.1.6 Wiring diagram of the Revolts system 58

5.1.7 I/O list Revolt 59
5.1.6 Classification of autonomous capabilities 65
Chapter 6 : Results of Research Question 1 66
6.1 Results FMVEA Analysis 67
6.1.1 Functions of embedded computer (Tank 720) 68
6.1.2 Failure Mode, Vulnerabilities and Effect Analysis of embedded computer (Tank 720) 69
Component element explanations 73
6.2 Results of STPA and STPA-SEC analysis 77
6.2.1 Concept - Define & Frame Problem 77
6.2.1.1 Losses/accidents and Hazards 79
6.2.2 Choosing control actions and connected process model variables and values 80
6.2.2.1 Create functional control structure 81
6.2.3 Identify Hazardous Control Actions 82
6.2.3.1 UCA summary 87
6.2.4 Generate Safety and Security constraints 88
6.2.4.1 Violation of Safety and Security constraints 90
6.2.4.2 Causal Factors leading violation of constraints and eventually Hazards 91
6.2.5 AC - accident cause 92
6.2.5.1 Generate casual scenarios & Mitigations and control 92
6.3 Results of CHASSIS analysis 98
6.3.1 Elicitation of functions and services 98
6.3.2 CHASSIS process 99
6.3.2.1 Use case: Operating and monitoring 100
6.3.2.2 Safety misuse case: Provide Operating and monitoring - Component failing 101
6.3.2.3 Security misuse case: Provide Operating and monitoring - Obtain access Revolt 103
6.3.2.4 Final misuse case with mitigations 105
6.3.3 Sequence Diagrams 106
6.3.3.1 Misuse Sequence Diagram 106
6.3.4 Perform HAZOP 108
6.3.4.1 HAZOP table for specifying safety requirements 109
6.3.4.2 HAZOP table for specifying security requirements 110
6.4 Results of Comparison of the three methods 111
6.4.1 Safety related hazards detected 112
6.4.2 Security related hazards/ threats detected 113
6.4.3 Risk assessment stages coverage 114
6.4.4 Qualitative vs Quantitative Risk analysis 114

6.4.5 Method origin 115
6.4.6 Discussion of comparisons 115
6.4.6.1 Cost-effectiveness 115
6.4.6.2 Security and safety hazards identified 116
6.4.6.3 Methods applicable to different situations 117
6.4.6.4 Security aspect – categorizing and measuring the effects of hazards 117
6.4.6.5 Summary of the methods based on experience 118
6.4.6.6 Addressing the challenges with autonomous systems 119
6.4.6.7 Possible limitations of the study 119
6.4.6.8 Limitations of STPA-sec in security analysis 119
Chapter 7 : Results of Research Question 2 120
7.1 Recap of STPA and STPA-sec analysis 121
7.2 Comparison of possible threat modeling methods for improving STPA-sec 122
7.2.1 Methods placement in the STPA-sec process - for STPA-sec incretion / data extraction 123
7.2.2 Methods for improving STPA-sec - Data input/output 124
7.2.3 Combination of each method 125
7.2.3.1 FMVEA with STRIDE – for Quantitative risk analysis 125
7.2.3.2 Misuse cases 127
7.2.3.3 Data Flow Diagram 129
7.2.3.4 Bow-tie diagram 131
7.2.3.5 Attack tree 133
7.2.3.6 BPMN for Threats 135
7.2.3.7 Socio-Technical Security modeling language (STS-ml) 137
7.2.4 Discussion of combining STPA-sec with threat modeling methods 139
7.3 Security first approach 140
7.3.1 Classification of Safety and security co-analysis methods 140
7.3.2 Method classification for Safety and security co-analysis methods 144
7.3.2 Data flow diagram – exploring the idea of starting from a Safety vs. Security approach 149
7.3.2.1 Safety starting point 150
7.3.2.2 Security starting point 151
7.3.2.3 Results comparison of starting point Safety vs Security 152
Chapter 8 : Conclusions and further work 154
8.1 Conclusions 154
8.2 Further work 155
8.2.1 Experiences working with an autonomous car 155
References 158

List of Figures

Figure 1 - Cyber Physical Systems 5
Figure 2 - How is a smartphone a CPS system 6
Figure 3- Automaton computing model 7
Figure 4 - Intelligent road infrastructure 8
Figure 5 - Autonomous Rover system developed APL [10] 9
Figure 6 – Autonomous modes in today's traffic that could help achieving goals [11] 10
Figure 7 – General hardware dependencies for autonomous systems 11
Figure 8 - Difference between autonomous and automatic 12
Figure 9 – Example of an Autonomous Decision system 13
Figure 10 -The 6 levels of autonomous driving 14
Figure 11 - The 5 grades of automation on trains 14
Figure 12 -Rio Tinto with the world's first fully autonomous rail journey 15
Figure 13 - Autonomy level for cyber-enabled ship 15
Figure 14 - Illustration of the Revolt project's cargo hold 16
Figure 15 - Autonomous Saildrone 17
Figure 16 - Aircraft operations in terms of four levels of automation 18
Figure 17 - BAE Systems successfully trial its autonomous aircraft 18
Figure 18 - Example of a multi-rotor UAV 19
Figure 19 - Drone components and I/O 19
Figure 20 - Autonomy related perspectives 22
Figure 21 - Goals of performing either a safety or security analysis 25
Figure 22 - Analysis methods focus on areas when targeting either safety or security 25
Figure 23 - Different definitions in different standards [31] [32] [33] 28
Figure 24 - Security and Safety distinctions 29
Figure 25 – summary of functional safety 34
Figure 26 - Jasper Juinen, Bloomberg via Getty Images 35
Figure 27 - Comparison of research security efforts on autonomous systems and some vulnerabilities found 36
Figure 28 - Demonstration of completely killing the Jeep remotely by hackers 37
Figure 29 -Attack surface for an autonomous automotive system 38
Figure 30 - Parrot AR.Drone 2.0 39
Figure 31 - GPS spoofing attack 41
Figure 32 - Basic Control Loop [52] 43
Figure 33 - Overview of the STPA method process 44
Figure 34 - Annotated control graph with scenarios for uncertain control actions 46
Figure 35- Overview of the FMVEA method 49
Figure 36 -Overview of the CHASSIS method 50
Figure 37 -Proposed methods for a case study of autonomous system – the Revolt platform 52
Figure 38 - Overview of the placement of the different components on the Revolt platform 54
Figure 39 - Overview of the placement of the different component on the Revolt 54
Figure 40 - Communication flow diagram of the Revolt hardware 57
Figure 41 - Communication system of the Revolt system [64] 57
Figure 42 - Complete wiring diagram of the Revolt 58

Figure 43 - Differences between manned, remote, automated and autonomous ships 65
Figure 44 - System level analysis FMVEA method 67
Figure 45 - Functional control structure of the Revolt 81
Figure 46 - Classification of how safety and security constraints could be violated 90
Figure 47 - Classification of causal factors leading to hazards 91
Figure 48 - Use case for Revolt 100
Figure 49 - Safety misuse case 101
Figure 50 - Security misuse case 103
Figure 51 - Final misuse case for the use case "operation and monitoring" with mitigations 105
Figure 52 -Misuse Sequence Diagram for the “Provide Operating and monitoring - Obtain access to the Revolt” security misuse case 106
Figure 53 - Failure Sequence Diagram for the «Provide Operating and monitoring - Component failing” safety misuse case 107
Figure 54 - Where STPA-sec analysis originates from 121
Figure 55 - Methods placement in the STPA-sec process 123
Figure 56 - Example of processes of data exchanges between the methods when integrating and improving the STPA-sec method with other methods 124
Figure 57 - Combination of the FMVEA and STPA-sec methods 125
Figure 58 - Example of misuse case from Case study: Revolt 127
Figure 59 - Example of combination of STPA-sec and misuse cases on the Revolt case study 128
Figure 60 - Example of Data Flow Diagram 129
Figure 61 - Control structure of the Revolt as a basis for Data flow diagram 130
Figure 62 - Example of a Data flow diagram based on Revolt control structure 130
Figure 63 - Example of Bow-tie diagram 131
Figure 64 - Example of a Bow-tie diagram based on general hazards of the Revolt case study 132
Figure 65 - Example of Attack tree 133
Figure 66 - Example of an attack tree diagram based on an unsafe control action of the Revolt case study 134
Figure 67 - Example of BPMN used for threats (Using escalation events to represent threats in a collaboration diagram) 135
Figure 68 - Example of an BPMN process diagram based on the controlled process from the control structure of the Revolt case study 136
Figure 69 - Example of Socio-Technical Security modeling language (STS-ml) 137
Figure 70 - Example of an STS-ml diagram based on the controlled process from the control structure of the Revolt case study 138
Figure 71 - SEMA referential framework 140
Figure 72 - Differences between taking a base in security or safety when using, designing or combining analysis methods 141
Figure 73 - interlink between safety and security 148
Figure 74 - Example of a Data flow diagram based on Revolt control structure – safety starting point 150
Figure 75 - Example of a Data flow diagram based on Revolt vessel– security starting point 151
Figure 76 - Kia Niro Hybrid 155
Figure 77 – Drivekit on autonomous car 156
Figure 78 - Stakeholders Kia Niro 156

List of Tables

Table 1 -Comparison of different Autonomous Systems used as examples 20
Table 2 - Test results from pilot 33
Table 3 - shellscripts and Linux files on the Parrot AR.Drone 2.0 [47] 40
Table 4 - STAMP based methods 43
Table 5 – Differences in the sequence between A-STPA and STPA-Sec 47
Table 6 - Stakeholders with the Revolt vessel 53
Table 7 - The different software used on the Revolt platform 55
Table 8 - Nodes on the ROS operating system on the Revolt system 55
Table 9 - Component list for the Revolt 56
Table 10 - Component I/O list: Embedded computer 59
Table 11 - Component I/O list: 4G Router 59
Table 12 - Component I/O list: Hard drive 59
Table 13 - Component I/O list: Xsens 60
Table 14 - Component I/O list: Arduino Mega 60
Table 15 - Component I/O list: Arduino Uno 61
Table 16 - Component I/O list: H-bridge 61
Table 17 - Component I/O list: RC receiver 62
Table 18 - Component I/O list: Motor controller 62
Table 19 - Component I/O list: DC-motor 62
Table 20 - Component I/O list: Servo 62
Table 21 - Component I/O list: Linear actuator 63
Table 22 - Component I/O list: Motor controller 63
Table 23 - Component I/O list: AC-motor 63
Table 24 - Component I/O list: Stepper motor 63
Table 25 - Component I/O list: Current meas. Sensor 63
Table 26 - Component I/O list: Inductive sensor 64
Table 27 - Component I/O list: Water sensor 64
Table 28 - Component I/O list: Battery 64
Table 29 - Component I/O list: Relay 64
Table 30 - Comparison of what input the different methods will have for the analysis 66
Table 31 - Hazard classification for determine if a hazard is safety or security related 66
Table 32 - Functions of embedded computer (Tank 720) 68
Table 33 - FMVEA analysis 69
Table 34 - Classification of Risk Severity level related to components 71
Table 35 - Classification of System susceptibility related to components 71
Table 36 - Classification of Unusualness level related to components 71
Table 37 - Classification of Motivation level related to components 72
Table 38 - Classification of Capabilities level related to components 72
Table 39 - Classification of Attack probability related to components 72
Table 40 - Classification of Failure probability related to components 72
Table 41 - Operating state variables for Revolt operation 78
Table 42 - List of Unacceptable losses/accidents for the Revolt vessel 79
Table 43 - List of System hazards and constraints for the Revolt vessel 79
Table 44 - Connection between what losses are possible with certain hazards 80
Table 45 - Control actions derived from main functions 80

Table 46 - Unsafe control table for control action 1 83
Table 47 - Unsafe control table for control action 2 84
Table 48 - Unsafe control table for control action 3 85
Table 49 - Unsafe control table for control action 4 86
Table 50 - Unsafe control actions summary 87
Table 51 -Generate Safety and Security constraints 88
Table 52- Generation of casual scenarios, causal factors and design recommendations for unsafe control action 1 92
Table 53 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 2 92
Table 54 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 3 93
Table 55 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 4 93
Table 56 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 5 93
Table 57 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 6 94
Table 58 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 7 94
Table 59 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 8 94
Table 60 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 9 95
Table 61 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 10 95
Table 62 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 11 95
Table 63 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 12 96
Table 64 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 13 96
Table 65 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 14 96
Table 66 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 15 97
Table 67 - CHASSIS analysis process 98
Table 68 - Abbreviation for analysis 98
Table 69 - Textual Use case for case: Operating and monitoring 100
Table 70 - Textual safety misuse case for case: Provide Operating and monitoring - Component failing 102
Table 71 - Textual Security misuse case for case: Provide Operating and monitoring - Obtain access to the Revolt 104
Table 72 - Guide Words for HAZOP method 108
Table 73 - Guide Words for specifying security requirements for HAZOP method 108
Table 74 - Parameters for specifying safety and security requirements for HAZOP method 108
Table 75 - HAZOP table for specifying safety requirements 109
Table 76 - HAZOP table for specifying security requirements 110

Table 77 - Comparison of the three methods used in case study 1 111
Table 78 - Comparison of safety related hazards detected by the three methods in case study 1 112
Table 79 - Comparison of security related hazards detected by the three methods in case study 1 . 113
Table 80 - Risk assessment stages coverage by methods 114
Table 81 - Method origin 115
Table 82 - Comparison of methods used in the Revolt case study 116
Table 83 - Comparison of possible methods for improving the STPA-sec 122
Table 84 -FMVEA methods on an example from Revolt case study 126
Table 85 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 13 134
Table 86 – Conclusion - possible ways of improving the STPA for security 139
Table 87 - Classification of method origin and interlink used for safety or security, for the attempted combined methods 144
Table 88- Other general methods that could be used in combination with Safety and security co-analysis methods 147
Table 89 - Software on Kia Laptop for control of autonomous car 156

List of Abbreviations

NTNU	Norwegian University of Science and Technology
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
DP	Dynamic Positioning
ROS	Robot Operating System
USB	Universal Serial Bus
STPA	Systems Theoretic Process Analysis
STAMP	System-Theoretic Accident Model and Processes
STPA-sec	Systems Theoretic Process Analysis for Security
FMEA	Failure mode and effects analysis
FMVEA	Failure Mode, Vulnerabilities and Effect Analysis
FTA	Fault Tree Analysis
CHASSIS	Combined Harm Assessment of Safety and Security for Information Systems
HAZOP	Hazards and Operability Analysis
SAE	Society of Automotive Engineers
ISO	International Organization for Standardization
ADI	Autonomous Driving Intelligence
CPS	Cyber-physical system
AC	Accident cause
UCA	Unsafe control actions
CA	Control actions
SC	System constraints
QRA	Quantitative Risk Analysis (Qualitative risk analysis?)
ROS	Robot Operating System
LIDAR	Light Detection and Ranging
IP	Internet Protocol
GUI	Graphical User Interface
ISO	International Organization for Standardization
SAE	Society of Automotive Engineers
FACT	Failure-Attack-Countermeasure
SAL	Security assurance level
SIL	Safety integrity level
MATCS	Approach for combining safety and security requirements techniques
CORAS	A Framework for Risk Analysis of Security Critical Systems
DBT	Design Basis Threat
GSN	Goal structuring notation
NFR	Non-functional requirements
EFT	Fault trees
CFTs	Component fault trees
BDMP	Boolean logic Driven Markov Processes
BBN	Bayesian belief network based approaches
UML	Unified Modeling Language
MBSE	Model-based system engineering
STS-ml	Socio-Technical Security modeling language
FAIR	Factor analysis of information risk
Sun	Framework for detecting safety and security conflicts.
AADL	Architecture analysis and design language
FACT graph	Frequency and Co-occurrence-based Trend Graph
IEC 61508	International standard published by the International Electrotechnical Commission

ISO 14971	ISO standard for the application of risk management to medical devices
CPS	Cyber-physical system
SAL	Security assurance levels
SIL	Safety Integrity Level
TVRA	Threat,. Vulnerability, Risk Analysis
SecL	Security Level
ASIL	Automotive Safety Integrity Level
HARA	Hazard Analysis and Risk Assessment

Chapter 1 : Introduction

In this chapter, I will introduce the foundation for the master thesis.

There are autonomous drones, cars, boats and trains. But what they all have in common is the ability to have various levels of autonomous capabilities. The key mechanism in an autonomous system is situation classification and the decision that follows when an unexpected event occurs. What was previously the human's responsibility, is now in different degrees the systems responsibility, and that is what makes the system more autonomous.

When mentioning the two terms "security" and "safety", in regard to autonomous systems, one might think that there are a specific meaning and purpose for them. The truth is that in the context of autonomous systems, these terms might be associated to different meanings. Therefore, ambiguities might be a challenge in this context. With security vulnerabilities, the system is open up to the possible attack and compromise of the systems. With safety issues, if a failure mode occurs, this could lead to hazardous situations for the environment.

The Revolt platform is an autonomous boat concept in a small-scale model, developed and implemented by DNV GL. The Revolt is a test platform for an autonomous ferry, in which purpose is to transport cargo with minimal energy consumption and costs associated with it. The Revolt also serve as an early test platform for sensors and control systems dedicated for autonomous vessels. The overall research design is focused on this empirical case study.

If the developers of autonomous systems choose to combine security and safety analysis, then there are significant advantages involved. Both the industries and researchers have started to acknowledge the need to fill the gap between these two areas. There has therefore been developed analysis methods in recent times that are designed for that purpose, and can probably be used for analyzing security and safety of autonomous systems.

In this thesis, I explore different methods for ensuring both safety and security on autonomous systems. With a case study on an autonomous boat concept, the Revolt platform, with a focus on comparing different security/safety methods for the Revolt platform and autonomous systems in general.

The research contributions in this thesis are:

1) Few empirical studies have been performed to compare and evaluate safety and security co-analysis methods. In this thesis, I have in a case study evaluated three methods using an autonomous boat.
2) An improved STPA and STPA-sec co-analysis using a Data flow diagram, taking a security first approach.
3) Complete list of comparison of existing methods for safety and security co-analysis methods and approaches, with a survey approach. Adopted from previous work [3], with the focus on finding out what the interlink between safety and security are in these methods.

1.1 Motivation

Cybersecurity of business-critical systems is becoming more and more important. Although there are several cybersecurity analysis methods, one method named STPA-Sec has recently been proposed from MIT. However, the STPA-Sec methods have not been thoroughly piloted in industrial case studies. There are also room to develop this method further. In addition, there are not many tools that can support the use of this method.

If the developers of autonomous systems choose to combine security and safety analysis, there are significant advantages involved. Both the industries and researchers has started to acknowledge the need to fill the gap between these two areas. In recent times, it has therefore been developed analysis methods that are designed for that purpose, and can probably be used for analyzing security and safety of autonomous systems.

Autonomous systems have seen a rise in development in the recent years. There are now being developed autonomous drones, cars, boats and trains. Perhaps maybe one day these systems will replace the existing ones which requires manual input from a human. Many of these systems are safety-critical.

The motivation for the tasks is to find ways to improve the safety of critical systems, as there is a great need for this in the future. The security of autonomous systems (self-driven cars, boats, etc.) will be particularly important to ensure because the consequences for attacks / security breaches on these systems can be very high. Therefore, I choose to write a case study in collaboration with DNV GL, and based on the Revolt model, an autonomous ship for cargo shipping [4].

To address these challenges, an autonomous boat i.e. the Revolt developed by DNV GL [1], is used as a case study. In this case study, I will focus on comparing different safety and security co-analysis methods for the Revolt platform and autonomous systems in general.

1.2 Research questions

To goal of the research is to investigate current methods used for safety and security analysis of autonomous systems with taking into consideration the challenges these systems have, both the safety and security aspects.

The motivation is to make contributions in improving the weakness with safety and security co-analysis methods, and as a result make them more useful and applicable for complex systems e.g. autonomous and CPS systems.

Research questions are as follows:

RQ1: How does existing approaches/ methods for safety and security co-analysis compare to each other?

RQ2: How to improve the weaknesses of STPA and STPA-sec for a better safety and security co-analysis?

1.3 Research design and tasks

Research design

The following section will describe the overall research design for this thesis and will include study decisions, methodology and application areas.

1) Application area for this thesis is an autonomous boat. I therefore need to find out more about autonomous systems and cyber-physical systems (CPS). Because the autonomous boat is considered to be both a CPS and autonomous system.

2) The methodology areas for this thesis, which is safety and security analysis methods, and possible co-analysis methods. Therefore, I need to describe the basic principles of this methodology.

3) An empirical case study must be performed on the autonomous boat with the base of theory described in the literature review. The aim will be to compare the methods to find out if they are applicable to be used on advanced and complex systems, like the autonomous boat. If they are not applicable, I need to work on finding possible ways of improving the existing methods.

Tasks

To address the research questions presented in this thesis, the following tasks are established:

1) A state of the art literature review of autonomous systems and cyber-physical systems (CPS).
2) Investigate cybersecurity analysis methods in general and the STPA-Sec method in particular.
3) A case study to pilot the STPA-Sec method and possibly other methods to investigate its applicability, effectiveness, and cost efficiency.
4) If weaknesses are found in the existing methods, how to address those weaknesses?

1.4 Structure of the thesis

This thesis is based around the problem statement and its tasks. Therefore, the chapters are as follows:

Chapter 1 - **Introduction**

The first chapter introduces the thesis and lays a foundation for the following chapters.

Chapter 2 - **Literature Review of Cyber-physical systems and Autonomous systems**

In chapter two I will perform a literature review on Cyber-physical systems and Autonomous systems, which is the selected topics of the thesis.

Chapter 3 - **Safety and Security analysis**

In the third chapter, the basic principles of this work are described. The topic is methods for Safety and Security Co-analysis.

Chapter 4 - **Research Motivation and Research Questions**

In chapter four, I will propose methods for a case study.

Chapter 5- **Research Design and Case study**

Chapter five consist of the practical analysis and the results of the case study. The basis for the analysis methods are being presented. I will introduce the selected case for a case study, the Revolt platform, and give the reader an overview of the platform.

Chapter 6 - **Results of Research Question 1**

In chapter six the results of research question 1 is presented.

Chapter 7- **Results of Research Question 2**

In chapter six the results of research question 2 is presented.

Chapter 8 - **Conclusions and further work**

At the very end in chapter eight, a summary of the case study is concluding the thesis work.

Chapter 2 : Literature Review of Cyber-physical systems and Autonomous systems

In this chapter the I will conduct a literature review of cyber-physical systems and autonomous systems, with a focus on the safety and security aspects, to give a basis for the case study, in which will involve an autonomous boat.

2.1 Cyber-physical systems

As previously mentioned, a Cyber-physical system (CPS) are systems that consist of both physical hardware and the virtual world in form of information being processed with algorithm connected to a network. The physical hardware often consists of sensors and actuators and they will interact with the physical world. What the software represents is the ability to connect to the virtual cyber world. Therefore, the system has a connection to both the virtual and physical world.

An example of a CPS system has been mentioned before and is from the automotive domain, a Cloud-Driven Traffic Monitoring [5] system connects the physical (cars) with a monitoring system (virtual world).

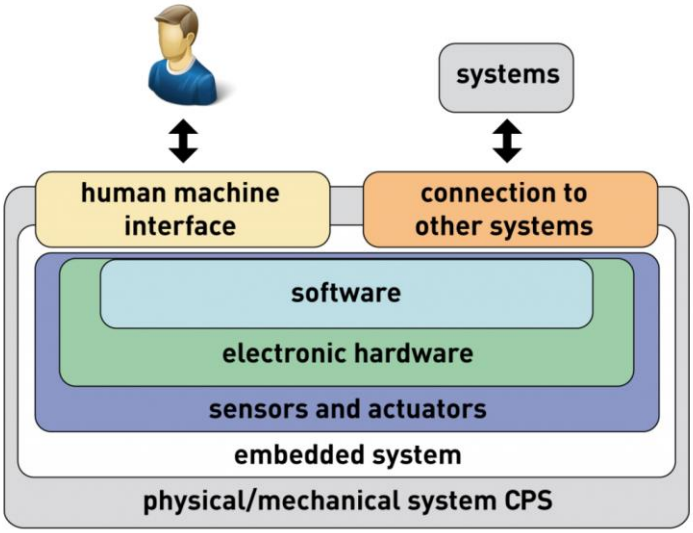


Figure 1 - Cyber Physical Systems

What a CPS system usually consist of is displayed in Figure 1 [6]. The inner layer consists of the software, hardware (including - sensors and actuators) and this makes an embedded system. What makes the system a CPS system is the connection it has to other systems and also a human machine interface. The safety and security vulnerabilities/hazards can be located on either of these layers, displayed in Figure 1.

2.1.1 A simple example of CPS system

One simple example of a CPS system is a smartphone.

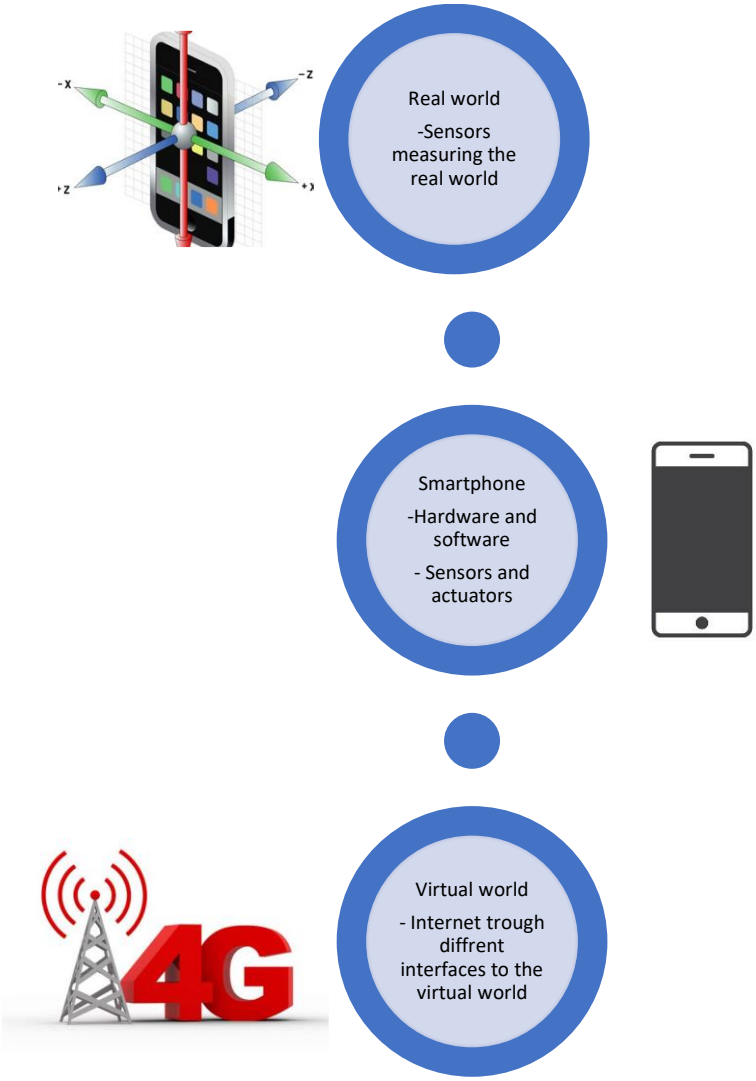


Figure 2 - How is a smartphone a CPS system

The CPS system, being a smartphone is illustrated in Figure 2. The smartphone is in the middle and has both the hardware and software to enable a CPS system. The smartphone is connected to the real world through the sensors. For example, an accelerometer, gyroscope or magnetometer, that are installed on the smartphone. These measures the physical world. The smartphone is also connected to the virtual world through the internet. Doing so trough wireless interfaces like 4G or WIFI networks. From there, the sensor data can be transferred from the physical world to the virtual world. For example, the accelerometer can be used with Google maps, to track the changes to the position of the smartphone.

2.1.2 CPS systems in the Automotive domain

I will now explore in what way Cyber-physical systems could be intergraded in the Automotive domain, with a focus on how the future is likely to be and connect this to the safety and security aspects.

Automotive Systems

An Automotive System is referring to the design of a system used on motorized vehicles, for example cars, trucks or motorcycles. In Automotive engineering concerns like design, operation, manufacturing, safety and security are considered.

The future of automobiles leading towards CPS integration

With the introduction of the concept of self-driven cars, or autonomous cars. The way vehicles are driven today, might be drastically changed in the further. In a report by the research company Gartner, it is predicted that by 2020, there will be a quarter of a billion connected vehicles on the road [7]. This will give the possibility to provide new in-vehicle services, automated driving, being one of them. With this new innovated way of using automobiles, the demand for the infrastructure to support this, will also arise.

One might also consider the other changes that will come with this new shift. An article written by Vladimir Hahanov and Wajeb Gharibi, discusses Cloud-Driven Traffic Monitoring [5]. The article discusses what types of cloud infrastructure the roads might have in the future. As it says in the article, there are already signs on the road that are controlled by the cloud. The need for more of these components to be connected to the cloud, will increase. As of now even the cars are beginning to depend more on it.

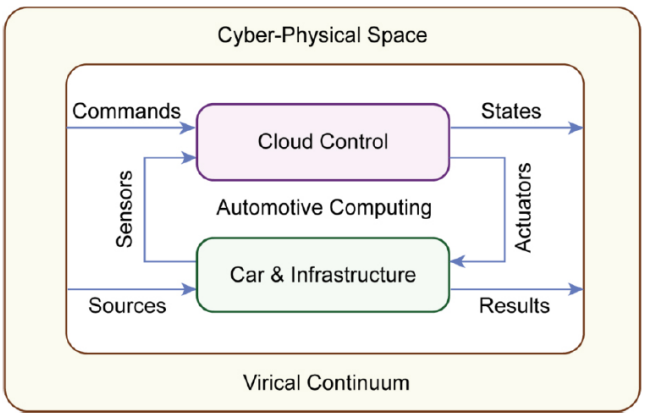


Figure 3- Automaton computing model

In Figure 3 as shown in the mentioned article [5], the frame of Cloud control and the cars and infrastructure, will be the Cyber-Physical space and the virtual-physical (virical).

The impact the “cloud” or “cyber world” has on today’s society are further discussed in an article from the international conference “Parallel and distributed Computing systems” [8]. There is argued that Cloud component can solve important challenges in our society regarding traffic safety. With machine learning and an infrastructure for monitoring, and control of the vehicles on the road. The possibility to reduce accidents are greatly increased. What might be a good point is the environmental problems, that could be reduced. For example, the cars won’t use as much fuel and reduce Carbon dioxide emissions overall.

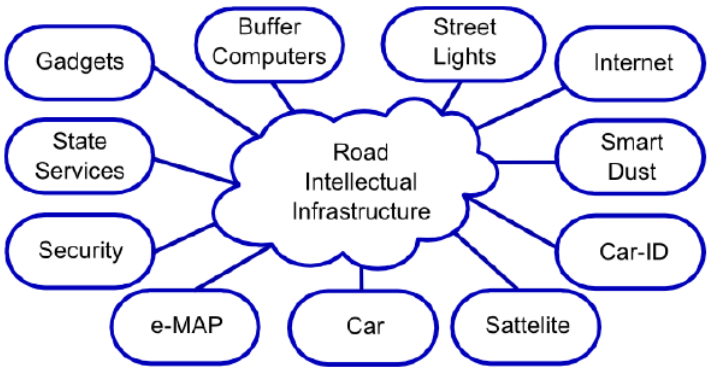


Figure 4 - Intelligent road infrastructure

The figure above (Figure 4) is collected from the article mentioned [8], and show what services the infrastructure for the “Green Wave Traffic” Cloud service in which is introduced.

2.2 Autonomous systems

In this section I will do a literature review of Autonomous systems, with a focus on trying to give a basis for the case study, in which will involve an Autonomous boat.

An Autonomous System is referring to the design of a system in which can change its behavior in response to unanticipated events during Its operation [9]. The essential concept is that the system is capable of performing actions individually, without interactions from humans.
These systems are today in use on motorized vehicles, for example the Google car, and also trains are becoming more and more autonomous. Autonomous Systems has also seen success in military applications, such as drones.

In Autonomous engineering concerns like design, operation, manufacturing, safety and security are considered when developing the system to be autonomously capable, in different degrees.

2.2.1 History

The first research phase of autonomous systems was initiated in 1940 by Norbert Wiener, a mathematics professor at MIT. He was working on automated rangefinders for anti-aircraft guns [9]. During his research, he discovered the intelligent behavior of these systems. This initiated the first wave of research of autonomous systems, systems that could through analog signals demonstrate intelligent behavior. In 1964, an autonomous rover system was developed by APL (Adaptive Machines Group). This rover could navigate through hallways and find power outlets, and charge when its battery was low.

Figure 5 - Autonomous Rover system developed APL [10]

We saw a growth in the interest for autonomous systems in the 1970, form a number of factors related new technologies. With the arrival of digital control electronics, and costs hardware (sensors, processors etc.) went down. At the same time, there was a growth of knowledge in the new field of AI (Artificial intelligence). This enabled the development of autonomous systems that could operate fairly complex tasks without little or no interaction from humans, as we can see today.

2.2.2 Background

In today's society, we have to face many critical problems. We have seen the threat of terrorism, global climate change and still occurring accidents in our traffic. The autonomous industry is going face to face, to tackle these issues.

In today's society, we are seeing the use of Autonomous military drones, in which removes the risk of life-or-death scenarios. The drones can for example scout the enemies' battlefield and collect valuable information, that could save lives. In Figure 6 we can see that there are many goals that could be achieved by using autonomous systems [11].

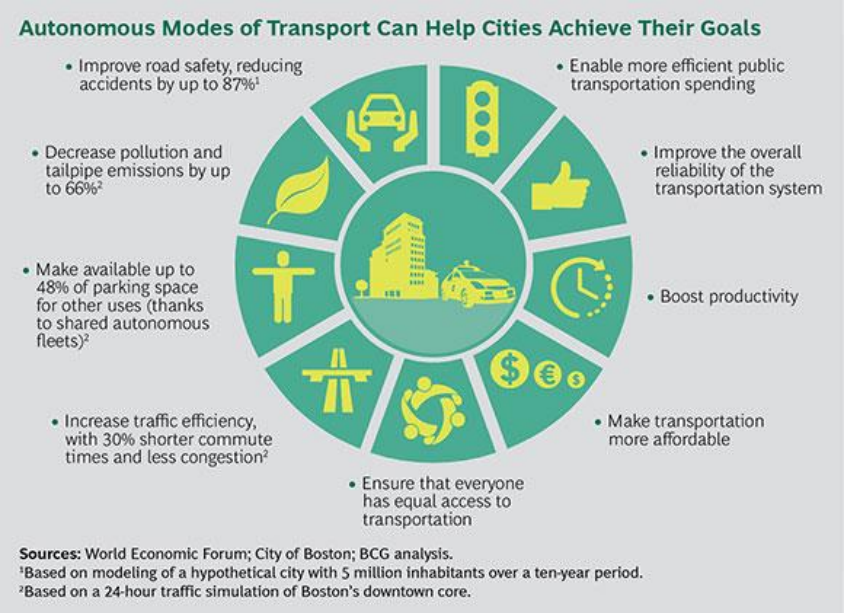


Figure 6 – Autonomous modes in today's traffic that could help achieving goals [11]

In the traffic, we have seen the development of the Google car. In a scenario where every car was autonomous, we could remove the risk of accidents happening in the traffic, from the cause of for example drunk driving. With the use of autonomous vehicles, the carbon dioxide emissions will also be reduced. For example, when autonomous systems is being used, the car always knows where it is and where to drive, without the loss of the GPS signal off course. There will be no lost drivers, and wasting fuel will be less occurring.

Other motivations for autonomous systems are that they can perform the following tasks:

- Replace humans:
  Perform tasks that are not possible because of unclear or hazardous environments. For example, space missions, deep sea diving and military actions.
- Human assistance:
  Perform tasks that are not necessary to be done by humans (repetitive tasks etc.). Assist humans with handicaps.

2.2.3 Hardware of Autonomous system

In an article published by the IEEE computer society in 2013, the basic hardware dependencies for a system to be called autonomous are described [12], and illustrated in Figure 7. The first hardware component an autonomous system needs is at least one embedded processor. The processor is responsible to coordinate and control the activities on the autonomous system platform, and will receive input from the sensor. The next component is an array of sensors to receive information about the physical world, and will send this information to the processor(s), to make decisions based on them. The last hardware component is a communication system, in which are used to send or receive information and instructions from the command center, where the autonomous system is controlled from, and also to have the possibility to receive information from other autonomous systems which could be nearby.

There is certain hardware that are necessary for a system to have autonomous capabilities, as described in the figure below.

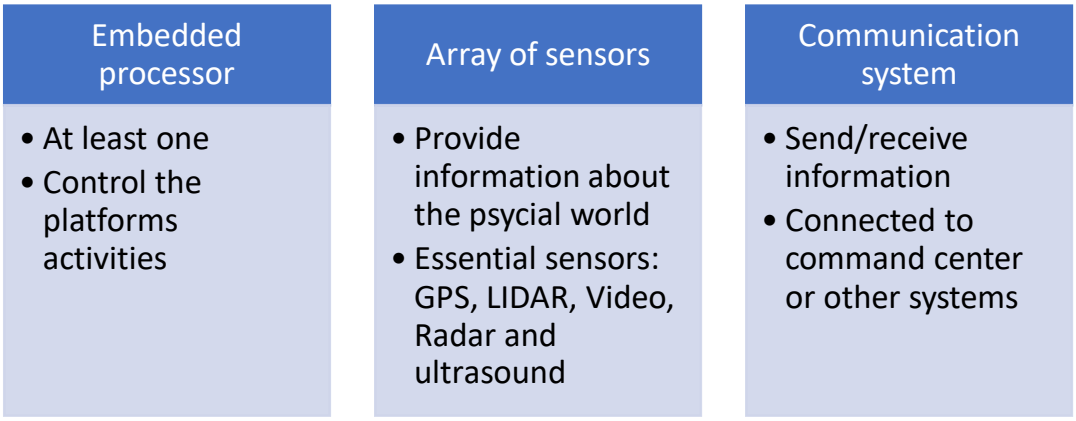


Figure 7 – General hardware dependencies for autonomous systems

2.2.4 Software of Autonomous system

The key mechanism in an autonomous system is situation classification and the decision that follows when an unexpected event occurs. What was before the human's responsibility, is now in different degrees the systems responsibility, and that is was makes the system more autonomous. This demands the system to be more intelligent to be able to make the decisions. This demands for a more complex system with more sensors and connectivity, and this hardware will be made use of by the software that runs the system.

Therefore, the autonomous system need to be capable to make very intelligent decisions on its own. Here is where an ADI (Autonomous Driving Intelligence) comes in to play. With an ADI, the information that is gathered is being processed and an action is taken based on that. What the ADI has in challenges could be complex. The environment this intelligence does need to navigate through could be difficult to predict and controlled. For example, different domains, users, scenarios and a large set of use cases. Also, the system itself has its challenges of being very intergraded and possibly having a bottom-up growth.

2.2.5 Difference between autonomous and automatic

Now that both automotive and autonomous systems have been introduced. The clear distinguishing factor we can see from them are the decision making. With a fully autonomous system, not only one or two tasks are performed automatically, like with an automatic system. The system is required to make decision completely independent and without interactions with humans at some point.

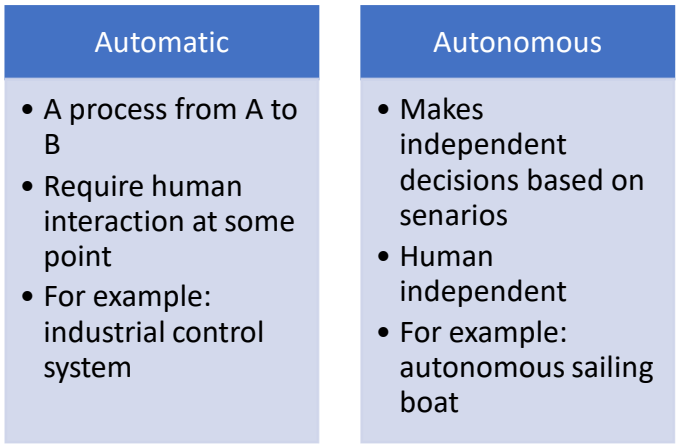


Figure 8 - Difference between autonomous and automatic

In Figure 8 the key differences between an autonomous and automatic system is described.

Autonomous Decision system

In Figure 9, we can see the decision process for solving problems, that exist in an autonomous ship [13]. The system is running with automatic functions, if a problem is detected the autonomous entity in the system is then set to solve the problem. If it can't solve the problem, a remote operator is contacted to solve the problem. If there is no contact to the operating central, a fail to safe mechanism is used.

I would conclude that the more automatic functions a system has, the more is it leaning towards being autonomous. However, what needs to be covered is an autonomous decision system.

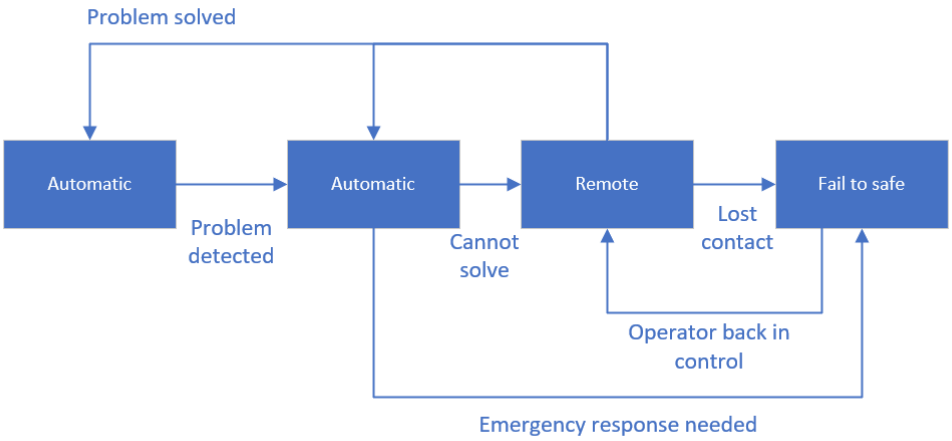


Figure 9 – Example of an Autonomous Decision system

2.2.6 Area of autonomous domains

Autonomous systems are relevant to different domains, in the article written by David P. Watson and David H. Scheidt, such domains are discussed [9]. With taking the base of what involvement APL, the company who developed the Autonomous Rover system (Figure 5), has had. APL has since 1964 developed autonomous systems for the maritime, ground, air and space domains. I will take a base in these domains, not only limited to APLs development, and discuss them here.

2.2.6.1 Ground

The first autonomous domain I will discuss are the ground domain, meaning autonomous systems that operate only on the ground. Today there are many examples of such systems. For example, autonomous automotive systems like the self-driven car.

The google car, in which became Waymo, is a relevant example we can see in today's society, they are working towards the goal of full autonomy on their vehicles [14].

With the introduction of the concept of self-driven cars, or autonomous car. The way vehicles are driven today, might be drastically changed in the future. A report by the research company Gartner, predicts that by 2020, there will be a quarter of a billion connected vehicles on the road [7]. This will give the possibility to provide new in-vehicle services, automated driving, being one of them. With this new innovated way of using automobiles, the demand for the infrastructure to support this, will also arise.

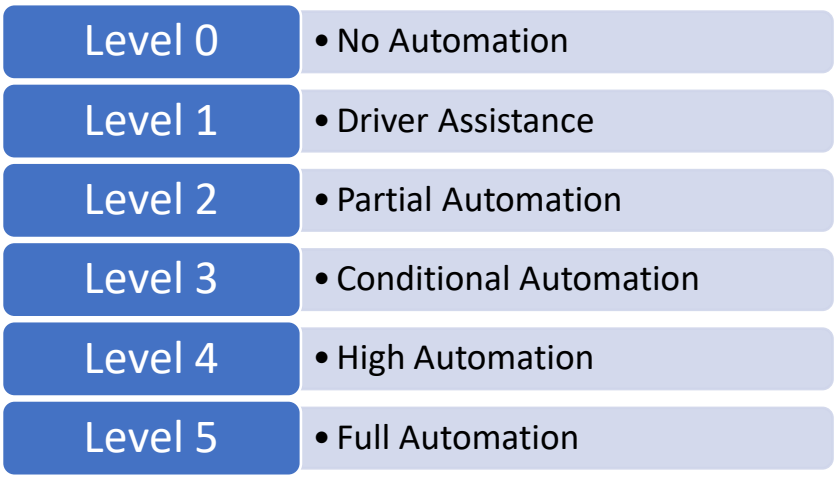


Figure 10 -The 6 levels of autonomous driving

The Society of Automotive Engineers (SAE), have described 6 levels of autonomous driving [15], as illustrated in Figure 10. The levels vary from level 0, where the vehicle has no autonomous capabilities and the human driver are responsible for all aspects of the driving task. To level 5, where the driving task are only managed by the autonomous driving system.

2.2.6.3 Trains

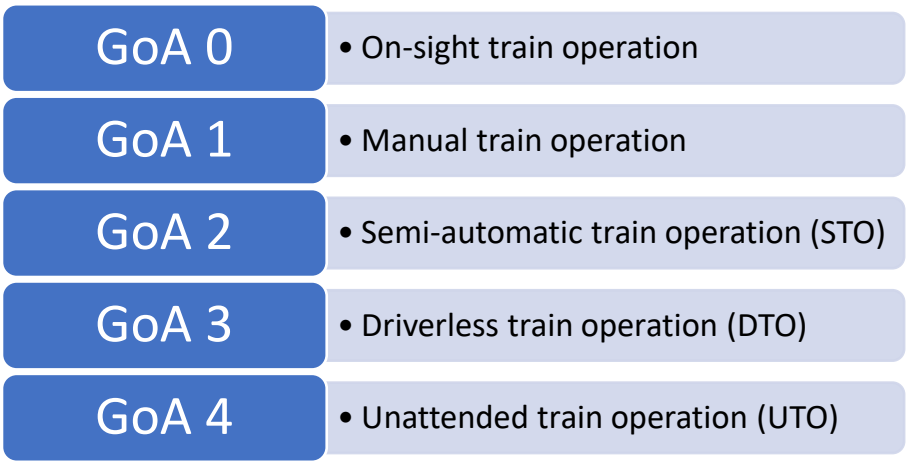


Figure 11 - The 5 grades of automation on trains

The International Association of Public Transport (UITP) have described 5 grades of automation on trains in Figure 11 [16].

Rio Tinto Autonomous train

Figure 12 -Rio Tinto with the world's first fully autonomous rail journey

In the figure above (Figure 12) we can see a picture of the first fully autonomous rail journey in Australia, by the company Rio Tinto [17]. This is the first of the kind of autonomous trains. This train has the highest level of automation (GoA 4), Unattended train operation (UTO) and there is no need to have any operator on the train.

2.2.6.4 Maritime

The second autonomous domain is the maritime domain, meaning autonomous systems that operate on the surface of the sea or in deep sea. We have seen development of such systems over time.

AL 0	•Manual – no autonomous function
AL 1	•On-ship decision support
AL 2	•On and off-ship decision support
AL 3	•'Active' human in the loop
AL 4	•Human on the loop – operator/supervisory
AL 5	•Fully autonomous - Unsupervised or rarely supervised operation
AL 6	•Fully autonomous - Unsupervised operation

Figure 13 - Autonomy level for cyber-enabled ship

In the Lloyd's Register guidance document, a procedure for autonomous ships describes seven autonomy levels (AL) [18], these are displayed in Figure 13.

Autonomous boats for cargo shipping

The Norwegian company DNV GL has done research within autonomous shipping. Their ReVolt project has the concept of an unmanned, zero-emission, shortsea vessel [19]. With this vessel, brings the possibility to transport containers, up to 100 TEU (Twenty-Foot Equivalent Unit). The range with only one battery is estimated to be 100 nautical miles [20].

