Vehicle Systems Integration: Defining Vetronics Mission-Critical Systems

By Kyriakos Houliotis

A thesis submitted in the fulfilment of the requirements of the University of Brighton for the degree of Doctor of Philosophy

June 2019

Vetronics Research Centre

School of Computing, Engineering and Mathematics

University of Brighton

Declaration

I declare that the research contained in this thesis, unless otherwise formally indicated within the text, is the original work of the author. The thesis has not been previously submitted to this or any other university for a degree and does not incorporate any material already submitted for a degree.

Signed: 

Dated: 14/06/2019

Abstract

Vetronics (vehicle electronics) systems have evolved for many years to integrate different electronic components and provide reliable services in most applications related to vehicles. Reliable vehicle services are primarily defined within the safety and security domains and are well documented and standardised (e.g. IEC 61508, ISO 26262, RTCA DO-178B and ISO/IEC 27001) within critical systems design cycles. However, in defence, systems that are critical to the success of a mission are not defined within the literature nor are there any guidelines in defining criticality in their design or operational capabilities. When it comes to Vetronics, a Mission-Critical system, is a system with much complexity and mixed criticality levels that are a part of the overall platform (military vehicle) offering reliably integrated system capabilities.

Therefore, the aim of this research is to provide a novel framework focusing on a generic approach to mission life-cycle activities for systems consisting of integrated Electrical and/or Electronic and/or Programmable Electronic (E/E/PE) components that are used to execute various objectives (missions). This offers a coherent early de-risking process for developing any mission-related system so as to enhance mission survivability and success.

he research concentrated on three main phases which are as follows,

Phase 1 - Investigation of Mission-Critical systems definitions used in various sectors; analyse and provide a clear Mission-Critical definition for military systems.

Phase 2 - Following Model Driven Architecture (MDA) design, a novel data model approach is used to describe a generic architecture for a Mission-Critical system, its data and components inter-relationships.

Phase 3 - Finally, as the framework proof-of-concept, a Defense Aid Suite (DAS) system is used as a case study to apply the proposed processes. Early de-risking modelling and techniques are used to model and simulate the Mission-Critical system so as to obtain qualitative and quantitative results.

The main objective of this research (novelty) is to provide a clearer definition of Mission-Critical systems to the overall programme's stakeholders and to de-risk any potential integration issues that may occur during the V-Cycle process of the system. Even though many standards and researchers worked on defining Mission-Critical systems, very few were reported for military platforms especially for platforms using the Interoperable Open Architecture (IOA) approach. From this study, it has been concluded that Mission-Critical systems can be defined but it is very challenging due to the fact that a mission cannot be easily described. Despite that, in this research, it has successfully presented that it is possible to define, specify, develop and de-risk Mission-Critical systems' development and integration throughout a framework and with fruitful qualitative and quantitative results.

This framework and can be potentially exploited for Mission-Critical system using the principles of Interoperable Open Architectures (IOA) and related defence standards such as Generic Vehicle Architecture (GVA) DefStan 23-009 and NATO GVA (NGVA) STANAG 4754.

Acknowledgements

My first gratitude words go to Prof. Stipidis Elias, the head of the VRC family. If ever in my life step on a lottery ticket and from that lottery ticket, win a big prize, it will be a lie if I say “I am lucky”, compared to the fact that I met Prof. Elias Stipidis. All of the motivation source for this thesis extracted from Prof. Stipidis’s attributes – thinking, observing, motivating, recommending, sharing his priceless knowledge and mostly his huge amount of patience and belief. At this moment, my current inspiration in life is to reach and even surpass Prof. Stipidis’s accomplishments. I do really thank you for everything.

Next, is Dr Charhalakis Periklis (Peri). Without him, the rationalism behind this thesis could not exist. Rule No.[1]: “Do not argue with Peri”. Rule No.[2]: “Repeat rule No.[1]”. Apparently, there is a misunderstanding in these rules. Periklis is not there to argue but is there to be convinced of what he’s listening throughout a discussion. Is a bit challenging but is worth a try to convince him through discussion. Although, sometimes his personal, (not academic), preferences might not be the best in the world…but thank you, Peri.