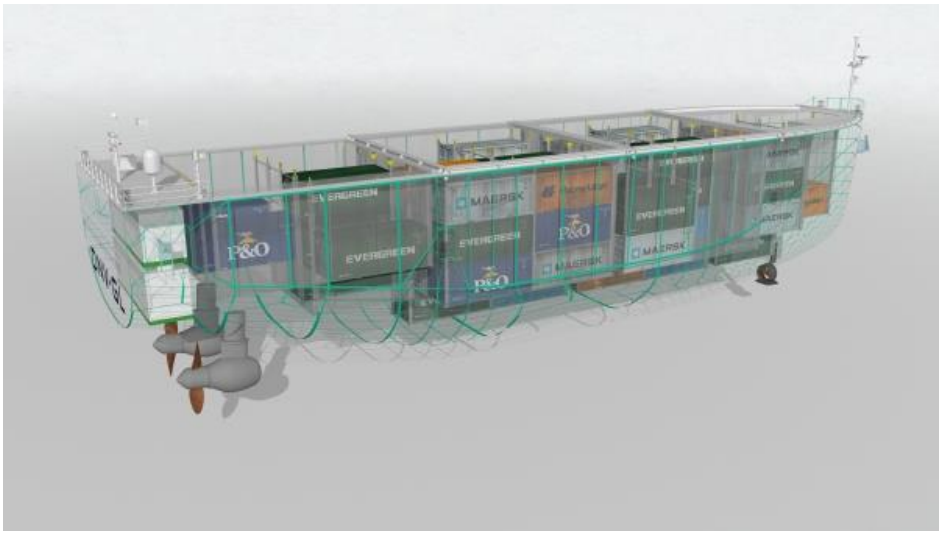


Figure 14 - Illustration of the Revolt project's cargo hold

DNV GL has established a research cooperation with NTNU, and has a demonstration platform in which are used to further development of the concept, called the Revolt – Demonstration platform [4]. This platform is in 1:20 scale to the original Revolt, and was built to test for different features.

Issues and missing features of the Revolt

The Revolt model is currently under development and is not yet a fully autonomous boat. The main functions the vessel currently has, is the GPS/GNSS and controller. This gives the operator the opportunity to control the vessel without any issues. But the decision making is still done by a human. There needs to be further development of the software (Revolt Intelligent System) being used and additional sensors e.g. LIDAR, radar.

However, this is still a valuable object to be used as a case study, for the following reasons:

- Explore Safety and Security issues related to autonomous steering of the vessel. Loss of control and steering of the vessel could result in loss of life or other damage.
- Explore Security issues related to data-communication between onshore and offshore systems. Sensitive data could be compromised.

Autonomous sailing boats

A concept that we might see in the future is autonomous sailing boats. There has been development in for example the cargo shipping industry, as mentioned before. The technology potential has been carried over to sailing boats. In 2015, an article published in the IEEE Journal of Oceanic Engineering, discussed this development [21] The possible solar and wind power sources are there, to support the system. The article proposes a hardware and software architecture for an autonomous sailing robot.

Figure 15 - Autonomous Saildrone

In Figure 15, we can see what is called a Saildrone [22]. This is an invention in the sailing boat industry. The sensors on this boat makes it possible to predict the world's weather with far more precision than before.

UUVs

In the 1980s, APL supported the development of UUVs (unmanned underwater vehicles) These underwater vehicles were a part of the DARPA (Defense Advanced research program), in which is an agency of the US Department of Defense, and they were used for military purposes. For example, these underwater vehicles could be used to clear mine fields under sea water. They are operated to have the task to prioritize intelligence, surveillance and reconnaissance (ISR). In addition to UUVs, there are also what is called ROVs - Remotely operated vehicles.

2.2.6.5 Air

The NBAA Automated Flight Deck Training Guidelines describes Aircraft operations in terms of four levels of automation [23], as shown below.

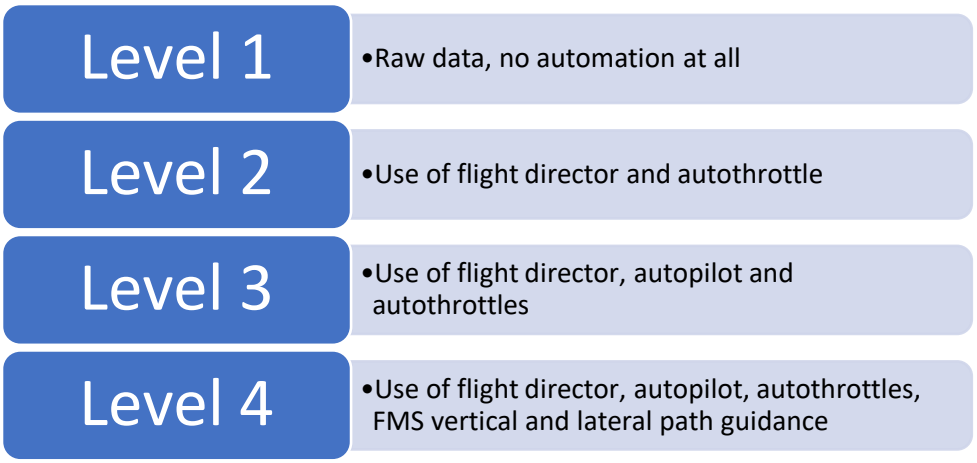


Figure 16 - Aircraft operations in terms of four levels of automation

Figure 17 - BAE Systems successfully trial its autonomous aircraft

In Figure 17, we can see an Autonomous aircraft developed by BAE systems [24]. This was originally a plane called Jetstream 31, but was transformed and equipped with hardware and software in which makes autonomous flying possible. In December of 2016, trails with this airplane has been successful – autonomous technology did take over control at 15000ft in the sky. BAE systems is working towards fully autonomous capabilities on these airplanes by 2020.

Autonomous drones

There are different kinds of drones, for the air, ground or maritime areas. Both military grade drones e.g. UAVs (Unmanned aerial vehicles) is the NASA Global Hawk. Drones for commercial use are getting popular, and example of this is in Figure 18 - the typical design of a multi-rotor UAV is displayed with its internal components.

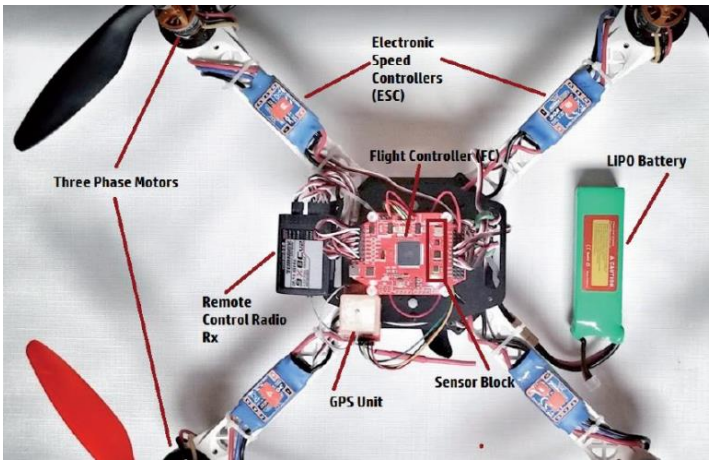


Figure 18 - Example of a multi-rotor UAV

All the different components that are installed the specific drone and the I/O is listed in Figure 19.

Drone components

- Electronic speed controllers – Controls the motor driver.
- Three phase motors - Screw propeller.
- Flight controller – CPU and firmware, connected to the radio receiver, on a power distribution board.
- Remote control radio rx – connected to a remote Controller via 2.4GHz radio.
- Sensor block - gyroscope, barometer, accelerometer, compass etc.
- GPS unit – Report the position to the remote controller.
- LIPO battery – last typically at least 15 minutes.

Drone I/0

- Micro-USB – used to connected to flight simulating program
- MicroSD – used for storage and firmware updates

Figure 19 - Drone components and I/O

2.2.6.6 Comparison of Autonomous Systems from different domains

We have now gone through the most common domains for Autonomous Systems. In the future, there is a high chance of other domains also adapting to include autonomous systems.

In Table 1, the different Autonomous Systems used as examples previously is compared to see how the levels of autonomy relates. However, each domain has its own levels, so they are not directly correlated. What is interesting to see is the different technologies used and the purpose of these systems are somewhat the same. Except for the autonomous Saildrone, in which has a different use case, collection of data for manned research ships. The research fields this Saildrone is being used for is currently in ocean data collection, fish stock analysis and environmental monitoring.

Table 1 -Comparison of different Autonomous Systems used as examples

	Autonomous car Google car	**Autonomous train**	**Autonomous boat Revolt**	**Autonomous Saildrone**	**Autonomous aircraft**
Company	Google	Rio Tinto	DNV GL	Saildrone	BAE systems
Year	2012	2017	2015	2014	2016
Driving assistant system costs	130000 USD	N/A	N/A	N/A	400000 GBP
Level of autonomy	Level 4 -High Automation	GoA 4 - Unattended train operation (UTO)	AL 4 -Human on the loop – operator/supervisory	AL 5 -Fully autonomous - Unsupervised or rarely supervised operation	Level 3 - Use of flight director, autopilot and autothrottles
Essential Technologies used	3D LIDAR GPS	3D maps On-board (computers operate independently) Real time data	GPS / GNSS Xsens Inductive, water and current. meas sensors	Carbon dioxide, acidity, currents and water temperature sensors	Infrared camera and seven other cameras Collision avoidance system
Purpose	Passenger transportation	Passenger and cargo transportation	Short-sea cargo shipping	Collection of data for manned research ships	Passenger transportation

2.2.7 Challenges with Autonomous Systems

1.**Intelligent System**

The key mechanism in an autonomous system is situation classification and the decision that follows when an unexpected event occurs. What was before the human's responsibility, is now in different degrees the systems responsibility, and that makes the system more autonomous. This demands the system to be more intelligent to be able to make the decision. This demands for a more complex system with more sensors and connectivity to other devices.

Therefore, the autonomous system need to be capable to make very intelligent decisions on its own. This is where an ADI (Autonomous Driving Intelligence) comes in to play. With an ADI, the information that is gathered is being processed and an action is taken based on that. What the ADI has in challenges could be complex. The environment this intelligence does have to navigate through could be difficult to predict and control. For example, different domains, users, scenarios and a large set of use cases. Also, the system itself has its challenges of being very intergraded and possibly having a bottom-up growth.

2. **Need for connectivity**

With the increasing level of capabilities the autonomous system has, the larger need it has to be connected to different components and other systems and entities. For example, an autonomous system like a driverless car, does communicate with various entities dependent on its location. With all this communication between systems, it also increases chances for a potential cyber security attack. We also need to consider all the different hardware, software, users and risk exposure these systems have. A system could potentially pose a risk to all the systems its connected to.

3. **Dependencies on a Global Positioning Systems (GPS)**

We also need to consider the dependencies the autonomous systems have to an GPS. These systems need data to navigate, and this poses a risk if the data received are not accurate. This area could be problematic, since there is various infrastructure for these GPS systems. The solution is often to increase the amount of satellites, but how long could this solution last with the increasing need for it?

The GPS is an open standard and is therefore accessible to the public, and therefore also a potential attacker could use this to take advantage when the architecture is known. There has been reports of spoofing attacks and this could be a real threat to upcoming autonomous systems.

4. **Testing environment**

With new technology there is always need for testing to see if the system behaves safe and is secure. This poses an extra challenge to autonomous systems, there is no human driver in the testing scenarios. There practically is an unlimited amount of scenarios the system could be exposed for. Therefore, often to ensure safety and security of these systems there is a need for billions of miles of testing.

5. **Different risk levels and performance of ADI**

There need to be performed more research to determine which requirements the autonomous system must have. The system is exposed to a variation of risk level, and some are acceptable, and some is not. The requirements need to ensure that the ADI performs to the level of a human or better.

2.2.8 Autonomy related perspectives

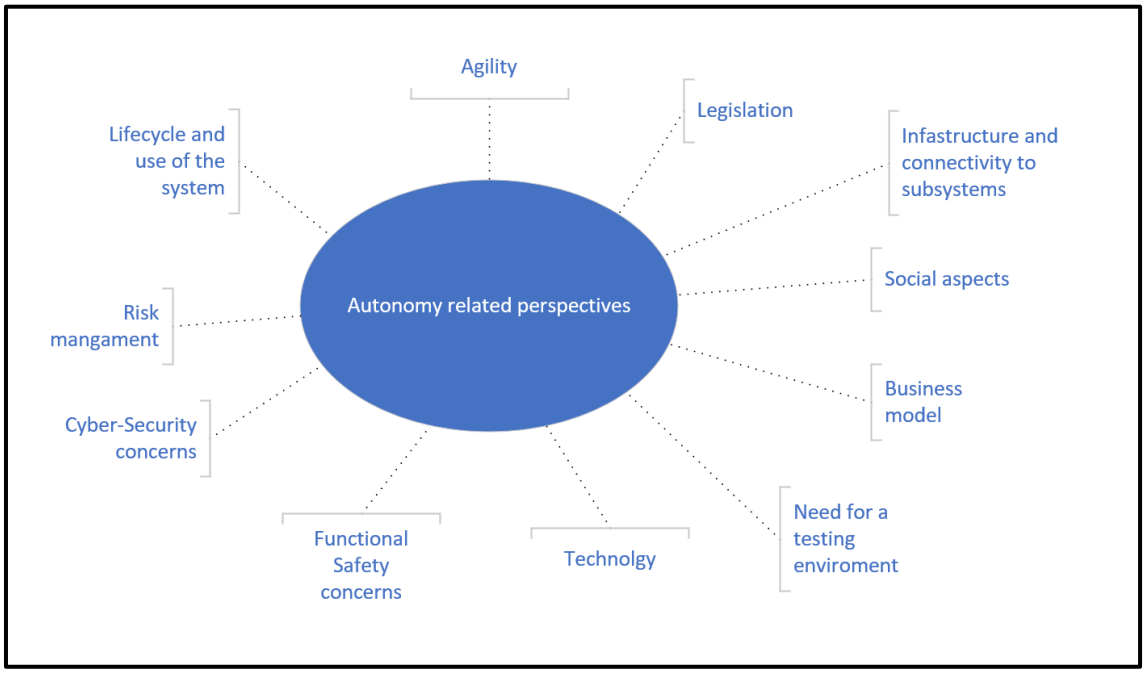


Figure 20 - Autonomy related perspectives

In Figure 20, different autotomy related perspectives are displayed. There are many perspectives and factors that needs to be addressed when developing an autonomous system and the challenges mentioned can come from any of these perspectives.

Chapter 3 : Safety and Security analysis

In this chapter the basic principles of this work are described, and terminology used.

3.1 Safety and Security analysis methods

There are two concepts that are essential to the work in this thesis; safety and security. These two terms could also have different meaning dependent on the industries they are being used. However, what is called the SEMA reference framework [25] have a way of distinguishing these two terms:

- Safety definition “the degree to which accidental harm is prevented, reduced and properly reacted to”
- Security definition “the degree to which malicious harm is prevented, reduced and properly reacted to.”

With the SEMA framework, these terms are distinguished with what kind of harm they prevent, accidental or malicious [26].

A way the security is measured is through a CIA model. CIA stands for confidentiality, integrity, availability and all these key principles should be considered for a system to be secure.

Functional safety and cyber-security ambiguities, overlap and differences, in regards to autonomous systems, are discussed in chapter 4.

3.1.1 Safety Analysis

A safety analysis method aims to ensure the safety of both the environment around the target case and the people. When for example taking the case of an autopilot function on a car. There has to be very high standards concerning this function. There cannot be any downtime or system failures, if they occur, it will have an effect on the autopilot when it is in operating mode. An error could result in a failure and then a malfunction of the system. This could lead to a fatal crash or worse.

The process of the safety analysis varies from the different analysis methods, but some common distinguishing factors has been outlined in an article published at SAFECOMP 1999 [27], and is as follows:

1. Functional and Technical Analysis
2. Qualitative Analysis
3. Quantitative Analysis
4. Synthesis and Conclusions

3.1.2 Security Analysis

Security is becoming more and more important to systems that are being developed today. For example, business critical systems and especially military grade systems have a high degree of security. The communication layer is a high focus when considering the security of the system. The messages being sent from system to system could be intercepted and information being compromised. If military intelligence agencies where to be subject for this, that could lead to secret strategies or tactics being known to the enemy and eventually this could decide the life and death of people affected by the information being lost.

A security incident is something that typically has an effect on the confidentiality, integrity or availability of a system. Also called the CIA model.

The article published at SAFECOMP 1999 [27], has also outlined some common distinguishing factors for the Safety Analysis:

1. Asset Identification
2. Vulnerability Analysis
3. Likelihood Analysis
4. Countermeasure Evaluation

3.1.3 Safety and security co-analysis methods

3.1.3.1 Goals and analysis targets

We have already distinguished between the safety and security terms. There is a difference between malicious and accidental risk. However, how does this translate to the analysis methods for safety and security?

Security threats are less triggered by what you do, but what the infrastructure and services is set up and provided to the users.

Safety, is more related to how the infrastructure is used. A lot of unsafe hazards are triggered by unsafe use of the system or infrastructure.

What are the general goals of performing either a safety or security analysis?

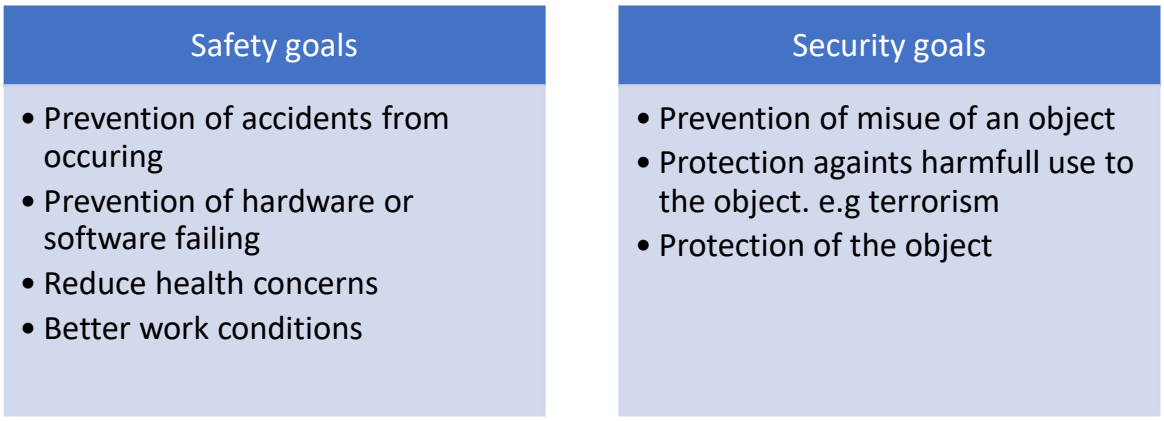


Figure 21 - Goals of performing either a safety or security analysis

I would therefore say that the analysis methods typically focus on these areas when targeting either safety or security.

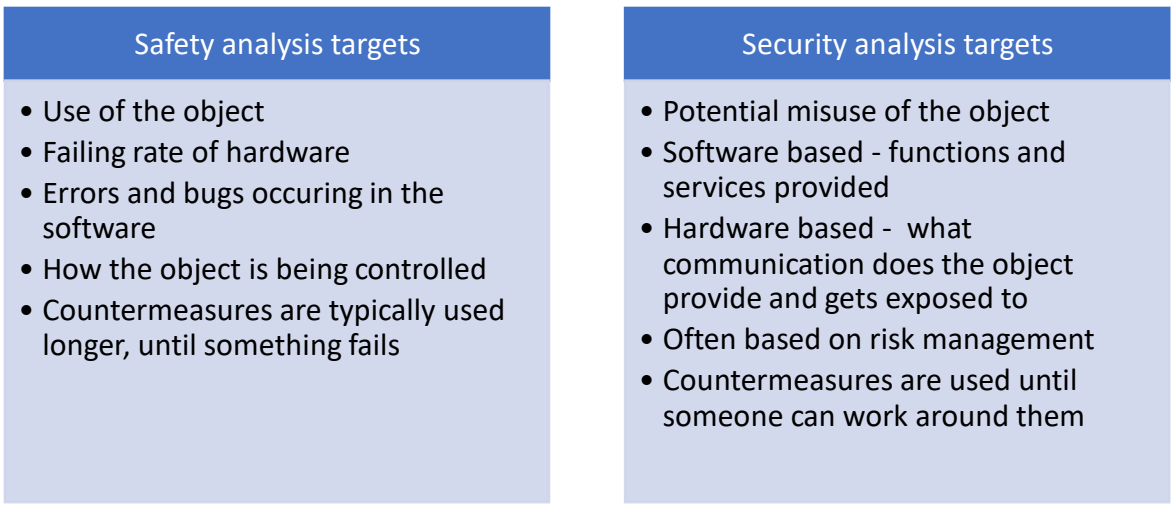


Figure 22 - Analysis methods focus on areas when targeting either safety or security

Therefore, when choosing an analysis method. The choice should be based on what the object is exposed to.

Method classification

A survey of methods for safety and security has created a classification of security and safety co-analysis methods [3]. The classifications are as follows:

- Generic
- Model-based Graphical methods (including Extended fault trees and Formal methods)
- Non-graphical methods (Informal and Formal)

Method origin

What should also be taken into consideration is the origin of the method.

Is the method an extension of an existing method or a combination of existing methods? Is the method Component-based or Systems-based?

This topic is further discussed in chapter 9, when comparing the results of the Revolt case study.

3.1.3.2 Combining security and safety analysis

An article published in IEEE Transactions on Industrial Electronics [28] argues that when developing systems, separating safety and security could have disadvantages. The consequences might be increased costs, longer Time-to-market, reduced performance and higher complexity. What is also argued for is that safety and security issues should be taken into consideration in the design, operation maintenance, and decommissioning phases. Essentially the whole life-cycle of the system. The truth is that there can be many interdependencies between safety and security, which is described more closely in the article published in Reliability Engineering and System Safety, in 2015 [3].

Advantages

There are different studies that propose that it is necessary to consolidate the security and safety co-analysis [3]. There is different reason for this, some of them are:

- Security breaches can bring risks to system safety
- We can learn from approaches from both sides, because safety and security are a duality
- There are dependencies between safety and security

An important reason is that security could affect safety, and there are interactions between them.

Disadvantages

There are also disadvantages to combined safety and security into a co-analysis. These disadvantages have been discussed [3] [27] and some reasons are:

- Could reduce developers' understanding of the system and prevent a thorough analysis
- Could hide the requirements conflicts that it aims to resolve

A countermeasure to the disadvantages mentioned, a classification for Safety–Security interactions could be used, as described in the next page.

Safety–Security interactions can be classified into four categories

- Conditional dependency: Satisfaction of safety requirements conditions security or vice-versa.
- Mutual reinforcement: Satisfaction of safety requirements or safety measures contributes to security, or vice-versa, thereby enabling resource optimization and cost reduction.
- Antagonism: When considered jointly, safety and security requirements or measures lead to conflicting situations.
- Independency: No interaction.

These categories have been described in the article by Piètre-Cambacédès [29].

3.1.4 Qualitative and quantitative analysis

What to have in mind, is the method qualitative and quantitative based?

This raises a general question about the differences between qualitative and quantitative research.

A quantitative analysis does depend more on what data is available, often this type of analysis is only performed with risks that already has been picked out for further analysis. Therefore, a qualitative analysis is often performed first to pick out some risks and then used as input for a quantitative analysis.

This topic is further discussed in chapter 9, when comparing the results of the Revolt case study.

3.1.5 Risk Management

Risk management is often the main aspect used in a security analysis. The main concept of a Risk Management process is finding out identifying risks, analyse their impact and evaluate how they can be mitigated and eliminated.

Risk assessment stages coverage – which are covered by each method?

- Risk Identification
- Risk Analysis
- Risk Evaluation

A way we can evaluate the different safety and security co-analysis methods are if they cover the different steps of a risk assessment, doing so based on their definitions their activities and their practical results.

It makes sense to evaluate the different risk assessment models, that are used in each method, because each could have their different weaknesses and strengths that could be used to improve each method.

This topic is further discussed in chapter 9, when comparing the results of the Revolt case study.

3.1.6 Functional safety and cyber-security ambiguities, overlap and differences

When we are mentioning the two terms “Security” and “Safety”, in regard to autonomous systems, one might think that there are a specific meaning and purpose for them. The truth is that in the context of autonomous systems, these terms might be associated with different meanings. The reasons behind these claims, I will now try to describe.

Different standards to different industries

Autonomous systems are today in use in vastly different industries. For example, we can first take the case of an Autonomous automotive system – the self-driven car in the Automotive industry. Then we take the case of Autonomous ships for cargo shipping, in the maritime industry. Both of these industries have different standards they use when developing, designing, operating and manufacturing their autonomous systems

There are not, as of today, any standards in use for specifically Autonomous systems. However, on the 5th of April, 2016, The IEEE Standards Association introduces the program “Global Initiative for Ethical Considerations in the Design of Autonomous Systems”. This program purpose is to identify need to develop standards, certifications and codes related to Autonomous Systems [30].

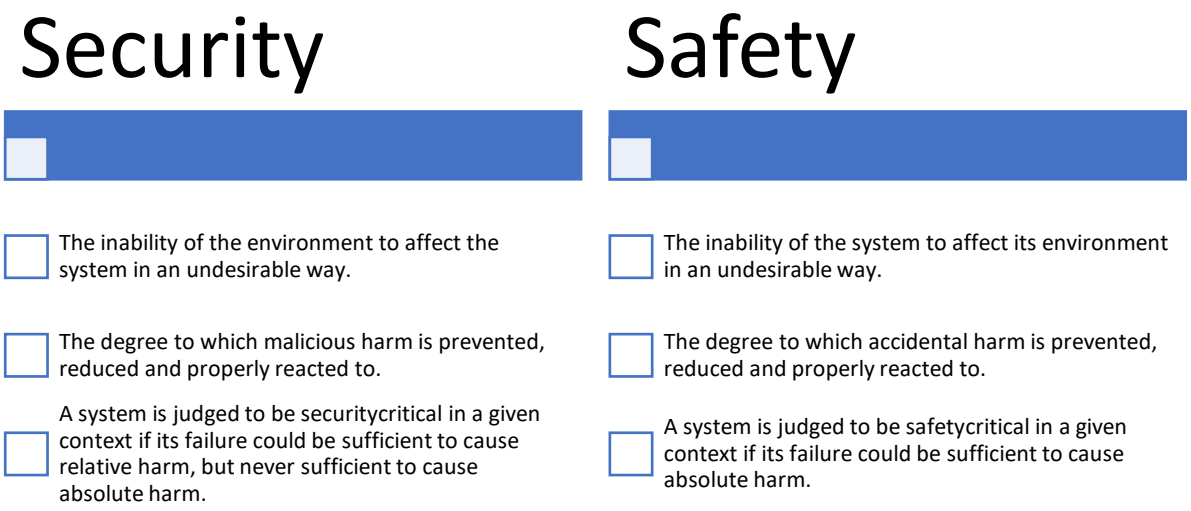


Figure 23 - Different definitions in different standards [31] [32] [33]

Therefore, when the different industries are developing their autonomous systems. They take in different standards, and the terms security and safety have different definitions in these standards, as illustrated in Figure 23. As a result, there ambiguities related to these terms.

Different engineering disciplines

What we also have to take into considerations when using these terms are the different engineering disciplines that are involved when developing these autonomous systems. Which is very well described in the article by Ludovic Pièrre-Cambacédès and Claude Chaudetb. As we can see with taking an example from the article.

> *"…situations such as the recent coordination between the US Federal Energy Regulatory Commission and the US Nuclear Regulatory Commission related to cyber security for nuclear power plants. This is a scenario where security and safety have to be considered from a triple perspective: power grid, nuclear power generation and control systems/telecommunications." [26]*

Now, if we transfer this example to our case with autonomous systems in a holographic context. The safety and security terms should have been seen from the perspective from all the engineering disciplines that it is related to.

What are the commonalities across the different professional disciplines?

I would say that there are some clear distinctions that separates the security and safety terms from each other, regardless of the professional disciplines. This has to do with what kind for risk the terms describe, analyses and takes into account. SEMA reference framework have a way of distinguishing these two terms and is described below [26].

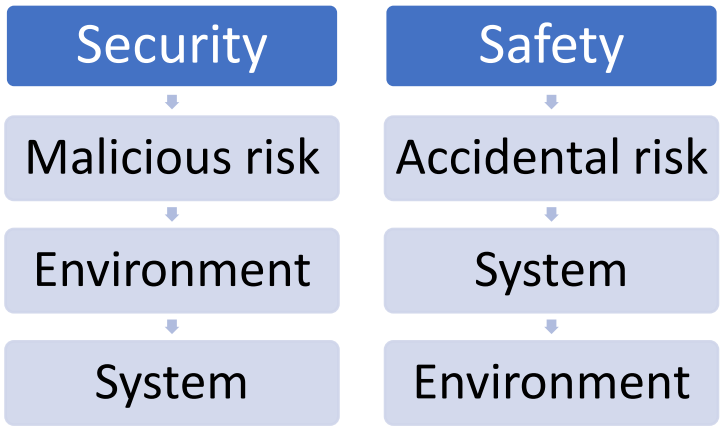
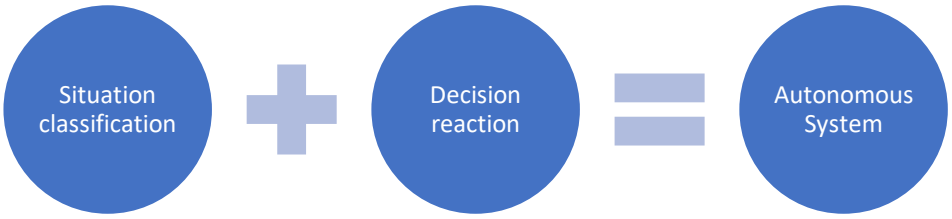


Figure 24 - Security and Safety distinctions

What is described in Figure 24, is what types of risk that gives a differentiation to the safety and security terms. The security topic takes into account malicious risk, for example unwanted persons like hackers and sabotagers. In this case, the environment exposes a risk to the system. On the other hand, the safety topic takes into account accidental risk in form of unwanted situations e.g. accidentally an error occurs on the system. In this case, the system exposes to a risk to the environment.

What are common in autonomous systems?

There are autonomous drones, cars, boats and trains. But what they all have in common is the ability to have various levels/grades of autonomous capabilities. The key mechanism in an autonomous system is situation classification and the decision that follows when an unexpected event occurs.



What was before the human's responsibility is now, in different degrees the systems responsibility, and that is was makes the system autonomous. For example, a self-driving car are driving on the road and an obstacle are in the road. When this situation occurs, the system has a set situation response and will react accordingly. Therefore, it makes the decision to break and stop the car to react to the situation. The variables for the autonomous systems are the hardware (different sensors, communication etc..) and the software protocols on how to react.

3.2 Safety and Security Standards specific to CPS systems

There are as mentioned specific standards for safety and security in the automotive domain. However, in recent times there have also been developed standards specific for CPS systems. What we need to take into consideration is that a system could be a CPS system and also an automotive system or other variations. Which standard or multiple standards these systems should follow is something that needs to be discussed for each system.

International Society of Automation (ISA) have proposed two standards to address safety and security for CPS systems:

- ISA84 standard (also called IEC 61511) on safety instrumented systems
- ISA99 standard (also called IEC 62443) on control system security

This has been discussed in a article by Giedre Sabaliauskaite [34]. What has been proposed in this article is an alignment of safety and security based on these standards for CPS systems. This has been done by merging the two standards, since ISA84 is related to safety and ISA99 to security. And creating a unified model – Failure-Attack-CounTermeasure (FACT) for the different lifecycle phases, based on both safety and security and this is specifically done in the development phase of a system.

3.3 Safety and security of Automotive Systems

3.3.1 Safety in Automotive Systems

The current definition on functional safety in automotive systems is stated in the ISO 26262 standard [35] as “absence of unreasonable risk due to hazards caused by malfunctioning behavior of the Electrical/Electronic systems “. In other words, there are very high standards that needs to be meet when coming to essential components in vehicles. Understandably so, a component failing or not failing could be a matter of life or death.

In the article written by Catalin-Virgil Briciu and Ioan Filip, these challenges are discussed, with taking into consideration the available safety standards to the car manufactures [36]. An interesting finding is that standardization of functional safety is useful to give clear methods when developing systems. But the development process is complex and not so easy when every fault causes must be considered. Therefore, strong analysis methods are very useful for finding faults, issues or bugs that are not that easy to detect with just regular testing and might have an impact to the end-users.

The ISO 26262 standard is an international standard for automotive functional safety [35]. In this standard, concepts and methods for safety in vehicles are described. For example, risk assessments and hazard analysis methods and ASIL, and the risk classification scheme.

The potential for use of the hazard analysis method STPA (System-Theoretic Process Analysis) in an ISO 26262 standard process has been analyzed in the article shown at the Springer International Publishing Switzerland in 2016 [37]. It is a known fact that the manufacturers of cars and their suppliers, strive to use the ISO 26262 standard with their products. The ISO standard contains both hazard analysis and risk assessment methods

(HARA). STPA can be argued to be a fairly new analysis method to the automotive industry. What the main difference between the current analysis methods in the ISO 26262 standard and the STPA method are the risk assessment, or the lack thereof in the STPA method.

However, an interesting discovery is that the STPA method does not interfere with the ISO 26262 standards risk assessment and only demand modest augmentation to be used in an HARA complaint process [37]. This was tested and implemented on an automotive subsystem, with cooperation from an automotive OEM (Original equipment manufacturer). The use case was a Battery Management System (BMS) of a Plug-in Hybrid Electric Vehicle. The STPA method was mainly implemented in the concept phase of development (ISO 26262) to see if it could also be used to include a risk assessment.

Thus, I believe that the STPA method can be used together with the existing standards in the automotive industry, and there is a great potential to take use of its advantages.

3.3.2 Cyber-security in Automotive Systems

When referring to security in the context of automotive Systems, it is regarding the protection of the vehicle system for undesired access. Cyber-security of critical systems is becoming more and more important, especially when it comes to the automotive industry. In which we have seen many innovations from, in the recent years.

In today's vehicles, the main network is based on a CAN bus (Controller Area Network) within this bus network, there are multiple ECUs (electronic control units) connected. These are used to control the components on the car e.g. control the lights, engine and other sensors. The article written by Sam Abbott-McCune and Lisa A. Shay explores different hacking techniques that can be used to exploit the CAN bus [38]. There were not many security concerns when the CAN bus was developed, because it was intended to be used in an isolated environment. In recent times, the CAN bus has been given access to CPS systems (cyber-physical systems). These CPS systems have given the capabilities to for example APS braking and cruise control.

In the article mentioned, to test ECUs and CAN bus network, there has been used a bench-top system to simulate the environment. Different hacking techniques are used to try to take control of the different ECUs on the simulated vehicles, and eventually take control of the vehicle.

Security vulnerabilities to physical object – Case: vehicle

The test result from the study are shown in Table 2. The test vehicle was hacked using an OBD-II connector, together with various cables and adapters. To capture the data on the CAN bus, an algorithm developed using python was used. We can see from the results that almost all CPS actions was completed.

In other words, the physical side of a CPS system could also have vulnerabilities. It is not only the communication layer and virtual world that could expose a risk.

Table 2 - Test results from pilot

Vehicle functionality	**Primary Vehicle**	**2014 model**
Remotely lock and unlock vehicle	Completed	Completed
Remotely open the trunk	N/A	Completed
Remotely start the vehicle	Completed	Completed
Remotely honk the horn	Completed	Completed
Roll down the windows	Partially Completed	Partially Completed
Adjust the driver's seat	Not Completed	Completed
Turn on/off the wipers	Completed	Completed
Turn on/off the lights, high/low beams	Completed	Completed
Adjust the mirrors	Completed	Completed
Adjust the radio controls	Completed	Completed
Adjust the AC/heat controls	Completed	Completed
Activate the rear camera	Completed	Completed

SAE J3061

The new Cybersecurity Standard SAE J3061 is a guideline for automotive cyber-security engineering. It was published in January 2016 at the SAE international, and was excepted to fulfill the need for guideline to security engineering for automotive systems in modern vehicles. The automotive industry has seen many innovations in recent time and has gone through a technological shift and a new standard for security of these new innovations was needed. An article from the Vienna University of Technology, discusses this new standard, with taking into consideration the existing ISO26262 standard, in which is more focused on safety [39]. In the article, there is described how to apply the new SAE J3061 Standard in a use case of an in-car ECU (electronic control unit), and is specifically used in the concept phase of the development life cycle. Useful experiences from applying this standard are mentioned, and the differences from the ISO26262 standard are pointed out and discussed.

Security against Road-to-vehicle communication system

In Japan, there has been implemented a system for communication between vehicles and control centers called Road-to-vehicle. An example of this is the ETC system (electronic toll collection), which is used for collecting tolls electronically. There are possibilities to exploit this system with side-channel attacks against computationally secure cryptographic circuits. These methods for spoofing or falsification of data has been considered in the article published by the Meijo University, Japan [40].

3.4. Safety and Security of Autonomous Systems

3.4.1 Safety in Autonomous Systems

I would say that the functional safety aspect is particularly important when regarding autonomous systems. Depending on the level of automation the system is capable of, as described in figure 1. The higher the level of automation, the human driver are less and less responsible of the safety aspect, and the system itself and the infrastructure it uses, are more and more responsible for the safety. Therefore, the key mechanism in an autonomous system is situation classification and the decision that follows.

If we take an example of the current definition on functional safety in automotive systems, as stated in the ISO 26262 standard: *"absence of unreasonable risk due to hazards caused by malfunctioning behavior of the Electrical/Electronic systems "* [35]. In other words, there are very high standards that need to be meet, when coming to essential components in vehicles. Especially in vehicles that are autonomous, and do not relay on humans. Understandably so, a component failing or not failing could be a matter of life or death.

In the article written by Catalin-Virgil Briciu and Ioan Filip, these challenges are discussed, with taking a base in the available safety standards to the car manufactures [36]. An interesting finding is that standardization of functional safety is useful to give clear methods when developing systems. But the development process is complex and not so easy when every fault causes must be considered. Therefore, strong analysis methods are very useful for finding faults, issues or bugs that are not that easy to detect with just regular testing, and might have an impact to the end-users.

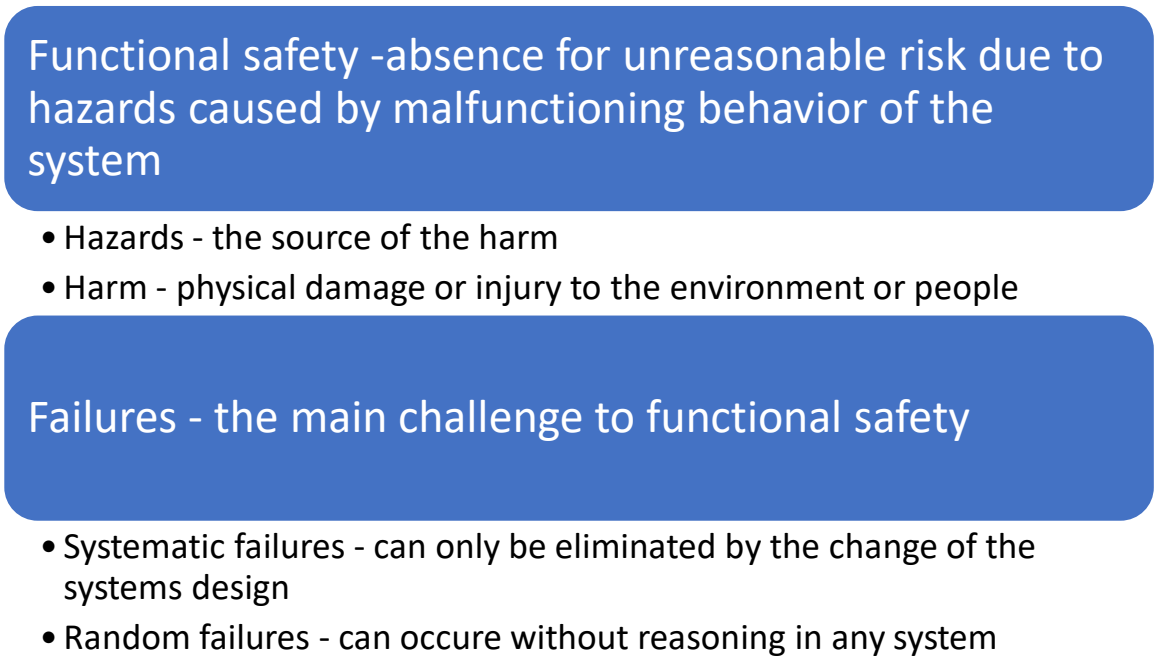


Figure 25 – summary of functional safety

3.4.1.1 Tesla driver killed in crash with Autopilot active

On May the 7th in 2016 in Williston, Florida. The car driver of a Tesla Model S was killed due to a tractor trailer that drove across the highway and the Tesla crashed into it [41]. The autopilot was on when this fatal crash happened, but the autopilot did not make any decision to try to stop for the tractor.

Figure 26 - Jasper Juinen, Bloomberg via Getty Images

This is reported to be the first fatality with a Tesla with autopilot mode activated.

> *"Autopilot is getting better all the time, but it is not perfect and still requires the driver to remain alert."- Tesla says to The Verge.*

In other words, there might be some time before cars can be operated only by an ADI. However, in recent times there have been devoted a lot of effort in developing driverless technology. In South Korea, Driverless cars have their own city [42]. This city is called "K-City" and is 3.45 million square feet of testing space for companies that need to test their driverless cars. This is a space that could further develop theses autonomous systems to make them safer, and incident like the one mentioned earlier can be avoided.

3.4.2 Security of Autonomous Systems

When referring to security in the context of autonomous Systems, it is regarding the protection of the system form undesired access. Cybersecurity of business-critical systems is becoming more and more important, especially when it comes to the industries that invest in autonomous capabilities. In which we have seen many innovations from, in the recent years.

If we take the example of the very complex system of the internet, this system has been around for a while and have been heavily studied on security threats. However, networks surrounding the autonomous systems has not been studied as much. I would say that the main reason is that many of these autonomous systems are still in development, and has not been exposed to as many threats. However, security issues and vulnerabilities regarding these systems are emerging.

According to an article published by the IEEE computer society in 2013 [12], most of the effort regarding the security design on autonomous vehicles, has been on encrypting of wireless channels. Both from the wireless channels and the embedded system itself.

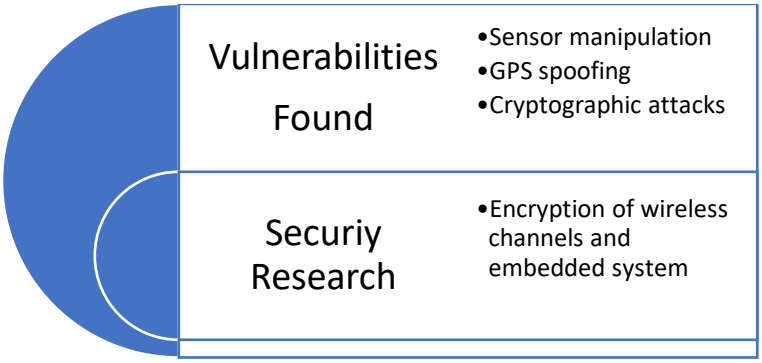


Figure 27 - Comparison of research security efforts on autonomous systems and some vulnerabilities found

As little research effort has been in made regarding security on autonomous systems, there have been published public articles regarding vulnerabilities on them. These demonstrates vulnerabilities on autonomous systems, in which are in use today, e.g. in an article that was presented at the 2010 IEEE Symposium on Security and Privacy [43]. Several demonstrations on manipulation on the sensor of an autonomous car, by nonconventional methods, e.g. through the entertainment system of the target vehicle.

In the next pages, I will describe three different examples of security incidents with autonomous systems. This will involve a car, drone and ship.

3.4.2.1 Jeep recalled 1.4 million Cherokees

What happened in 2015 was a transparent example of how the autonomous vehicles can be remotely controlled and hijacked. A demonstration of this was in a reportage from the American web and paper-based magazine i.e. Wired [44]. In which focuses on the IT and wireless industry, content industry and telecommunications. The hackers who demonstrated this vulnerability were Charlie Miller and Chris Valasek. They used a feature that was built into the vehicle's "entertainment system" or "head unit", called "Uconnect". The Uconnect feature provide a service that is connected to the internet through a standalone LTE connection.

The Uconnect feature could be compromised by the following security threats:

- Query for information – GPS coordinates, Vehicle Identification Number and IP addresses
- Run commands
- CAN messages (CAN bus - Controller Area Network)

What was an even more disturbing discovery, is demonstrated when the Jeep was below a certain speed and in reverse. The attacker could control the steering wheel, set the speed and disable the breaks. The results are shown in figure 5.

Figure 28 - Demonstration of completely killing the Jeep remotely by hackers

What eventually happened at the 24th of July 2015, was that Fiat Chrysler Automobiles order recall of 1.4 million vehicles that was vulnerable to the threat demonstrated. This recall was pressured from not only their costumers, but also the government of the United states. The congress had a real concern for the safety of the drivers of these cars, as the hacking method had been made public. Therefore, was this first of its kind recall initiated. What the so-called fix for this recall was a software update of certain radios that could be vulnerable for hacking [45].

The attack surface of an autonomous automotive system, in form of a Jeep vehicle, are displayed in Figure 29, collected from an article published by the IEEE computer society in 2013 [12]. One security incident in 2015, caused the recall of 1.4 million Jeep Cherokees [45]. The hackers connected to the vehicle through its wireless transmission hardware, in form of an LTE standalone connection, 10 miles away. From there, the onboard vehicles electronics where targeted- the entertainment system, and specifically the Uconnect feature. Uconnect controls the vehicles entertainment and navigation system. By having control of this software, they could perform query for information and run commands on the CAN bus.

Figure 29 -Attack surface for an autonomous automotive system

In today's vehicles, the main network is based on a CAN bus (Controller Area Network) within this bus network, there are multiple ECUs (electronic control units) connected. These are used to for example control the lights, engine and other sensors. The article written by Sam Abbott-McCune and Lisa A. Shay explores different hacking techniques that can be used to exploit the CAN bus [7]. There were not many security concerns when the CAN bus was developed, because it was intended to be used in an isolated environment. In recent times, the CAN bus has been given access to CPS systems (cyber-physical systems). These CPS systems have given the capabilities to for example APS braking and cruise control.

What is evident is that there is both techniques to hack the CAN bus of modern autonomous automotive vehicles, both with the physical use of wires [38] and also remotely through an LTE connection [44].

3.4.2.2 Security vulnerabilities on the Parrot AR.Drone 2.0

Unmanned aerial vehicle (UAV) or more commonly named drones have traditionally been related to military use, when the initial drones were costly to make. In recent times, the costs and advances in the components of which the drones compose of, has been lowered. The market of drones has been made open to the regular consumers. With that comes safety and security concerns, as we have seen multiple incidents related to these drones.

As drones have been more and more easily available, there has been done research in security vulnerability related to them. In 2013, Samy Kamkar demonstrated with the Parrot AR. Drone 2.0, as shown in Figure 30. That there is a possibility to hijack other drones, with a special drone called SkyJack [46]. To create this special drone, he used amongst other things a Raspberry Pi, wireless transmitter, USB battery and a combination of software. What this drone was capable of was to seek out and detect other drones, then hack the drones over the WIFI signal it broadcasts. The drone will be completely under the control of the attacker and basically creating an army of zombie drones.

Figure 30 - Parrot AR.Drone 2.0

Also, an interesting discovery was made in the article published at the conference of SPIE (the international society for optics and photonics) [47]. In the article there is performed a security threat analysis on the Parrot AR.Drone 2.0, as previously mentioned. What is demonstrated is multiple security problems in the design of the drone. Essential points that are brought out is the obvious lack of encryption of the WIFI connection to the drone, in which makes it possible to eavesdrop the videostream transmitted. An external network connection is showed to secure the drone from attacks on the WIFI connection.

GNU/Linux is used as user management on the drone, in which did have issues and there was discovered a backdoor. The first step was to connect to the hotspot the drone creates, in which is unencrypted. Then to perform a port scan (Nmap) on the IP-address to the drone. A number of interesting ports were discovered, for example a telnet server port, in which would give access to the root shell of the device (root account not password encrypted). After connecting to the telnet port, access to the root files where given and there was found

shell scripts. These could be used, changed and given commands. Therefore, the drone is essentially under the attacker's command.

What we can see from Table 3, are the results after connecting to the telnet port on the drone and an attacker is given multiple possibilities to take advantage of weak software engineering. For example, by changing the parameters on the "reset_config.sh" file, the attacker might choose to change what files the reset button effects. By that way, other malicious code that the attacker has inserted to the filesystem will still be affected after the owner tries to reset the device.

Table 3 - shellscripts and Linux files on the Parrot AR.Drone 2.0 [47]

Filepath	Explanation
/bin/check_update.sh	Update script
/bin/init_gpios.sh	Initialization of GPIO ports used for connecting the navigation board to the SoC
/bin/mount_usb.sh	Mounting of USB devices
/bin/pairing_setup.sh	Shell script for pairing using the Smartphone app
/bin/parallel-stream.sh	Camera streams
/bin/reset_config.sh	Reset config.ini while keeping total flighttime value
/bin/umount_usb.sh	Unmounting USB devices
/bin/Wifi_setup.sh	Start of Wi-Fi connection and other services
/sbin/udevd.sh	Start udevd with udev_init launcher
/lib/udev/rndis.sh	Hook script called by udhcpc on rndis interfaces-related events
/usr/sbin/loadAR6000.sh	Additional Wi-Fi settings
/etc/inetd.conf	Superserver for FTP (/update and /data/video)
/etc/udhcpd.conf	DHCP server configuration for Wi-Fi network
/data/config.ini	Main config file

3.4.2.3 GPS spoofing or jamming attack – Manipulation of 20 ships in the Black Sea

With a jamming attack the goal is to make part of a system not serviceably or the whole system. In other words, this could be a form of sabotage. However, with the target of the communication system, the potential for performing some kind of physical attack could be possible and more doable. For example, stealing cargo or information of the ship when the communication system is not serviceable.

In Figure 31, such a spoofing attack is illustrated, and anti-spoofing methods are discussed in the article [48]. Stanford University has in many years done research on how to make the

scrutiny around the GPS technology more secure and protected against spoofing attacks. Among the methods is to use WAAS messages for authentication.

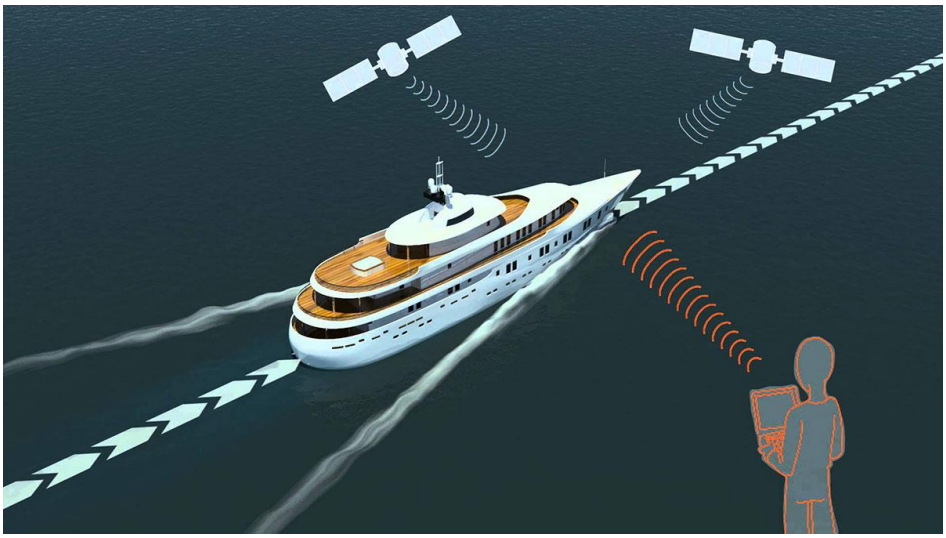


Figure 31 - GPS spoofing attack

With GPS spoofing there has in recent times been an interesting incident. At the 16th of August 2017, a GPS spoofing attack manipulated the location of 20 ships in the Black Sea [49].

This was a more sophisticated attack, by having the control of the ships location, the attacker had a lot of possibilities (stealing, sabotage etc.). As it says in the article, this was probably the first incident of its kind. However, in the future as autonomous ship are being developed. This could be a real threat.

3.4.3 Methods to combine security and safety analysis in autonomous systems

As previously discussed, there are both advantages and disadvantages involved when combining safety and security for a co-analysis. However, what to keep in mind is what application area the method will be applied for. There have been performed empirical case studies investigating methods for safety and security co-analysis for automotive cyber-physical systems [50], with promising results.

If the developers of autonomous systems choose to combine security and safety analysis, as there are significant of advantages in doing. Both the industries and researchers has started to acknowledge the need to fill the gap between these two areas. In recent time, there has therefore been developed analysis methods that are designed for that purpose, and can probably be used for analyzing security and safety of autonomous systems.

I would say that autonomous systems are in special need to develop a framework to combine security and safety. Not only because of the high risks these systems have to the environment. There seem to be a trend – there has already been made an effort in the safety area when designing these systems, but the Cyber-security has been neglected. I would therefore say there is great potential for progress by combining the two.

Chapter 4 : Research Motivation and Research Questions

In the article published in Reliability Engineering and System Safety, there has been performed a survey of approaches that tries to combine safety and security for industrial control systems [3]. The article aims to create an overview over the different approaches, and doing so, classifies them into generic, model based graphical and non-graphical methods. When the methods were typically used in the system life-cycle was described, either in the development or operational phase. Also, if the method was regarded as qualitative or quantitative. This was done with a total of 42 methods. An interesting point is that when the article was written (February 2015), there was not any specific achievements made in mastering this concept. There is still the challenge of dealing with the dependence between safety and security.

I hereby first introduce three safety and security co-analysis methods. Then I will present my research motivation and research questions.

Different co-analysis methods for safety and security are discussed and two recent approaches are especially focused on, FMVEA and CHASSIS. The authors arrive to the conclusion that these methods provide what the STPA-Sec methods lacks – high level concepts and action points during the process. However, FMVEA and CHASSIS also has its weaknesses. For example, with CHASSIS the same methods are used for analyzing security and safety, but they are done separately and there are no interactions between them. Another concern is that both FMVEA and CHASSIS do not demonstrate how to continuous perform safety and security co-analysis throughout the life-cycle of the system. This might be an issue when new security threats arise, and the risk assumptions change.

In the article published at the ACM Workshop on Cyber-Physical System Security in 2015 [50]. There is a case study of the FMVEA and CHASSIS methods used for safety and security co-analysis on automotive Cyber-physical systems (systems with interacting computational components and physical systems). I would argue that autonomous systems fall under the Cyber-physical system domain. What is argued by the authors is that events in the “cyperspace” and the physical world, poses challenges to safety and security, and a holistic approach should be considered. What is common in both of these systems, is that they rely on software for communication. This brings up an interesting point. There is a known fact that because security vulnerabilities of software and communication could give adversaries the possibility to attack and challenge both the safety and security of a system. I will therefore suggest, that when analyzing safety system of a system, security must be co-analyzed. The system is not safe until it is secure.

4.1 STPA and STPA-sec

4.1.1 STAMP

STAMP stands for Systems-Theoretic Accident Model and Processes and was developed by Prof. Nancy Leveson in 2012 [51].

STAMP is different from the existing chain-of-events model, in which was heavily used at the time STMAP was developed. In these events models, the main goal is to attempt to avoid accidents. This is not the goal with STAMP, the focus is on controlling the processes of the target system. The main philosophy with STAMP is that accidents are not caused by specific events, but the lack of control of processes in the system. If the components within the process are controlled, the process is safe. The essential logic with this model is that hazards are not classified as a control problem but rather a reliability problem [51]. However, what is not originally considered is that security could affect the safety of a system.
The control loop which STAMP is based upon is shown in Figure 32.

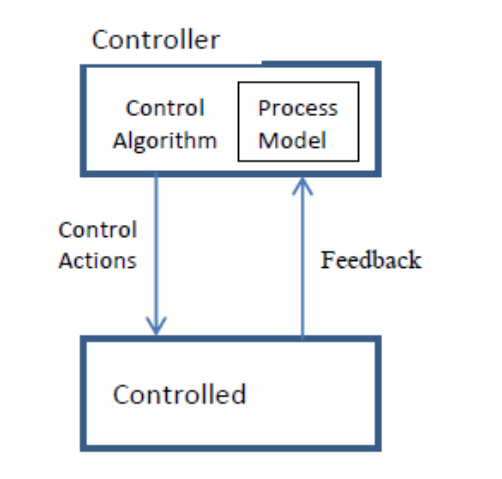


Figure 32 - Basic Control Loop [52]

"The system is conceived as a hierarchical structure, where each level enforces constraints on the behavior of components at the next lower level. These constraints control emergent system behavior, such as safety and security. Control loops operate between each level of this hierarchical control structure." [52].

The control loop is the essential element in the STAMP methodology. As the method is used, errors are detected. However, not only by components failing, but through interactions between several components in the control loop.

Table 4 - STAMP based methods

Name	Stands for	Type
STPA	Systems Theoretic Process Analysis	Hazard Analysis
STPA-Sec	Systems Theoretic Process Analysis for Security	Hazard Analysis security focused
CAST	Causal Analysis using System Theory	Accident/Event analysis
STECA	Systems Theoretic Early Concept Analysis	Safety-Guided design/Hazard identification method

In Table 4, the different STAMP based methods are listed. A STAMP-based method is the application of a process, and this process is based on the STAMP accident causality model. These methods will be discussed in this chapter, but also methods form other origins.

CAST

CAST stands for Causal Analysis using System Theory and is a casual analysis based on STAMP. CAST is used to determine the cause of occurring accidents of a system. Therefore, the insight this analysis method provides is finding out which events need to occur for the accident to be triggered. Then this information is use this for insight when developing new systems. The theory is that the accident happened based on a series of events and the circumstances of the accident needs to be changed in order to prevent it for happening again.

4.1.2 STPA

STPA stands for System-Theoretic Process Analysis [51], and is a Hazard Analysis based on STAMP. What the STPA method does is performing a risk analysis based on control actions, requirements and constraints. The approach for this analysis method is to identify causal factors of the control structure of the system and what interactions the system components has, then model the system in a hierarchical structure. What at the end is generated is system-level scenarios which can lead to losses.

STPA is an analysis method for detecting safety hazard. The foundation is from systems theory described in STAMP. With established safety approaches, hazards are treated as a control problem and not a reliability problem.

STPA is particularly useful in the development phase of the systems lifecycle, to create a foundation to understand the system while it is created and at the same time also documenting the requirements for the system.

Figure 33 - Overview of the STPA method process

An overview of the STPA method process are described in Figure 33. There are in total seven steps with the process STPA method provides.
Critics of this method is focused around the lack of details and describing the use case to be large-scale systems. This might be a too general analysis to be used for very complex systems e.g. autonomous systems. If the method can't find the necessary details, then high-level security and safety can't be assured for the system. The lack of details in the techniques the method describes, makes this a method with a high level of abstraction. One might therefore argue that this method needs improvements before being appropriate for real-world scenarios.

4.1.3 STPA-Sec

STPA-Sec does stand for the same as the STPA method does, but also includes the security aspect [52] [53]. Compared to other security analysis methods, STPA-sec does not focus on countermeasures that should be taken, but mainly on identifying scenarios that could lead to losses [1].

STPA-Sec is an extension of the STPA, in which extends the safety analysis method with security considerations. The same principles from the STPA method is applied. STPA and STPA-sec could be used together for a safety and security co-analysis.

The STPA-Sec extends the STPA method with these elements:

- Designing and framing the security problem
- Control structure: Identifying **unsecure** control actions with corresponding process model variables.
- Accident cause – identifying scenarios that could lead to unsafe or **unsecure** control actions
- Developing new requirements, control and design features to mitigate **unsecure** and unsafe scenarios.

Introduction and origin of STPA-sec

The article written by William Young and Nancy Leveson first introduces the new STPA-Sec method [52]. Challenges with the existing approaches and methods that are facing security professionals are discussed. These challenges are the main motivation for development and introduction of this method.

> *"Despite increased funding and resources, we do not appear to be making satisfactory progress in our ability to secure the complex systems that we are increasingly able to create. Arguably, new approaches are needed. This paper presents one such approach." [52]*

In the first section of this article there is a discussion of the current approach in the cyber security field, known as a chain-of-events model, where the main theme is to attempt to avoid accidents. Limitations of this method is thoroughly discussed, and the rest of the article are centered around introducing the STPA-sec method. The scope of this article is focusing on integrity and availability violations.

> "The scope of the paper is limited in that it focuses on losses resulting from violations of integrity and availability but not confidentiality violations. We believe these can be handled equally well within this framework." *[52]*

An interesting fact is that confidentiality violations was not considered. That's maybe the reason for development of the STAP-priv method [54]. In the article written by Schmittner, Ma and Puschner [55], limitations of the first version of the STPA-sec method are discussed, and with a focus on confidentiality concerns.

Challenges with STPA-sec

I find the article written by Schmittner, Ma and Puschner to take an interesting approach to analyzing the STPA-sec method [55]. Limitations are found when testing the method on a real-world use case, a Battery Management System for hybrid vehicles. The article then present possible improvements and applies these to the test case.

Other safety and security methods are considered and analyzed, such as SAHARA, FMVEA and CHASSIS. A thoroughly review of the STPA-Sec process are presented, and possible weak spots are commented.

We are presented with a figure of a control loop with potential starting points for the identification of unsafe control actions (Figure 34). A good point that is brought up in the article is the lack of security related elements, and does not capture the scenarios for an potential attack. This is an issue with the current approach with combining the STPA with STPA-sec for a safety and security co-analysis.

Extension of STPA-sec - In the test case, they have specifically improved the annotated control loop used in STPA for casual analysis for identifying unsafe control actions due to security attacks.

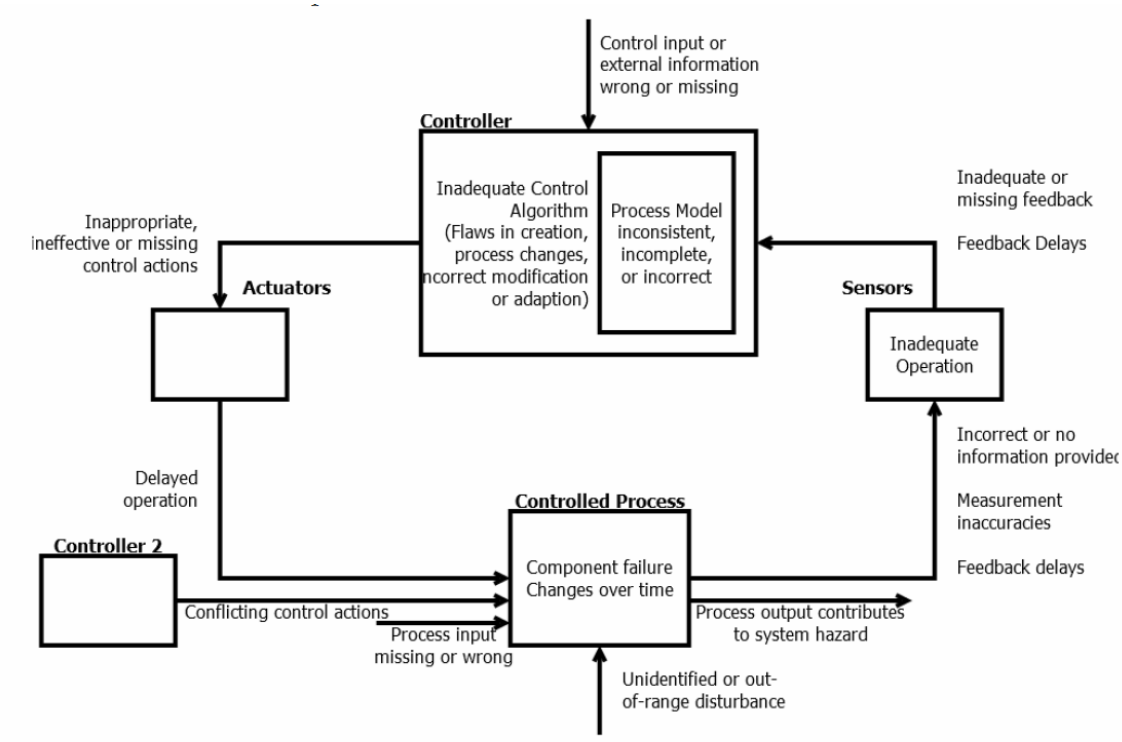


Figure 34 - Annotated control graph with scenarios for uncertain control actions

Case studies using STPA-sec

The article written by William Young and Nancy Leveson discusses different safety analysis techniques and approaches currently being used in cybersecurity. An interesting subject in the article is using strategy vs. tactics in security. The relationship between safety and security is discussed and gives references to STPA and STPA-Sec [53].

The article uses the case of a ballistic missile defense system, and introduced STPA-Sec in that scenario. In the article they published in 2013 [52], “Systems Thinking for Safety and Security”, they simply introduced the STPA-sec method, without any case scenarios. Therefore, a proven use case was necessary to get validity to the approach.

4.1.4 Comparison between STPA and STPA-sec

Table 5 – Differences in the sequence between A-STPA and STPA-Sec

STPA	STPA-Sec	Differences
Establish Fundamentals	Establish Fundamentals	No
System Description	System Description	No
System Goals	System Goals	No
Accidents	Losses	Yes
Hazards	Vulnerabilities	Yes
Linking of Accidents and Hazards	Linking of Losses and Vulnerabilities	Yes
Design Requirements	Design Requirements	No
Control Structure	High Level Control Structure Model	Yes
Unsafe Control Actions	Unsafe Control Actions	Yes
Unsafe Control Actions Table	Unsecure Control Actions Table	Yes
Corresponding Safety Constraints	Corresponding Security Constraints	Yes
Causal Analysis	Causal Analysis	No
Control Structure with Process Model	Control Structure with Process Model	No
Context Tables	Context Tables	Yes
Refined Unsafe Control Actions	Refined Unsafe Control Actions	Yes
Refined Safety Constraints	Refined Security Constraints	Yes
Causal Factors Table	Causal Factors Table	Yes
LTL Formula Table	LTL Formula Table	No

In Table 5 the differences between STPA and STPA-sec are displayed with the base in the sequence the methods provide, collected from [56]. The table does show if there are any differences in the way the steps are performed if one was to do a separate analysis of either STPA or STPA-sec. There seems to be significant differences between them, but to be noted, this is the process the method provides in the view that has been interpret by the author [56]. This might vary from the authors, and there seems to be more of a formal difference between STPA and STPA-sec. For example, “Refined Safety Constraints” and “Refined Security Constraints”. There are only different terms used, what they mean is a different subject and there is no clear guideline for this provided by the methods.

4.1.5 Related work with the STPA-sec

There has already been done work in the research field, in trying to improve the STPA-sec method. Two approached has been developed to improve the weaknesses of the STPA-sec.

STPA-SafeSec

The article published in the journal of information and security and applications introduced the method STPA-SafeSec [57]. In this method, they have reviewed the current approaches to cybersecurity and taken the backbone in the STPA-Sec method created by Young and Levison [52]. From there, they have identified in their views what the limitations of the STPA-sec method are and attempted to improve these areas.
A point they bring out is that there are separate methods for safety (STPA) and security (STPA-Sec). Therefore, they present a novel analysis method for combining both safety and security into the STPA-SafeSec method. Therefore, this methods could have better interactions between safety and security, than the current version of the STPA and STPA-sec.

STPA-priv

The STPA-priv method takes a base in the STPA method and focuses on privacy-based concerns [54]. Doing this by adapting the current method and changing it to be a more focus towards privacy throughout the process. For example, the word adverse replaces the word loss in the existing method, because this term is well-known in the privacy field. A major change is the introduction of “Privacy-compromising control actions” and “Privacy constraints”, these properties seems to be explicitly used for privacy and no other aspects.

4.1.6 Supported tools for STPA

STPA tool requirements:

What I think a STPA tool should have, based on testing different tools in this thesis:

Basic Text for the concept of the system – system description, goals, losses, hazards.
Linking objects. Because each activity in the STPA is connected to the previous activity – For example:

- Losses and hazards
- Unsafe Control Actions to Safety and Security constraints
- Control actions and process variables

Tool for creating the control structure, model process variables, arrows...
Table for UCA – linked to objects in the control structure. Process Model Variables mainly.

XSTAMPP

XSTAMPP is an open source platform tool that supports both the STPA and STPA-sec methods [58]. I would suggest some improvements on these points below.

The current XSTAMPP (version 2.5.0) tool don’t have the following:

- Correct UCA tables (as described in the thesis by John Thomas [59])
- Can’t add Causal Factors leading violation of Safety and Security constraints (and eventually Hazards) on the control structure
- Don’t directly link CA and Process model variables
- There are also a lot of bugs in XSTAMPP

4.2 FMVEA

FMVEA does stand for Failure Mode, Vulnerabilities and Effects Analysis [60].

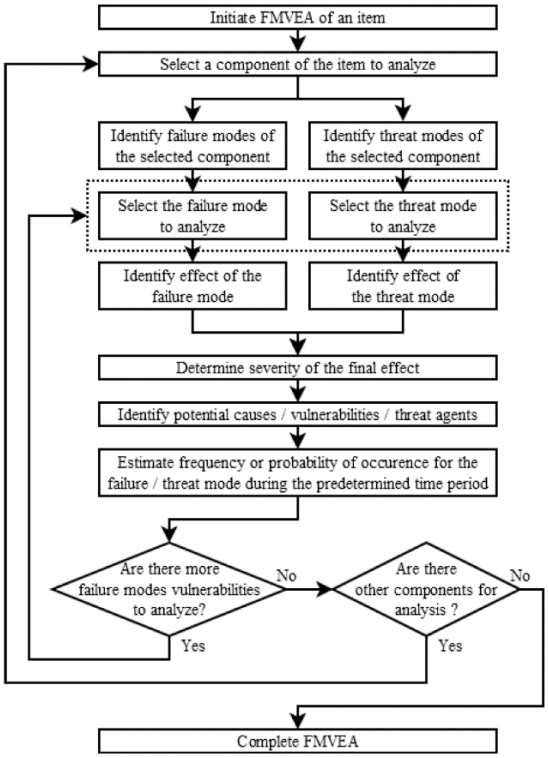


Figure 35- Overview of the FMVEA method

An overview of how the FMVEA method is implemted on a system, is displayed in Figure 35, in wich is based on a three-level data flow diagram(DFD). The method involves in its first step modelling of the system and then identifing failure and threat modes to each component of the system. The failure mode cover the safety ascpect, by describing the way the compoentent could potentially fail(Level 1). The threat mode covers the security ascpety, describing the way the compentent could be potentially missued (Level 2). The threat modes are based on the STRIDE model, in wich was developed my Mirosoft in 2002 [61]. What is dependent on creating failure and threat modes is knowledge about the system. The potential risks and the effect they could have, are each reletad to a component (context level).

FMEA and FMVEA differences

The FMVEA is based on the FMVEA method but extended to include the security related aspect. The key concept is that failure and threat modes are analyzed separately, and the effects are predicted. Both of these methods are Quantitative methods and use numerical values to describe risks.

Challenges with FMVEA

A challenge with FMVEA is that there is no possibility to continuously perform this safety and security co-analysis throughout the life cycle of the target system. Also, the activities in FMVEA is similar to the ones defined in the EBIOS methodology.

4.3 CHASSIS

CHASSIS does stand for Combined Harm Assessment of Safety and Security for Information Systems [62].

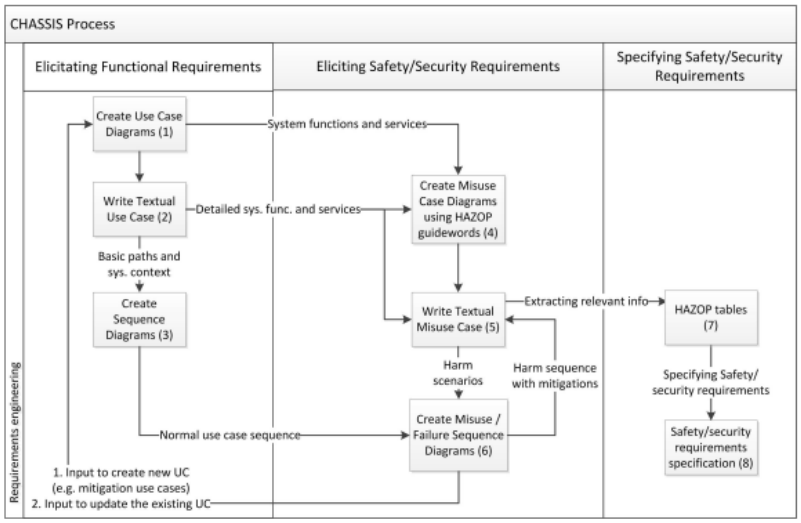


Figure 36 -Overview of the CHASSIS method

An overview of the CHASSIS method, is displayed in Figure 36. From what we can see, are sequence diagrams and use cases used to create functional and safety and security requirements. Step 1 is where the appropriate functional requirements is described so that they can be used for safety and security purposes. In step 2 the safety and security requirements are elected based on the requirements in step 1. The goal of the method is to identify possible safety hazards and mitigations for them. Equally, security threats and mitigations is incldued.

The CHASSIS method defines a complete combination of existing methods for safety and security assessments based on UML notations.

Existing methods used with CHASSIS

- Use cases (UCs),
- Misuse cases (MUCs),
- Sequence diagrams (SDs)
- Misuse/failure sequence diagrams (MUSDs/FSDs)
- HAZOP (hazard and operability)

Challenges with CHASSIS

A Challenge for the CHASSIS is that it requires a decent amount of expert knowledge before performing the analysis. The method includes different UML notation methods. Also, security and safety are done separately and there is no interaction. CHASSIS has the same challenge as FMVEA – can't perform safety and security co-analysis throughout the life cycle of the target system.

4.4 Research Motivation

Some factors that has been already established in this thesis:

- There is a known fact that because security vulnerabilities of software and communication could give adversaries the possibility to attack and challenge both the safety and security of a system. There are as previously discussed significant advantages by doing so. I will therefore suggest, that when analyzing safety system of a system, security must be co-analyzed. **The system is not safe until it secure.**

With the introduced methods: FMVEA, STPA and STPA-sec and CHASSIS:

- All of the methods have **challenges related to having valuable interactions between safety and security.**

To summarize, the challenges with safety and security co-analysis methods is interactions between safety and security. In which is the whole point of having a co-analysis method. If there are no interactions, it may as well have been separate analysis methods for safety and security.

If I can make any contributions in improving the weakness with safety and security co-analysis methods. They may be more useful and applicable for complex systems e.g. autonomous and CPS systems. But also, for systems in general in which demands a high control for safety and security.

To make any recommendation for improvements, I first need to compare the weaknesses and strengths of the existing methods and approaches for safety and security co-analysis. Based on the experiences I will make, I will gain insight to what might be the best interactions/interlink between safety and security, and how they could be used in its best form.

This research project could be a major contribution to the safety and security research fields. This could result in possibly more and better use of safety and security co-analysis methods in the future, once we better understand how to design them for the best results and coverage. Also, by having a complete list of how safety and security co-analysis methods are created, we have a better opportunity to compare how they are designed and understand them better.

A process goal for this thesis is to gain deep knowledge about safety and security. To tackle safety and security for autonomous systems is a major challenge, because the systems inside these objects are very complex. By learning about these systems and methodology used, I can this apply knowledge for other systems in my work.

4.5 Research Questions

The following methods, FMVEA, STPA combined with STPA-sec and CHASSIS, is proposed to be useful methods for safety and security analysis of autonomous systems with taking into consideration the challenges these systems have, both the safety and security aspects.

The reasoning behind picking these three methods, where that the FMVEA and CHASSIS have been used in a case study before, and these systems have similar challenges as the Revolt. These methods have been used for a case study on intelligent and cooperative vehicles, in 2015 [50]. This case study had some interesting results, and the case seemed comparable to our targeted case. Therefore, further research would be valuable for the FMVEA and CHASSIS methods, to see if they are applicable for autonomous systems.

The reasoning for picking STPA-sec is that this method has recently been proposed from MIT [63], but has not been thoroughly piloted in industrial case studies. By combining STPA with STPA-sec, both the safety and security aspects is covered. This is also a system-based approach, and a challenge is to try to understand the intelligence part of autonomous system. The STPA and STPA-sec will be piloted to test if they can help with this issue.

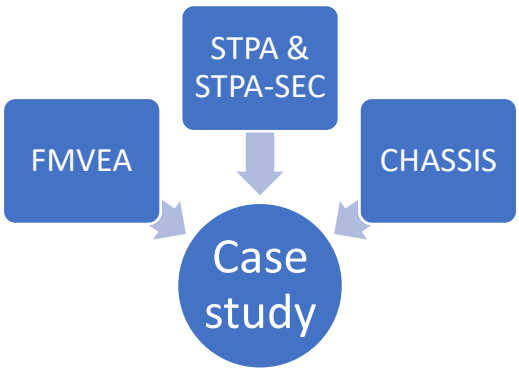


Figure 37 -Proposed methods for a case study of autonomous system – the Revolt platform

Another factor for picking these methods was that these three are from different categories of methods, as described in a survey of methods for safety and security. In this survey there has been created a classification of security and safety co-analysis methods [3]. FMVEA is picked from the *Generic approach* category, CHASSIS is from the *Model-based Graphical methods* category and STPA plus STPA-sec is from the *Model-based Non-graphical methods* category. Therefore, an approach from each category will be compared to measure applicability, efficiency, and hazards identified. With significantly different methods being used, the result might be interesting, and lessons can be learned from each method.

Research questions are as follows:

RQ1: How does existing approaches/ methods for safety and security co-analysis compare to each other?

RQ2: How to improve the weaknesses of STPA and STPA-sec for a better safety and security co-analysis?

Chapter 5 : Research Design and Case study

5.1 Introduction

The Revolt platform is an autonomous boat concept in a small-scale model, developed and implemented by DNV GL [1]. The Revolt is a test platform for an autonomous ferry, in which purpose is to transport cargo with minimal energy consumption and costs associated with it. The Revolt also serve as an early test platform for sensors and control systems dedicated for autonomous vessels. Therefore, the Revolt will be under ongoing development and is not a finished product. The safety and security analysis that will be performed will be taking a case of Revolt as the current version of 3rd of November 2017.

However, the revolt platform is still a valuable object to be used as a case study, for the following reasons:

- Explore Safety and Security issues related to autonomous steering of the vessel. Loss of control and steering of the vessel could result in loss of life or other damage.
- Explore Security issues related to data-communication between onshore and offshore systems. Sensitive data could be compromised.

The following figures and information is derived from documents on courtesy of DNV GL. Not all information needed is provided to perform the mentioned analysis methods. Therefore, some assumptions needs to be made. Also, some further figures and information are created from looking at the design of the Revolt model. Only the information considered necessary for the analysis to be performed are included in this chapter.

Many thanks to Jon Arne Glomsrud at DNV GL for valuable knowledge, and for allowing me the opportunity to use the Revolt model for this case study.

Stakeholders Revolt

Table 6 - Stakeholders with the Revolt vessel

Name	Role
DNVGL Operators	Operate the Revolt at sea and in harbors, docks
DNVGL Maintenance workers	Perform Maintenance on the Revolt
Cargo Customers	Buy cargo units on the Revolt and transport their cargo to the Revolt for transport
Dock workers	loading and dispatch of cargo on the Revolt

5.1.1 Overview of the Revolt platform

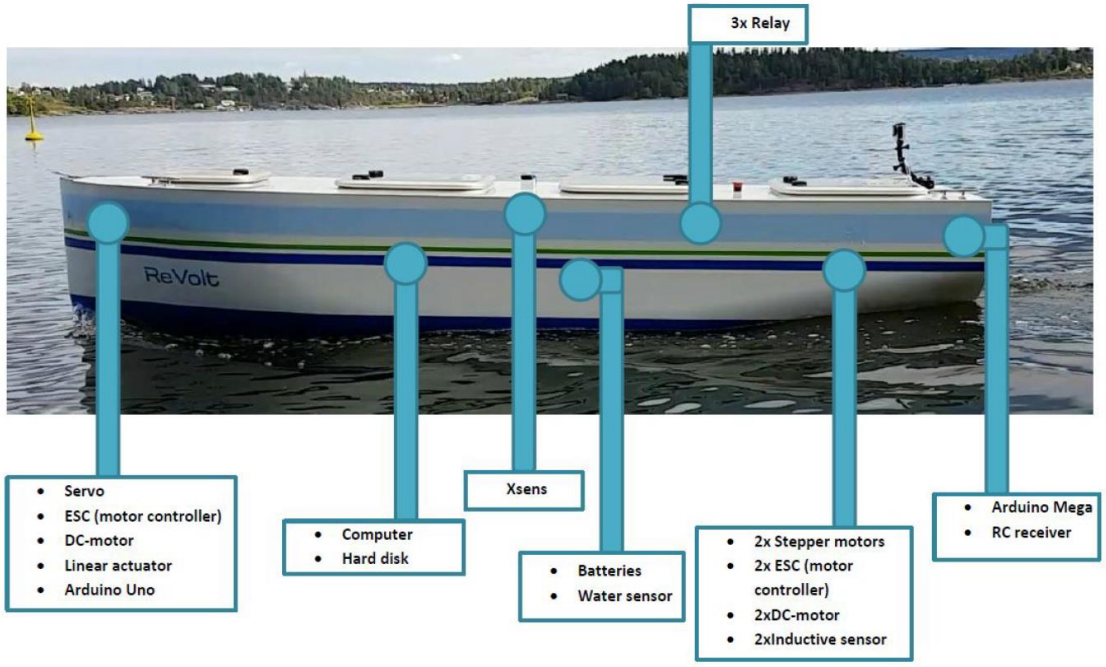


Figure 38 - Overview of the placement of the different components on the Revolt platform

The Revolt model as shown in Figure 38, this hardware was made by Stadt Towing Tank (STT), on a mission from DNV GL in 2014. The model is a 1:20 scale model of the concept ship. The model ship has a length of 3 meters and weighs 257kg.

No.	Name	No.	Name
1	Arduino Mega	2	ESC - AC Robbe Roxxy
3	Stern Thruster	4	Batteries
5	Tank-720 PC	6	Xsens MTI-G-710
7	Arduino Uno	8	Bow Thruster
9	ESC - DC Robbe Roxxy	10	Spektrum AR400 RC Receiver
11	Satel Radio Link	12	Hemisphere Vector VS330

Figure 39 - Overview of the placement of the different component on the Revolt

The different components that are placed inside the Revolt vessel are displayed in Figure 39. A number is connected to each component. This overview is collected from a previous thesis on the Revolt [64]. All of the components are listed and described in table 7.

System

The autonomous boat is also a CPS system, since it has both a connection to the physical world. Through sensing the environment around it with its installed components, it has a connection to the virtual world, trough sending/receiving messages from the operation central. Therefore, aspects designed for both autonomous and CPS systems should be considered in the case study.

5.1.2 Software

Table 7 - The different software used on the Revolt platform

Name	Version	Installed/used on component
Linux Ubuntu	14.04 LTS	Embedded computer
ROS	Indigo Igloo	Embedded computer
Python	2.7	Embedded computer
Java	1.7.0 121	Embedded computer
SSH	N/A	Remote computer
Archer MR200 firmware	N/A	4G Router

The different software that are used on the various components on the Revolt model are listed in Table 7.

Nodes on the ROS operating system

Table 8 - Nodes on the ROS operating system on the Revolt system

Node name	Package	Programming language
ControllerNode	controller	C++
rosserial_server_uno	actuators	C++/Arduino
rosserial_server_mega	actuators	C++/Arduino
RC Remote node	rcremote	Python
refFilterNode	dp_controller	Python
Stepper node	actuators	Python
Reference node	headingcontrol	Python
Translate node	headingcontrol	Python
headingControl	headingcontrol	C++
Xsens node	xsens_driver	Python
vectorVS330	nmea_navsat_driver	Python
DPcontrollerNode	dp_controller	Python
thrusterAllocNode	dp_controller	Python
Observer	observer	Python

5.1.3 Component list

Table 9 - Component list for the Revolt

ID	Category	Name	Placement	Model	Role
1	Thruster	Motor controller	Bow	Robbe NavyControl535R	Speed controller with forward, stop and reverse functions
2	Thruster	DC-motor	Bow	Robbe Roxxy Starmax 48	Rotate the propeller
3	Thruster	Linear actuator	Bow	Firgelli L16	Retract/lower the propeller house
4	Thruster	Servo	Bow	HiTEC HS-5485HB	Rotate propeller house to get desired thrust direction
5	Thruster	H-bridge	Bow	L293NE	The Arduino does not have a 12V output, therefore a simple circuit was created using a h-bridge
6	Thruster	Motor controller	Stern	Robbe Roxxy Control 900	Speed controller with forward, stop and reverse functions
7	Thruster	AC-motor	Stern	Robbe Roxxy BL-outrunner 5055-45	Rotate the propeller
8	Thruster	Stepper motor	Stern	Nanotec PD2-N41	Rotate propeller to get desired thrust direction
9	Sensor	Current meas. sensor	2x stern, 1x bow	Phidgets 1122_0	30 Amp Current Sensor, AC/DC In-Line
10	Sensor	Inductive sensor	Stern	XS618B1PAL2	Sensor for detecting metal targets approaching the sensor
11	Sensor	Xsens	Middle, top	Xsens MTi-G 710	Provides position measurements, accelerations and angular velocity. Provides: IMU, GPS, GNSS, INS, and magnetometer
12	Sensor	Water sensor	Under batteries	Homemade	Water sensor
13	Controller	Embedded computer	Middle, port side	Tank 720	Installed Robot Operating System (ROS), provides low level device control, message passing and more
14	Controller	Hard drive	Middle, port side	Verbatim 500GB	Hard drive for the embedded computer, read/write data
15	Controller	4G Router	Stern, port side	TP-Link MR200	Provide 4G internet by broadcasting a WIFI network, which the embedded computer connects to
16	Controller	Arduino Uno	Bow	Arduino Uno R3	Microcontrollers for handling analog input/output to some of the actuators, low level communication signals such as Pulse Width Modulation (PWM) signals to the ESC and servomotor, and radio receiver
17	Controller	Arduino Mega	Stern	Arduino Mega	
X	Sensors	Video Cameras	2x not yet mounted	Muvi K2 Sport	Provide video feed from the vessel
18	Power	Battery	Middle	Exide 12V 40Ah	Provide power to all the components

19	Power	Relay	Middle, starboard side	-	Switch on/off power to bow, port and starboard
20	Remote control	RC remote	-	Spektrum DX6i	For operating the Revolt by a remotely
21	Remote control	RC receiver	Stern	Spektrum AR610	Remote control receiver on the Revolt

The different components that are inside the Revolt model are listed in Table 9. The components are divided into the following categories Thruster, Sensor, Controller and Remote control.

5.1.4 Communication flow diagram

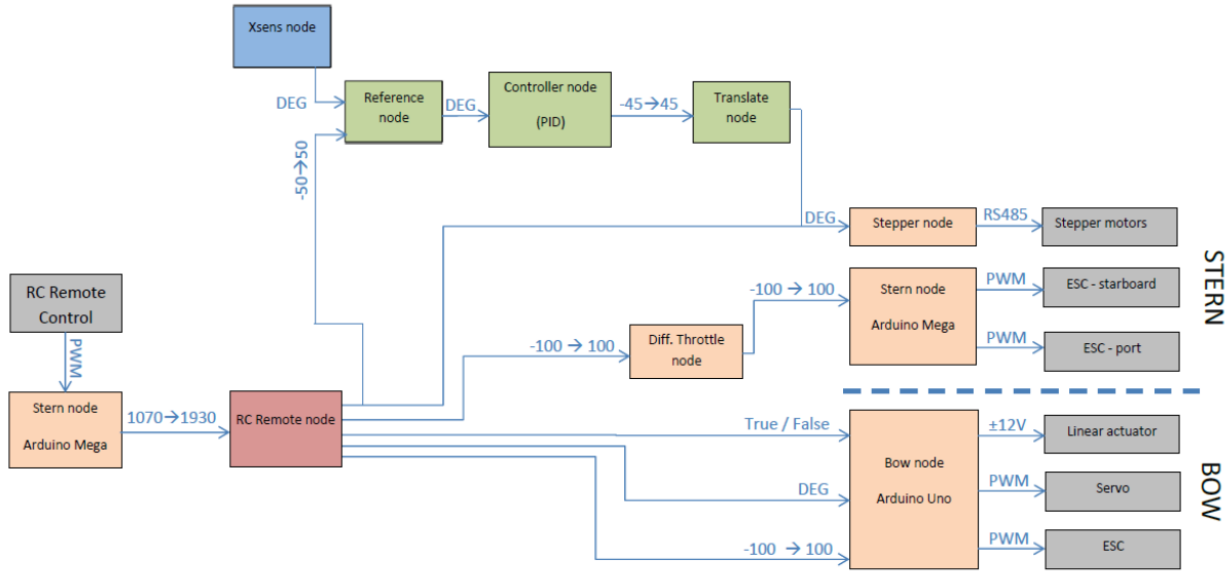


Figure 40 - Communication flow diagram of the Revolt hardware

5.1.5 Communication system

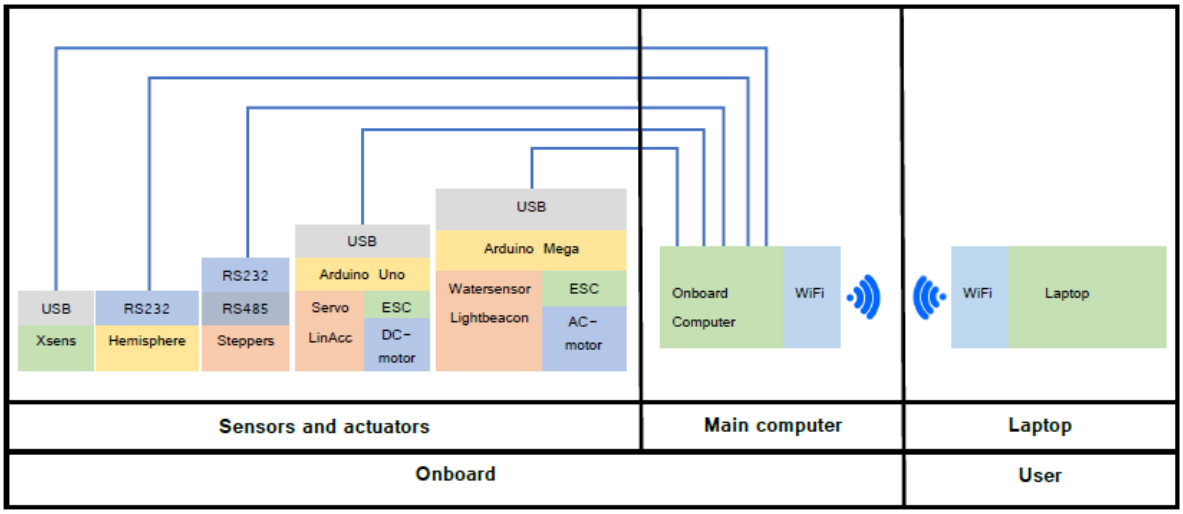


Figure 41 - Communication system of the Revolt system [64]

5.1.6 Wiring diagram of the Revolts system

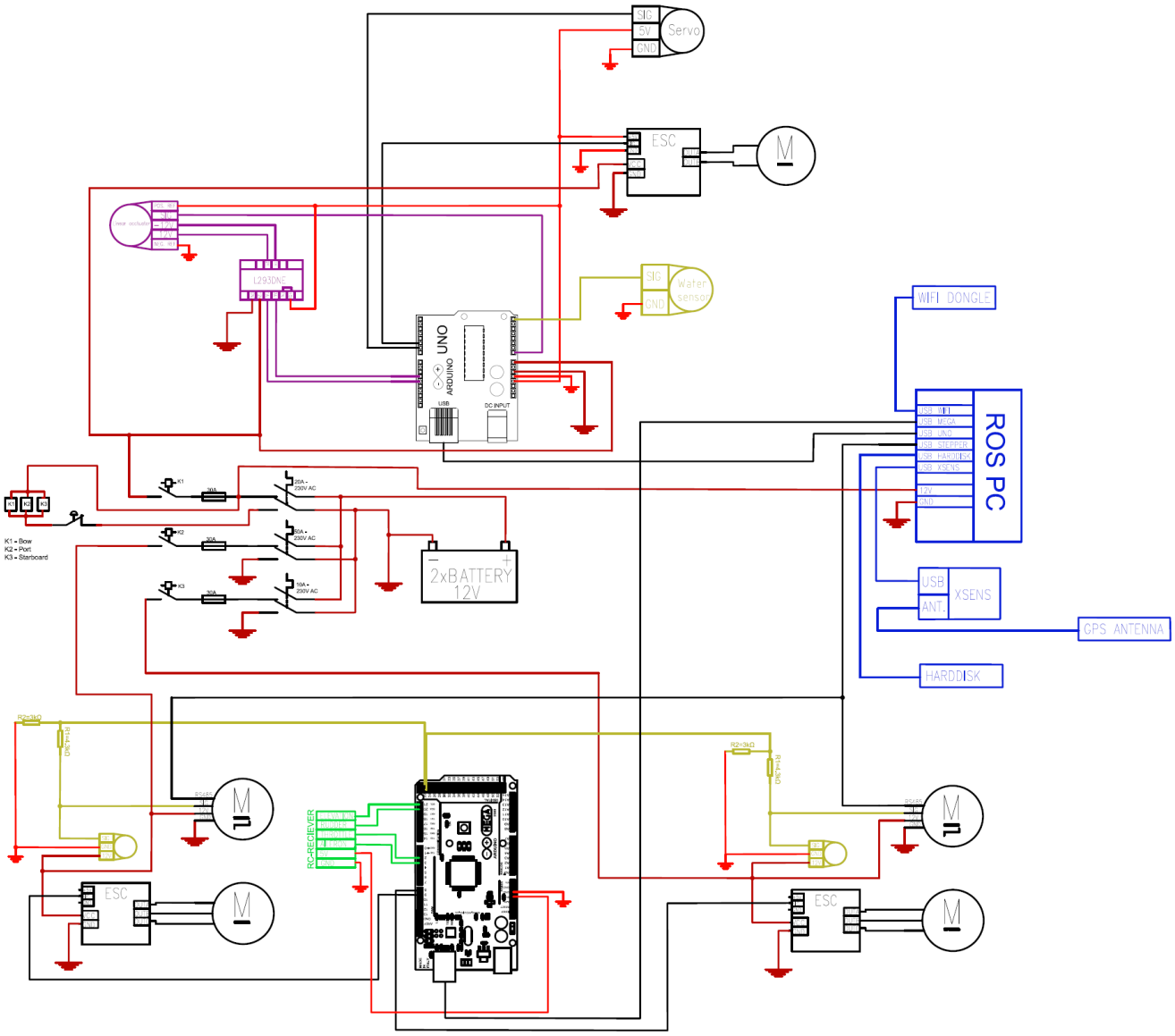


Figure 42 - Complete wiring diagram of the Revolt

In Figure 42, there is a complete wiring diagram of all the components within the Revolt vessel. From this there is created an I/O list. The color of the wires represents which component the wire originates from.