I am most grateful to my good friend/landlord/“boss”, Mr. Ian Watts. Mr. Watts, was the reason to meet Prof. Stipidis and Dr. Charchalakis and therefore, achieving what has been achieved. I am very happy that I met you, very grateful for all the support and the faith you had in me managing an entire module with almost 150 engineering students by myself.

A big thank goes to the VRC family, current and ex-members. You guys helped implement this, in your own unique way. I really appreciate your valuable time for listening and supporting me.

And lastly and most honourable mention goes to my family (Christos, Angela, Antonis, Theodoros and Giagia) and to my closest friends (Laura, Andreas, Eleni and Manolis). Without you, this would be impossible to be completed. God bless you all.

Table of Contents

Abbreviations

[n]	Requirement's Number
ADAS	Advanced Driving Assistance System
APDS	Armour-Piercing Discarding Sabor
API	Application Programming Interface
ASIL	Automotive Safety Integrity Level
AT[n]	Action Taken
B[n]	Benefits
BAE	British Aerospace – Macroni Electronic
BMW	Bayerische Motoren Werke
c.	Circa
C2	Command and Control
C2IS	C2 Information Systems
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
CAN	Controller Area Network
CDM	Conceptual Data Model
CM	Computation Module
CNC	Computer Numerical Control
CPA	Consumer Protection Act
CPU	Central Processing Unit
D[n]	Detection
DAS	Defence Aid Suite
DLOD	Defence Lines of Development
DM[n]	Detection Mode
DOD	Department of Defense
DST	Defence Science and Technology
E/E/PE	Electrical, Electronic and Programmable Electronic

EADS	European Aeronautic Defence and Space Company
ECM	Effector Control Module
ECU	Electronic Control Unit
EL_	Effectiveness Level_
ETA	Event Tree Analysis
FACE	Future Airborne Capability Environment
FK	Foreign Key
FMEA	Failure Mode and Effects Analysis
FPGA	Field-Programmable Gate Array
FSA	Functional Safety Assessment
FTA	Fault Tree Analysis
FTT-CAN	Flexible-TTCAN
GmbH	Gesellschaft mit beschränkter Haftung
GPS	Global Positioning System
GVA	Generic Vehicle Architecture
HARLID	High Angular Resolution Laser Irradiance Detector
HAZOP	Hazard and Operability Study
HMI	Human Machine Interface
HUMS	Health and Usage Monitoring
HW	Hardware
i	Sequential Number
IDEF0	Icam DEFinition – Function Modelling
IDEF1X	Integration DEFinition
IE	Information Engineering
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council of Systems Engineering

IOA	Interoperable Open Architecture
IoT	Internet of Things
IPL	Independent Layer of Protection
IR	Infrared
IT	Information Technology
LAN	Local Area Network
LAV	Light Armoured Vehicle
LBR	Reflective Object Sensor
LDM	Land Data Model
LDM	Logical Data Model
LIN	Local Interconnect Network
LWR	Laser Warning Receiver
M[n]	Mission
MAC	Mission Assurance Category
MAW	Missile Approaching Warner
MAWS	Missile Approach Warning System
MBMA	Model-Based Mission Assurance
MBSE	Model-Based System Engineering
MBT	Main Battle Tank
MC	Mission-Critical
MCDM	Mission-Critical Data Model
MCF[n]	Mission-Critical Function
MCS[n]	Mission-Critical System
MDA	Model Driven Architecture
MDE	Model-Driven Engineering
MIL	Mission Integrity Level
MIP	Multilateral Interoperability Programme

MOD	Ministry of Defence
MoE	Measurement of Effectiveness
MoP	Measurement of Performance
MOST	Media Oriented Systems Transport
MP[n]	Mitigation Process
MS[n]	Mission System
MW	Middleware
n	Real/Last Number
NATO	North Atlantic Treaty Organisation
NF	Normal Form
NFOV	Narrow Field of View
O[n]	Occurrence
OEM	Original Equipment Manufacturer
OMG	Object Management Group
OMT	Object Modelling Technique
OOD	Object-Oriented Design
OOST	Object-Oriented Software Technique
P2P	Point-to-Point
PDM	Physical Data Model
PFD	Probability of Failure on Demand
PI[n]	Potential Impact
PIM	Platform Independent Model
PK	Primary Key
PLC	Programmable Logic Controller
PSI	Platform Specific Implementation
PSM	Platform Specific Model
QFD	Quality Function Deployment

QM	Quality Management
QoS	Quality of Service
R[n]	Responsibility
RAM	Random Access Memory
Req	Requirement
RF	Radio Frequency
RPG	Rocket-Propelled Grenade
RPN	Risk Priority Number
RRF	Risk Reduction Factor
RWR	Radar Warning Receiver
SA[n]	System Analysis
SAE	Society of Automotive Engineers
SCOBA	*Not defined
SE	Systems Engineering
SE[n]	Severity
SEP	Systems Engineering Process
SIF	Safety Instrumented Function
SNR	Signal to Noise Ratio
SO[n]	Source
SoSa	System-of-Systems Architecture
SPM	Sensor Processing Module
SRS	Safety Requirements Standard
SW	Software
SysML	Systems Modelling Language
T[n]	Threat
TC	Technical Committee
TD[n]	Target Date

TDMA	Time Division Multiple Access
TETRA	Terrestrial Trunked Radio
TL	Threat Level
TPM	Technical Performance Measurement
TRD	Towed Radar Decoys
TRL	Technology Readiness Level
TS[n]	Threatening System
TTCAN	Time-Triggered CAN
TTEthernet	Time-Triggered Ethernet
TTP	Time-Triggered Protocol
UK	United Kingdom
UML	Unified Modelling Language
US	United States
USL	Universal Systems Language
UV	Ultraviolet
V&V	Verification and Validation
Vetronics	Vehicle Electronics
VICTORY	Vehicle Integration for C4ISR Interoperability
WFOV	Wide Field of View
XML	Extensible Mark-up Language

List of Figures

List of Tables

List of Publications

“An efficient approach to designing mission-critical systems: Case study: Defensive Aid Suite (DAS) systems”

2017 International Conference on Military Technologies (ICMT), May 31 – June 2, 2017, Brno, Czech Republic, pp. 402-409.

“Mission-Critical Systems Design Framework”

Advances in Science, Technology and Engineering Systems Journal, vol. 3, no. 2, pp. 128-137 (2018)

Chapter 1 Introduction

1.1 Research Motivation

Mission is a formal summary of the aims and values of an activity. A mission may be considered successful when the primary objective is accomplished. During a mission, there are other sub-mission elements that enhance the possibility of the success capability of the mission's primary objective. These sub-mission elements are known as Mission-Critical. Mission-critical can be either referred to as,

> "*Any factor that is crucial to the successful completion of an entire activity*" or,
>
> "*Any factor that is vital to the mission of the organisation which attempts it*" [1]

In the military domain, specifically within the land military vehicles, there are some Mission-Critical electronic components used to enhance the success capability of a mission. These electronic components are known as Mission-Critical systems. These systems are considered to be the essential electronic contributors of the core mission. A well-known Mission-Critical system in military applications is the Global Positioning System (GPS) that provides information on positioning, in which then processed for speed and time on the vehicle which is integrated to.

Recently, military procurement agencies established the Systems Engineering (SE) principles on their platforms' electronic architecture design in order to embrace "design aims" such as modularity and openness. With this move, the results from the traditional design approach, such as stove-piped upgrades with integration conflicts, the proliferation of crew controls and displays, lack of exploitation of data and lack of standardisation across the fleet with training and maintenance issues were addressed. The idea behind this approach was to define a generic architecture that requires open implementation standards (Open Architecture), to support cost-effective integration of (sub)-systems on land platforms electronically, electrically and physically.

Data exchange and integration of (sub)-systems are fundamental to achieve the goals of Open Architecture. Commonality and coherence across the data exchange are needed to enable data interoperability between different (sub)-systems. This is accomplished with the aid of the Data Model approach, which represents the data structure definitions and semantics for data interaction between (sub)-systems in land military vehicles. This is what is known today as "message specification".

However, the data models that these architectures describe are falling short in defining Mission-Critical elements. When it comes to Vetronics (Vehicle Electronics), a Mission-Critical system is a system with much complexity and mixed criticality, that is a part of the overall platform (military vehicle) offering integrated system capabilities. Despite that, the level of effectiveness of the Mission-Critical system's capability can be changed from mission to mission.