5.1.7 I/O list Revolt

In the following section a complete I/O list of the Revolt system is presented.

Digital

The Onboard Computer (OBC) in Revolt is a Tank-720. This is fanless and robust embedded computer running on the Linux Ubuntu operating system. A program called Robot Operating System (ROS) is installed on this computer. ROS is an open source operating system for providing low level device control, message passing and other functions [65]. This is what enables the sensors and actuators to be controlled.

Table 10 - Component I/O list: Embedded computer

#13 Component I/O list: **Embedded computer** - Tank 720			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	USB	WIFI	Digital
Output	USB	Arduino Mega	Digital
Output	USB	Arduino Uno	Digital
Output	USB (maybe RS-485)	Stepper	Digital
Input	USB	Harddisk	Digital
Input	USB	XSENS	Digital
Input	Power	12V	
Input	GND	Ground	

Table 11 - Component I/O list: 4G Router

#15 Component I/O list: **4G Router** - TP-Link MR200			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	Ethernet	Embedded computer	Digital
Input	Power		

Table 12 - Component I/O list: Hard drive

#14 Component I/O list: **Hard drive** - Verbatim 500GB			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	USB	embedded computer	Digital
Input	Power		

Table 13 - Component I/O list: Xsens

#11 Component I/O list: **Xsens** - Xsens MTi-G 710			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	USB	Embedded computer	Digital
Input	Some type of connector	GPS Antenna	Digital

Analog

For analog Input/output (I/O)s there are two microcontrollers, called Arduino Mega and Arduino Uno. These handle low level communication signals such as Pulse Width Modulation (PWM) signals to the actuators - Electronic speed controller (ESC), servomotor, stepper motors, radio receiver and other components.

Table 14 - Component I/O list: Arduino Mega

#17 Component I/O list: **Arduino Mega** - Arduino Mega			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Output	PWM 9	To ESC	Analog
Output	PWM 8	To ESC	Analog
Output	PWM 3	To RC-Receiver AILERON	Analog
Output	PWM 2	To RC-Receiver Throttle	Analog
Output	Communication SDA 20	To RC-Receiver Rudder	I2C (TWI) communication
Output	Communication SCL 21	To RC-Receiver Elevation	I2C (TWI) communication
Input	Digital 22	To sensor	Analog
Input	Digital 24	To yellow sensor	Analog
Output	Power 5V	To RC-Receiver	
Input	GND	Ground	
Input	USB Embedded computer - Tank 720		Digital

Table 15 - Component I/O list: Arduino Uno

#16 Component I/O list: **Arduino Uno** - Arduino Uno R3			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	USB Embedded computer - Tank 720	Digital	Digital
Output	PWM 3	Servo	Analog
Output	PWM 4	Ground	Analog
Output	PWM 5	Ground	Analog
Output	PWM 7	H-bridge	Analog
Output	PWM 8	Line actuator	Analog
Input	AREF	Water sensor	Analog
Input	GND -SIG	ESC	
Input	GND - SIG	Servo	
Input	Analog In A3 – 2A	H-bridge	Analog
Input	Analog In A4 – 1A	H-bridge	Analog

Table 16 - Component I/O list: H-bridge

#5 Component I/O list: **H-bridge** - L293NE			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	1Y Motor terminal 1	Line actuator	
Input	2Y Motor terminal 2	Line actuator	
Input	1A Motor Logic pin 1	Arduino Uno	
Input	2A Motor Logic pin 2	Arduino Uno	
Input	Ground	Ground to disable motor	
Output	+5V	IC Power	Power to enable motor
Output	12V	Motor Power supply	Power to enable motor

Remote

Table 17 - Component I/O list: RC receiver

#21 Component I/O list: **RC receiver** - Spektrum AR610			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Output	Elevation - Uno Mega	PWM	
Output	Rudder - Uno Mega	PWM	
Output	Throttle - Uno Mega	PWM	
Output	Aileron - Uno Mega	PWM	
Input	5V – Uno Mega		
Input	GND		

Table 18 - Component I/O list: Motor controller

#1 Component I/O list: **Motor controller** - Robbe NavyControl535R			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	Battery direct	6 V … 12 V lead-acid	
Output	Motor direct	Motor current: 35 A Pulse frequency: 1kHz	Forward / stop / reverse

Table 19 - Component I/O list: DC-motor

#2 Component I/O list: **DC-motor** - Robbe Roxxy Starmax 48			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	Out A	ESC	Power
Input	Out B	ESC	Power

Table 20 - Component I/O list: Servo

#4 Component I/O list: **Servo** - HiTEC HS-5485HB			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	Power for Motor 3 Pole Ferrite	Operating Voltage: 4.8-6.0 Volts	

Table 21 - Component I/O list: Linear actuator

#3 Component I/O list: **Linear actuator** - Firgelli L16			
Type	**Name**	**Signal to**	**Protocol / Signal type**
Input	Power	0-15 VDC. Rated at 12VDC. Stall Current 650mA @ 12V	

Table 22 - Component I/O list: Motor controller

#6 Component I/O list: **Motor controller** - Robbe Roxxy Control 900			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	Out A	ESC	Power
Input	Out B	ESC	Power

Table 23 - Component I/O list: AC-motor

#7 Component I/O list: **AC-motor** - Robbe Roxxy BL-outrunner 5055-45			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	Out A	ESC	Power
Input	Out B	ESC	Power
Input	Out C	ESC	Power

Table 24 - Component I/O list: Stepper motor

#8 Component I/O list: **Stepper motor** - Nanotec PD2-N41			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	Out A	ESC	Power
Input	Out B	ESC	Power
Input	Out C	ESC	Power

Sensors

Table 25 - Component I/O list: Current meas. Sensor

#9 Component I/O list: **Current meas. Sensor** - Phidgets 1122_0			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Output	SIG	Arduino Mega	Analog

Input	GND	Arduino Mega /Ground	
Input	12V	ESC	Power

Table 26 - Component I/O list: Inductive sensor

#10 Component I/O list: **Inductive sensor** - XS618B1PAL2			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Output	SIG	Arduino Mega	Analog
Input	GND	Arduino Mega /Ground	
Input	12V	ESC	Power

Table 27 - Component I/O list: Water sensor

#12 Component I/O list: **Water sensor** - Homemade			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	SIG	Arduino Uno	Analog
Input	Ground		

Power

Table 28 - Component I/O list: Battery

#18 Component I/O list: **Battery** - Exide 12V 40Ah			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Output	+ -	K1, K2, K1	Power output
Input	Ground		

Table 29 - Component I/O list: Relay

#19 Component I/O list: **Relay**			
Type	**Name**	**Signal to/ from**	**Protocol / Signal type**
Input	K1	Bow	Power switch
Input	K2	Port	Power switch
Input	K3	Starboard	Power switch

5.1.6 Classification of autonomous capabilities

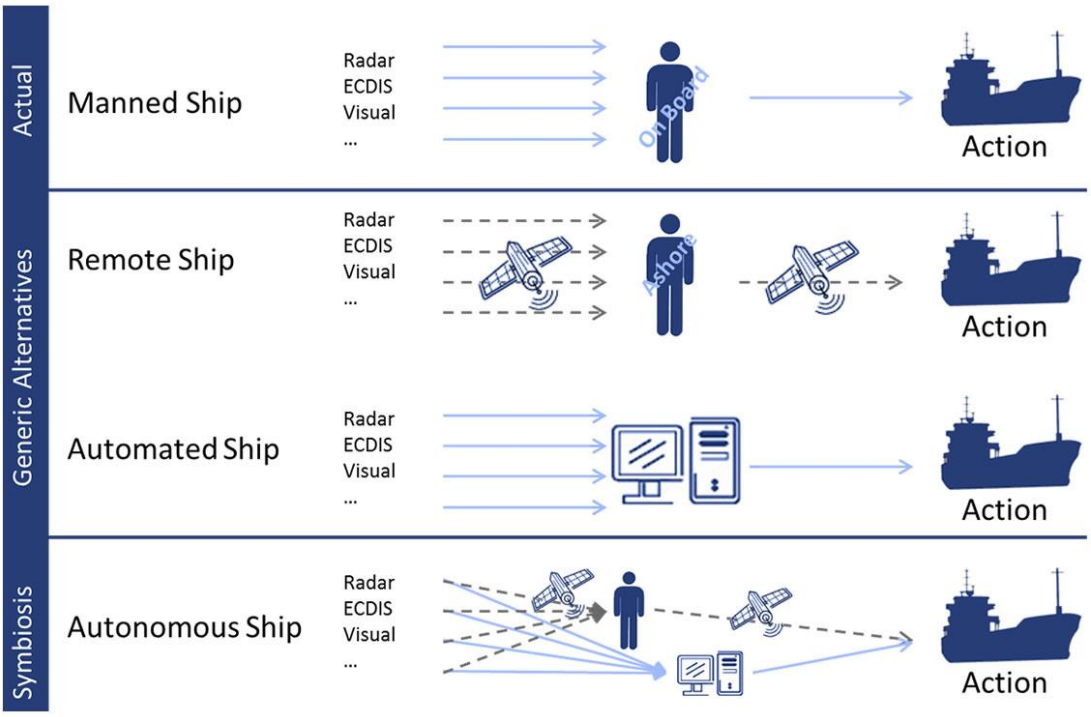


Figure 43 - Differences between manned, remote, automated and autonomous ships

In the picture above (Figure 43) the different levels of autonomous capabilities are illustrated [13].

So, in which level of autonomous is the revolt according to the picture?

The Revolt with the present design and sensor fitting is not autonomous, but a remotely operated dynamically positioned ship, missing sensors/functions for tracking other objects. This is the work to be done in the future. However, as the Revolt model would be further developed, additional sensors, such as a LIDAR, camera and radar might be implemented. This would naturally increase the risk of a sensor attack. For example, a LIDAR sensor has been proven to have vulnerabilities with autonomous cars.

I would conclude that the Revolt Model is currently, based on the figure above, a remote ship, enabled by dynamic positioning. But in the future, it will be an autonomous ship.

In the Lloyd's Register guidance document, a procedure for autonomous ships describes 7 autonomy levels (AL) [18]. According to this standard, described further in chapter 3. The Revolt will have autonomy level 4 - Human on the loop – operator/supervisory.

Chapter 6 : Results of Research Question 1

Input for analysis methods

What is the set rule for each analysis is that the process in which the analysis method provides, will only be performed once. With the aim of creating a most accurate comparison of the methods as possible.

Table 30 - Comparison of what input the different methods will have for the analysis

Methods	FMVEA	STPA-sec	CHASSIS
Starting point variable	Component	Control actions	Use case
Variable Input	Component 1: **Embedded computer** on the Revolt	**CA1**: Control the Position of the vessel **CA2**: Control the Speed of the vessel **CA3**: Control the Course of the vessel **CA4**: Control the Access to the vessels system	Use case 1: **Operating and monitoring** the Revolt remotely by Operating central at sea and docking of cargo by DNGVL operating crew and the Revolt intelligent system.

Hazard classification

Table 31 - Hazard classification for determine if a hazard is safety or security related

Hazard type	Safety related hazard	Security related hazard
Risk type	Accidental risk	Malicious risk
Potential hazardous situations	Contact damage ❖ Being too close to an object ❖ Heading towards an object with too high speed, coming too close too soon	Theft of data ❖ Someone getting access to the data
	System error ❖ System behaving incorrectly and having issues/errors	System accessibility ❖ Someone prevents the operator from accessing the system

6.1 Results FMVEA Analysis

Mission statement: Environmentally friendly short-sea shipping.

System level analysis: functional tree, the functions based on the mission statement of the Revolt System (Figure 44).

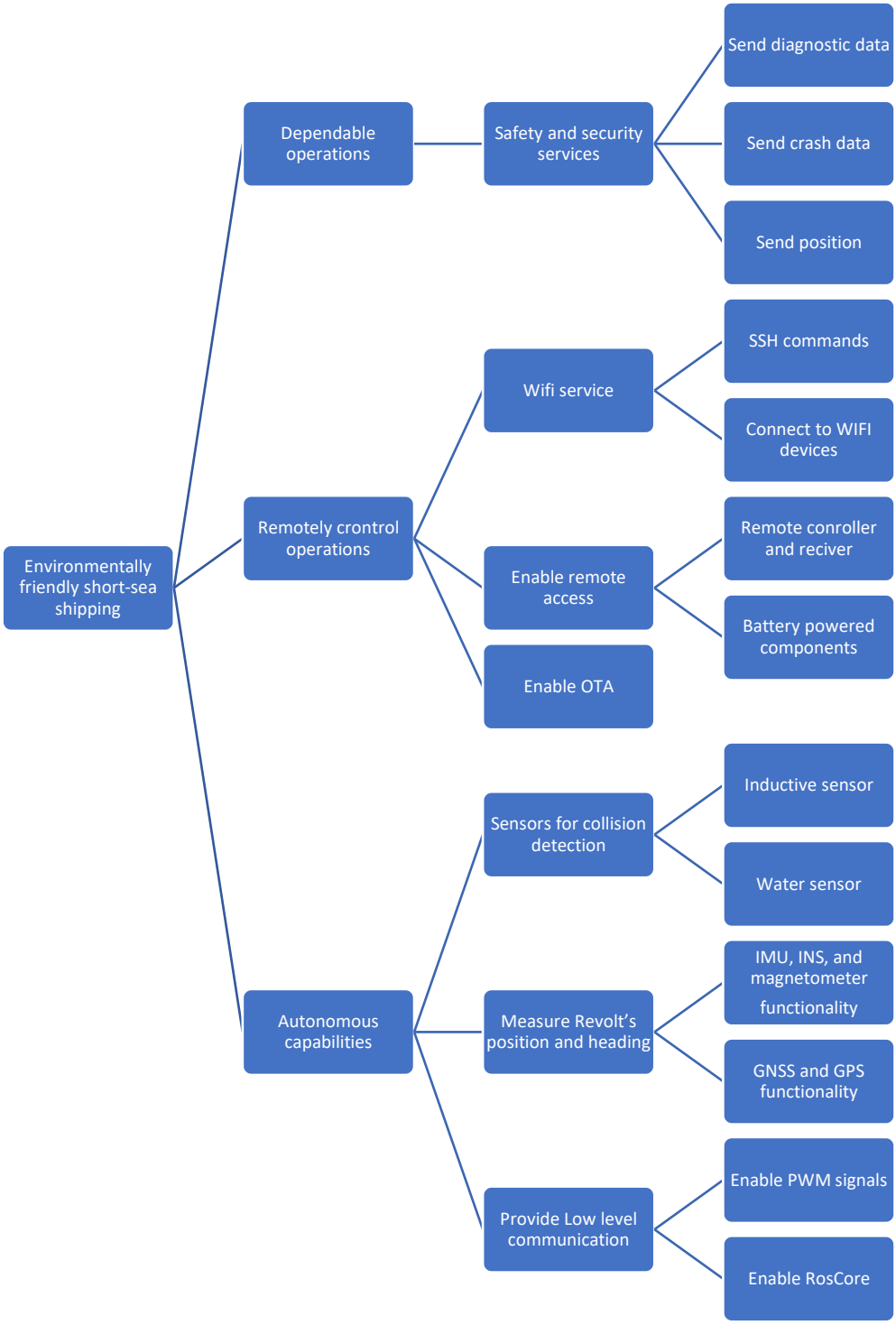


Figure 44 - System level analysis FMVEA method

6.1.1 Functions of embedded computer (Tank 720)

The FMVEA analysis will focus on the embedded computer. The reasoning for picking this component, is because the attack surface is the highest on the embedded computer, of all the components on the Revolt. The embedded computer is connected to every other component in some way.

Microcontrollers are connected to the embedded computer via USB, and analog components (water sensor etc.) are connected to the microcontrollers. Microcontrollers are controlled by the embedded computer.

The embedded computer has access to both low level device communication, trough the USB bus to the two microcontrollers. Also, the embedded computer has access to stepper motors and GPS/GNSS with Hemisphere through RS232 connectors.

Therefore, by gaining control over the embedded computer. The attacker has full control over the system on the Revolt and can access all components. All the functions on the embedded computer is listed in Table 32.

Table 32 - Functions of embedded computer (Tank 720)

Safety and Security Services	Operating microcontrollers USB bus	WIFI service	Nodes registration service - RosCore	Diagnostics services	Operating GNSS receiver and steppers RS232 bus
Send crash data	Provide Low level communication with - Pulse Width Modulation (PWM) signals	Execute commands from outside laptop via WIFI	Keeps track of all nodes and messages published and subscribed to	Send and receive diagnostic data	Measure Revolt's position and heading
Send position		Commands via SSH	Communication between nodes Messages (topics) with TCP (TCPROS)	Receive over the air(OTA) firmware updates	
		Connect to WIFI devices			

6.1.2 Failure Mode, Vulnerabilities and Effect Analysis of embedded computer (Tank 720)

The complete FMVEA analysis is performed on the functions of the embedded computer(listed in Table 32) in the following table (Table 33).

Table 33 - FMVEA analysis

ID	component / element	Vulnerability/ Failure Cause	Threat Mode/ Failure Mode	Threat Effect/ Failure Effect	System Status	System Effect	Severity	System Susceptibility	Threat Properties	Attack/Failure Probability	Risk
1	WIFI connection	Wireless connection is targeted to jamming	Attacker interrupts connection between operator and Revolt	Revolt is unreachable	Remote operation	Attacker has control over the Revolt's system	Critical:4	4	5	9	36
2	WIFI connection	No device verification, man in the middle attack with access to RosCore	Attacker is pretending to be 4G router, spoofing	Revolt system sends and receives false data	Normal operation	System is no longer reliable	Critical:4	4	4	8	32
3	WIFI connection	Cracking WPA2-PSK with dictionary attack	Attacker receives access to the WIFI network	The Revolts embedded computer is compromised	Normal operation	System integrity is hurt	Critical:4	4	6	10	40
4	Xsens sensor: GNSS/GPS	GPS spoofing attack	Spoofing a satellite's signal with a false signal sent from a ground station	Spoofing causes the receiver to lie, the attacker can manipulate the Revolts location	Normal operation	Attacker has control over the Revolt's location	Catastrophic :5	5	6	11	55
5	OTA	Connection is lost	Data missing for update	Update is corrupt because of interruption	Updating	None	Negligible:1	..	..	5	6

6	OTA	Update causes faults	Components don't work as intended	System could have critical faults	Updating	System is no longer reliable	Moderate:3	..	..	4	12
7	Transmit diagnostic data	Man in the middle attack on GSM base station	The attacker is manipulating diagnostic data	Wrong data sent	System receiving wrong diagnostic	Reduced functionality of system	Marginal:2	3	4	7	14
8	Wrong sensor input data	Faulty sensors	The sensors are giving wrong input data to the computer, causing it to make wrong decisions	The sensors for making correct navigation decisions cannot function as normal	Normal operation	System is no longer reliable	Critical:4	..	..	3	12
9	Wrong GNSS/GPS input data	Faulty GNNS or GPS	The GPS/GNNS are giving wrong input data to the computer, causing it to make wrong decisions	The navigation system cannot function as normal	Normal operation	System is no longer reliable	Critical:4	..	..	2	8
10	System error - causes execution of command delays or system failure	System error/ services unavailable	System services have stopped working	The embedded computer cannot function as normal and the Revolt cannot function as normal	Normal operation	System is no longer reliable	Catastrophic :5	..	..	3	15

Explanations for each function and how the quantitative analysis is performed, is listed in the pages below.

Explanations – how to perform Quantitative analysis

Classification of the following terms are collected from the IEC 61812 standard [66].

Severity

Severity level is determined by "Significance or grading of the failure mode's effect on item operation, on the item surrounding, or on the item operator; failure mode effect severity as related to the defined boundaries of the analyzed system" [66].

Table 34 - Classification of Risk Severity level related to components

Severity level	Severity type	Description
1	Negligible	Not resulting in any harm on the object
2	Marginal	The result might cause inconveniences or minor harm to the object
3	Moderate	The results might cause moderate harm or problems to the object
4	Critical	The result might cause serious harm to the object and environment
5	Catastrophic	The result might cause catastrophic consequences to the object, environment and nearby objects and possible permanent harm

System susceptibility

The sum of the Reachability (Table 35) and Unusualness (Table 36) of the related component, properties characterizes the system susceptibility.

Table 35 - Classification of System susceptibility related to components

Reachability level	Network type
1	No network
2	Private network
3	Public network

Table 36 - Classification of Unusualness level related to components

Unusualness level	Unusual type
1	Restricted
2	Commercially available
3	Standard

Threat properties

The threat properties are determined by both motivation and capabilities and the sum of both is the threat property.

Table 37 - Classification of Motivation level related to components

Motivation level	Motivation type
1	Opportunity target
2	Mildly interested
3	Main target

Table 38 - Classification of Capabilities level related to components

Capabilities level	Capabilities type
1	Low
2	Medium
3	High

Attack probability

Table 39 - Classification of Attack probability related to components

System Susceptibility						
6	8	9	10	11	12	
5	7	8	9	10	11	
4	6	7	8	9	10	
3	5	6	7	8	9	
2	4	5	6	7	8	
	2	3	4	5	6	**Threat properties**

Failure probability

Table 40 - Classification of Failure probability related to components

Rating	Description	Definition
5	Very high probability: failure is most inevitable	1 failure in 5 attempts
4	High: repeated failures	1 failure in 50 attempts
3	Moderate: occasional failures	1 failure in 500 attempts
2	Low: relatively few failures	1 failure in 5000 attempts
1	Remote: failure is unlikely	<1 failures in 500,000 attempts

Rating scales can help to standardize the team members' responses. Below is the probability rating scale adapted from the Healthcare Failure Mode and Effects Analysis (HFMEA) model developed by the National Center for Patient Safety of the Veterans Health Administration [67].

Risk

Severity * Attack/Failure Probability = Risk

Component element explanations

The following will explain how each property are calculated for each component in the FMVEA analysis.

#1: WIFI connection - Wireless connection is targeted to jamming
Severity – Critical (4): With this type of attack, the WIFI connection on the Revolt will be unbailable, and can't be reached in that form. The only way to control the Revolt is by a Remote control, if enabled.

System Susceptibility (4): The WIFI connection that the 4G router provides is commercially available (2) and the network used is a private network (2).

Threat Properties (5): The attacker is to be considered to have a high degree of motivation, most definite is this vessel the main target (3). However, this is not the most advance type of attack, so the capabilities of the attacker are considered to be at a medium level (2).

Attack Probability (9).

Risk: 4*9=36

#2: WIFI connection - No device verification, man in the middle attack with access to RosCore
Severity – Critical (4): In this situation, the system is not reliable anymore and the attacker can manipulate the system.

System Susceptibility (4): The WIFI connection that the 4G router provides is commercially available (2) and the network used is a private network (2).

Threat Properties(4): The attacker is to be considered to be mildly interested, since this attack is not that resource demanding. This attack requires a medium level of capabilities to execute.

Attack Probability (8).

Risk: 4*8=32

#3: WIFI connection - Cracking WPA2-PSK with dictionary attack
Severity – Critical (4): With this type of attack, the attacker receives access to the WIFI network and from there, the attacker could attempt to take control of other components, that are connected to the WIFI.

System Susceptibility (4): The WIFI connection that the 4G router provides is commercially available (2) and the network used is a private network (2).

Threat Properties (6): The attacker is to be considered to have a high degree of motivation, most definite is this vessel the main target (3), and this type of attack requires a high level of technical insight (3).

Attack Probability (10).

Risk: 4*10=40

#4: Xsens sensor: GNSS/GPS - GPS spoofing attack

Severity – Catastrophic (5): This type of attack is of the highest severity. If the attacker has control over the Revolt's location, meaning controlling its position. The risk of life increases when untrained individuals are controlling the vessel. Safety protocols are not followed.

System Susceptibility (5): The Xsens sensor is commercially available (2), the GNSS or GPS connection is publicly accessible (3)

Threat Properties: Threat Properties (6): The attacker is to be considered to have a high degree of motivation, most definite is this vessel the main target (3), and this type of attack requires a high level of technical insight (3).

Attack Probability (11).

Risk: 5*11=55

#5: OTA - Connection is lost

Severity – Negligible (1): In this situation, the update fails, because of interruption. This has however no effect on the system.

System Susceptibility & Threat Properties: For this case, there is no malicious attack. But a failure mode.

Failure Probability (5).

Risk: 1*5=5

#6: OTA - Update causes faults

Severity – Moderate (3): When an update causes faults, the components in the Revolt might not work as intended and causes faults.

System Susceptibility & Threat Properties: For this case, there is no malicious attack. But a failure mode.

Failure Probability (4).

Risk: 3*4=12

#7: Transmit diagnostic data - Man in the middle attack on GSM base station

Severity – Marginal (2): With this Man in the middle attack, the attacker only succeeds in potentially reducing the functionality of system.

System Susceptibility (5): The GSM connection is considered to be not that common for non-commercial applications, but commercially available (2). The wireless GSM connection is publicly accessible (3).

Threat Properties: Threat Properties (4): The attacker is to be considered to be mildly interested, since this attack is not that resource demanding. This attack requires a medium level of capabilities to execute.

Attack Probability (7).

Risk: 2*7=14

#8: Wrong sensor input data

Severity - Critical (4): When a wrong input data from a sensor occurs. This might lead to faulty maneuvering, and the revolt might crash into objects.

System Susceptibility & Threat Properties: For this case, there is no malicious attack. But a failure mode.

Failure Probability (3).

Risk: 4*3=12

#9: Wrong GNSS/GPS input data

Severity - Critical (4): When a wrong input data from the GPS occurs. This might lead to faulty maneuvering, and the revolt might crash into objects.

System Susceptibility & Threat Properties: For this case, there is no malicious attack. But a failure mode.

Failure Probability (2).

Risk: 4*2=8

#10: System error - causes execution of command delays or system failure

Severity – Catastrophic (5): This failure mode has the highest severity. When system errors occur on this component, it might not work as intended and causes faults and can eventually lead to system failure and the controller has lost the control of the vessel.

System Susceptibility & Threat Properties: For this case, there is no malicious attack. But a failure mode.

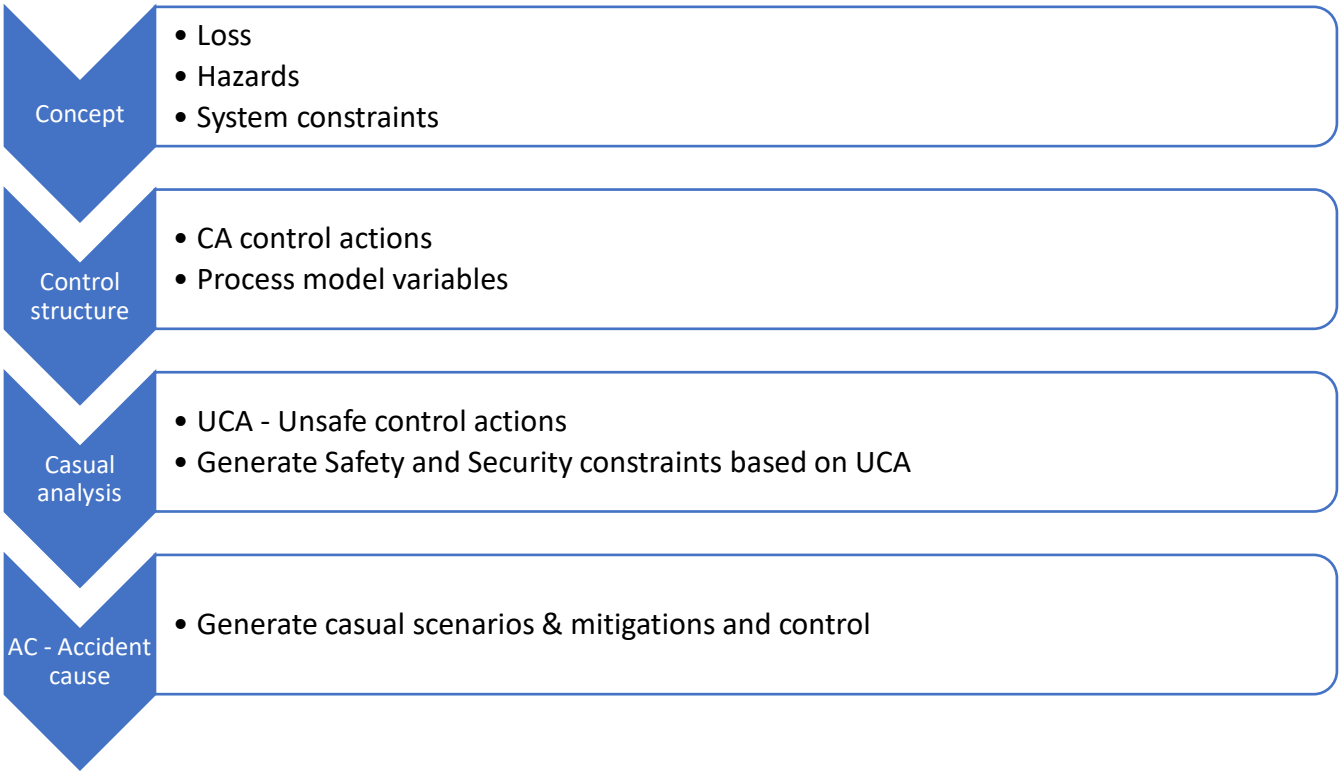
Failure Probability (3).

Risk: 5*3=15

6.2 Results of STPA and STPA-SEC analysis

I will now perform a combined STPA and STPA-SEC analysis on the Revolt case. This analysis will focus on generation casual scenarios of the system and from there design recommendations, in which will mitigate certain issues of the system.

This analysis will be divided into the following activity format:



6.2.1 Concept - Define & Frame Problem

This is the part of the analysis where the concept of the system is described.

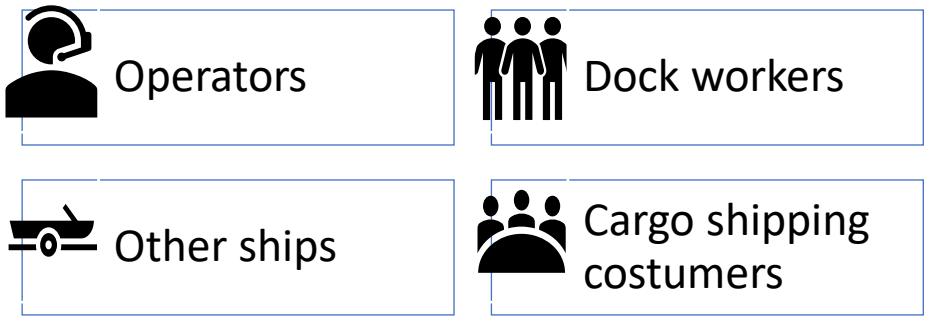
Scenario

Assure a safe and secure transport of cargo from point A to point B, doing so in an unmanned vessel. This will include protection from cyber-attacks, in which could be derived from industry espionage, terrorism or theft. Also, this requires a high level of functional safety from the system and vessel, both the software and hardware.

Mission

Environmentally friendly short-sea shipping.

Key stakeholders

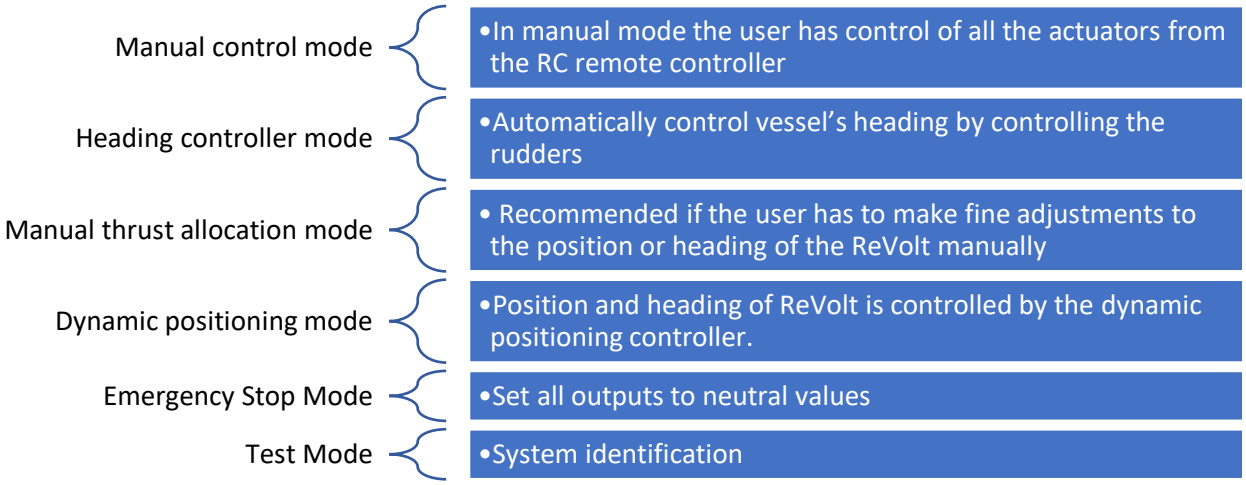


System purpose and goals

The Revolts' system is based on a Robot Operating System (ROS), in which gives the system a possibility to be very modular, and various hardware can be used. The system is equipped with necessary sensors to adapt to the weather conditions at sea. The purpose of this system is to make the Revolt ship function as optimal as possible during operation and maintenance, and at the same time being safe – collision avoidance and have protection against attacks. The goal is to be an environmentally friendly option for short-sea shipping

Operating state variables

Table 41 - Operating state variables for Revolt operation



The Revolt vessel has six different operating states that it could be in, as shown above. Accidents could occur during all these operating states.

6.2.1.1 Losses/accidents and Hazards

Unacceptable losses/accidents

Table 42 - List of Unacceptable losses/accidents for the Revolt vessel

ID	**Unacceptable losses/accidents**
L1	Collision with vessels, objects, humans/mammals, structures, grounding
L2	Fire or explosion
L3	Foundering (sinking, failing or plunging)
L4	Loss of cargo
L5	Loss of mission objectives
L6	Loss of information

System hazards and constraints

Table 43 - List of System hazards and constraints for the Revolt vessel

Accidents	System hazards	System constraints	Operating state variables
Contact damage	H1: Being too close to an object	SC1: The operation crew of vessel must always have control over the ship SC2: The ship should never sail off route SC3: The operation crew of the vessel must follow safety protocols	Manual control mode Manual thrust allocation mode
	H2: Heading towards an object with too high speed, coming too close too soon	SC4: The operation crew of the vessel must never violate minimum distance the ship shall have to other ships, docks or other objects	Dynamic positioning mode Heading controller mode
Theft of data	H3: Someone unauthorized getting access to the vessels data	SC5: No access to the ships hardware or software (remotely or physical) without permitted authorization SC6: The operation crew of the vessel must follow security protocols	Test Mode Emergency Stop Mode

Table 44 - Connection between what losses are possible with certain hazards

	L1: Collision with vessels, objects, humans/mammals, structures, grounding	L2: Fire or explosion	L3: Foundering (sinking, failing or plunging)	L4: Loss of cargo	L5: Loss of mission objectives	L6: Loss of information
H1: Being too close to an object	X	X	X	X		
H2: Heading towards an object with too high speed, coming too close too soon	X	X	X	X	X	
H3: Someone unauthorized getting access to the vessels data						X

6.2.2 Choosing control actions and connected process model variables and values

To choose the right control actions for this vessel. The main functions must be the basis.

Main functions for Revolt: guidance function (to decide and generate a path to follow) and a propulsion function (to command the thrusters/rudder).

These functions must only be performed by authorized persons. Controlling the access to the system on the vessel is therefore also a main function.

Control actions derived from main functions:

Control actions are the position, speed and course command executed by the boat. Control the access to the vessels system.

Table 45 - Control actions derived from main functions

ID	Control actions	Process model variables	Values
CA1	Control the Position of the vessel	Position	* Aligned with plan * Not aligned with plan * Unknown
CA2	Control the Speed of the vessel	Speed	* Speed up * Slow down * Unknown
CA3	Control the Course of the vessel	Course	* Safe Course for situation * Unsafe Course for situation * Unknown
CA4	Control the Access to the vessels system	Access	* Access control enforced * Inappropriate access enforcement * Unknown

6.2.2.1 Create functional control structure

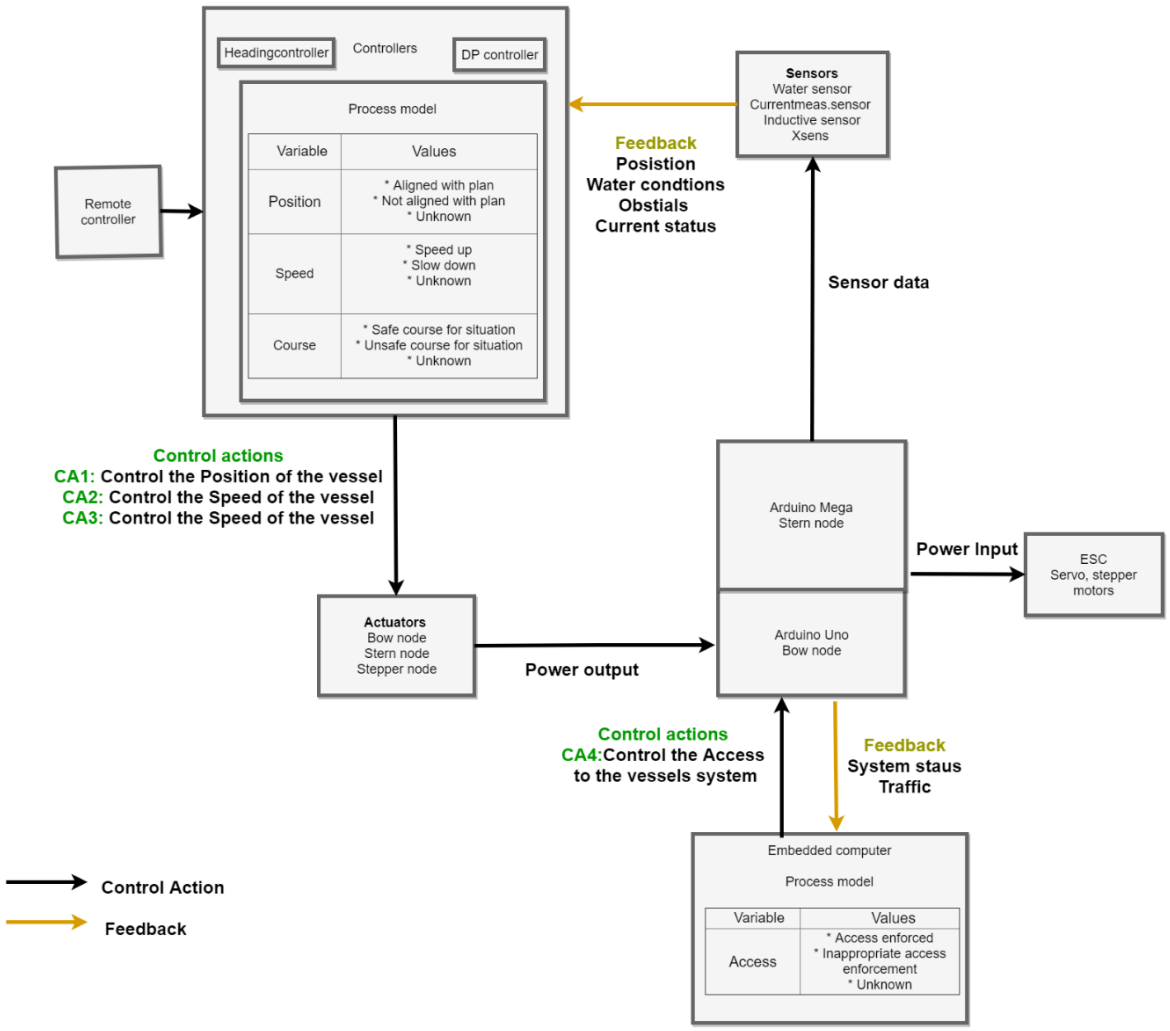


Figure 45 - Functional control structure of the Revolt

In Figure 45, there is a functional control structure of the Revolt system, in which is contained inside the vessel.

6.2.3 Identify Hazardous Control Actions

Control actions dependencies

It is important to note that the set of values defined for each variable does not necessarily need to be detailed, but they must be complete so that every possibility is included. For example, the speed “slow down” and “speed up” is complete because the set includes every possibility. The speed can essentially only be increased or decreased.

However, CA3: control the course is dependent on the speed. For example, there could be an unsafe course correction if the speed is set to “speed up” at the same time as a heavy course correction is being done (the vessel might tip over if the speed is too high). Therefore CA3: control the course is dependent on CA2: control the speed.

Also, if the speed is dependent on the course. For example, if the course is set to being close to another ship. There should be a lower speed if another vessel is close.

This is their mutual unsafe dependency.

Therefore, some control actions are mutually dependent and should be issued in pairs. A possible solution for analyse mutually dependent control actions is to add the control action as a process control variable of another control action, if another control action has dependency with it.

New entity in UCA table: CA Dependency

If there is a condition where a control action need to have information about another control action (variable sate of the CA) before making a decision and sending the input to the controller. The control actions are dependent on each other.

Otherwise they are not dependent.

This is a new approach that has been developed during this case study and the results can be seen in the modified UCA tables further down in this analysis.

In the UCA tables, I have listed which other CA is dependent, the CA itself is not listed in the UCA table, it is a self-granted that the CA is dependent on itself.

UCA for CA1

Table 46 - Unsafe control table for control action 1

Controller				Heading controller		H1	H1: Being too close to an object		
CA1 - Control Action 1				Control the Position of the vessel		H2	H2: Heading towards an object with too high speed, coming too close too soon		
						H3	H3: Someone unauthorized getting access to the vessels data		
	Process Model Variables			**CA Dependency**		**Control Actions (CA) hazardous?**			
	Position (Aligned with plan)	Speed (Slow down)	Course (Safe for situation)	Dependable CA	NOT Dependable CA	CA NOT provided	CA provided	CA provided too late/early	CA stopped too late/early
1	Yes	Yes	Yes	CA2, CA3	CA4	H1		Too early (H1, H2)	
2	Yes	Yes	No	CA2	CA3, CA4	H1		Too early (H1, H2)	Too early (H1,H2)
3	Yes	No	Yes	CA3	CA2, CA4	H1		Too early (H1, H2)	Too early (H1,H2)
4	Yes	No	No		CA2, CA3, CA4	H1		Too early (H1, H2)	Too early (H1,H2)
5	No	Yes	Yes	CA2, CA3	CA4	H1	H1, H2	Too early (H1, H2)	Too early (H1,H2)
6	No	Yes	No	CA2	CA3, CA4	H1	H1, H2	Too early (H1, H2)	Too early (H1,H2)
7	No	No	Yes	CA3	CA2, CA4	H1	H1, H2	Too early (H1, H2)	Too early (H1,H2)
8	No	No	No		CA2, CA3, CA4	H1	H1, H2	Too early (H1, H2)	Too early (H1,H2)

UCA for CA2

Table 47 - Unsafe control table for control action 2

Controller				Heading controller		H1	H1: Being too close to an object		
CA2 - Control Action 2				Control the Speed of the vessel		H2	H2: Heading towards an object with too high speed, coming too close too soon		
						H3	H3: Someone unauthorized getting access to the vessels data		
	Process Model Variables			**CA Dependency**		**Control Actions (CA) hazardous?**			
	Position (Aligned with plan)	Speed (Slow down)	Course (Safe for situation)	Dependable CA	NOT Dependable CA	CA NOT provided	CA provided	CA provided too late/early	CA stopped too late/early
1	Yes	Yes	Yes	CA1, CA3	CA4			Too early (H1, H2)	Too soon(H1, H2)
2	Yes	Yes	No	CA1	CA3, CA4			Too early (H1, H2)	Too soon(H1, H2)
3	Yes	No	Yes	CA1	CA3, CA4		H2	Too early (H1, H2)	Too soon(H1, H2)
4	Yes	No	No	CA1	CA3, CA4		H2	Too early (H1, H2)	Too soon(H1, H2)
5	No	Yes	Yes	CA3	CA1, CA4	H2		Too early (H1, H2)	Too soon(H1, H2)
6	No	Yes	No		CA1, CA3, CA4	H2		Too early (H1, H2)	Too soon(H1, H2)
7	No	No	Yes	CA3	CA1, CA4	H2	H2	Too early (H1, H2)	Too soon(H1, H2)
8	No	No	No		CA1, CA3, CA4	H2	H2	Too early (H1, H2)	Too soon(H1, H2)

UCA for CA3

Table 48 - Unsafe control table for control action 3

Controller				Heading controller		H1	H1: Being too close to an object		
CA3 - Control Action 3				Control the Course of the vessel		H2	H2: Heading towards an object with too high speed, coming too close too soon		
						H3	H3: Someone unauthorized getting access to the vessels data		
	Process Model Variables			**CA Dependency**		**Control Actions (CA) hazardous?**			
	Position (Aligned with plan)	Speed (Slow down)	Course (Safe for situation)	Dependable CA	NOT Dependable CA	CA NOT provided	CA provided	CA provided too late/early	CA stopped too late/early
1	Yes	Yes	Yes	CA1, CA2	CA4			Too early (H1, H2)	
2	Yes	Yes	No	CA1, CA2	CA4			Too early (H1, H2)	
3	Yes	No	Yes	CA1	CA2, CA4		H2	Too early (H1, H2)	
4	Yes	No	No	CA1	CA2, CA4		H2	Too early (H1, H2)	
5	No	Yes	Yes	CA2	CA1, CA4	H2		Too early (H1, H2)	Too soon(H1)
6	No	Yes	No	CA2	CA1, CA4	H2		Too early (H1, H2)	Too soon(H1)
7	No	No	Yes		CA1, CA2, CA4	H2	H2	Too early (H1, H2)	Too soon(H1)
8	No	No	No		CA1, CA2, CA4	H2	H2	Too early (H1, H2)	Too soon(H1)

UCA for CA4

Table 49 - Unsafe control table for control action 4

Controller		Embedded computer		H1	H1: Being too close to an object		
CA4 – Control action 4		Control the Access to the vessels system		H2	H2: Heading towards an object with too high speed, coming too close too soon		
				H3	H3: Someone unauthorized getting access to the vessels data		
ID	**Process Model Variables**	**CA Dependency**		**Control Actions (CA) hazardous?**			
	Position (Access enforced)	Dependable CA	NOT Dependable CA	CA NOT provided	CA provided	CA provided too late/early	CA stopped too late/early
1	Yes	CA1	CA2, CA3	H2,H3	H3	Too early (H2, H3)	Too early (H3)
2	No		CA1, CA2, CA3	H3	H2,H3		

6.2.3.1 UCA summary

Table 50 - Unsafe control actions summary

Control actions	**Hazardous control actions**			
	Not providing CA	Providing CA	Providing CA too soon or too long	Providing CA in the wrong sequence or order (too early/late)
CA1: Control the Position of the vessel	Not providing CA1 when an emergency situation is occurring in the area of the vessel (e.g. oil leak, storms or other) [H1]	Providing CA1 when the revolt is in the middle of a hazardous situation and the system has already set commands for collision avoidance [H1] [H2]	Too soon: Providing CA1 when sensors/components are not operating correctly at the time [H1] [H2]	Too early: Providing CA1 when system components are being updated[H1]
CA2: Control the Speed of the vessel	Not providing CA2 when the speed is unsafe for current situation and the system and this is not detected by the system [H2]	Providing CA2 when there is vessel is on route and there is a possibility of losing the connection and the speed is set too high for upcoming situations [H2]	Too soon: Providing CA2 before the system reports to be functioning correctly[H1] [H2]	Too early: Providing CA2 when sensors are not calibrated yet[H1] [H2]
CA3: Control the Course of the vessel	Not providing CA3 when the course is unsafe for current situation and the system and this is not detected by the system [H2]	Providing CA3 when the speed is too high for a heavy correction of the course [H2]	Too soon: Providing CA3 manually before having taken the current situation into consideration [H1] [H2]	Too soon: Providing CA3 when the shipping dock has not permitted the action (other ships are dispatching at the same time) [H1]
CA4: Control the Access to the vessels system	Not providing CA4 when ships WIFI connection is not encrypted [H3] Not providing CA4 when updates are necessary to mitigate a security issue [H3] Not providing CA4 when a spoofing or jamming attack is occurring [H2] [H3]	Providing CA4 when revolt ship protocols for dispatch has not been followed [H2] [H3]	Too early: Providing CA4 before revolt has been authorized [H3] Too early: Providing CA4 when Revolt are on a sailing mission, operating and can't be interfered with [H2] [H3]	Too early: Providing CA4 when system components are being updated and before having done firmware testing to see the results of the update [H3]
Hazards	H1: Being too close to an object H2: Heading towards an object with too high speed, coming too close too soon H3: Someone unauthorized getting access to the vessels data			

6.2.4 Generate Safety and Security constraints

The following process is used for generating safety and security constraints based on unsafe/unsecure control actions:

Unsafe control actions → summarized and translated → Safety or Security constraints

This process has been performed in Table 51.

Table 51 -Generate Safety and Security constraints

Unsafe/Unsecure Control Actions	Safety or Security constraints	Safety or Security related?
[UCA1] Not providing CA when an emergency situation is occurring in the area of the vessel (e.g. oil leak, storms or other) [H1]	[SC1] Operator must enable DP mode when available, this mode should be used for best stabilization at sea	Safety
[UCA2] Providing CA when the revolt is in the middle of a hazardous situation and the system has already set commands for collision avoidance [H1] [H2]	[SC2] All operator members must have safety and security training before operating the revolt	Safety
[UCA3] Too soon: Providing CA when sensors/components are not operating correctly at the time [H1] [H2]	[SC3] The system Revolt system will have backup components for communication of emergency	Safety
[UCA4] Too early: Providing CA when system components are being updated[H1]	[SC4] Systems updates must only be performed when the revolt is dispatched, not operational	Safety
[UCA5] Not providing CA when the speed is unsafe for current situation and the system and this is not detected by the system [H2]	[SC5] Operator must enable DP mode when available, this mode should be used for best stabilization at sea. However, the operator must always be alert of hazardous situations and step in and override system if needed.	Safety
[UCA6] Providing CA when there is vessel is on route and there is a possibility of losing the connection and the speed is set too high for upcoming situations [H2]	[SC6] System will be implemented with fail-safe method.	Safety
[UCA7] Too soon: Providing CA before the system reports to be functioning correctly[H1] [H2]	[SC7] Alerts will go off to operating central when the Revolts system is not functioning correctly	Safety
[UCA8] Too early: Providing CA when sensors are not calibrated yet[H1] [H2]	[SC8] Safety procedures for calibrating the sensors must be followed	Safety
[UCA9] Not providing CA when the course is unsafe for current situation and the system and this is not detected by the system [H2]	[SC9] Safety procedures for use of different modes on the Revolt must be followed at all times	Safety
[UCA10] Providing CA when the speed is too high for a heavy correction of the course [H2]	[SC10] System will override if unsafe course change is being performed	Safety

[UCA11] Too soon: Providing CA manually before having taken the current situation into consideration [H1] [H2]	[SC11] Operator must enable manual mode when necessary, this mode should only be used when an emergency is occurring because of a sensor or other components is failing	Safety
[UCA12] Too soon: Providing CA when the shipping dock has not permitting the action (other ships are dispatching at the same time) [H1]	[SC12] Safety and security procedures must be followed when the revolt is near a shipping dock	Safety
[UCA13] Not providing CA when ships WIFI connection is not encrypted [H3]	[SC13] Security procedures for WIFI service must be followed	Security
[UCA14] Not providing CA when updates are necessary to mitigate a security issue [H3]	[SC14] Security updates must be prioritized on the revolts system	Security
[UCA15] Not providing CA when a spoofing or jamming attack is occurring [H2] [H3]	[SC15] The Revolts system must have components equipped for protection against jamming and spoofing attacks	Security
[UCA16] Providing CA when revolt ship protocols for dispatch has not been followed [H2] [H3]	[SC16] Safety procedures for dispatch of the revolt must be followed at all times	Safety
[UCA17] Too early: Providing CA before revolt has been authorized [H3]	[SC17] Security procedures for authorization of the revolt must be followed	Security
[UCA18] Too early: Providing CA when Revolt are on a sailing mission, operating and can't be interfered with [H2] [H3]	[SC18] Security procedures for severe access changes must be done when vessel is not operating	Security
[UCA19] Too early: Providing CA when system components are being updated and before having done firmware testing to see the results of the update [H3]	[SC19] There must always been done testing of the effects of the update in a testing environment, before being performed on the Revolt	Security
Total Safety related hazards: **13** Total Security related hazards (or threats): **6** Total Safety and Security related hazards: **19**		

6.2.4.1 Violation of Safety and Security constraints

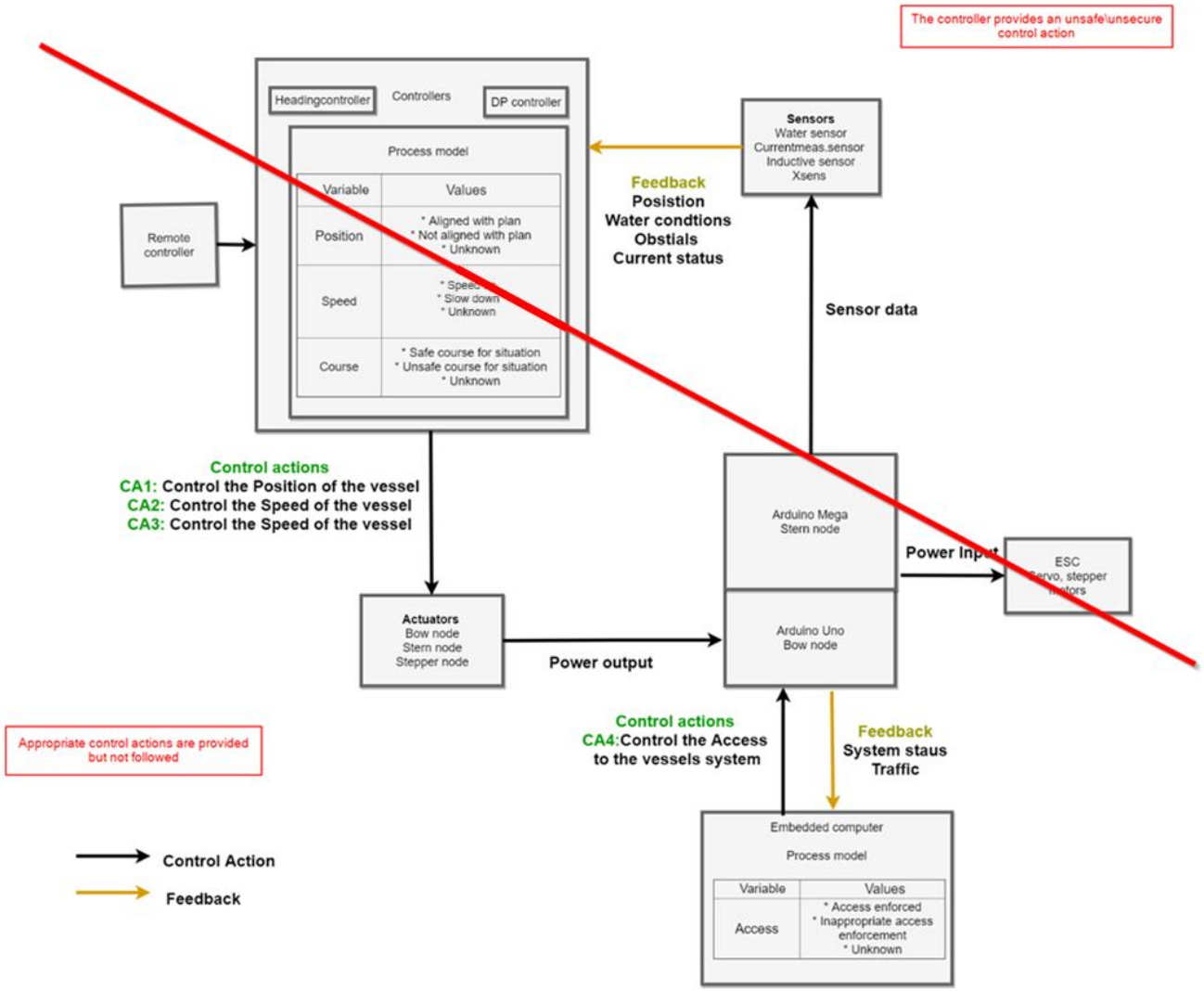


Figure 46 - Classification of how safety and security constraints could be violated

There are two ways that a safety or security constraint can be violated:

- The controller provides an unsafe\unsecure control action
- Appropriate control actions are provided but not followed

In the Figure 46, the red stripe is dividing the two ways a safety or security constraint can be violated. Top of the red line is related to hazards that produced when the controller provides an unsafe\unsecure control action. Below the red line is related to violations that occur when appropriate control actions are provided but not followed.

6.2.4.2 Causal Factors leading violation of constraints and eventually Hazards

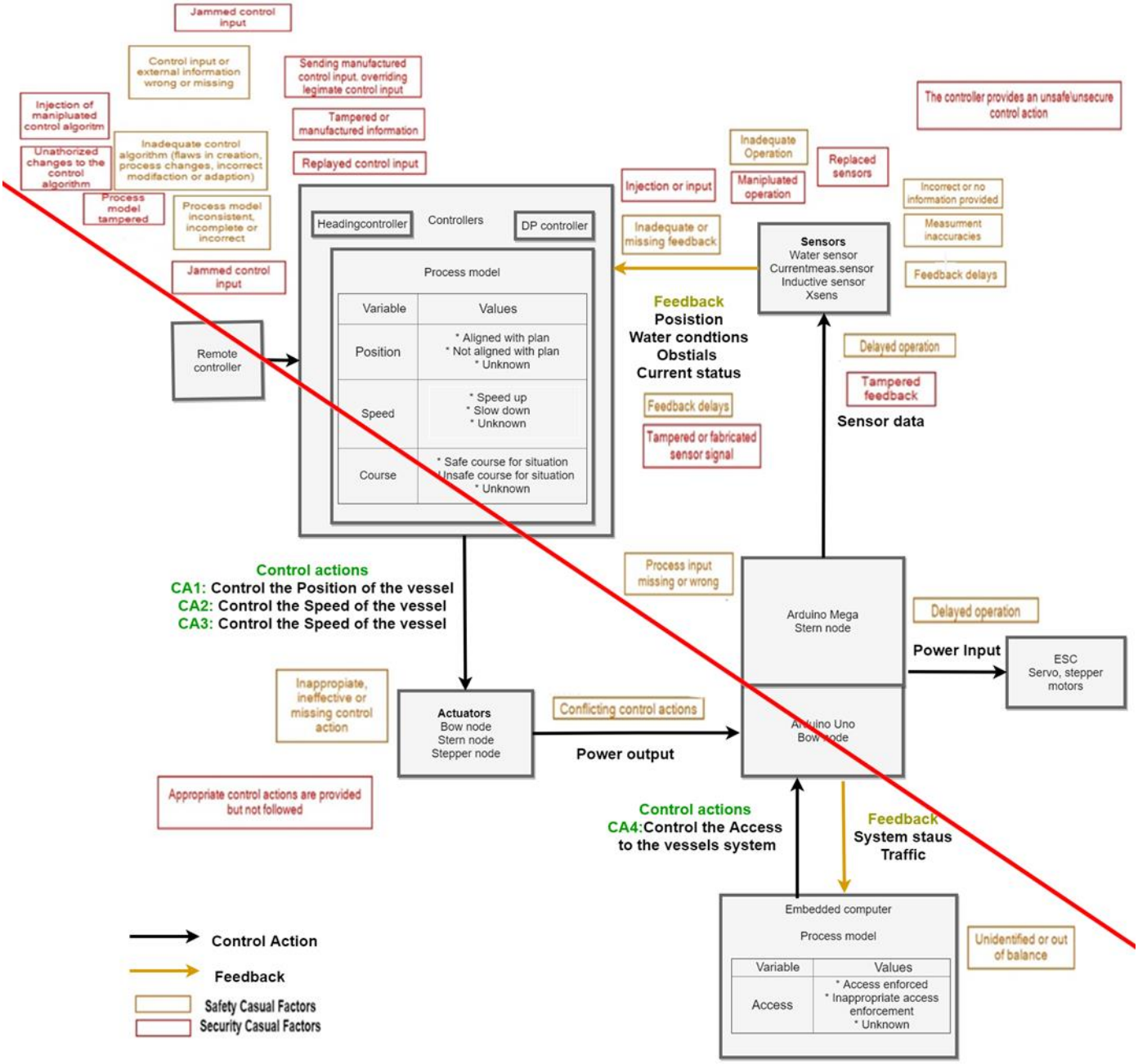


Figure 47 - Classification of causal factors leading to hazards

In Figure 47, the causal factors, both safety and security related, that could lead to hazards or threats are described and placed where they could happen in the control loop of the Revolt system.

6.2.5 AC - accident cause

In the part of the analysis the accident cause for these unsafe control actions will be attempted to be discovered. Doing so with generating casual scenarios to unsafe control actions and the connected a underlying causal factor. For this analysis to be useful for creating a better and more safe and secure vessel, design recommendations / requirements are also included.

6.2.5.1 Generate casual scenarios & Mitigations and control

Table 52- Generation of casual scenarios, causal factors and design recommendations for unsafe control action 1

[UCA1] Not providing CA when an emergency situation is occurring in the area of the vessel (e.g. oil leak, storms or other)		
Scenarios	Causal Factors	Design recommendations / requirements
Revolt can't hold its current position when at sea and when there is high wind and waves	There is no safety mechanism for enabling DP mode	There should be a mechanism for automatically enabling the DP mode, when there are very high waves
Revolt can't be controlled and crashes in nearby ship	Operator has not followed safety procedures for DP mode	There should be safety and security messages (from the safety and security procedures of the system) displayed in the operator system, before different actions are performed

Table 53 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 2

[UCA2] Providing CA when the revolt is in the middle of a hazardous situation and the system has already set commands for collision avoidance		
Scenarios	Causal Factors	Design recommendations / requirements
The Revolts crashes in shallow water	There is no functionality for forcing a specific mode on the Revolt	Force manual mode in certain situations

Table 54 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 3

[UCA3] Providing CA when sensors/components are not operating correctly at the time		
Scenarios	Causal Factors	Design recommendations / requirements
The 4G router on the Revolt fails	There is no functionality for recovering from the 4G router failing	Safety recovery from losing 4G connection must be implemented
The water sensors fail	There is no functionality for recovering from sensors failing	Safety recovery from sensor input must be implemented

Table 55 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 4

[UCA4] Providing CA when system components are being updated		
Scenarios	Causal Factors	Design recommendations / requirements
There are updates being installed on the revolt and it is causing the components to fail	Updates should never be installed at sea, Revolt operating	Block updates from being installed at sea

Table 56 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 5

[UCA5] Not providing CA when the speed is unsafe for current situation and the system and this is not detected by the system		
Scenarios	Causal Factors	Design recommendations / requirements
The Revolt ship crashes into a small island at sea	The operator has not enabled manual mode when necessary	The manual mode should only be used when an emergency is occurring because of a sensor or other components is failing. Therefore, implementation for this should be implemented.

Table 57 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 6

[UCA6] Providing CA when there is vessel is on route and there is a possibility of losing the connection and the speed is set too high for upcoming situations		
Scenarios	Causal Factors	Design recommendations / requirements
The revolt system is being updated and it loses its connection to the operating central	❖ Too weak signal on antennas ❖ Signal being block/ noise	Implement stronger antennas and/or a backup system on the revolt in case of failure. Fail-safe method.

Table 58 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 7

[UCA7] Providing CA before the system reports to be functioning correctly		
Scenarios	Causal Factors	Design recommendations / requirements
The Revolt leaving the dock with low battery and getting stranded at sea	There is no functionality for stopping the operator from starting the Revolt without having fully charged the battery	Implantation of functionality for forcing the battery to be fully charged before dispatch
The customer gets the cargo shipment too late	There is no alert system for the revolt operation system	There should be implemented an alert system, so that alerts will go off to operating central when the Revolts system is ready for cargo shipment for customer
A component is failing, and an emergency signal is not being sent to operating central	❖ Jamming of signal from the revolt by an attacker ❖ Interference in communication channel	Implement different frequencies communication channel

Table 59 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 8

[UCA8] Providing CA when sensors are not calibrated yet		
Scenarios	Causal Factors	Design recommendations / requirements
Sensors giving wrong feedback to the controller and wrong predictions are made by the system or operator	No safety procedures for calibrating the sensors must be followed on system level	Implementation of calibration functionality on the revolts system

Table 60 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 9

[UCA9] Not providing CA when the course is unsafe for current situation and the system and this is not detected by the system		
Scenarios	Causal Factors	Design recommendations / requirements
Heading towards another ship with too high speed, coming too close due to the wrong course, impact could occur	Insufficient operator awareness or insufficient alert system.	Implementation of alerts for every object that gets detected within a area set.

Table 61 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 10

[UCA10] Providing CA when the speed is too high for a heavy correction of the course		
Scenarios	Causal Factors	Design recommendations / requirements
The vessel turns over too much on one side of the vessel a possibly tips over	Too much thrust and course correction used at the same time	System override function that sets in if unsafe course change is being performed with too high speed

Table 62 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 11

[UCA11] Providing CA manually before having taken the current situation into consideration		
Scenarios	Causal Factors	Design recommendations / requirements
The operator makes a mistake and loses contact with the revolt The Revolt crashes into a nearby object	❖ No access level on different parts of the Revolts system ❖ No authorization of operator before operating the Revolt	There should be different levels of authorization for different part of the revolts system Implement authorization functionality for the Revolt

Table 63 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 12

[UCA12] Providing CA when the shipping dock has not permitting the action (other ships are dispatching at the same time)		
Scenarios	Causal Factors	Design recommendations / requirements
The Revolt dispatch from dock and crashing into another ship	Not sufficient communication with the docks operation central	Implement functionality for commination with the docks operation central, and approval must be granted before dispatch

Table 64 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 13

[UCA13] Not providing CA when ships WIFI connection is not encrypted		
Scenarios	Causal Factors	Design recommendations / requirements
An attacker use cracking WPA2-PSK with dictionary attack and Attacker receives access to the WIFI network	❖ Too weak encryption ❖ Too weak passphrase	Use more secure password on WIFI network Use a more secure encryption method

Table 65 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 14

[UCA14] Not providing CA when updates are necessary to mitigate a security issue		
Scenarios	Causal Factors	Design recommendations / requirements
An attacker exploits a certain known security issue on the system	The system has not been updated	Implement an update schedule. Update at certain times, downtime etc.

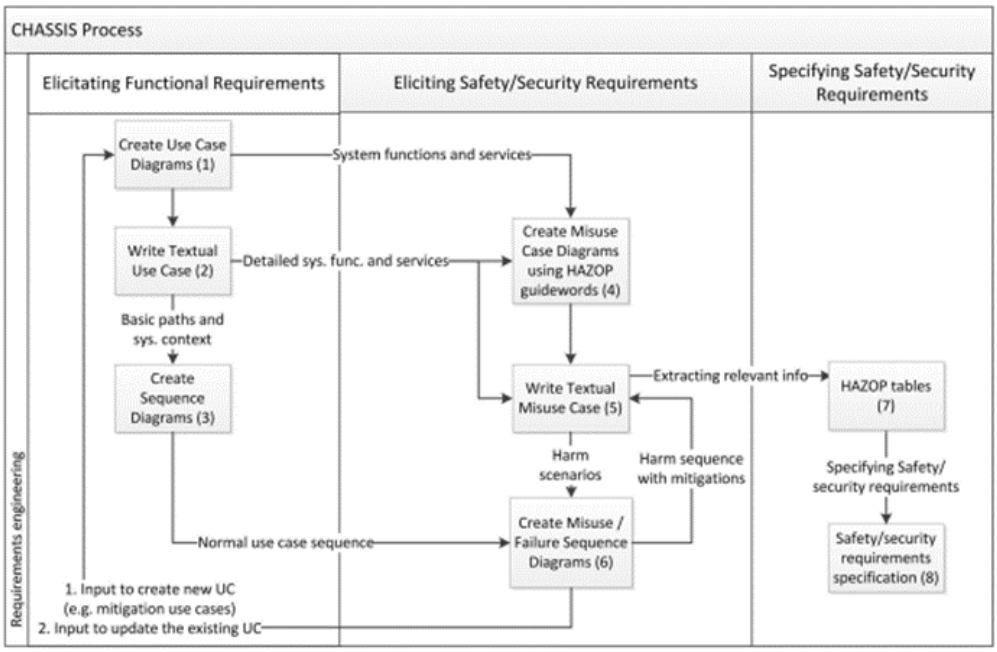
Table 66 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 15

[UCA15] Not providing CA when a spoofing or jamming attack is occurring		
Scenarios	Causal Factors	Design recommendations / requirements
Spoofing attack is occurring, Man in the middle attack on GSM base station	The system has no proception against spoofing attack	The revolts system must follow security standards for protection against spoofing attacks, this must be implemented on system level
The Revolt get attacked by a jamming attack in which take the remote receiver out of operation	The Revolts system doesn't have components equipped for protection against jamming attacks	Installation of components for protection against jamming attack Possibly switching frequencies

6.3 Results of CHASSIS analysis

I will now perform a CHASSIS analysis on the Revolt case. I will take the base in the most recent CHASSIS model. The CHASSIS analysis will focus the specific task of the main stakeholders of the Revolt system.

Table 67 - CHASSIS analysis process



6.3.1 Elicitation of functions and services

Based on stakeholder table of the Revolt, the following use case is most relevant for the Revolt vessel, and is used in this CHASSIS analysis:

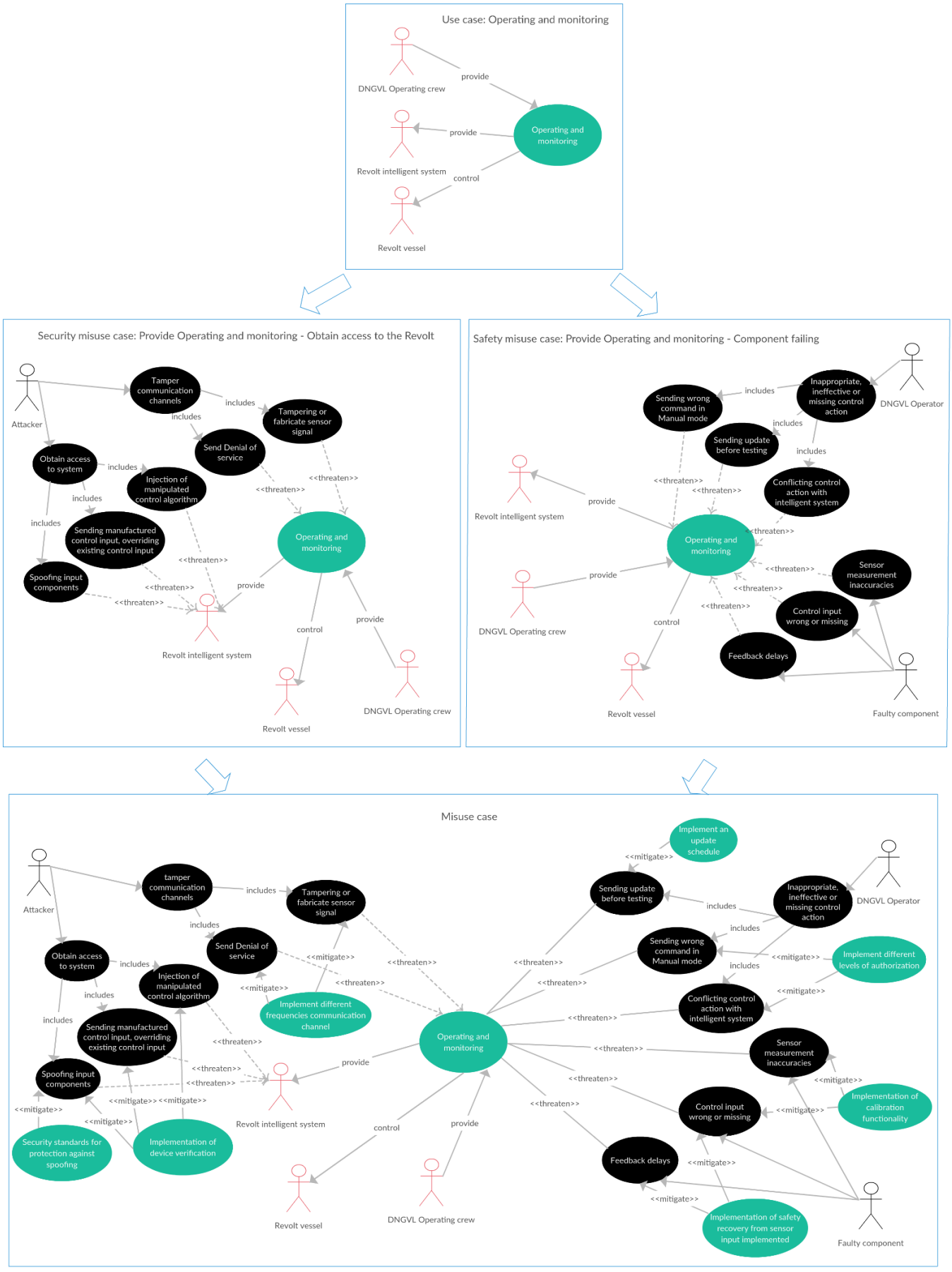
Use case 1:
Operating and monitoring the Revolt remotely by Operating central at sea and docking of cargo by DNGVL operating crew and the Revolt intelligent system.

In the following section are the different use cases and misuse cases for the CHASSIS analysis. The are created using the Creately tool [68].

Table 68 - Abbreviation for analysis

Abbreviation	Description
DOC	DNGVL Operating crew
RIS	Revolt intelligent system
RV	Revolt vessel

6.3.2 CHASSIS process



6.3.2.1 Use case: Operating and monitoring

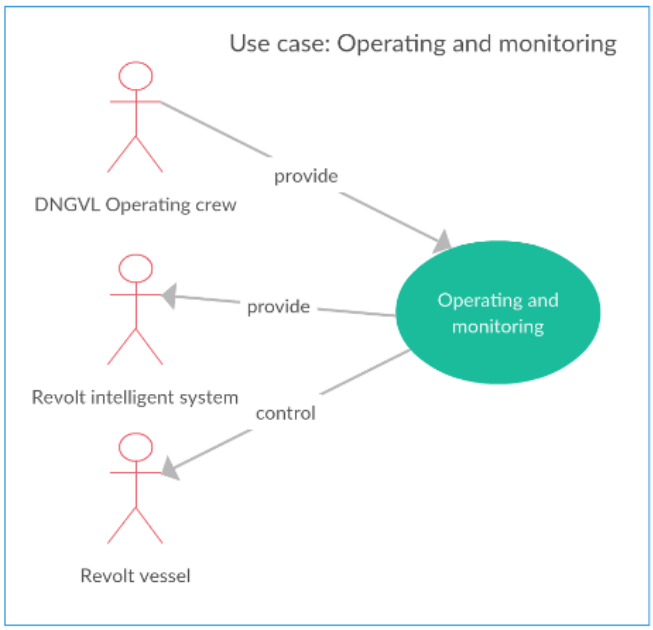


Figure 48 - Use case for Revolt

Textual Use case for case: Operating and monitoring

Table 69 - Textual Use case for case: Operating and monitoring

Name	Operating and monitoring
Iteration	1
Summary	DNGVL Operating crew (DOC) provides Operation and monitoring to the Revolt intelligent system (RIS) to and enables the control of Revolt vessel (RV)
Basic path	Bp1. DOC gives commands to RIS and then gets executes on RV Bp2. RIS makes decisions itself and executes on RV
Alternative paths	Ap1. DOC override RIS by enable manual mode and execute on RV
Exception paths	Ep1. In bp1, if command is unrecognizable the RIS will override the control of RV
Extension points	..
Triggers	1. Changes necessary during operation of RV 2. Alerts of changes from component (sensors etc.) 3. Unexcepted events in RV environment
Assumptions	1. System working as excepted 2. Communication only through Remote control or WIFI
Preconditions	..
Postconditions	..
Related business rules	..
Author	Erik Nilsen Torkildson
Date	16.11.2017

6.3.2.2 Safety misuse case: Provide Operating and monitoring - Component failing

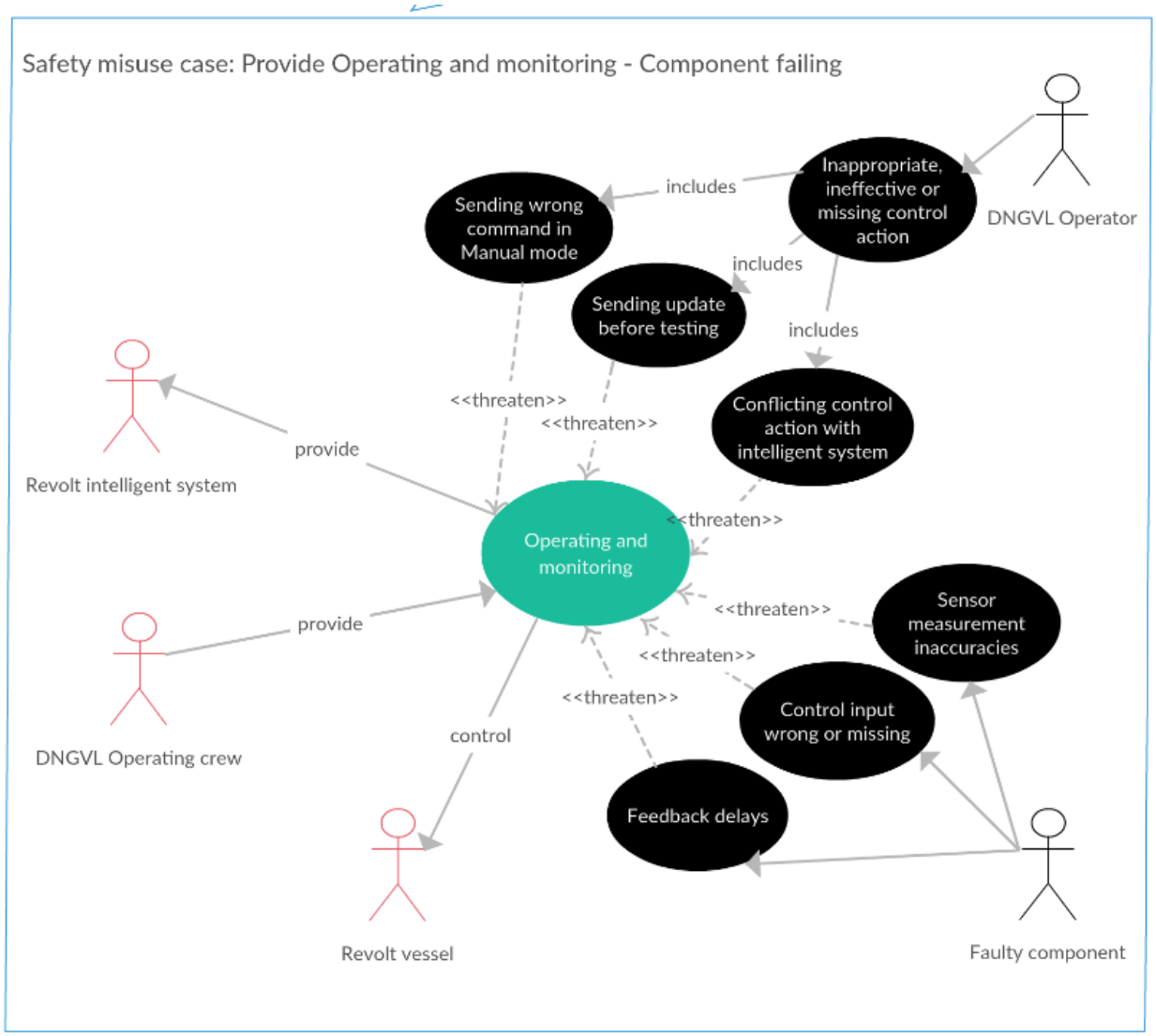


Figure 49 - Safety misuse case

Figure 49 is a safety misuse case based on the general use case the for the Revolt vessel (Figure 48).

Textual safety misuse case for case: Provide Operating and monitoring - Component failing

Table 70 - Textual safety misuse case for case: Provide Operating and monitoring - Component failing

Name	Provide Operating and monitoring – DNGVL operating crew are having faulty components on the RV. Component failing.
Basic path	Bp1. The DOC is sending commands to the RIS but gets no answer Bp2. The RIS is behaving abnormal Bp3. The DOC is sending commands to the RIS but unexcepted result occurs
Mitigation points	❖ Implement an update schedule ❖ Implement different levels of authorization ❖ Implementation of calibration functionality ❖ Implementation of safety recovery from sensor input implemented
Assumptions	• Software used for communication is used is assumed • Broadcast hardware of communication is assumed • Hardware for actuators is assumed
Preconditions	
Misuser profile	Faulty communication system
Stakeholders, risks	❖ Loss of life or serious injury ❖ Loss of cargo ❖ Loss of control of the Revolts ship and system ❖ Loss of credibility in the maritime shipping industry
Author	Erik Nilsen Torkildson
Date	16.11.2017

Table 70 is a textual safety misuse case for the graphical safety misuse case (Figure 49).

6.3.2.3 Security misuse case: Provide Operating and monitoring - Obtain access Revolt

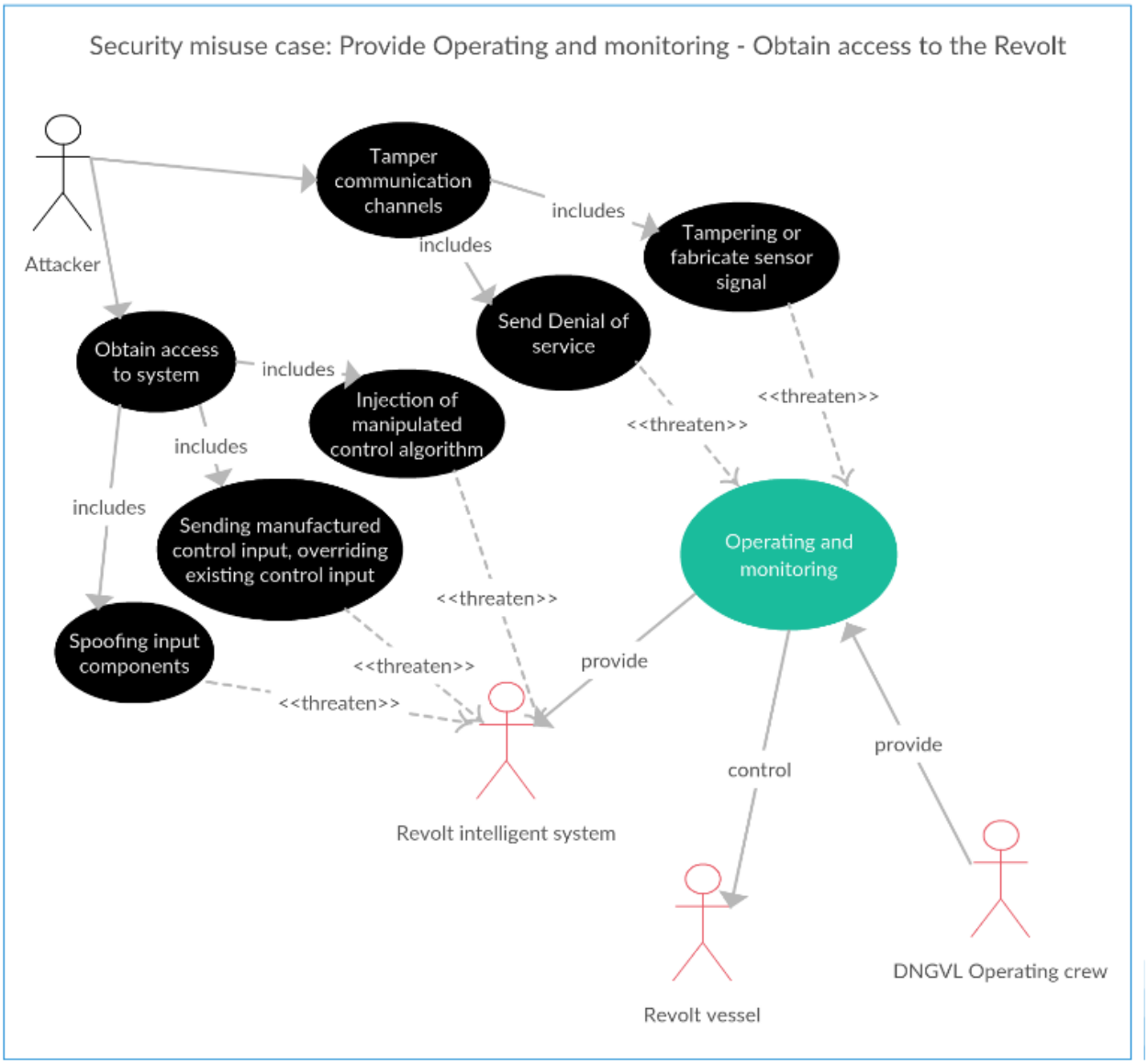


Figure 50 - Security misuse case

Figure 50 is a security misuse case based on the general use case the for the Revolt vessel (Figure 48).

Textual Security misuse case for case: Provide Operating and monitoring - Obtain access Revolt

Table 71 - Textual Security misuse case for case: Provide Operating and monitoring - Obtain access to the Revolt

Name	Provide Operating and monitoring – DNGVL operating crew are having attacks on components on the RV. Obtain access to the RV.
Basic path	Bp1. The DOC is sending commands to the RIS but gets no answer Bp2. The RIS is behaving abnormal Bp3. The DOC is sending commands to the RIS but commands are being blocked Bp3. The DOC is sending commands to the RIS but
Mitigation points	❖ Implement different frequencies communication channel. Monitoring communication channel and when jamming attack occurs - switching frequencies ❖ Implementation of device verification ❖ Security standards for protection against spoofing
Assumptions	1. Broadcast hardware of communication is assumed 2. Software used for communication is used is assumed 3. Hardware for actuators is assumed
Preconditions	
Stakeholders, risks	❖ Loss of control of the Revolts ship and system ❖ Loss of credibility in the maritime shipping industry ❖ Loss of confidential information of customer
Author	Erik Nilsen Torkildson
Date	16.11.2017

Table 71 is a textual security misuse case for the graphical security misuse case (Figure 50).

6.3.2.4 Final misuse case with mitigations

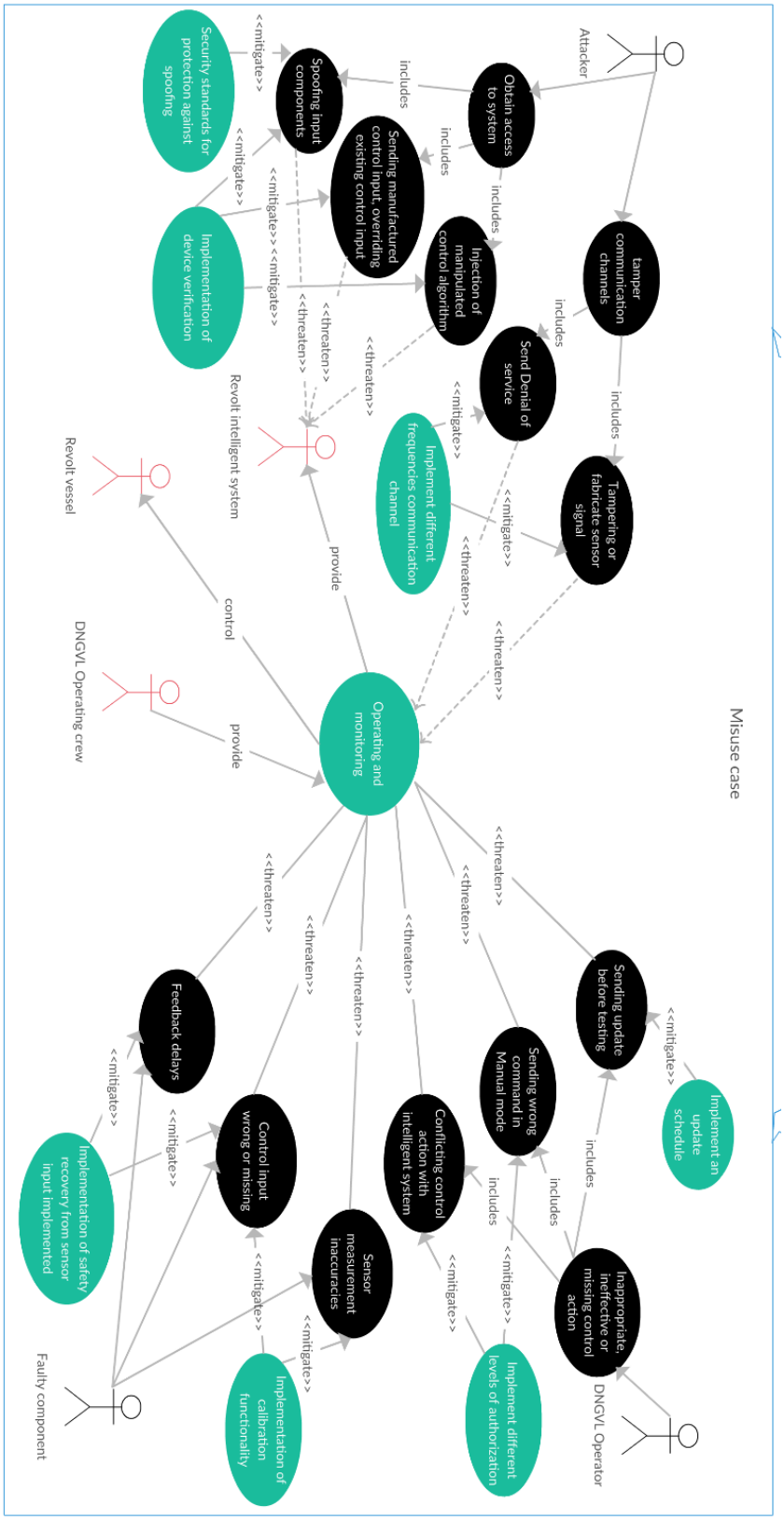


Figure 51 - Final misuse case for the use case "operation and monitoring" with mitigations

In Figure 51, the final misuse the use case "operation and monitoring" are displayed with mitigations.

6.3.3 Sequence Diagrams

In the following section are the sequence diagrams for the CHASSIS analysis. They are created using the Lucidchart sequence diagrams tool [69].

6.3.3.1 Misuse Sequence Diagram

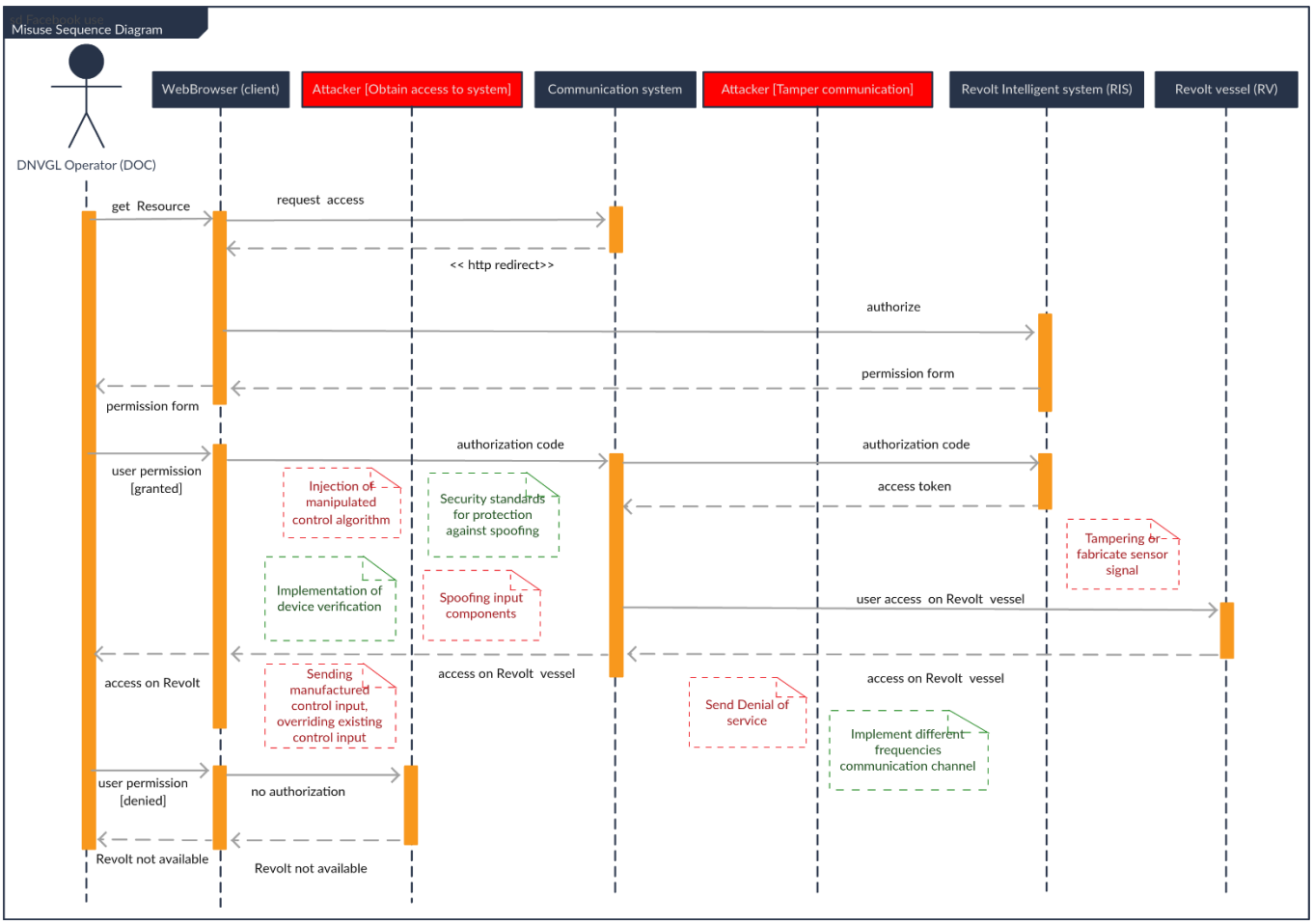


Figure 52 -Misuse Sequence Diagram for the "Provide Operating and monitoring - Obtain access to the Revolt" security misuse case

In Figure 52, there is a misuse sequence diagram for the connected misuse case diagram (Figure 50)

6.3.3.2 Failure Sequence Diagram

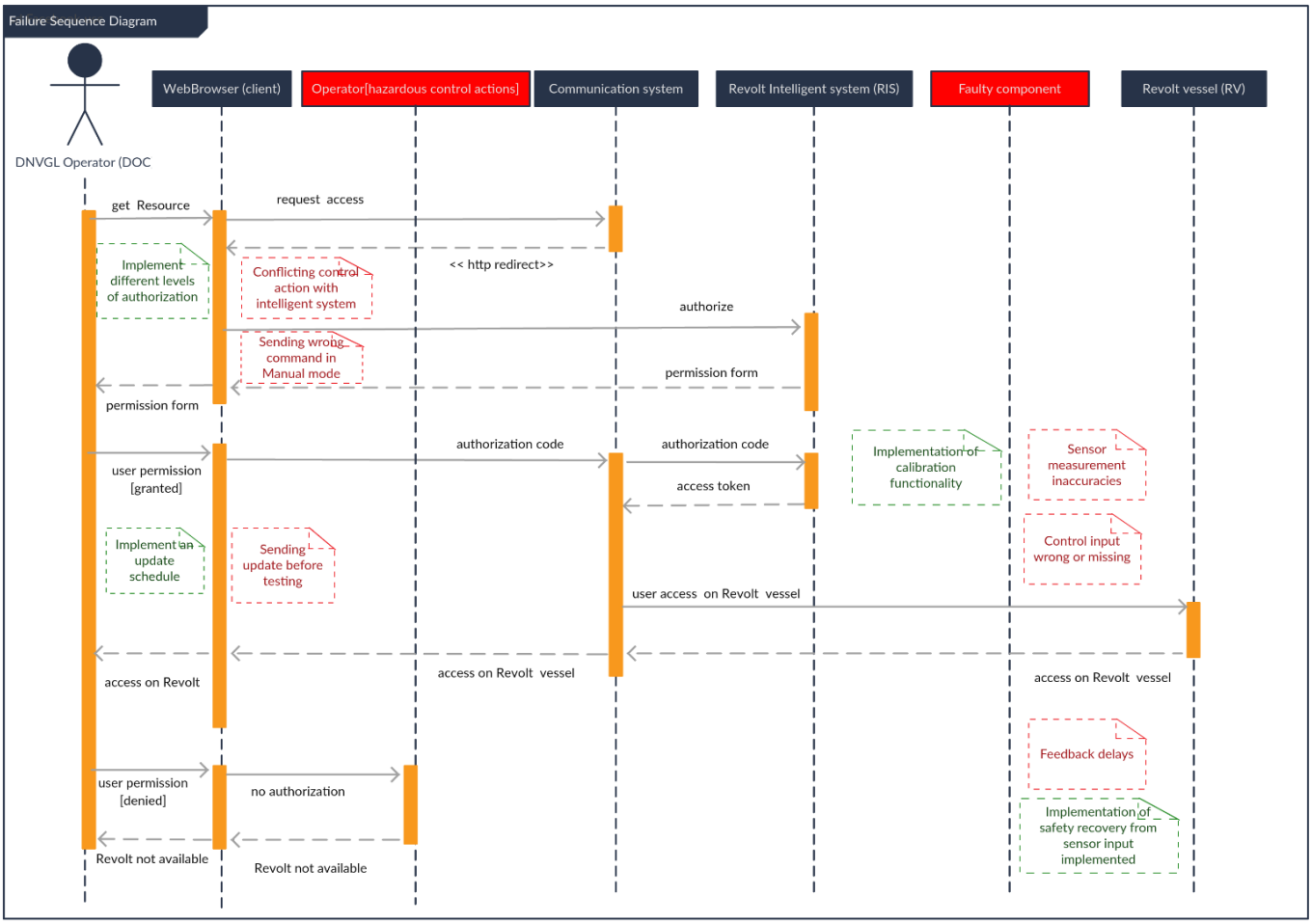


Figure 53 - Failure Sequence Diagram for the «Provide Operating and monitoring - Component failing" safety misuse case

In Figure 53, there is a failure sequence diagram for the connected safety misuse case diagram (Figure 49).

6.3.4 Perform HAZOP

Guide Words

The following are the guide words used for the HAZOP analysis [70]. They are based on the developed uses cases and sequence diagram words (previous steps in the CHASSIS analysis).

Table 72 - Guide Words for HAZOP method

Guide Words for specifying safety requirements	
Word	Description
Early	A situation where a control action has been done too early
Late	A situation where a control action has been done too late
Before	A situation where a control action has been done before its appropriate
Other than	A situation where something has happened, and outcome not expected

Table 73 - Guide Words for specifying security requirements for HAZOP method

Guide Words for specifying security requirements	
Word	Description
Flood	Attack the target repeatedly in order to overload its capacity and make the services its provide unavailable
Authenticate	Provides identification of a certain induvial, in order to grant or deny access to a resource
Spoof	A person or component Masquerade Itself In order to appear as something real in the communication process
Bypass	Avoid authentication or to access a resource on the target
Scan	Perform a set of tests on a specific target in order to extract certain valuable information of the targets characteristics
Modify	Change the characteristics of a target

Parameters

Table 74 - Parameters for specifying safety and security requirements for HAZOP method

Parameters for specifying safety and security requirements	
Parameter	Description
Request user access on Revolt vessel	The operator request user access on Revolt vessel
Access on Revolt vessel	The operator has access to perform commands on the Revolt vessel and change system files
User permission granted	The operator has access granted to the system
User permission denied	The operator has access denied to the system

6.3.4.1 HAZOP table for specifying safety requirements

Table 75 - HAZOP table for specifying safety requirements

Function	Parameter	Guideword	Consequence	Cause	Harm	Recommendation
Provide Operating and monitoring	Request user access on Revolt vessel	**Early**	System components are being updated too early	There is no environment / functionality for firmware/updates testing	The updates are causing system failure when the Revolt is at sea	There should be an environment for testing changes to the Revolts system and implementation of an update schedule
Provide Operating and monitoring	Access on Revolt vessel	**Late**	The operator has enabled manual mode too late	There is no functionality for forcing a specific mode on the Revolt	The sensors for collision detection is failing and the Revolt crashes	Force manual mode in certain situations
Provide Operating and monitoring	Access on Revolt vessel	**Before**	The operator performs operations on the revolt before having done security and safety procedures	No access level on different parts of the Revolts system before operation	The operator makes a mistake and loses contact with the revolt	There should be different levels of authorization for different part of the revolts system for different individuals
Provide Operating and monitoring	Access on Revolt vessel	**Late**	The Revolts components are having feedback delays and commands are executed to late	There could be too weak signal on antennas Or signal being block/ noise. Components could also fail	The position of the Revolt could not be predictable and therefore collision could occur	Implementation of safety recovery from sensor input and backup system/components on the revolt in case of failure
Provide Operating and monitoring	Access on Revolt vessel	**Other than**	The sensors on the revolt could give data other than is considered correct	There is no calibration functionality for correcting sensors	The operator might perform hazardous decisions based on wrong data from sensors	Implementation of calibration functionality

6.3.4.2 HAZOP table for specifying security requirements

Table 76 - HAZOP table for specifying security requirements

Function	Parameter	Guideword	Consequence	Cause	Harm	Recommendation
Provide Operating and monitoring	Request user access on Revolt vessel	**Spoof**	The system has no proception against spoofing attack and can be **spoofed**	An attacker is Spoofing input components	Spoofing attack is occurring, Man in the middle attack on GSM base station	Implement different frequencies communication channel
Provide Operating and monitoring	Access on Revolt vessel	**Modify**	There is no device verification and a man in the middle attack could happen and from there **modify** system files	An attacker is Tampering or fabricate sensor signal	The Revolt is being attacked physically and a sensor component is being spoofed	Implementation of device verification
Provide Operating and monitoring	User permission denied	**Flood**	The system is not equipped to prevent a **flooding** attack on either network components	An attacker is sending an Denial of service attack	The operator might not reach the Revolt because components and services are overloaded	Security standards for protection against spoofing and Installation of components for protection against jamming attack
Provide Operating and monitoring	Access on Revolt vessel	**Modify**	The communication system might have vulnerabilities that could lead to **modification** of system files	An attacker is injecting manipulated control algorithm	The attacker might change system files/algorithms and take control of the system	Follow security standards for protection encrypting of communication signals
Provide Operating and monitoring	User permission denied	**Bypass**	The system might have components or services that could be **bypassed**	An attacker is sending manufactured control input, overriding existing control input Spoofing a satellite's signal with a false signal	The attacker might for some time take control over the revolts location	Implementation of device verification and implement different frequencies on communication channel

In Table 75 and Table 76, the HAZOP analysis is performed, based on the guide words.

6.4 Results of Comparison of the three methods

Table 77 - Comparison of the three methods used in case study 1

	FMVEA		**STPA and STPA-sec**		**CHASSIS**	
Safety related hazards detected	5		13		5	
Security related hazards detected (threats)	5		6		5	
Total hazards detected	10		19		10	
Time used method / total time used	55 hours / 242,5 hours		105 hours / 242,5 hours		82,5 hours / 242,5 hours	
Time used on methods	22,68%		43,29%		34,02%	
Time used on each step of method in hours	System level analysis	5h	Define & Frame Problem	7h	Elicitation of functions and services	4h
	Selection of component	3h	Losses/accidents and Hazards	5h	Use case diagram and textual	9h
	List functions of selected component	10h	Create functional control structure	30h	Safety misuse case diagram and textual	9h
	Failure Mode, Vulnerabilities and Effect Analysis	27h	Identify unsafe and Hazardous Control Actions	20h	Security misuse case diagram and textual	9h
	Risk assessment	10h	Generate casual scenarios	20h	Final misuse case with mitigations	15h
			7. Mitigations and control	23h	Misuse Sequence Diagram	8h
					Failure Sequence Diagram	8h
					Perform HAZOP	20,5h
Documents needed	List of system components Wiring diagram of system		List of system components Wiring diagram of system Input/output list Software functionality		List of system components Stakeholder information and usage of the system	
Knowledge required	Basic usage and functionality of components. The system architecture is used for rating risks		STAMP knowledge Functionality provided by the system to create a functional control structure		UML notation Knowledge about various security and safety methods misuse case, hazop etc.	
Domain knowledge recommended	Security expert Safety expert		Security expert Safety expert Software engineer		Security expert Safety expert Requirement engineer	
Difficulty degree	Low		Medium/High		High	

In Table 77, the results for the Revolt case study regarding hazards, time consumption, documents needed, and knowledge required/ recommended.

6.4.1 Safety related hazards detected

Table 78 - Comparison of safety related hazards detected by the three methods in case study 1

Hazard category	FMVEA	STPA and STPA-sec	CHASSIS
Hazards related to updates	-Over the air updates- Connection is lost -Over the air updates- Update causes faults	-Providing CA1 when system components are being updated	-System components are being updated too early
Hazards related operation		-Providing CA3 manually before having taken the current situation into consideration -Providing CA3 when the shipping dock has not permitted the action (other ships are dispatching at the same time) -Providing CA4 when revolt ship protocols for dispatch has not been followed	-The operator has enabled manual mode too late
Hazards related to safety procedure		-Providing CA1 when the revolt is in the middle of a hazardous situation and the system has already set commands for collision avoidance	-The operator performs operations on the revolt before having done security and safety procedures
Hazards related to faulty sensors/ components	-Wrong sensor input data	-Providing CA1 when sensors/components are not operating correctly at the time -Not providing CA2 when the speed is unsafe for current situation and the system and this is not detected by the system -Providing CA2 when sensors are not calibrated yet	-The sensors on the revolt could give data other than what is considered correct
Hazards related to GPS/GNSS	-Wrong GNSS/GPS input data -System error - causes execution of command delays or system failure	-Providing CA2 before the system reports to be functioning correctly -Not providing CA3 when the course is unsafe for current situation and the system and this is not detected by the system	-The Revolts components are having feedback delays and commands are executed too late
Hazards related to environmental factors		-Not providing CA1 when an emergency situation is occurring in the area of the vessel (e.g. oil leak, storms or other) -Providing CA2 when the vessel is on route and there is a possibility of losing the connection and the speed is set too high for upcoming situations -Providing CA3 when the speed is too high for a heavy correction of the course	
Control actions related to the STPA and STPA-sec method: CA1: Control the Position of the vessel CA2: Control the Speed of the vessel CA3: Control the Course of the vessel CA4: Control the Access to the vessels system			

Table 78 illustrates a comparison of the different safety related hazards detected with the three methods. If the hazard is placed in the same category they are somewhat related to each other or has the same subject. There are five hazard categories: updates, operation, safety procedure, faulty sensors/components, GPS/GNSS or environmental factors.

6.4.2 Security related hazards/ threats detected

Table 79 - Comparison of security related hazards detected by the three methods in case study 1

Threat category	**FMVEA**	**STPA and STPA-sec**	**CHASSIS**
Threats related to GPS/GNSS	-GPS spoofing attack: Man in the middle attack at GSM base station	-Not providing CA4 when a spoofing or jamming attack is occurring	-The system has no proception against spoofing attack and can be spoofed
Threats related jamming	-Wireless connection is targeted to jamming		-The system is not equipped to prevent a flooding attack on either network - components
Threats related to WIFI	-Cracking WPA2-PSK with dictionary attack	-Not providing CA4 when ships WIFI connection is not encrypted	-The system might have components or services that could be bypassed
Threats related to access control	-No device verification: Man in the middle attack with access to RosCore -Transmit diagnostic Data - Man in the middle attack	-Providing CA4 before revolt has been authorized	-There is no device verification and a man in the middle attack could happen and from there modify system files
Threats related to no action towards known vulnerabilities		-Not providing CA4 when updates are necessary to mitigate a security issue -Providing CA4 when system components are being updated and before having done firmware testing to see the results of the update. -Providing CA4 when system components are being updated and before having done firmware testing to see the results of the update.	-The communication system might have vulnerabilities that could lead to modification of system files
Control actions related to the STPA and STPA-sec method: CA1: Control the Position of the vessel CA2: Control the Speed of the vessel CA3: Control the Course of the vessel CA4: Control the Access to the vessels system			

Table 79 illustrates a comparison of the different safe security related hazards/threats detected with the three methods. If the hazard is placed in the same category they are somewhat related to each other or has the same target. There are four threat categories: GPS/GNSS, jamming, WIFI, access control or no action towards known vulnerabilities.

Similar results between methods?

From comparing all the hazards/threats detected, I can see that there are some similarities between the hazards and threats being detected from each method e.g. the first column in Table 78, has both FMVEA, STPA and STPA-sec and CHASSIS hazards related to updates, with some differences related to formulation, but they are very similar.

Similar results between safety and security?

There are also similarities between safety and security detected hazards/threats e.g. there is both a column in Table 78 and Table 79 with hazards/threats related to GPS/GNSS. Therefore, I can be sure to say that there is both hazards and threats related to this component, and all of the methods have some detection of them.

6.4.3 Risk assessment stages coverage

Table 80 - Risk assessment stages coverage by methods

Method	Risk Identification	Risk Analysis	Risk Evaluation
FMVEA	X	X	
STPA and STPA-sec	X	X	X
CHASSIS	X	X	

The table above illustrates what the different methods have coverage over, in regard to the steps in a risk assessment, based on their definitions. All of the three methods involve a step where risk is identified. The FMVEA uses quantitative risk analysis and the STPA-sec uses qualitative risk analysis. I would argue that CHASSIS also uses qualitative risk analysis with the use of the HAZOP method. However, only the STPA-sec method has a risk evaluation process, which determines if a risk is acceptable or not. An interesting discovery is that the risk evaluation stage is not regarded as important in the FMVEA and CHASSIS methods.

6.4.4 Qualitative vs Quantitative Risk analysis

What has been made clear in this thesis is that both FMVEA, STPA & STPA-sec and CHAISSIS includes a risk analysis. However, the FMVEA is based on a quantitative analysis, whilst STPA & STPA-sec and CHAISSIS is qualitative based. This raises a general question about the differences between qualitative and quantitative research.

A quantitative analysis does depend more on what data is available, often this type of analysis is only performed with risks that already has been picked out for further analysis. Therefore, a qualitative analysis is often performed first to pick out some risks and then used as input for a quantitative analysis. Therefore, I would suggest that the FMVEA method can be useful to combine with either STPA-sec or CHASSIS. To for example, further analyze the effects of certain hazards/threats.

6.4.5 Method origin

Table 81 - Method origin

	Extension of existing method	Combination of existing method(s)
Component-based	FMVEA	
System-based	STPA-sec	CHASSIS

The CHASSIS method is considered a system-based method, because it focuses on interactions between entities, which could also include human actors. The STPA-sec is a system-based method, it focuses on the functionality provided by the system to create a functional control structure. The FMVEA focuses explicitly on components.
The FMVEA is based on the FMEA method, and STPA-sec is based on STPA and STAMP. However, the CHASSIS does not origin from a specific method, but a combination of methods – use cases, misuse cases and HAZOP.

6.4.6 Discussion of comparisons

6.4.6.1 Cost-effectiveness

From this case study there has emerged some interesting results. The total amount of time used on this case study is 232,5 hours. Documents are provided by DNV GL, some adoption of these documents had to be done before performing the analysis methods. The analysis methods have been done with a base in black box testing [71].

The FMVEA method seems to require the least amount of resources and time for its analysis. However, the method also detects the least number of hazards. The input of this method is not much when considering the other methods. Only a list of components on the system and how they are connected. This is a limitation but might be an advantage for early analysis of the system development.

The STPA and STPA-sec method seems to require a moderate amount of resources, but at the same time being a time-demanding method. The method detects the highest number of hazards, and I would say that main reason is the generation identification of hazardous control actions with the use of process model variables, which might generate scenarios that would otherwise not be thought of. This process can lead to a high number of unsafe scenarios, and their practical risk need to be considered.

The CHASSIS method seems to require a high degree of interdisciplinary collaboration between the safety and security domains. Also, knowledge about UML notation and methods like use cases, misuse cases and HAZOP are an advantage to have. If there is limited knowledge, this method might be more time demanding than maybe necessary. A restriction I would say that the method have is the starting point with a use case. This use case can't be as broad, otherwise the methods that follows in the process might be difficulty to use, with this input.

Table 82 - Comparison of methods used in the Revolt case study

FMVEA	STPA-Sec	CHASSIS
❖ Generic Method ❖ Based on safety analysis method (FMEA) ❖ Based on Components (Software or hardware)	❖ Model-Based Non-graphical method ❖ Formal method ❖ Based on safety analysis method (STPA) ❖ Based on system-level scenarios	❖ Model-based graphical method ❖ UML based ❖ Based on requirements engineering

What is clear is that all the methods can analyze both the safety and security aspects but take significantly different approaches in doing so. The FMVEA method focuses on only the components and how they are connected. The STPA-sec focuses on a top-down system level analysis with generation of system scenarios. The CHASSIS method focuses on UML and combining different existing methods e.g. use cases, misuse and at the end a HAZOP analysis for risk assessment and possible mitigations.

Time consumption vs. hazards detected

The FMVEA methods can require very high time consumption if every component needs to be analyzed. In this situation only one component was chosen, and a relatively low amount of time was used. However, in a very complex system with variously different components, the outcome might be different if every component were to be analyzed. One might therefore choose to not analyze certain components, but then risks to not discover hazards that you thought were not risk in a certain component, but possibly still is. In other words, there needs to be done a fair amount of effort in the selection of components for analysis, before actually performing an analysis.

6.4.6.2 Security and safety hazards identified

The FMVEA method seems to produce more specific security hazards but lacks the amount of safety hazards compared to the other two methods. A reason for that might be that the method focuses on specific threats modes on one the selected component, and therefore is not as abstract.

The STPA and STPA-sec methods seem to produce a high amount of safety hazards. A possible reason for this is the way the early steps of the process are performed and focuses on. When scoping out a certain frame and problem, the following unacceptable losses and selection of the process variables sets the foundation for what is generated of the method.

The CHASSIS method did produce an equal amount of safety and security hazards. The method does have a very separate focus between the safety and security aspects. The final misuse case does give the possibility exchange interactions between the aspects, if the entities are relatable. The selection of the use case can be very limiting dependent on how broad it is. By selecting a use case that potentially can have both safety and security issues, the number of hazards seems to be the highest.

6.4.6.3 Methods applicable to different situations

Since the FMVEA method require the least amount of knowledge and experience. The method can be appropriate at the planning and design phase of a system. Where maybe different components of the system is being considered.

I would say that the STPA and STPA-sec method are more applicable later in the system life-cycle. The method requires a more established system for a good analysis. When considering that the method requires an I/O of the system, the method is more applicable for the testing and deployment phase.

The CHASSIS method takes usage of expert knowledge and different stakeholders of the system, discussing different use cases with the users of the system. I would say that CHASSIS is more resource demanding than the other methods. Therefore, this method is more applicable to the later steps of the systems life-cycle, similar to the STPA and STPA-sec method. The use cases are essential to the method, and therefore can probably be useful for an evaluation of a system.

6.4.6.4 Security aspect – categorizing and measuring the effects of hazards

When regarding the effect of an attack, I think the STRIDE model, used in the FMVEA method was useful for categorizing different types of attacks. However, when finding out the effects of these attacks, this becomes more case specific. For example, if an unauthorized person obtains access to the WIFI network, in which the Revolt broadcasts, there is a potential for the attacker to obtain access to devices connected to the WIFI. For determining how likely this is to occur, the FMVEA method has an advantage by determining:

- System susceptibility
- Threat properties
- Attack probability

The STPA and STPA-sec does take a different approach, by looking at the Hardware and interface level, and developing a control loop of the system. With STPA-sec, I think the key for having a good security analysis is to have the right process model variables, to generate the right casual scenarios and security causal factors.

6.4.6.5 Summary of the methods based on experience

FMVEA

Benefits	Drawbacks
Requires the least amount a resources and knowledge for performing an analysis.	Does not take any consideration into how the components/methods are used. With the exception of how it is wired, in which could give some, but limited information.
Could give detailed information about hazards/ issues based on a specific component.	Little effort, but also limited discovery of vulnerabilities
Good at categorizing and measuring the effect of security related hazards	

STPA and STPA-Sec

Benefits	Drawbacks
Detects a considerable number of hazards based on the time used.	Hazards could lack details, more in-depth information about the system.
Take the system as a hole into consideration and can better understand the intelligence part of the system.	Could be time demanding.
The method is adaptable, with the changing of model process variables.	Does only detect aspects that are relevant for the control loop

CHASSIS

Benefits	Drawbacks
High usage of the different stakeholders of the system, to map how it is actually used.	Requires a considerable amount of resources.
Takes advantage of existing methods that are proven to be useful.	Requires knowledge about various existing methods.
	The use case is the limiting factor.

6.4.6.6 Addressing the challenges with autonomous systems

I would argue that these three methods are addressing some of the challenges that autonomous systems have regarding safety and security. All of the methods can be used for risk identification and analysis. However, the methods that set itself apart is the STPA and STPA-sec method, in which also includes the risk evaluation step.

6.4.6.7 Possible limitations of the study

As stated before, a big challenge with safety and security co-analysis on autonomous systems, is to understand the intelligence part. I would say that the STPA and STPA-sec method tackles this challenge the best, with taking a system-level approach. A big challenge for autonomous system is the risk management of the intelligent part of the system. The development of a functionally control structure gives the best foundation for understanding how decisions are made on a very intelligent system, in which autonomous systems certainly need to be. After a control structure is established, control actions is used for generating casual scenarios and eventually mitigations.

Verdict

With the currently iteration, none of the methods used in this case study are suitable and efficient enough to be used on autonomous systems. Therefore, the next focus in thesis will be to explore how the STPA-sec method can be improved, so that it might be better suited to cover the security aspect of a system and might be better applicable for autonomous systems.

6.4.6.8 Limitations of STPA-sec in security analysis

Why does STPA-sec not detect all threats? Related to the control loop?

Security threats do not so much get triggered by what you do, but what the infrastructure and services is set up and provided to the users.

Safety is more related to how the infrastructure is used. A lot of unsafe hazards are triggered by unsafe use of the system or infrastructure.

The control structure the STPA and STPA-sec method provides, does not make it natural to include all the security features. For example, what encrypting is used for communication? This is not discovered, because it is not relevant to how the system is controlled. Therefore, some security aspects are missing, but could show up in the AC -accident cause step of the STPA-sec process.

This is a major weakness with the STPA-Sec method and will therefore be a focus area for Research question 2, i.e. how to improve the weaknesses of STPA and STPA-sec for a better safety and security co-analysis?

Chapter 7 : Results of Research Question 2

Interesting discoveries has been made when answering research question 1 and advantages and drawbacks for each method has been made. From these results, I will attempt to find ways of improving the STPA for security, since this method has different weaknesses.

Examples of method that could be combined will be based on the Revolt case study, where I explore possible solutions for improving the STPA-sec methods. Based on experiences from the Case study.

I will try to improve the STPA and STPA-sec from two aspects.

- Combination of methods - I will try to use the proposed methods in some aspect, to try to improve the already established weaknesses of the STPA-sec method. Threat modelling methods will also be explored.
- Often is the approach with safety and security co-analysis methods to start from a safety standpoint. Could we go the opposite route, and take a base in security? STPA and STPA-sec starts with safety analysis. Could I go the opposite route, e.g. start with security analysis, taking a base in the target assets related risks/ consequences, and then include safety?

Threat modeling methods will be used for attempting to try to improve the weaknesses of the STPA-sec method. I have been inspired to use some of the threats models from a presentation at by Per Håkon Meland, SINTEF Digital and NTNU. The presentation was held at NTNU and the topic was Threat modeling [72].

7.1 Recap of STPA and STPA-sec analysis

STPA-Sec [52] [53] extends STPA, which is a safety analysis method, a System-Theoretic Process Analysis [51]. The extension is to includes security analysis, as shown in Figure 54.

The main steps of STPA plus STPA-Sec are:

- Identifying what essential services and functions must be protected or what represents an unacceptable loss.
- Identifying system hazards and constraints.
- Drawing the system control structure, physical hardware and network structure, and identifying unsafe control actions.
- Determining the potential causes of the unsafe control actions. The potential causes could be security vulnerability and threats. To facilitate the security analysis, some guide words like tampered feedback, injection of manipulated control algorithm, and intentional congestion of feedback path, are added [55].

Compared to other security analysis methods, STPA-sec does not focus on countermeasures that should be taken, but mainly on identifying scenarios that could lead to losses [73].

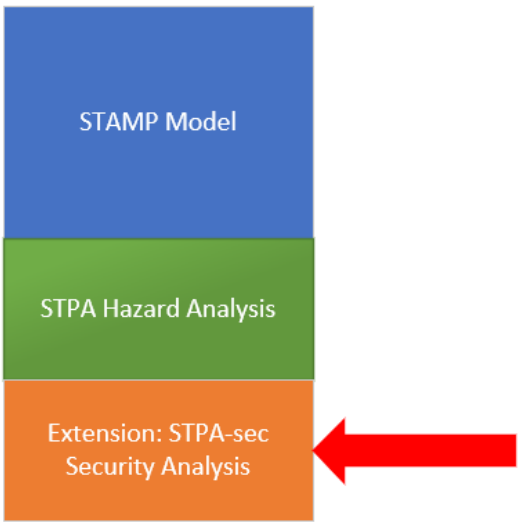


Figure 54 - Where STPA-sec analysis originates from

7.2 Comparison of possible threat modeling methods for improving STPA-sec

Table 83 - Comparison of possible methods for improving the STPA-sec

Method	Description	STPA-sec incretion / data extraction	Will improve what aspect
FMVEA	Further analysis the effects of certain hazards/threats. More specific: Discovered Threats: Security Causal Factors	After step 4: Identify unsafe and Hazardous Control Actions	• Categorizing and measuring the effect of a security related hazards • Better Risk management of discovered threats
Misuse cases / Misuse case map	Use control actions as use cases and finding other vulnerabilities. Then translate these vulnerabilities into unsafe control actions (and extracted back to the STPA-sec process)	After Step 3: Create functional control structure and control actions	• Find unsafe control actions that otherwise might not be find, relevant for security
Data Flow Diagram	Use a data flow diagram based on the control structure of the target system. Information input to and output from the system (data flow) is translated into control actions	After Step 3: Create functional control structure	• Adding the communication layer to the control structure and therefore considering aspects such as encryption.
Bow-tie diagram	Use the general hazards of the system to perform the bow tie method	After step 2: Losses/accidents and Hazards	• Safety and security integration
Attack tree	Further analysis the effects of all discovered hazards/threats	After step 4: Identify unsafe and Hazardous Control Actions	• The tree structure the method provides can be leveraged by STPA-Sec to connect and present the final analysis results and then extended the method by the results for in-depth security analysis.
BPMN + Threats	Model the controlled process from the control structure with BPMN and find possible threats. These threats will be translated into unsafe control actions	After Step 3: Create functional control structure	• Find more security related unsafe control actions
Socio-Technical Security modeling language (STS-ml)	Use the control actions to create an STS diagram to find the social dependencies trough the social interactions the system will be exposed to.	After Step 3: Create functional control structure and control actions	• Find more security related unsafe control actions trough social aspects

7.2.1 Methods placement in the STPA-sec process - for STPA-sec incretion / data extraction

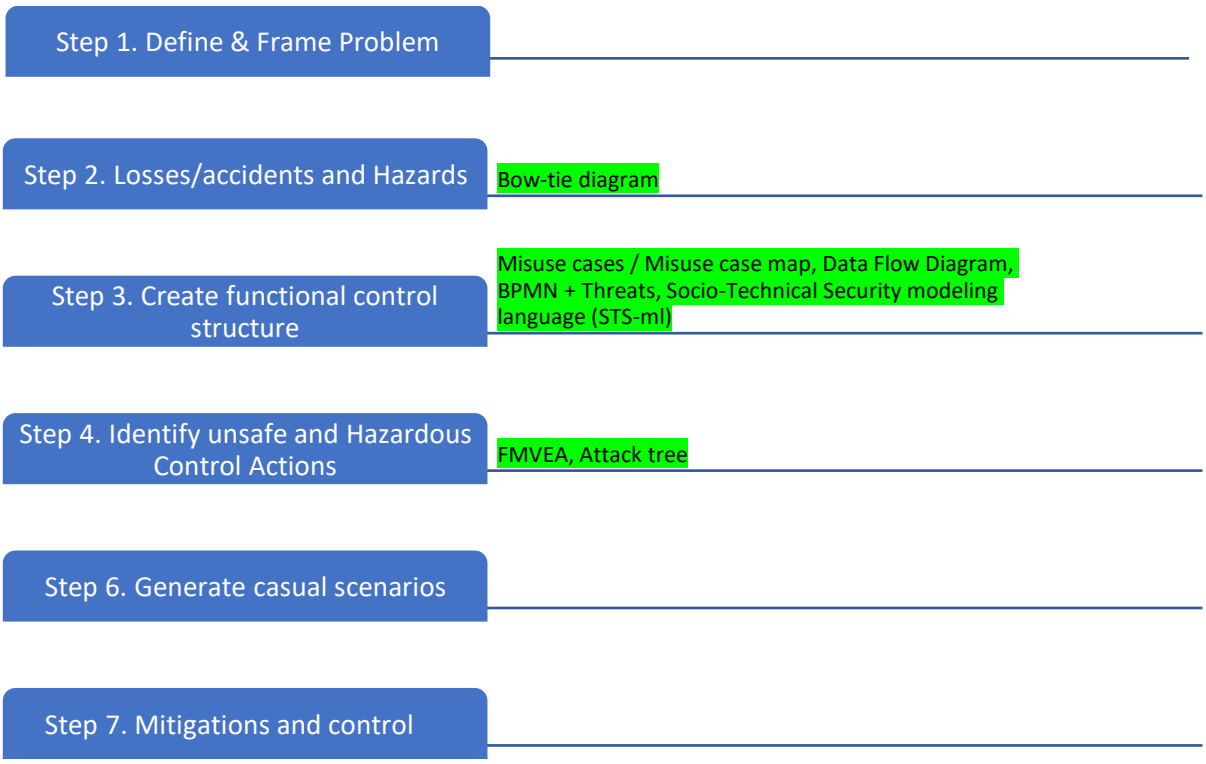


Figure 55 - Methods placement in the STPA-sec process

The figure above shows where the different methods will be used in the STPA-sec process to improve certain parts of it. We can see that all the methods are being used between step 2 and 4. The intent for these methods is to improve the STPA-sec method and naturally there has to be some data already produced before these methods can have any use. Therefore, these methods are in the middle steps of the STPA-sec method. As of some data has been produced and will be used for the method that will improve the STPA-sec process. When these data have been translated and issued as input on the method, they will eventually be translated and put back into the STPA-sec process.

7.2.2 Methods for improving STPA-sec - Data input/output

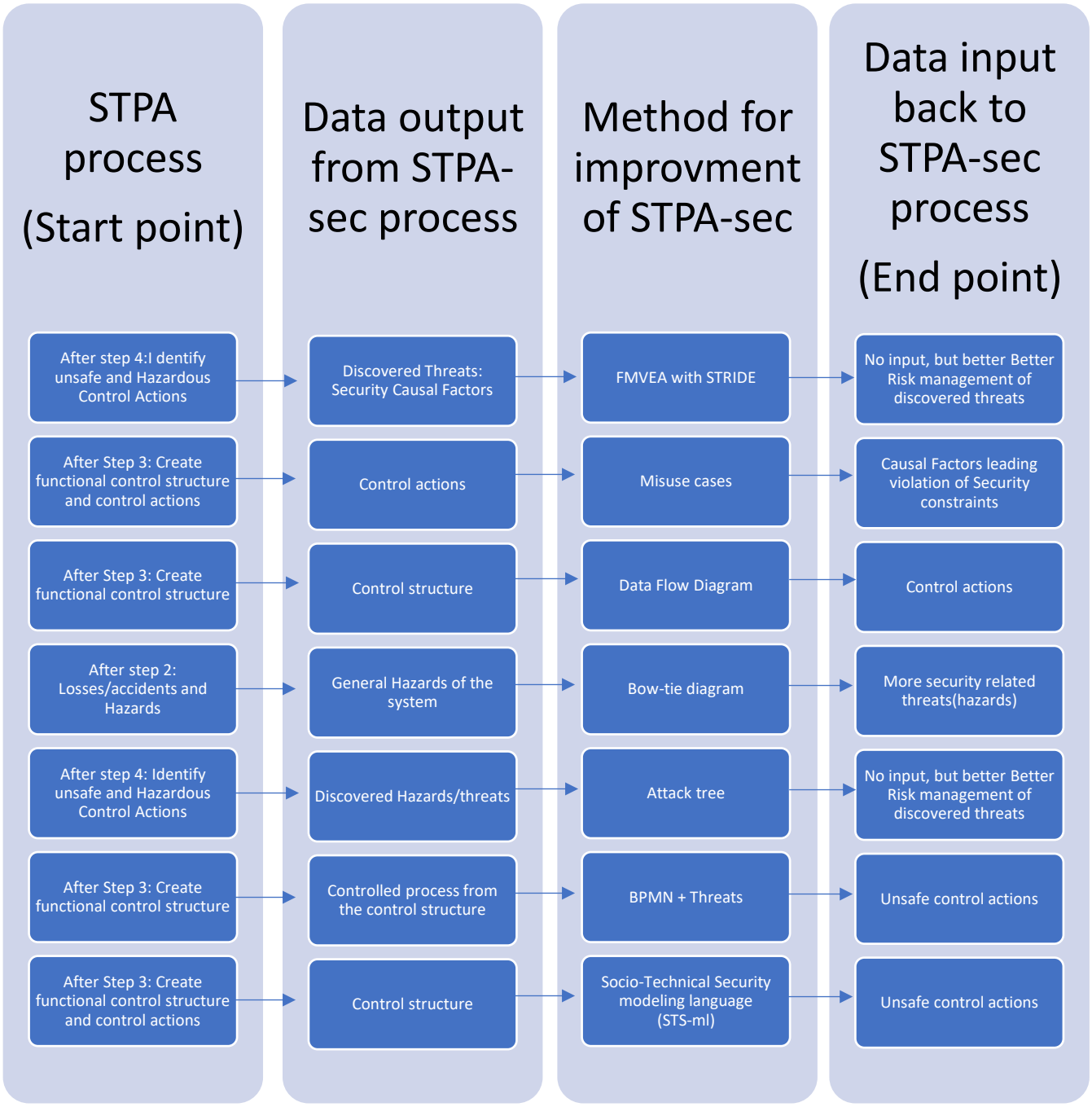


Figure 56 - Example of processes of data exchanges between the methods when integrating and improving the STPA-sec method with other methods

The figure above shows what might be the different processes when integrating and improving the STPA-sec method with other methods, specifically the data flow, output and input between the methods.

7.2.3 Combination of each method

I will now try to use the proposed methods in some aspect, to try to improve the already established weaknesses of the STPA-sec method.

The method will first be introduced and a possible approach for combination for the STPA-sec method will be discussed. Then a real-world example will be included, with an example from Revolt case study. At the end the results will be discussed, the suitability for combination with the STPA-sec will be considered.

7.2.3.1 FMVEA with STRIDE – for Quantitative risk analysis

From these results, I will argue that there could be useful to combine FMVEA and STPA-sec methods. The FMVEA method has the advantage of categorizing and measuring the effect of a security related hazards, in which STPA-sec lacks in this area. The STPA-sec on the other hand takes a system-level approach in which could better understand the system by developing casual scenarios and security causal factors. The STPA method detects the highest number of safety related hazards, and I would say that main reason is the generation identification of hazardous control actions with the use of process model variables, which might generate scenarios that otherwise would not be thought of.

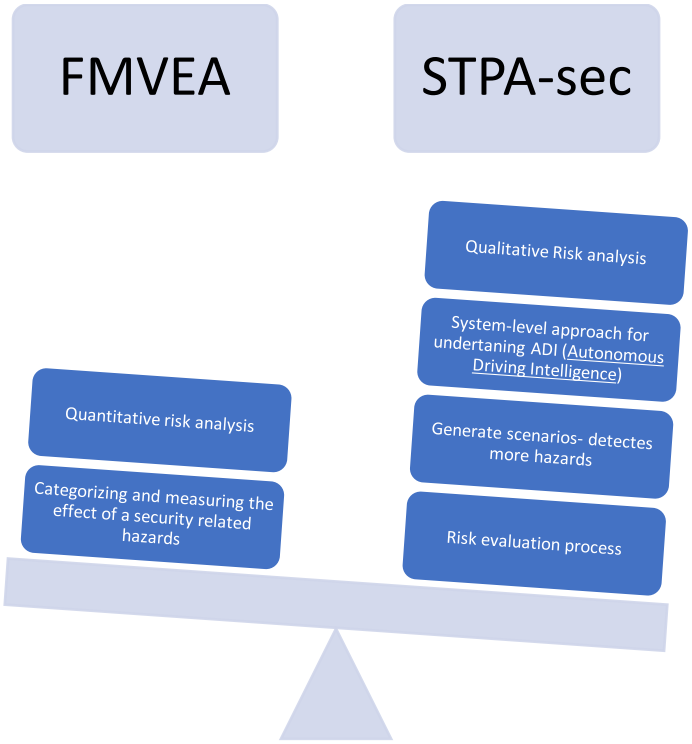


Figure 57 - Combination of the FMVEA and STPA-sec methods

I would also say that the FMVEA gives also more detailed hazards, where the STPA-sec method might lack a bit in this area.

The FMVEA include a quantitative risk analysis and does depend more on what data is available, often this type of analysis is only performed with risks that already has been picked out for further analysis. Therefore, a qualitative is often performed first to pick out

some risks and then used as input for a quantitative analysis. Therefore, I would suggest that the FMVEA method can be useful to combine with STPA-sec. To for example, further analyze the effects of certain hazards.

In other words, the FMVEA methods, can give the STPA-sec method some aspects, in which it lacks. And the combination will give a good understanding of safety and security related risks that autonomous systems are exposed for.

FMVEA combined with STPA-sec

Security Causal Factors from Revolt case study:

- Too weak signal on antennas. Signal being block/ noise
- Too weak encryption
- The system has not been updated

Table 84 -FMVEA methods on an example from Revolt case study

ID	component / element	Vulnerability / Failure Cause	Threat Mode/ Failure Mode	Threat Effect/ Failure Effect	System Status	System Effect	Severity	System Susceptibility	Threat Properties	Attack/Failure Probability	Risk
1	GPS/ GNSS antenna	GPS/ GNSS connection is targeted to jamming	Attacker interrupts connection between operator and Revolt	Revolt is unreachable	Remote operation	Attacker has control over the Revolt's system	Critical:4	4	5	9	36
3	WIFI connection	Cracking WPA2-PSK with dictionary attack	Attacker receives access to the WIFI network	The Revolts embedded computer is vulnerable	Normal operation	System integrity is hurt	Critical:4	4	6	10	40
6	Update system	Lack of updates causes faults	Components don't work as intended	System could have critical faults	Normal operation	System is no longer reliable	Moderate:3	..	..	4	12

What we can see from trying to combine the FMVEA method with the STPA-sec, is that this combination of methods provides a better risk evaluation process, including a qualitative risk analysis. This is something that is useful to better prioritize the threats/hazards according to high or low risk and make requirements based on.

7.2.3.2 Misuse cases

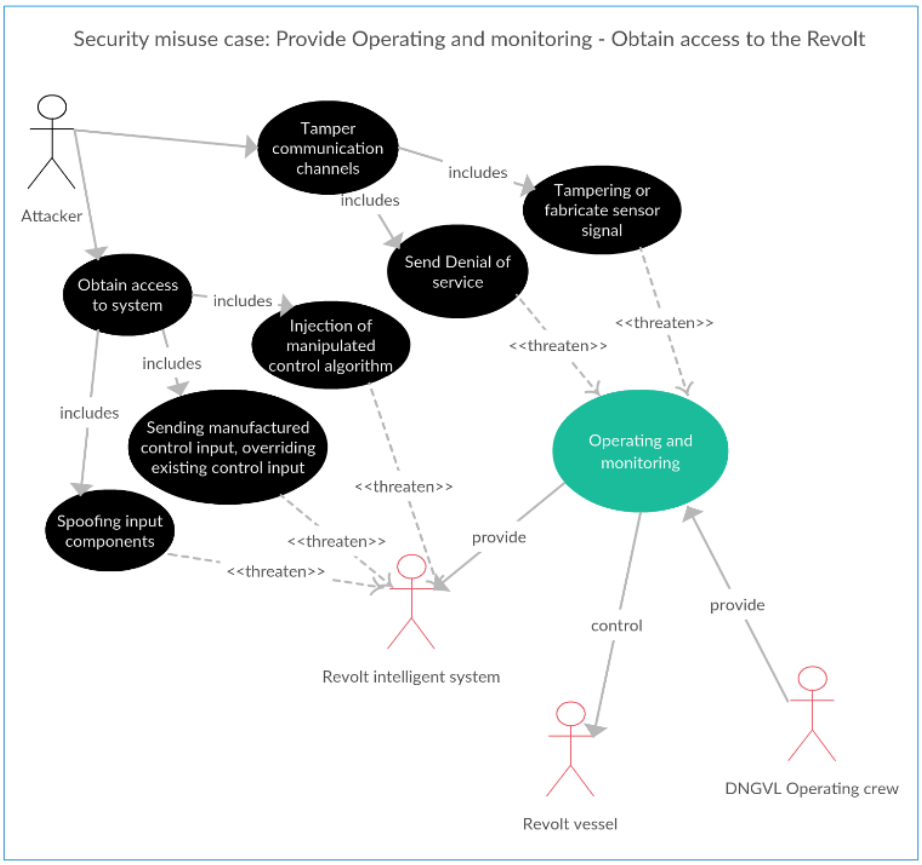


Figure 58 - Example of misuse case from Case study: Revolt

We have previously tested using misuse cases in the CHASSIS method for the Revolt vessel. However, CHASSIS is a relative complex and long process. Therefore, to integrate this with STPA-sec, selecting only the misuse case method [74], could be a better option. What could improve the STPA-sec method is to find security vulnerabilities like for example improper encryption for the target case. A way this can be done is to use existing control actions as use cases and misuse cases. After this is done and a misuse case is created. Then certain relevant vulnerabilities will be translated into causal factors leading violation of security constraints, and therefore extracted back to the STPA-sec process.

Misuse cases combined with STPA-sec

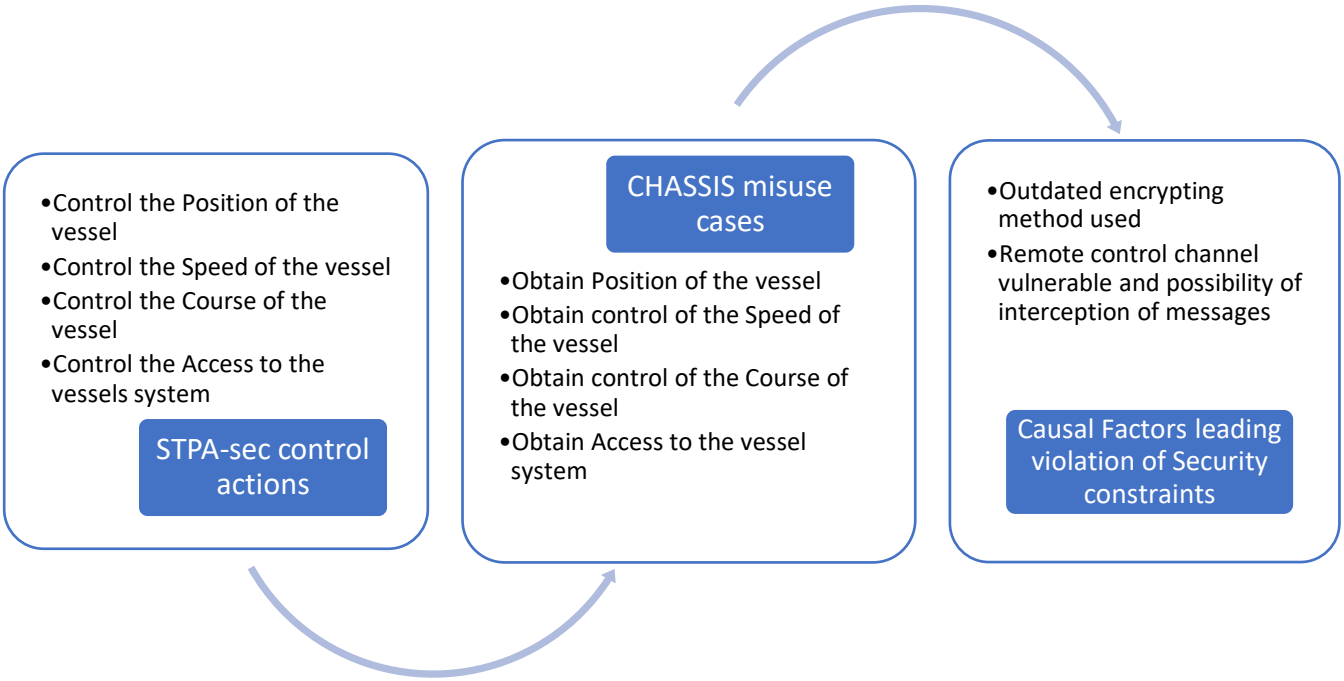


Figure 59 - Example of combination of STPA-sec and misuse cases on the Revolt case study

From the STPA-sec process on the Revolt case study, the control actions are used as input for the Misuse case method.

In the example in Figure 59, the use cases from the Revolt case study has been used as an example of how the information can be extracted from the use cases.

Combing a misuse case with the STPA-sec method did discover more security related threats. These threats where related to the weaknesses of the STPA-sec method – the data flow and encrypting related to the communication the system uses. The method discovers two casual factors that could lead to violation of security constraints. These were based on the control action of the system.

7.2.3.3 Data Flow Diagram

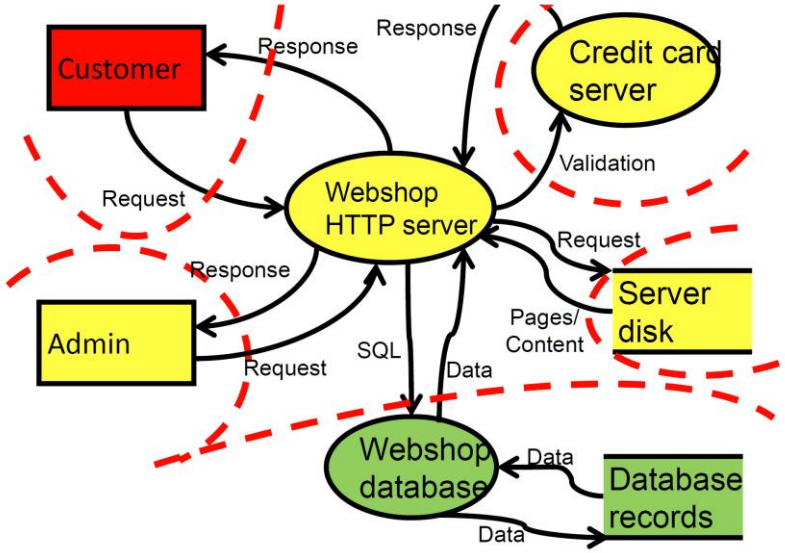


Figure 60 - Example of Data Flow Diagram

A Data flow diagram is intended to better understand the system via documenting the data flow between subsystems or different components of the system [75]. The method illustrates the attack surface of the system and potentially critical components. The symbols this diagram uses are intended to show data inputs, outputs, storage points and the interaction between them.

A big weakness with STPA-sec is that the control loop the STPA method provides, does not make it natural to include all the security features. For example, what encrypting is used for communication. Because, this is not relevant to how the system is controlled. This is exactly what the data flow has, and what the STPA-sec is missing.

By adding the communication layer to the control structure from the STPA-sec. The STPA-sec method will be strengthened when it also could consider aspects such as encryption.

The intention is to use a data flow diagram based on the control structure of the target system. Then this information that the data flow diagram provides e.g. input to and output from the system (data flow) is translated into control actions and will be if it is an unsafe control action.

Data Flow Diagram combined with STPA-sec

From the STPA-sec process on the Revolt case study, the Control structure of the Revolt is used as a basis for the Data Flow Diagram.

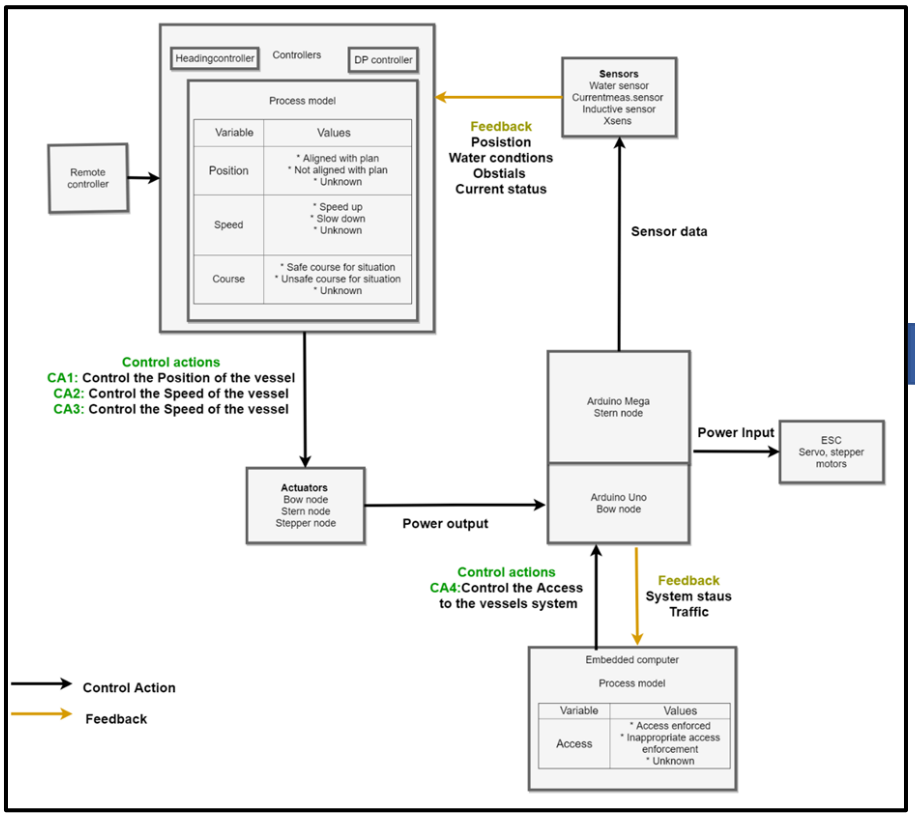


Figure 61 - Control structure of the Revolt as a basis for Data flow diagram

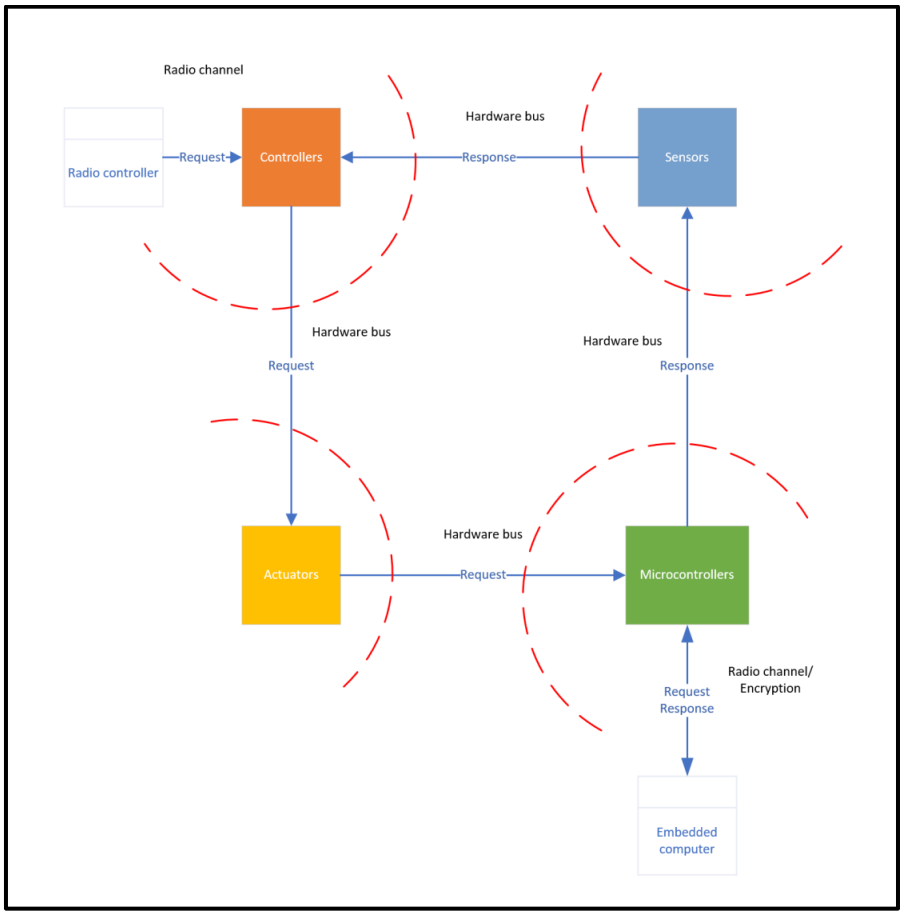


Figure 62 - Example of a Data flow diagram based on Revolt control structure

With this approach control actions are created from messages on the data flow diagram. After new control actions are added, based on the data flow diagram. We create a new control structure of the Revolt, with a complete coverage of security. The following control actions could be added to the STPA analysis:

- Request/receive encrypted message from embedded computer to microcontrollers/ actuators
- Request/receive encrypted message from Remote controller to controllers

7.2.3.4 Bow-tie diagram

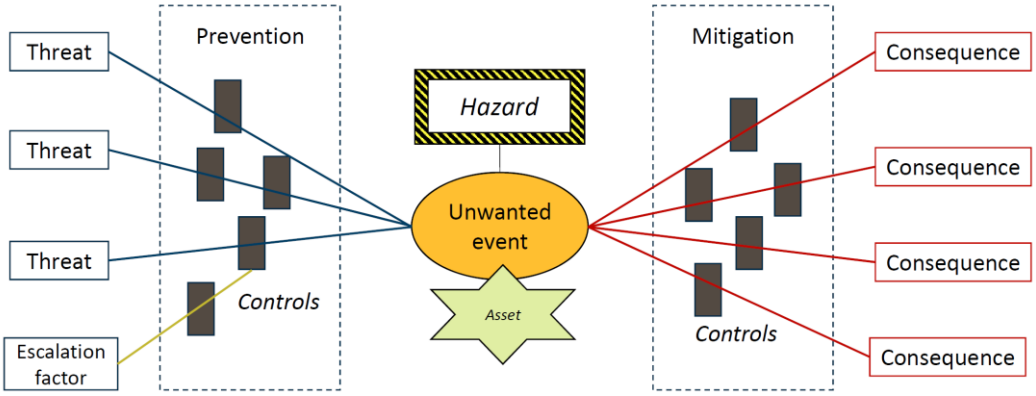


Figure 63 - Example of Bow-tie diagram

With the Bow-tie modelling method [76], I get the possibility to model a single unwanted event at the time. This gives the option to get in-depth data about this event, since there are different causes/threats, and consequences to each event. The diagram this method produces illustrates the related risk the certain event represents.

The Bow-tie term comes from how it looks, in the middle is the unwanted event, to the left is the prevention measures and to the right is the mitigation measures. This creates a clear distinction between proactive and reactive risk management. Also, a risk the diagram alone could otherwise be difficult to explain. This is easier to understand and is a strength with this method.

The Bow tie method could be used to improve the Safety and security integration, since there is an area in the STPA-sec method that could be improved. Integrating the bow tie method with STPA-sec, could be done with using the hazards of the system (After step 2: Losses/accidents and Hazards) to perform the bow tie method. This method could find other relevant threats from the specific events (selected hazards) and find more security related aspects of the system.

Bow-tie diagram combined with STPA-sec

From the STPA-sec process on the Revolt case study, hazard number 3 is extracted and used as input for the Bow-tie method:

- H3: Someone unauthorized getting access to the vessels data

Also, the operating state variables relevant for the hazard as relevant:

- Test Mode
- Emergency Stop Mode

The relevant loss for the hazard is relevant:

- L6: Loss of information

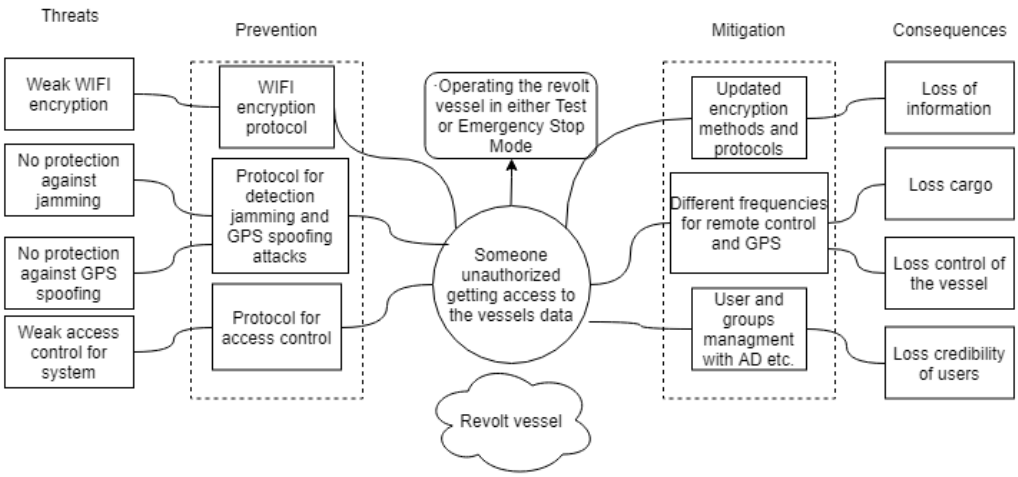


Figure 64 - Example of a Bow-tie diagram based on general hazards of the Revolt case study

The bow tie methods discovered four threats based on hazard number three from the STPA-sec method in the Revolt case study. The following threats where found:

- Weak WIFI encryption
- No protection against jamming attack
- No protection against GPS spoofing attack
- Weak access control for system

7.2.3.5 Attack tree

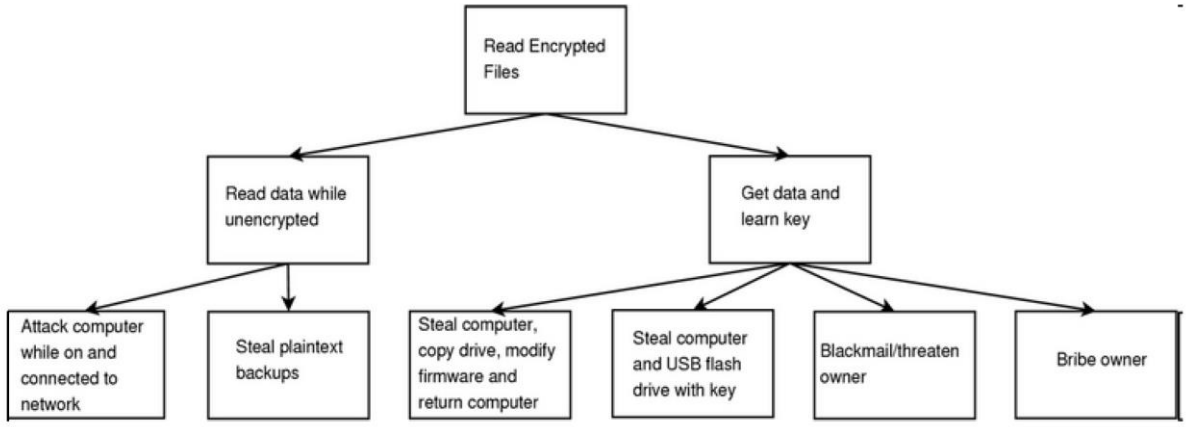


Figure 65 - Example of Attack tree

An attack tree is a formal method that is often used when concerning security of critical systems [77]. This method produces a tree diagram which illustrated an attack of a system. A typical attack tree will include:

- Root – The source of the attack
- Leaves (Nodes) – Various ways of arching the attack goal. The leaves can be AND or OR leaves:
  - AND leaves – means that there are children leaves that must be fulfilled to achieve the goal
  - OR leaves - means at least one child leaves must be fulfilled to achieve the goal

The attack tree method will produce a tree structure of an attack of the system. This can be leveraged by STPA-Sec to connect and present the final analysis results and then extended the method by the results for in-depth security analysis. This will further analyze the effects of all hazards/threats and possibly find other vulnerabilities of the target system.

Advantages:

- Useful for small or well-known attack types, because good attack vectors are provided.
- The structure of the attack tree can be re-used

Disadvantages:

- Identification of possible scenarios can be subjective
- Attack scenarios could be limited

Attack tree combined with STPA-sec

From the STPA-sec process on the Revolt case study, unsafe control action number 13 is extracted and used as input for the attack tree method:

Table 85 - Generation of casual scenarios, causal factors and design recommendations for unsafe control action 13

[UCA13] Not providing CA when ships WIFI connection is not encrypted		
Scenarios	Causal Factors	Design recommendations / requirements
An attacker use cracking WPA2-PSK with dictionary attack and Attacker receives access to the WIFI network	❖ Too weak encryption ❖ Too weak passphrase	Use more secure password on WIFI network Use a more secure encryption method

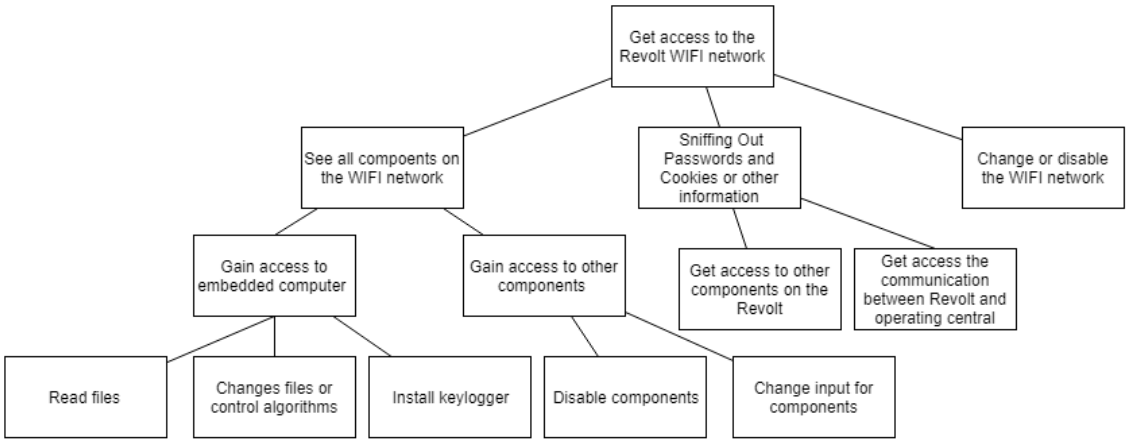


Figure 66 - Example of an attack tree diagram based on an unsafe control action of the Revolt case study

The attack tree method have shown to provide better risk management of discovered threats based on unsafe control action number 13 from the STPA-sec method of the Revolt case study. However, some of the tree branches could also be new discovered threats. For example, by getting access to the WIFI network. The attacker has access to all the components on the network. This gives the possibility to read files, change files and install a keylogger on a computer. This is a new threat of loss of information and should be added to the STPA-sec analysis as an unsafe control action, for example.

7.2.3.6 BPMN for Threats

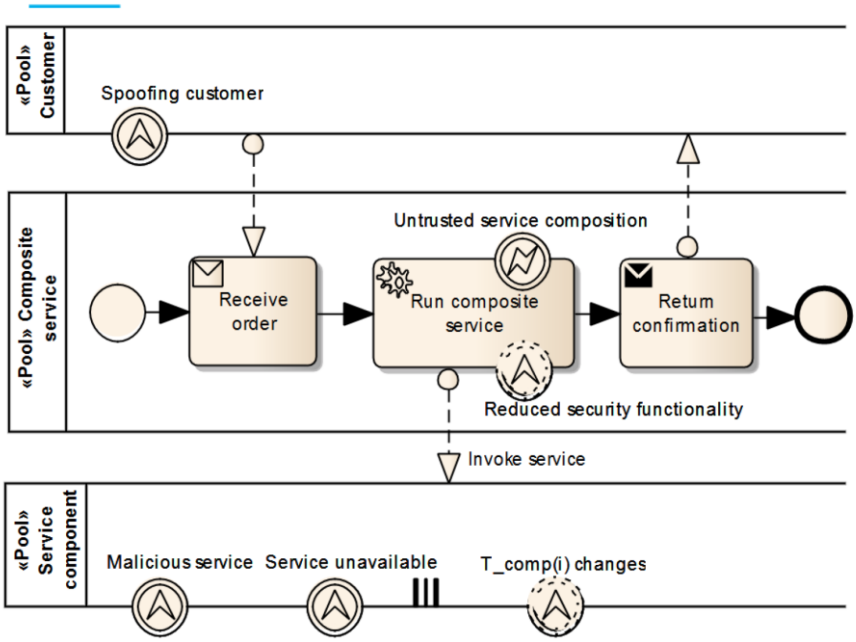


Figure 67 - Example of BPMN used for threats (Using escalation events to represent threats in a collaboration diagram)

Using the Business Process Modeling Notation (BPMN) in the context of threats and security has been explored in previous studies [78]. The result shows that traditionally BPMN has been used for regular business processes but could be very suitable for security and threat modeling.

This gives an opportunity to combine and integrate BPMN with the STPA-sec process. What could be the goal when using the BPMN with STPA-sec is to find more security related unsafe control actions.

To find more unsafe control actions with the BPMN process. An approach could be to model the controlled process from the control structure with BPMN and find possible threats. These threats will be translated into unsafe control actions.

BPMN combined with STPA-sec

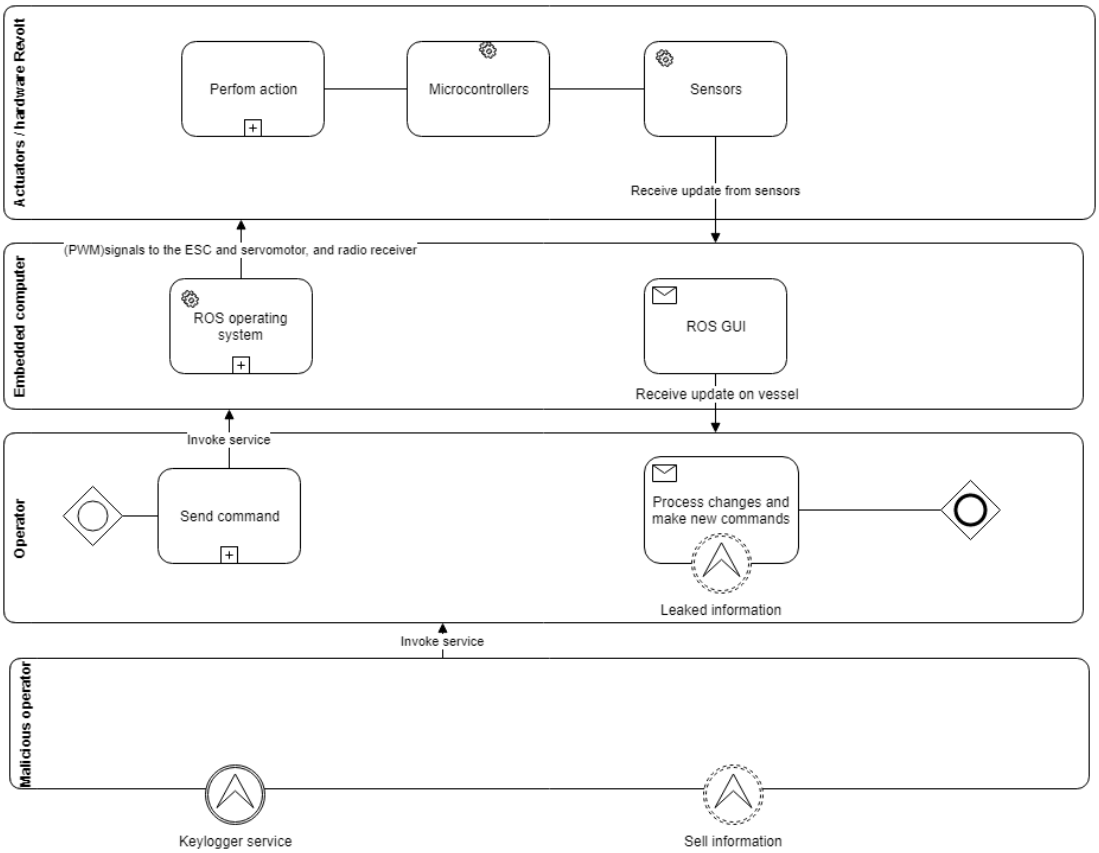


Figure 68 - Example of an BPMN process diagram based on the controlled process from the control structure of the Revolt case study

From the STPA-sec process on the Revolt case study, the Control structure of the Revolt is used as a basis for the BPMN method. In particular, the operation of the embedded computer is used for the BPMN process.

The BPMN method is traditionally used for business processes. However, in this case the method has the possibility to find threats related the operating the Revolt vessel. What is discovered is the possibility to steal information about how the Revolt is maneuvered and what cargo it has. Doing this in the form of an keylogger installed by a malicious operator. The unsafe control actions could be:

- Invoke service without checking current services running on system
- Invoke service without regularly performing security scans on system

7.2.3.7 Socio-Technical Security modeling language (STS-ml)

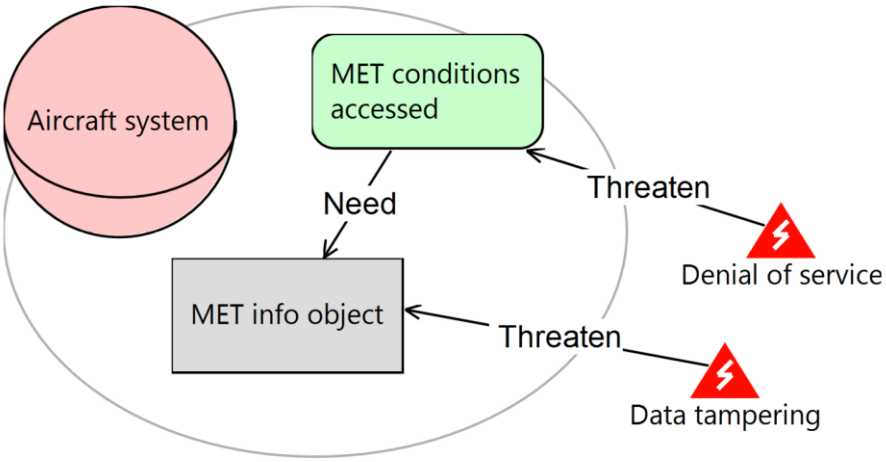


Figure 69 - Example of Socio-Technical Security modeling language (STS-ml)

When using the modelling language Socio-Technical Security modelling language (STS-ml) [79], the security aspect could be improved by adding the service-oriented perspective. This is done by adding the information about the services the system provides in terms of what the goal each actor has and what the services exchange of information. By doing this we can relate security requirements to social interactions the system has. What is being illustrated in these diagrams the method provides is the constrains related to the way different actors exchange data, and which goals the actors has. This helps to specify which security requirements are related to each actor or service.

An STS diagram could be used to improve the threats that are exposed to the system from a social standpoint. A solution could be to use the basic control structure to create an STS diagram to find the social dependencies through the social interactions the system will be exposed to.

The goal with this method is to find more security related unsafe control actions through social aspects and insert these back into the STPA-sec process.

STS-ml method combined with STPA-sec

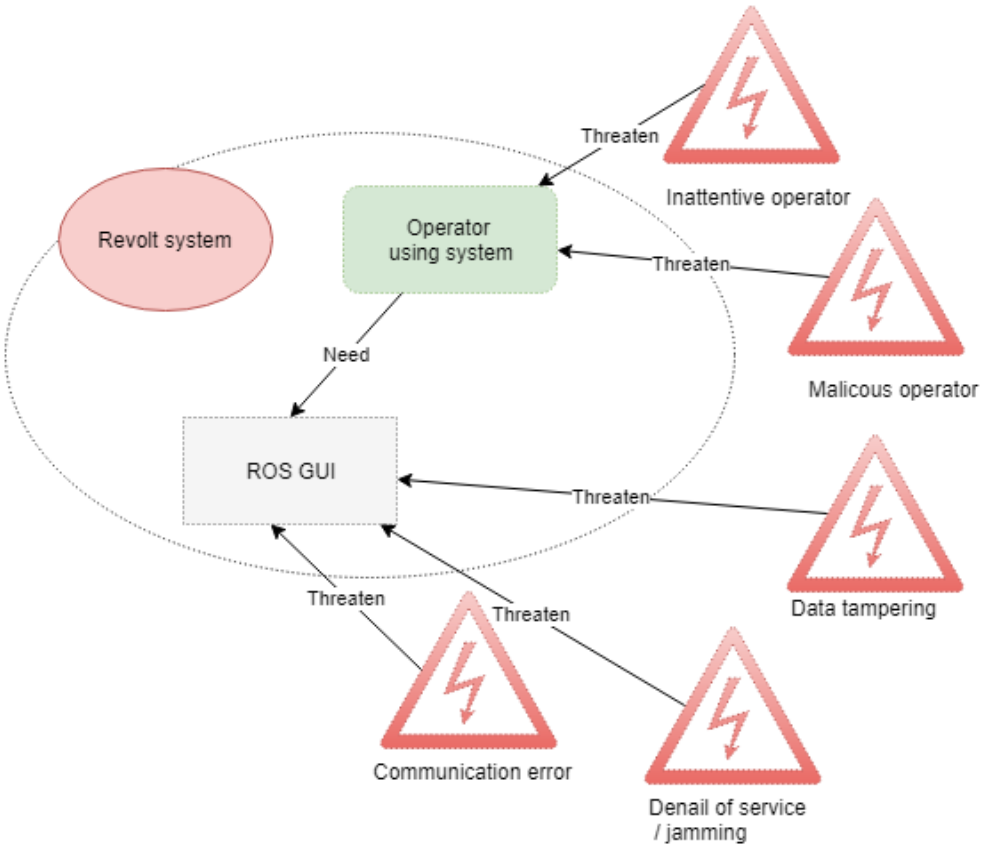


Figure 70 - Example of an STS-ml diagram based on the controlled process from the control structure of the Revolt case study

From the STPA-sec process on the Revolt case study, the Control structure of the Revolt is used as a basis for the STS-ml method.

The STS-ml method does show promising results. However, in the case with the Revolt case study, the STS-ml method did not find any new threats related to the social aspect. There were no new threats found with this method, but this is case specific. So, this method could be a benefit to combine with the STPA-sec methods in another case study.

7.2.4 Discussion of combining STPA-sec with threat modeling methods

Table 86 – Conclusion - possible ways of improving the STPA for security

Method	Results	What does the method add to the STPA-sec	Suitability for combination with STPA-sec
FMVEA	Provides a better risk evaluation process, including a qualitative risk analysis. This is something that is useful to better prioritize the threats/hazards according to high or low risk and make requirements based on. However, does not find any new security vulnerabilities.	Risk evaluation of existing threats. No new threats.	Provides better risk evaluation process.
Misuse cases from CHASSIS	Combining a misuse case with the STPA-sec method did discover more security related threats. These threats where related to the weaknesses of the STPA-sec method – the data flow and encrypting related to the communication the system uses. The method discovers two casual factors that could lead to violation of security constraints.	2 new casual factors that could lead to new security vulnerabilities	Takes a stand in security and the target assets, the usage of the assets.
Data Flow Diagram	Based on the data flow diagram we have created from the control structure of the Revolt. The following control actions could be added to the STPA analysis: -Request/receive encrypted message from embedded computer to microcontrollers/ actuators -Request/receive encrypted message from Remote controller to controllers	5 new Control actions that could lead to new security vulnerabilities	Covers explicitly a large weakness with the STPA-sec.
Bow-tie diagram	The bow tie methods found four threats based on hazard number three from the STPA-sec method in the Revolt case study. These threats were found: -Weak WIFI encryption -No protection against jamming attack -No protection against GPS spoofing attack -Weak access control for system	No new threats	Based on accident scenarios that could exist with a specific Hazard. More safety related.
Attack tree	The attack tree method have shown to provide better risk management of discovered threats. could also be new discovered threats. For example, new threat of loss of information and should be added to the STPA-sec analysis as an unsafe control action.	1 new unsafe control actions	Threat focused. Takes a base in an asset.
BPMN for Threats	In this case the method has the possibility to find threats related the operating the Revolt vessel. What is discovered is the possibility to steal information about how the Revolt is maneuvered and what cargo it has. Doing this in the form of a keylogger install by a malicious operator. UCA: -Invoke service without checking current services running on system & Invoke service without regularly performing security scans on system	2 new unsafe control actions	Could be used to find more threats. But there are options that are more suitable.
Socio-Technical Security modeling language (STS-ml)	The STS-ml method does show promising results. However, in the case with the Revolt case study. The STS-ml method did not find any new threats related to the social aspect. There were no new threats found with this method, but this is case specific. So, this method could be a benefit to combine with the STPA-sec methods in another case study.	No new threats	There were no new threats found with this method.

Based on the results in Table 86, the methods that seems to be most suitable to combine with STPA-sec is FMVEA, Misuse cases, Data Flow Diagram and Attack trees. However, the one that sticks out is the Data flow diagram. With the data flow diagram there was found more exchange of messages, these could be translated to five more control actions and added to the STPA-sec process and going through the STPA-sec process again there can be discovered more security vulnerabilities.

7.3 Security first approach

7.3.1 Classification of Safety and security co-analysis methods

Origin - safety or security

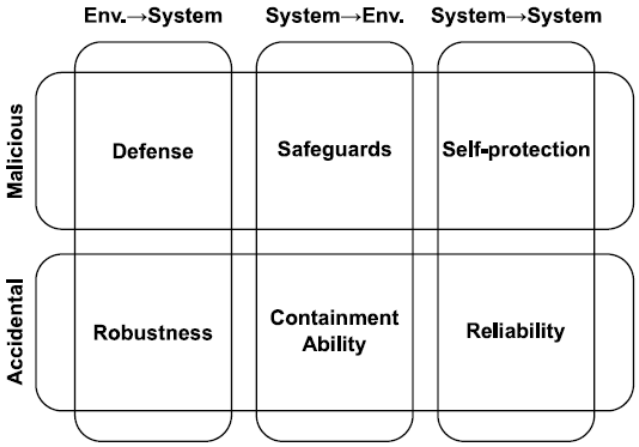


Figure 71 - SEMA referential framework

With the SEMA referential framework as described in Figure 71 [25]. The aim is to avoid ambiguities when using the terms safety and security in different industries. Safety and security does have differences and commonalities dependent on each industry, as already established. However, the main difference regardless of the context of them according to the SEMA framework, is the origin of risk.

When performing a safety analysis, we can discover hazards. The hazards represent how an asset (for example a system) can harm the environment.

When performing a security analysis, we can discover threats. The threats represent how the environment can harm an asset (for example a system).

To be clear, if the security of an asset is compromised, the safety is compromised. For example, if an unwanted person has gained access of a system. This person has control of the system, and the system might affect the environment. This does not go the other way around. The safety of an asset does not affect the security.

The consequences is therefore different from safety and security. The tools and framework is also different, since there is a difference in risk and consequences, and there are different risk management between hazards and threats. Therefore different methods has been used for analysing safety and security of an asset.

This is the basis for a diagram I have created on the next page, to better understand why safety and security methods have been developed, in the way they have been in the recent years. I would say that since there has been focus on safety from an earlier stage than security. Many of the analysis methods today are based on safety, including security methods. They often do not start from looking at the environment (malicious aspect) and the target assets.

How to better understand the current approach

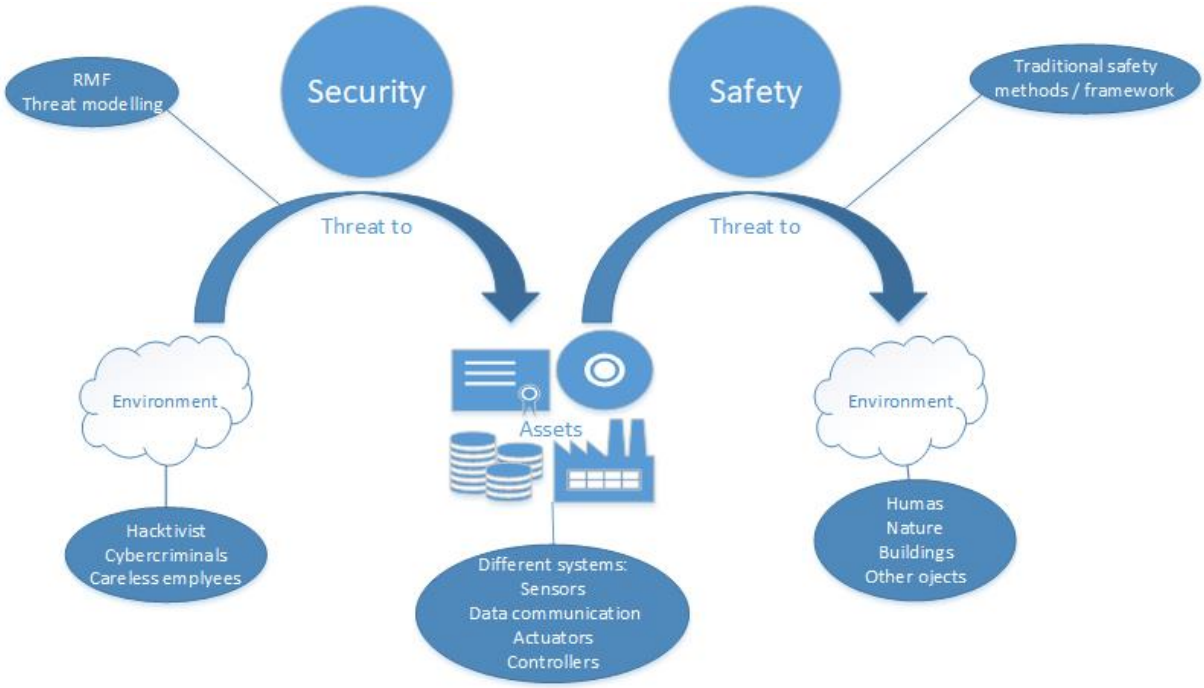


Figure 72 - Differences between taking a base in security or safety when using, designing or combining analysis methods

The figure above shows the differences between taking a base in security or safety when using, designing or combining analysis methods. When considering the security aspect, the environment is a threat to assets. For examples a hacktivist might be e threat to a system including sensors, actuators, controllers and is communication with other systems. To analyze this aspect, the Risk Management Framework (RMF) [80] is used and also methods for threat modelling.

When considering the safety aspect, we are going the opposite route, than with security. The assets could be a threat to the environment. For example, a system, let's say an autonomous car, could be a threat to the driver (humans) and the general environment (nature, buildings and other objects) if a crash occurs. To analyze this aspect, traditionally safety methods and framework are used. For example, different ISO standards and methods like HAZOP.

What approach is traditionally used for safety and security co-analysis methods?

Let's take the example of the STPA. When this method was created it was only intended for safety analysis, it was later extended to include security. The base has been taken in safety, the STPA method, and there has been attempted to find aspects or methods to improve it to include security. Based on the available research in these fields, I would say there has most of the time been taken a base in safety when trying to improve or extend a method to include security.

Could we go the opposite route, and take a base in security – to include security?

This is not an idea that has been thoroughly explored in research recently published in the security and safety field. We could first take a base in the environment that poses a threat to the assets. Then find the safety methods/framework that suits the assets, and already established security methods.

I will now try to find out which method that have taken a base in safety or security. In the specific case to the selected methods that has been previously attempted to be combined with STPA-sec to improve the weaknesses of the methods. I will do this with creating a classification of methods origin, either safety or security.

New approach - Interlink between safety and security

There will usually be different people working with safety and security, even if the methods focus on a safety and security co-analysis. Different experts will be used for safety and security. However, there need to be an interlink between safety and security – some kind of exchange of data, for the analysis cooperative. What I propose that this interlink could be is the data from:

- Attack to the assets and consequences from them

Is this idea original?

Form examining two relevant papers that have discussed and classified safety and security co-analysis methods into categories [3] [25], there has not been taken an approach that I am proposing in this thesis. There is little research that discusses this subject. Therefore, this is recommended to be prioritized for further research.

Unification vs. Integration

What is one of the categorization criteria's when performing the survey in [3] is if the approach is focused around unification or integration. The distinction between unification and integration has been described in [27]. Unification meaning that something being made or become united. Integration meaning that something will be combined with another to form a whole.

With the unification approach there is a unified security–safety framework in place. This framework is based on risk taxonomy, using different standards form both safety and security, and combining them. For example, the term "mishap" has a unified definition for both safety (hazards) and security (threats).

In other words, this approach aims to unify terms related to risks and requirements, and this will be the interlink between safety and security.

The other approach is focused on integration. This approach uses separate risk analysis processes for safety and security. The purpose of using this approach is to determine requirements, by cross-referenced documentation from both disinclines, and identify interactions between them.

In other words, this approach has the aim to have separate risk processes for the target objects and after creating requirements also separately. Then using documentation to find interactions between the requirements and integrate safety in security and vice versa. These interactions of requirements will be the interlink.

To summarize, there has been thoughts on identifying what the different aspects of the safety and security co-analysis methods are. However, the specific interlink has not had much focus. The methods could have focus on having risk or requirements as the interlink, but what are these based upon? Are the specific assets used or are there other factors? Therefore, there need to be another categorization criteria for the safety and security co-analysis methods, for determining where the risks and requirements are generated from in the analysis.

The next table (Table 87) will therefore be created to address this aspect.

7.3.2 Method classification for Safety and security co-analysis methods

Table 87 - Classification of method origin and interlink used for safety or security, for the attempted combined methods

Type	Method / Approach	Safety origin	Security origin	Unification	Integration	Interlink between safety and security
Generic	Stoneburner [81]	N/A	N/A	X		Framework for unifying definitions based on risk taxonomy. The definitions are the interlink
	Aven [82]	N/A	N/A	X		Framework for risk and vulnerability analysis for security and safety. The interlink is risk and vulnerabilities analysis (common terminology)
	Derock [83]	N/A	N/A	X		Based on merging safety and security processes based in two standards. Interlink is development and transferring of requirements
	Woskowski [84]	X		X	X	Based on IEC14971 standard, extended to include security analysis. Risk definition and mitigation is the interlink.
	Eames [85]	N/A	N/A		X	Approach for combining safety and security requirements techniques “MATCS”. Different system models, documentation and results for safety and security. Requirements is the interlink.
	Johnson [86]		X		X	Based in cybersecurity to increasing resilience for safety critical systems. This method does not include capturing safety–security interdependencies.
	Kornecki [87]	N/A	N/A		X	Method include security in a V-shaped development model, originally used in software engineering. There are interactions between safety and security when developing systems. However, most focus on security.
	Novak [88] [89] [90]	N/A	N/A		X	This approach is a lifecycle model for early development phase of systems, the authors argue that it is not enough to consider safety and security interactions with requirements, but the whole lifecycle. The interlink is therefore the different activities in the lifecycle of the system.
	Hunter [91]	N/A	N/A		X	This approach is called Lifecycle Attribute Alignment, and promotes safety and security interactions throughout the whole lifecycle of a system
	Sørby [92]	N/A	N/A		X	A development process for integration safety and security in critical systems. IEC 61508 standard (safety) and CORAS (security) is used.
	Ostby [93]		X		X	Uses what is called Design Basis Threat (DBT) for ensuring safety. Method promotes common understanding if requirements and this is the interlink
	Bieber [94]	N/A	N/A		X	Development of a standard for developing safe and secure embedded systems (Lifecycle approach) standard for aerospace safety where used tighter with traditional security standards.
	Schmittner - FMVEA [60]	X			X	Originates from the FMEA method for safety analysis. Threat modes are introduced. The interlink between is the components of a system, not a complete object.
	SAHARA [95]	X			X	An extension of the safety HARA method, to be security-aware by including the STRIDE approach. Security threats are included to the HARA method using the interlink SecL and ASIL.

Model-based Graphical methods	Goal structuring notation(GSN) [96] [91] [97] [98]	X			X	GSN is a graphical argumentation Notation and can represent any safety or security element (requirements etc.) and the relationship the elements has to each other. The interlink is the notation. Was created to present safety assurance arguments.
	Non-functional requirements (NFR) [99]	N/A	N/A		X	NFR is a goal-oriented approach that was created to evaluate security and safety - if objectives are achieved by a design. Similar to GSN. Enables co-evaluation for safety and security and the "softgoal" is the interlink
Extended fault trees	Fovino - Fault trees(EFT) and Attack trees [77] [100]	X			X	This method is based on Fault trees to explain how a fault in a system can occur, for the safety part. Attacks trees are security based and are integrated with the fault tree. Takes a base in an asset and explains how it might be attacked. The interlink is the development of the combined tree. Fault trees and attack trees are combined with this method. Formal definitions are unified. Events are used for linking security incidents with random failures.
	Bezzateev [101]	X			X	Uses fault trees with integrating a new security module. Safety and security hazards are in the same tree, and development of these is the interlink.
	Kornecki [102]	N/A	N/A		X	The process starts with the asset. Not any interactions until development of safety/security requirements. Requirements is the interlink.
	Steiner - Extended CFT [103]	X			X	This approach uses extended the component fault trees (CFTs) with attack trees to include security. This should be more intended for larger systems. Development of the CFTs to include security concerns that could impact system safety is the interlink.
Informal	Boolean logic Driven Markov Processes - BDMP [104]	X		X	X	Method originates from safety and reliability assessment. Combination of fault trees and Markov processes. Development of the tree is the interlink. Example: accidental (safety leaves) or malicious events (security leaves)
	Bayesian belief network based approaches - BBN [105]	X			X	Originally used for safety assessment. Interlink between safety and security by measuring how they impact each other and the reliability of the system.
	CHASSIS [62]		X		X	CHASSIS is a unified process for safety and security assessment. Originates from use cases. Shows how the environment effect the assets. The interlink is the target object and how it is used. Separate diagrams for safety and security
	Misuse cases [74]		X		X	Interlink is the same as CHASSIS.
	UMLsec/ UMLsafe [106]	X			X	Method being used in early development of critical system design. Has tools for modelling of scenarios, requirements and other relevant aspects, most focus on safety. Requirements is the interlink.
	SysML-Sec [107]		X		X	Method for security assessment of requirements and includes mechanisms for system safety. This method is mostly security focused, and ensures safety by providing a secure system. Limited interactions between safety and security.
	Stochastic Petri nets [108] [109] [110]		X	X		Method for CPS systems. Provides a SPN model for a CPS. Ensures safety by having a secure system. Quantitative analysis for safety and security.

	Model-based system engineering - MBSE [111]	N/A	N/A			Model based approach for analyzing the architecture of a system, viewpoints used for safety and security interactions
	Bow-tie diagram [76]	X			X	Safety based. Based on accident scenarios that could exist with a specific Hazard. The interlink is the Hazards.
	Socio-Technical Security modeling language (STS-ml) [79]		X	X		Security based. Is used to crate security requirements. Based on service-oriented settings in terms of goal-oriented actors. The actors are the interlink.
Formal methods	Zafar [112] - GSE method	N/A	N/A		X	Software engineering method that ensures design by having a set of properties (interlink) that must be fulfilled.
	Approaches for electrical networks [113] [114] [70]	N/A	N/A		X	Methods used for Electrical systems – "CRUTIAL" approach and "PIA". Little interactions between safety and security
	FACT Graph [34]	N/A	N/A	X		The model the FACT graph provides merging safety and security lifecycle phases and these phases are the interlink.
Non-graphical methods Informal	Reichenbach [115]		X		X	Approach uses integrity level (SIL) form IEC 61508 standard to combine safety and security (extended security methods TVRA)
	Holstein [116]	N/A	N/A		X	Description of research by ISA 99. SAL and SIL is used as the interlink between safety and security
	Depoy [117]		X	X		A top-down functional assessment methodology for Risk analysis., it combines risk related to physical and cyber attacks and this is the interlink.
Formal	Pieters - Factor analysis of information risk [118]		X		X	FAIR is framework for safety and security risk assessment that combines frequentist and adversarial approaches. Little interactions between safety and security
	Sun [119] - Maude language	N/A	N/A		X	Sun is a Framework for detecting safety and security conflicts. The conflicts are therefore the framework.
	Simpson [120] CSP/non-interference		X		X	Uses a method "non-interference concept" from security used to model the properties of a system. The properties are the interlink.
	Architecture analysis and design language (AADL) [121]	N/A	N/A		X	ADAL focuses on nonfunctional Aspects. For example, the interactions between components, that could be either safety or security related and this is the interlink.
	STPA and STPA-sec [52]	X			X	The interlink is control actions, because when these control actions are being developed they can be either security or safety related. However, no interactions after that step.
	Unified Security and Safety Risk Assessment [122]	N/A	N/A	X		This is a novel method that has nine steps for unifying a risk assessment for safety and security of a target system. The software and hardware are taken as base for the analysis, and each step involves both security and safety aspects.

Some of the methods in Table 87 are collected from a recent survey about Integrated Safety and Security Risk Assessment Methods, 2017 [123] and used as reference and comparison

for the methods already discussed in this thesis. These methods are SAHARA [95], FACT Graph [34], Extended CFT [103], EFT [100] and Unified Security and Safety Risk Assessment [122]. The rest of the methods are collected from [3].

The table is an adaptation from previous work [3], with the focus on finding out what the interlink is between safety and security are in these methods.

The differences between Unification and integration is described in [3], and has been previously discussed in this chapter. What is interesting to see is what the Interlink between safety and security is, for these methods. And also, what the starting point is, before the exchange of data is occurring.

N/A – meaning that the method/ approach cannot be classified in either safety or security origin. There are two reasons for this, and they are as follows:

- Because of the method has a base in other disciplines/ fields unrelated to safety or security.
- The method is equally based in safety and security. For example, adaption of two standards/framework, one from the safety field and one from security.

New approaches for this classification:

- Interlink between safety and security. I explore the differences between starting from a security vs. safety approach and what the interlink or interaction between safety and security is, in the approach/ method.
- Classification of Safety and security co-analysis methods based on Origin - safety or security and other factors from related studies.

Other general methods that could be used in combination with Safety and security co-analysis methods

Table 88- Other general methods that could be used in combination with Safety and security co-analysis methods

Method	Safety origin	Security origin	Interlink between safety and security
Data Flow Diagram [75]	N/A	N/A	Often used with business analysis. A general method that does not cover either a complete safety or security analysis. Often used as a tool in a larger analysis method.
BPMN for Threats [78]	N/A	N/A	A general method that is used to explain general business processes. Not a base in either safety or security.

The methods in Table 88 are other general methods that could be used in combination with Safety and security co-analysis methods. These methods have previously been attempted to

be combined with safety and security co-analysis methods, but they are separately not capable of covering both aspects.

Differences between starting from a security vs. safety approach

From the methods that has been attempted to be combined with STPA-sec, the one that seems most promising is the Data flow diagram. Because this has a clear interlink between the current approach taken in safety with STPA (creating the control structure of the system) and then adding/improving the security approach with the Data flow diagram.

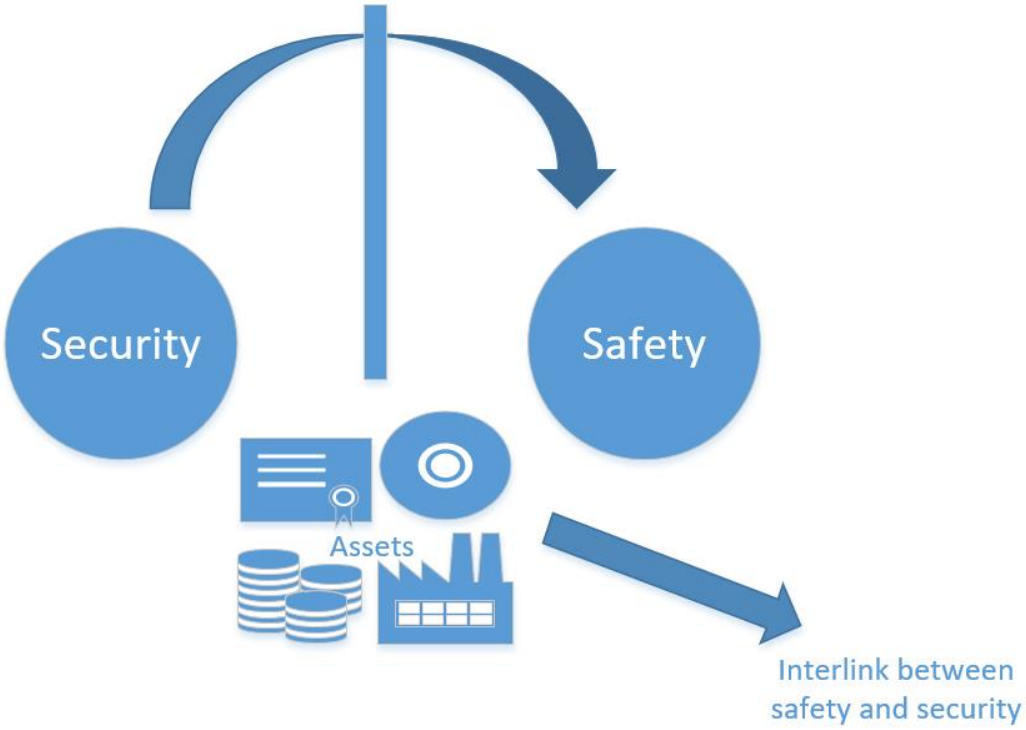


Figure 73 - interlink between safety and security

However, what will be the results if we start with the other point of view? We start with security. This means that I will start with creating a data flow diagram and then go back to STPA.

As already established, I propose that the interlink between safety and security should be

- Attack to the assets and consequence should be the interlink

This will be the interlink in this case also.

7.3.2 Data flow diagram – exploring the idea of starting from a Safety vs. Security approach

Why data flow diagram with STPA-sec?

What the data flow diagram provides has similarities to the control structure used in the STPA-sec process. Therefore, the transition between creating a control structure and data flow diagram will be clear to translate.

The data flow diagram does also address what has shown to be in my opinion the clear weakness with STPA-sec. The STPA-sec does not detect all threats, only those related to the control loop of the target system. The control loop the STPA method provides, does not make it natural to include all the security features as discussed previously in this case study. The data flow diagram includes all communication of data for the target system, not only what is relevant for controlling it. In that way there is a possibility to detect more security vulnerabilities that would otherwise be missed with the existing STPA-sec approach.

With a data flow diagram, I can find attack surface and critical components for the target system and subsystems. This makes it easier to give access control rules.

To summarize I think this approach could give:

- Better discovery of security threats (not only related to the control of system)
- Uncover larger attack surface
- Give clear access control rules for personnel using the system, and also privilege boundaries for requirements
- Better understanding of the whole system, including the subsystems used
- Easy to transition between creating a control structure and data flow diagram. Also, this information will be clear to translate - from data flow actions to control actions.

7.3.2.1 Safety starting point

As previously demonstrated in Figure 74, a data flow diagram based on the control structure of the Revolt system, does detect more aspects with data communication.

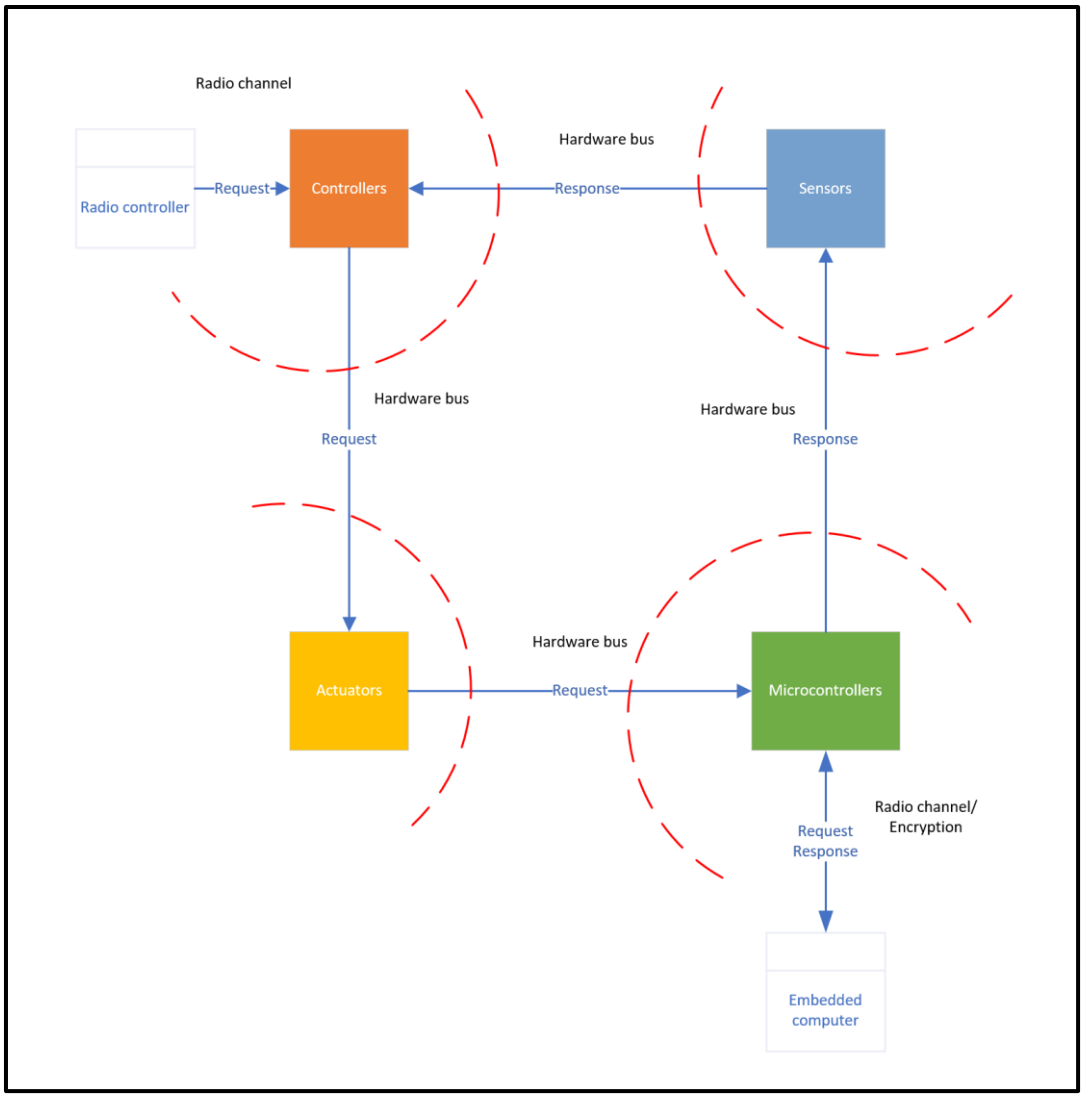


Figure 74 - Example of a Data flow diagram based on Revolt control structure – safety starting point

7.3.2.2 Security starting point

Now I will try to go the opposite route and start from a security stand point.

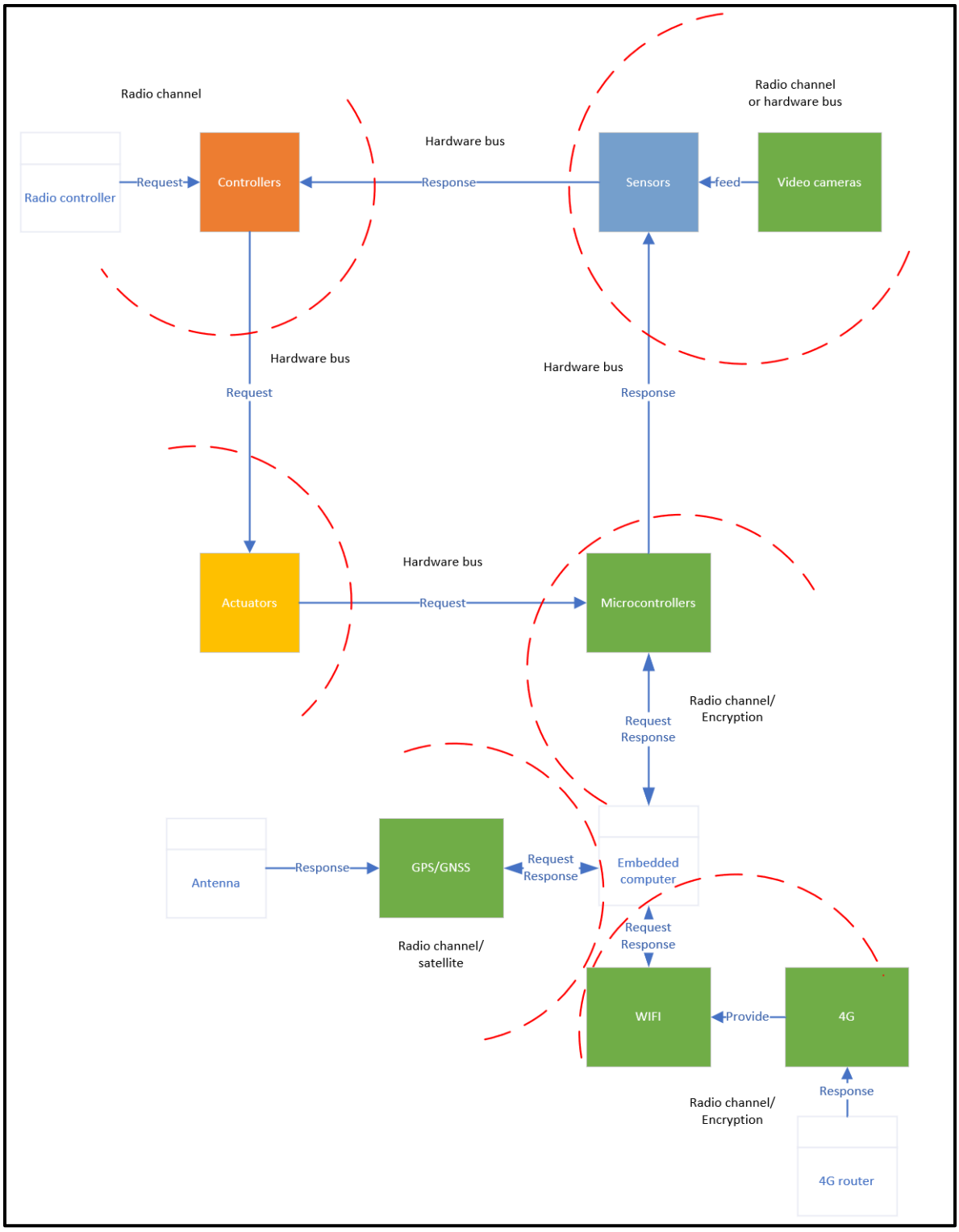


Figure 75 - Example of a Data flow diagram based on Revolt vessel– security starting point

7.3.2.3 Results comparison of starting point Safety vs Security

Safety starting point

Control actions based on data flow diagram (Figure 74). Based on the data flow diagram we have created from the control structure of the Revolt, the following control actions could be added to the STPA-sec analysis:

- Request/receive encrypted message from embedded computer to microcontrollers/ actuators
- Request/receive encrypted message from Remote controller to controllers

Security starting point

Control actions where based on data flow diagram (Figure 75). The following control actions could be added to the STPA analysis:

- Request/receive encrypted message from embedded computer to microcontrollers/ actuators
- Request/receive encrypted message from Remote controller to controllers
- Request/receive encrypted message from GPS/GNSS to embedded computer
- Request/receive encrypted message from WIFI to embedded computer
- Request/receive encrypted message from Video cameras to controllers/microcontrollers

To conclude, with the example of combining a data flow diagram with STPA-sec shown to discover more data flow in which could be used as control actions in the STPA-sec analysis, than the approach which takes base in the existing control structure of the Revolt (in which is safety based).

- **Safety approach: 2 Control actions**
- **Security approach: 5 Control actions**

As an example, there could be vulnerabilities connected to interception of all the new messages found in the data flow diagram, and probably more. These has not been considered in the original STPA-sec process. I would therefore say that there would be a minimum of 5 more detected threats with this approach. The original STPA-sec had 6 threats detected. However, the main purpose of using the data flow diagram simultaneously with creating the control structure, is to have a more complete list of control actions. That will also include security aspects, and not only safety aspects.

I will therefore recommend combining a data flow diagram with the STPA-sec method, for a better security analysis. Taken a base in the target asset (security approach) and not the control structure the STPA-sec process provides (safety approach).

From performing this experiment, the one approach that seems most promising to improve the STPA-sec method is the combination with a data flow diagram. However, taken a base in security and the asset itself seems to discover more security related aspects than taking a base in safety and the existing STPA method, specifically the existing control structure the method provides.

The results of combining the STPA-sec method with the Data flow diagram shows to have advantages.

The data flow diagram does discover security aspects that the control structure from the STPA-sec process is missing. These methods being used together during the creation of the control structure (STPA-sec Step 3. Create functional control structure). In which really is the step where there is a need to get the overview of the target system, is therefore recommend being further researched to discover an implementation that could be permanently used for security critical systems.

Chapter 8 : Conclusions and further work

8.1 Conclusions

The main objective of this thesis was to compare existing methods used for safety and security analysis of autonomous systems, and if possible find ways of improving them. The following are the contributions and conclusion from this thesis.

Contributions

RQ1 - Many security and safety co-analysis methods have been proposed from academia and industry. However, few empirical studies have been performed to compare and evaluate the methods. In this thesis, I have evaluated three methods using an autonomous boat, called Revolt, as a case study. Results of the study show advantages and disadvantages of each method, and have been published in [73]. The further steps are to check validity of the methods used in the case study, based on observing performance and incidents of the Revolt system. However, this demands more physical time being spent with the Revolt. The next research question is focused on extending and strengthen existing methods to analyze safety and security issues of intelligent and complex control actions of autonomous systems.

RQ2 - With the current iteration of methods used in the case study of this thesis, none of the methods are suitable and efficient enough to be used on autonomous systems. However, with the focus on finding methods or approaches to combine with STPA-sec, for an improved method. The results indicate that the STPA-sec method combined with a data flow diagram, taken a base in security, is better suited to cover the security aspect of a system and might be better applicable for autonomous systems.

Conclusion

The approach I have taken in this thesis, is prioritizing security first, in contrast to existing approaches. This approach for safety and security co-analysis are considered to be at an early stage, and not much research is to be found at this time. This makes my research, in my opinion, a high degree of originality in our research field. It is therefore important, to continue the research on methods for safety and security co-analysis methods and find out how it can be used in its best form.

To conclude, this thesis might be relevant for four audiences. The first are manufacturers of various autonomous systems. They can learn about which methods best ensures their own autonomous system. Secondly, the Norwegian government can gain insight into safety and security perspectives when constructing laws about operation of autonomous systems. Thirdly, this could be relevant for researchers working on safety and security methodology. This could be a base to continue the research on methods for safety and security co-analysis and find out how it can be used in its best form. Lastly, although this study was focused on autonomous systems, the research has also been a study of the consequences of safety and security breaches of systems in general. These consequences are therefore relevant to systems used in other fields. Relevant readers of this research can, thus, be researchers in these fields.

8.2 Further work

The following are some ideas for further work related to safety and security on autonomous systems.

8.2.1 Experiences working with an autonomous car

Many thanks to Professor Frank Lindseth at NTNU for valuable knowledge, and for allowing me the opportunity to work on setting up the software for the autonomous car.

I was very fortunate to get the opportunity to work with a relevant project related to autonomous systems. The project I was working on in the spring of 2018, was an autonomous car being developed by NTNU. This is a project lead by Professor Frank Lindseth. This project is a cooperation amongst different manufactures and NTNU.

The main manufacturer is Polysync, in which has provided what is called a Drivekit [124]. The Drivekit will provide a complete by-wire control of a car, in which will control the steering, breaking and throttle. This kit enables anybody that want to, to create an autonomous car on their own, if the car is supported. The car itself chosen for this project was a Kia Niro Hybrid 2018 model [125]. Other sensors like for example LIDAR, radar, cameras and GPS-IMU must be bought and mounted on the car from other manufactures, since Polysync only provides the Drivekit.

Kia Niro Hybrid

Figure 76 - Kia Niro Hybrid

Drivekit

The major hardware component on this platform is what is called a Drivekit, in which is added to the Kia Niro Hybrid (Figure 76), in which aims to make the car autonomous.
In Figure 77, the different components a Drivekit from Polysync includes is displayed [124]. The Drivekit is the brain of the autonomous car, all sensors are connected to the vehicle control module. This module is then either controlled by a laptop, or a joystick controller.

The other sensors installed on the autonomous car is necessary to make the Drivekit function as intended. The major components for sensing the environment around the car are LIDAR, Radar, Cameras and GPS-IMU.

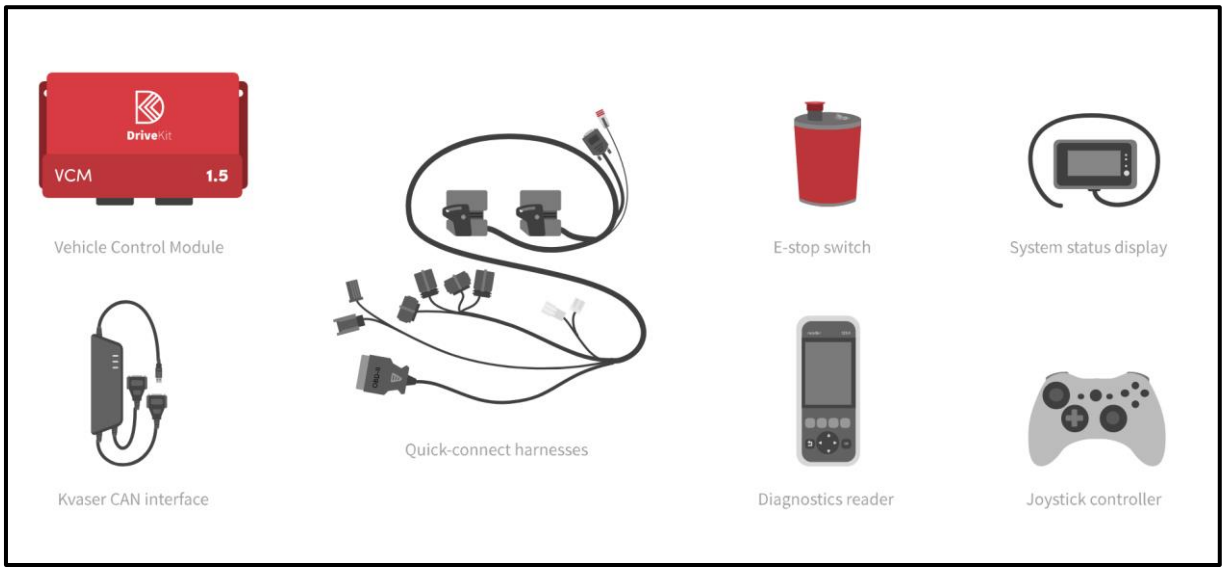


Figure 77 – Drivekit on autonomous car

Software on Kia Laptap

The different software that are used on the various components on the model are listed in Table 89.

Table 89 - Software on Kia Laptop for control of autonomous car

Name	Version	Installed/used on component
Linux Ubuntu	16.04 LTS	Kia Laptop computer
ROS	Melodic Morenia	Kia Laptop computer
Python with Anaconda Cloud	3.6.5	Kia Laptop computer
PolySync software -PolySync Core -PolySync Studio	N/A	Kia Laptop computer Connected to all sensors
SSH	N/A	Remote computer

Stakeholders Kia Niro

Name	Role
PolySync	Provide Drivekit and related software for the autonomous car [124]
Kia Norway	Perform maintenance on the autonomous car Mount components on the car
NTNU – Employees and students	Install software and perform research projects

Figure 78 - Stakeholders Kia Niro

I would recommend that when the autonomous car is finished with getting assembled and is ready for different research projects, this car would be a good case study in another project. Unfortunately, the car was still in assembly stage when I was working on this thesis. However, I gained valuable experiences from installing the software for the autonomous car.

With the autonomous car there is a great possibility to perform experiments with real-world scenarios. The control actions of the car could be used in the same way as with the case study in this thesis. It would be interesting to see if there where comparable results, when testing the FMVEA, STPA and STPA-sec and CHASSIS methods in a case study involving the autonomous car.

Also, what could be e contribution is testing if the proposed method in this thesis – combination of STPA-sec and Data flow diagram will give promising results on the autonomous car, comparing it to the original STPA and STPA-sec co-analysis method.

References

[1] DNVGL, "The ReVolt - A new inspirational ship concept," 2017. [Online]. Available: https://www.dnvgl.com/technology-innovation/revolt/. [Accessed 1 September 2017].

[2] NTNU, " European Safety and Reliability Conference - Welcome to ESREL 2018," [Online]. Available: https://www.ntnu.edu/esrel2018. [Accessed 2018 May 13].

[3] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," *Reliability Engineering & System Safety,* p. 156–178, 17 February 2015.

[4] Autosea, "ReVolt – Demonstration platform," 22 April 2017. [Online]. Available: http://autosea.github.io/misc/2017/04/22/revolt/. [Accessed 7 September 2017].

[5] V. Hahanov and W. Gharibi, "Cloud-Driven Traffic Monitoring and Control Based on Smart," 28 march 2017.

[6] ADDI-DATA, "CPS Cyber Physical Systems," ADDI-DATA, [Online]. Available: http://addi-data.com/cps-cyber-physical-systems/. [Accessed 5 February 2018].

[7] C. STAMFORD, "Gartner," 26 January 2015. [Online]. Available: http://www.gartner.com/newsroom/id/2970017. [Accessed 29 August 2017].

[8] V. Hahanov, S. Chumachenko, L. E.I., D. Farid and D. Sergey, "Intellection Traffic Control on Cloud," in *Parallel and Distributed Computing Systems*, Ukraine, Kharkiv, 2013.

[9] D. P. Watson and D. H. Scheidt, "Autonomous Systems," *Johns H opkins APL Technical Digest, Volume 26, Number 4,* pp. 368-376, 2005.

[10] "Cybernetic Zoo," [Online]. Available: http://cyberneticzoo.com/cyberneticanimals/1961-ferdinand-autonomous-robot-mod-i-jhu-apl-american/. [Accessed 4 September 2017].

[11] N. Lang, M. Rüßmann, J. Chua and X. Doubara, " Making Autonomous Vehicles a Reality: Lessons from Boston and Beyond," bcg, 17 October 2017. [Online]. Available: https://www.bcg.com/publications/2017/automotive-making-autonomous-vehicles-a-reality.aspx. [Accessed 18 January 2018].

[12] A. M. Wyglinski, X. Huang, T. Padir, L. Lai, T. R. Eisenbarth and K. Venkatasubramanian, "Security of Autonomous Systems Employing Embedded Computing and Sensors," *IEEE Micro,* vol. 33, no. 1, pp. 80-86, 2013.

[13] MUNIN, "The Autonomous Ship," 2016. [Online]. Available: http://www.unmanned-ship.org/munin/about/the-autonomus-ship/. [Accessed 14 January 2018].

[14] Waymo. [Online]. Available: https://waymo.com/journey/. [Accessed 6 Septmeber 2017].

[15] S. o. A. E. (SAE), "Automated Driving," [Online]. Available: http://www.sae.org/misc/pdfs/automated_driving.pdf. [Accessed 2017 September 7].

[16] UITP, "International Association of Public Transport," [Online]. Available: http://www.uitp.org/. [Accessed 2017 September 13].

[17] D. Songer, "Rio Tinto breaks new ground with world's first autonomous rail journey," smartrailworld, 9 October 2017. [Online]. Available: https://www.smartrailworld.com/rio-tinto-breaks-new-ground-with-worlds-first-autonomous-rail-journey. [Accessed 18 January 2018].

[18] L. Register, "ShipRight procedure – autonomous ships," July 2016. [Online]. Available: http://info.lr.org/l/12702/2016-07-07/32rrbk. [Accessed 14 September 2017].

[19] D. GL, "The ReVolt," [Online]. Available: https://www.dnvgl.com/technology-innovation/revolt/. [Accessed 7 September 2017].

[20] J. Hartkopf-Mikkelsen, "ShippingWatch," 23 September 2016. [Online]. Available: http://shippingwatch.com/suppliers/article9024890.ece. [Accessed 7 September 2017].

[21] F. Plumet, C. Pêtrès, M.-A. Romero-Ramirez, B. Gas and S.-H. Ieng, "Toward an Autonomous Sailing Boat," *IEEE Journal of Oceanic Engineering,* vol. 40, no. 2, pp. 397-407, 2015.

[22] Saildrone, "Saildrone," [Online]. Available: http://saildrone.com/#About. [Accessed 15 January 2018].

[23] I. National Business Aviation Association, "NBAA Automated Flight Deck," [Online]. Available: https://www.nbaa.org/ops/safety/200010-nbaa-automated-flight-deck-training-guidelines.pdf. [Accessed 14 September 2017].

[24] J. Billington, "ibtimes," 6 December 2016. [Online]. Available: http://www.ibtimes.co.uk/self-flying-pilotless-plane-completes-test-flight-could-be-landing-airports-by-2020-1594987. [Accessed 15 January 2018].

[25] L. Piètre-Cambacédès and C. Chaudet, "The SEMA referential framework : avoiding equivocations on security and safety issues," *International Journal of Critical Infrastructure Protection,,* vol. 3, pp. 55-66, 2010.

[26] L. PiètreCambacédès and C. Chaudet, "The SEMA referential framework: Avoiding ambiguities in the terms “security” and “safety”," 10 June 2010.

[27] D. P. Eames and J. Moffett, "The Integration of Safety and Security Requirements," *Proceedings of the 18th international conference on computer safety,* Vols. SAFECOMP’99, LNCS 1698, p. 468−480, 1999.

[28] "Safety- and Security-Critical Services in Building Automation and Control Systems," *IEEE Transactions on Industrial Electronics,* vol. 57, no. 11, pp. 3614 - 3621, 2010.

[29] L. Pière-Cambacédès, M. Bouissou and C. Chaudet, " Des relations entre sûreté et sécurité (The relationships between safety and security)," 2010.

[30] IEEE, "News & Events: Press Releases," 5 April 2016. [Online]. Available: http://standards.ieee.org/news/2016/ieee_autonomous_systems.html. [Accessed September 14 2017].

[31] A. Burns, J. McDermid and J. Dobson, "On the Meaning of Safety and Security," *The Computer Journal,* vol. 35, no. 1, pp. 3-15, 1992.

[32] M. B. Line, O. Nordland, L. Rostad and I. A. Tondel, "Safety vs. Security?," in *Proceedings of the 8th International Conference on Probabilistic Safety Assessment and Management*, New Orleans, 2006.

[33] D. G. Firesmith, "Common Concepts Underlying Safety, Security, and Survivability Engineering," Vols. Technical Note CMU/SEI-2003-TN-033, 2003.

[34] G. Sabaliauskaite and A. P. Mathur, "Aligning Cyber-Physical System Safety and Security," *Complex Systems Design & Management Asia, LNCS,* pp. 41-53, 2014.

[35] "Online Browsing Platform (OBP)," [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:26262:-1:ed-1:v1:en. [Accessed 30 August 2017].

[36] C.-V. Briciu and I. Filip, "The Challenge of Safety and Security in Automotive Systems," in *9th IEEE International Symposium on Applied Computational Intelligence and Informatics*, 2014.

[37] A. Mallya, V. Pantelic, M. Adedjouma, M. Lawford and A. Wassyng, "Using STPA in an ISO 26262 Compliant Process," *Springer International Publishing Switzerland,* p. 117–129, 2016.

[38] S. Abbott-McCune and L. A. Shay, "Techniques in hacking and simulating a modern automotive controller area network," in *2016 IEEE International Carnahan Conference on Security Technology*, Orlando, FL, USA, 2016.

[39] C. Schmittner, Z. Ma, C. Reyes, O. Dillinger and P. Puschner, "Using SAE J3061 for Automotive Security Requirement Engineering," 29 November 2016.

[40] M. Yoshikawa, Y. Nozaki, T. Asai and K. Asahi, "Frequency Domain aware Power Analysis Attack against Random Clock LSI for Secure Automotive Embedded Systems," *IEEE,* 2015.

[41] T. verge, "Tesla driver killed in crash with Autopilot active, NHTSA investigating," 30 June 2016. [Online]. Available: https://www.theverge.com/2016/6/30/12072408/tesla-autopilot-car-crash-death-autonomous-model-s. [Accessed 13 Desember 2017].

[42] T. Mogg, "digitaltrends," 11 July 2017. [Online]. Available: https://www.digitaltrends.com/cars/driverless-cars-k-city-south-korea/. [Accessed 1 January 2018].

[43] K. Koscher, A. Czeskis, F. Roesner, S. Patel and T. Kohno, "Experimental Security Analysis of a Modern Automobile," in *IEEE Symposium on Security and Privacy*, 2010.

[44] A. Greenberg, "Hackers Remotely Kill a Jeep on the Highway—With Me in It," 21 July 2015. [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/. [Accessed 13 September 2017].

[45] D. Shepardson, "Fiat Chrysler will recall vehicles over hacking worries," 24 July 2015. [Online]. Available: http://www.detroitnews.com/story/business/autos/2015/07/24/us-pushing-guard-vehicle-cyberhacking/30613567/. [Accessed 13 September 2017].

[46] S. Kamkar, "SkyJack," 2 December 2013. [Online]. Available: http://samy.pl/skyjack/. [Accessed 20 September 2017].

[47] J.-S. Pleban, R. Band and R. Creutzburg, "Hacking and securing the AR.Drone 2.0 quadcopter - Investigations for improving the security of a toy," in *SPIE*, 2014.

[48] S. Engineering, "Anti-Spoofing," [Online]. Available: https://gps.stanford.edu/research/current-research/anti-spoofing. [Accessed 21 January 2018].

[49] D. Hambling, "New Scientist," 16 August 2017. [Online]. Available: https://www.newscientist.com/article/mg23531394-300-ships-fooled-in-gps-spoofing-attack-suggest-russian-cyberweapon/. [Accessed 2018 January 21].

[50] C. Schmittner, Z. Ma, E. Schoitsch and T. Gruber, "A case study of FMVEA and CHASSIS as safety and security co-analysis method for automotive cyber-physical systems," in *ACM Workshop on Cyber-Physical System Security*, 2015.

[51] N. G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety, Cambridge: MIT press, 2012.

[52] Y. William and L. Nancy, "Systems Thinking for Safety and Security," *Association for Computing Machinery (ACM),* 22 August 2013.

[53] Y. William and L. Nancy, "An integrated approach to safety and security based on systems theory," *Communications of the ACM,* no. 2, pp. 31-35, 2014.

[54] S. S. Shapiro, "Privacy Risk Analysis Based on System Control Structures: Adapting System-Theoretic Process Analysis for Privacy Engineering," *IEEE Security and Privacy Workshops,* p. 17–24, 2016.

[55] S. Christoph, M. Zhendong and P. Peter, "Limitation and Improvement of STPA-Sec," in *Lecture Notes in Computer Science*, Scopus (Elsevier B.V), 2016, pp. 195-209.

[56] D. Mikalkinas, "Developing an Eclipse plug-in for STPA for security analysis," Stuttgart, 2017.

[57] F. Ivo, M. Kieran, S. Paul, L. David and S. Sakir, "STPA-SafeSec: Safety and security analysis for cyber-physical systems," 30 June 2016.

[58] "XSTAMPP is an open source platform for Safety Engineering," [Online]. Available: http://www.xstampp.de/. [Accessed 2018 May 17].

[59] J. Thomas, "Extending and automating a systems-theoretic hazard analysis for requirements generation and analysis," *Thesis (Ph. D.)--Massachusetts Institute of Technology, Engineering Systems Divisio,* 2013.

[60] C. Schmittner, T. Gruber, P. Puschner and E. Schoitsch, "Security Application of Failure Mode and Effect Analysis (FMEA)," *SAFECOMP : Computer Safety, Reliability, and Security,* pp. 310-325, 2014.

[61] Microsoft, "The STRIDE Threat Model," 2002. [Online]. Available: https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx. [Accessed 22 September 2017].

[62] C. Raspotnig, P. Karpati and V. Katta, "A Combined Process for Elicitation and Analysis of Safety and Security Requirements," *Enterprise, Business-Process and Information Systems Modeling,* vol. 113, pp. 347-361.

[63] W. Y. Jr, "System-Theoretic Process Analysis for Security (STPA-SEC): Cyber Security and STPA," 27 March 2017. [Online]. Available: http://psas.scripts.mit.edu/home/wp-content/uploads/2017/04/STAMP_2017_STPA_SEC_TUTORIAL_as-presented.pdf. [Accessed 2018 May 23].

[64] H. Alfheim and K. Muggerud, "Development of a Dynamic Positioning System for the ReVolt Model Ship," NTNU, Trondheim, 2017.

[65] R. organization, "About ROS," [Online]. Available: http://www.ros.org/about-ros/. [Accessed 7 January 2018].

[66] "IEC 60812: ANALYSIS TECHNIQUES FOR SYSTEM RELIABILITY – PROCEDURE FOR FAILURE MODE AND EFFECTS ANALYSIS (FMEA)," *International Electrotechnical Commission. 4. MIL-P-1629: Procedures for Performing a failure mode, effects and Criticality analysis. Department of Defense (US)..*

[67] "Affairs, US. Department of Veterans. VA National Center for Patient Safety: Healthcare Failure Mode and Effect Analysis (HFMEA)," [Online]. Available: https://www.patientsafety.va.gov/professionals/onthejob/hfmea.asp. [Accessed 27 May 2018].

[68] "Tool Creately," [Online]. Available: https://creately.com/app/#. [Accessed 1 December 2017].

[69] Lucidchart, "Sequence Diagram Tool," [Online]. Available: https://www.lucidchart.com/pages/landing/sequence_diagram_tool?utm_source=google&utm_medium=cpc&utm_campaign=sequence_diagram_tool_broad_norway&km_CPC_CampaignId=227607017&km_CPC_AdGroupID=23538069977&km_CPC_Keyword=sequence%20diagram%20tool&km_CPC_MatchT. [Accessed 1 December 2017].

[70] R. Winther, O.-A. Johnsen and B. A. Gran, "Security Assessments of Safety Critical Systems Using HAZOPs," *Voges (Ed.), Computer Safety, Reliability and Security, vol. 2187, Springer, Berlin, Heidelberg,* pp. 14-24, 2001.

[71] W. Chatham, "White Box, Black Box, and Gray Box Vulnerability Testing: What's the Difference and Why Does It Matter?," 19 January 2018. [Online]. Available: https://codedx.com/2018/01/19/black-white-and-gray-box-vulnerability-testing-code-dx-blog/. [Accessed 29 May 2018].

[72] P. H. Meland, "Threat modeling - Slides for subject TDT4237," SINTEF Digital and NTNU, 6 Mars 2018. [Online]. [Accessed 12 Mars 2018].

[73] E. N. Torkildson, J. Li, S. O. Johnsen and J. A. Glomsrud, "Empirical Studies of Methods for Safety and Security Co-analysis of Autonomous Boat," *European Safety and Reliability Conference 2018.*

[74] G. Sindre, "A Look at Misuse Cases for Safety Concerns," *IFIP — The International Federation for Information Processing book series, IFIPAICT,* vol. 244, 2007.

[75] S. Swidersky, "Threat modeling," *Microsoft Press,* 2004.

[76] K. Bernsmed, C. Frøystad, P. H. Meland, D. A. Nesheim and Ø. J. Rødseth, "Visualizing Cyber Security Risks with Bow-Tie Diagrams," *Computer Science book series, LNCS,* vol. 10744, 2018.

[77] B. Schneier, "Attack trees," *Dr. Dobb's journal : software tools for the professional programmer,* vol. 24, no. 12, pp. 21-21, 1999.

[78] P. H. Meland and E. A. Gjære, "Representing Threats in BPMN 2.0," in *2012 Seventh International Conference on Availability, Reliability and Security*, Trondheim, 2012.

[79] U. o. Trento, "A SOCIAL AND ORGANISATIONAL APPROACH TO SECURITY ENGINEERING," 2014. [Online]. Available: http://www.sts-tool.eu/. [Accessed 5 May 2018].

[80] N. S. P. 800-37, "Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach," 2010. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-37r1.pdf. [Accessed 1 January 2018].

[81] G. Stoneburner, "Toward a unified security–safety model," *Computer,* vol. 39, no. 8, pp. 96-97, 2016.

[82] T. Aven, "A unified framework for risk and vulnerability analysis covering both safety and security," *Reliability Engineering and System Safety,* vol. 92, no. 6, p. 745–754, 2007.

[83] A. Derock, P. Hebrard and F. Vallee, "Convergence of the Latest Standards Addressing Safety and Security for Information Technology," *On-line proceedings of embedded real time software and systems (ERTS2 2010),* 2010.

[84] C. Woskowski, "A pragmatic approach towards safe and secure medical device integration," *Computer Safety, Reliability, and Security (SAFECOMP 2014),* vol. 8666, pp. 342-353.

[85] D. P. Eames and J. Moffett, "The Integration of Safety and Security Requirements," *Computer Safety, Reliability and Security (SAFECOMP 1999),* vol. 1698, pp. 468-480.

[86] C. Johnson, "CyberSafety: CyberSecurity and Safety-Critical Software Engineering," *Achieving Systems Safety,* pp. 85-95, 2012.

[87] A. J. Kornecki and J. Zalewski, "Safety and security in industrial control," *Proceedings of the sixth annual workshop on cyber security and information intelligence research, New York, NY, USA,* p. 77:1–77:4, 2010.

[88] T. Novak and A. Gerstinger, "Safety- and Security-Critical Services in Building Automation and Control Systems," *IEEE Trans Ind Electron,* vol. 57, no. 11, pp. 3614-3621, 2010.

[89] T. Novak, A. Treytl and P. Palensky, "Common approach to functional safety and system security in building automation and control systems," *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007),* pp. 1141-1148, 2007.

[90] T. Novak and A. Treytl, "Functional Safety and System Security in Automation Systems –," *Proceedings of the IEEE international conference on emerging technologies and factory automation,* p. 311–318., 2008.

[91] H. B, "Integrating safety and security into the system lifecycle," *Improving systems and software engineering conference (ISSEC),* p. 147, 2009.

[92] K. Sørby, "Relationship between security and safety in a security–safety critical system: Safety consequences of security threats," M.Sc .thesis, NTNU, Trondheim, Norway, 2003.

[93] S. M. PA Ostby, "Topical report on security and safety integration. Energy facility contractors group, report prepared for the safety and security interface technology initiative," 2006.

[94] P. Bieber and L. Leonardon, "Security and Safety Assurance for Aerospace Embedded Systems," *Proceedings of the 6th international conference on embedded real time software and systems (ERTS2 2012),* pp. 1-3.

[95] G. Macher, A. Höller, H. Sporer, E. Armengaud and C. Kreiner, "A Combined Safety-Hazards and Security-Threat Analysis Method for Automotive Systems," *SAFECOMP 2015 Workshops, LNCS,* vol. 9338, p. 237–250, 2015.

[96] T. Cockram and S. Lautieri, "Combining security and safety principles," *Proceedings of the 2nd institution of engineering and technology international conference on system safety,* p. 159–64, 2007.

[97] S. Lautieri, D. Cooper and D. Jackson, "SafSec: Commonalities Between Safety and Security Assurance," *Constituents of Modern System-safety Thinking ,* pp. 65-75, 2005.

[98] C. W. Johnson, " Using assurance cases and Boolean logic driven Markov processes to formalise cyber security concerns for safety–critical interaction with global navigation satellite systems," *Electronic Communications of the EASST 45,* 2011.

[99] N. Subramanian and J. Zalewski, "Assessment of Safety and Security of System Architectures for Cyberphysical Systems," *Proceedings of the IEEE international systems conference (SysCon),* p. 634–41, 2013.

[100] I. N. Fovino, M. Masera and A. D. Cian, "Integrating cyber attacks within fault trees," *Reliability Engineering & System Safety,* vol. 93, no. 9, pp. 1394-1402, 2009.

[101] S. Bezzateev, N. Voloshina and P. Sankin, "Joint Safety and Security Analysis for," in *roceedings of the 13th conference of FRUCT (Finnish–Russian University Cooperation in Telecommunications) association*, 2013.

[102] A. J. Kornecki and M. Liu, "Fault Tree Analysis for Safety/Security Verification in Aviation Software," *Electronics,* vol. 2, no. 1, pp. 41-56, 2013.

[103] M. Steiner and P. Liggesmeyer, "Combination of Safety and Security Analysis - Finding Security Problems That Threaten The Safety of a System," *SAFECOMP 2013 - Workshop DECS (ERCIM/EWICS Workshop on Dependable Embedded and Cyber-physical Systems),* p. NA, 2013.

[104] L. Piètre-Cambacédès, Y. Deflesselle and M. Bouissou, "Security modeling with BDMP: From theory to implementation," in *Proceedings of the conference on network and information systems security (SAR-SSI)*, 2011.

[105] A. J. Kornecki, N. Subramanian and J. Zalewski, "Studying interrelationships of safety and security for software assurance in cyber-physical systems: approach based on Bayesian belief networks," in *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems pp. 1393–1399*, 2013.

[106] J. J¨urjens, "Developing Safety-Critical Systems with UML," *Lecture Notes in Computer Science book series (LNCS),* vol. 2863, pp. 360-372, 2003.

[107] L. Apvrille and Y. Roudier, "Towards the Model-Driven Engineering of Secure yet Safe Embedded Systems," in *Pre-proceedings of the international workshop on graphical models for security*, 2014.

[108] M. Roth and P. Liggesmeyer, "Modeling and Analysis of Safety-Critical Cyber Physical Systems using State/Event Fault Trees," in *Proceedings of Workshop DECS (ERCIM/EWICS workshop on dependable embedded and cyber-physical systems) of the 32nd international conference on computer safety, reliability and security, Toulouse, France*, 2013.

[109] R. Mitchell and I.-R. Chen, "Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems," *IEEE Trans Reliab,* vol. 62, no. 1, pp. 199-2010, 2013.

[110] F. Flammini, U. Gentile, S. Marrone, R. Nardone and V. Vittorini, "A Petri Net Pattern-Oriented Approach for the Design of Physical Protection Systems," in *Computer safety, reliability, and security, Springer International Publishing, Florence, Italy (2014), pp. 230-245*.

[111] J. Brunel, D. Chemouil, L. Rioux, M. Bakkali and F. Vallée, "A Viewpoint-Based Approach for Formal Safety & Security Assessment of System Architectures," in *Proceedings of MoDeVVa*, 2014.

[112] S. Zafar and R. G. Dromey, "Integrating safety and security requirements into design of an embedded system," in *Proceedings of 12th Asia-Pacific software engineering conference, APSEC '05. p. 8*, 2005.

[113] M. Beccuti, G. Franceschinis, S. Donatelli, S. Chiaradonna, F. D. Giandomenico and P. Lollini, "Quantification of Dependencies in Electrical and Information Infrastructures:the CRUTIAL approach∗," in *Proceedings of the fourth international conference on critical infrastructures, CRIS. p.1-8*, 2009.

[114] S. Chiaradonna, F. D. Giandomenico and P. Lollini, "Case Study on Critical Infrastructures: Assessment of Electric Power Systems," *Case study on critical infrastructures: assessment of electric power systems,* pp. 365-390, 2012.

[115] F. Reichenbach, J. Endresen, M. M. R. Chowdhury and J. Rossebø, "A pragmatic Approach on Combined Safety and Security Risk Analysis," in *Proceedings of the IEEE 23rd international symposium on software reliability engineering workshops (ISSREW) p. 239–44*, 2012.

[116] S. B. Holstein DK, "Quantitative security measures for cyber safety and security," in *Proceedings of the ISA safety & security symposium*, 2010.

[117] J. Depoy, J. Phelan, P. Sholander, B. Smith, G. Varnado and G. Wyss, "Risk assessment for physical and cyber attacks on critical infrastructures," *Proceedings of the IEEE military communications conference, MILCOM,* vol. 3, p. 1961–9, 2005.

[118] W. Pieters, Z. Lukszo, D. Hadziosmanovic and J. v. d. Berg, "Reconciling Malicious and Accidental Risk in Cyber Security," *Journal of internet services and information security,* vol. 4, no. 2, pp. 4-26, 2014.

[119] M. Sun, S. Mohan, L. Sha and C. Gunter, "Addressing safety and security contradictions in cyber-physical systems," *Proceedings of the 1st workshop on future directions in cyber-physical systems security (CPSSW'09), Newark, NJ, USA,* 2009.

[120] A. Simpson, J. Woodcock and J. Davies, "Safety through Security," *Proceedings of the 9th international workshop on software specification and design, Washington, DC, USA. p. 18,* 1998.

[121] J. Delange, L. Pautet and P. Feiler, "Validating Safety and Security Requirements for Partitioned Architectures," *F. Kordon, Y. Kermarrec (Eds.), Reliable Software Technologies – Ada-Europe 2009,* pp. 30-43.

[122] Y.-R. Chen, S.-J. Chen, P.-A. Hsiung and I.-H. Chou, "Unified Security and Safety Risk Assessment - A Case Study on Nuclear Power Plant," *Proceedings of the International Conference on Trusted Systems and their Applications (TSA),* pp. 22-28, 2014.

[123] S. Chockalingam, D. Hadˇziosmanovi, W. Pieters, A. Teixeira and P. v. Gelder, "Integrated Safety and Security Risk Assessment: A Survey of Key Characteristics and,"

Faculty of Technology, Policy and Management, Delft University of Technology, The Netherlands, 2017.

[124] Polysync, "Inside the Box.," [Online]. Available: https://drivekit.polysync.io/. [Accessed 4 April 2018].

[125] Kia, "KIA NIRO PLUG-IN HYBRID TEKNISKE DATA," [Online]. Available: http://www.kia.com/no/modeller/niro-phev/tekiske-data/. [Accessed 2018 April 4].