Other activities have been deployed to assure the required result and derive the best possible outcome from the intended system. Activities for safety, such as the IEC 61508, ISO 26262, RTCA DO-178B and ISO/IEC 27001, are well documented and standardised within system design cycles. Furthermore, mission assurance activities, such as the NASA-STD 8709.22, for the reliability, maintainability, quality assurance and risk management have been employed, to support effective communication between procurement agencies and its contractors. The UK Ministry of Defence (MOD) established the Defence Standard (Def-Stan) 23-009 Generic Vehicle Architecture (GVA) for its military land vehicle's system interoperability and interface unification. However, systems that are critical to the success of a mission are not defined within the literature nor are there any guidelines in defining criticality in their design or operational capabilities.

For these reasons, there is a need for an approach that allows the components from different manufacturers to be integrated and be able to exchange mission-criticality levels in data depending on the mission's requirements. When building a Mission-Critical system, the system designer should have the freedom to the intended use as well as enable integration to any legacy Mission-Critical system or sensor/actuator that exist onboard the vehicle. Additionally, the approach should be easily adaptable from different disciplines for an effective and efficient Mission-Critical system design and to de-risk the development process from the very early stages. Furthermore, the approach should be able to enhance the decision making on the bespoke Mission-Critical system to ensure that the core mission is not compromised.

In this research a novel framework is presented as the approach required from the above sections and is capable of defining, specifying, developing and de-risk Mission-Critical systems' development and integration throughout a framework and with fruitful qualitative and quantitative results.

1.2 Research Challenges

In this section, the challenges of this work will be defined and discussed. Main, the challenge of this research is how accurately the mission can be defined and how to de-risk the

development process of a Mission-Critical system that is going to be integrated into a new or an existing platform. Below, the challenges are analysed further.

1.2.1 To Specify Mission-Critical Systems Effectively and Efficiently

This challenge is referring to the system that is considered as Mission-Critical. Usually, a Mission-Critical system is the system that is the most essential system for operation. However, in this research, a Mission-Critical system can be considered as any system that exists in the mission. Therefore, if any system is considered as Mission-Critical then this result into a complex and a time-driven development process of a platform.

1.2.2 To Describe Mission-Criticality Between Systems Integrated in a Land Military Platform

Currently, the industry and defence are attempting to use the principles of the commercial plug and play approach for their platforms. Meaning that as long as a system is integrated into an interoperable E/E architecture and shares the same vocabulary, in which that will be discussed in Sections 2.6, 2.7 and 2.8, with the rest of the integrated systems the system can contribute to the mission with its capabilities. However, it is challenging for a system that shares the same vocabulary with other systems to declare its Mission-Criticality when different systems use the data for different purposes.

1.2.3 To Estimate a Platform's Mission Success/Impact Prior the Design and Development Phase

During the development process of a Mission-Critical system, many Mission-Critical elements might not be considered and as a result, when the development reach some level of completion, amendments and modifications can be extremely costly. Firstly, if a system or a platform needs amendment and/or modification, the overall process might be affected; and second, if the system is supplied by a Tier 1 or Tier 2 supplier, in both instances the time and budget might be very costly.

1.2.4 To De-risk the Integration Process of a Mission-Critical System

It is important that the development of a Mission-Critical system is developed with consistency. When the development phase of a Mission-Critical system is not well defined, especially from the concept phase, it means that when the system reaches the verification and validation process it is very likely that technical issues may occur. When more issues occur during the verification and validation process, the development will be going backwards and forwards until the system meets its objectives. Therefore, it is necessary to keep consistency during the development from the concept phase to the verification and validation process.

1.3 Research Questions

In this section, research questions will be discussed and analysed. These questions appeared before and during this research. Alternatively, these questions were mostly concerns about whether research has to follow a certain path. Below are some questions asked so that this research meets its objective.

1.3.1 What Would be Best Standard to Follow for Standardising Mission-Critical Systems

There are various standards for various application around the industry and defence. A set of well-known and well-defined standards are for safety. For example, the IEC 61508 is for industrial applications and the ISO 26262 is for the automotive. There are other standards for safety and mission assurance. The main question is, what would be the best standard to follow as a reference standard that covers mission involving interoperable architectures. In this research, various standards were reviewed but the safety standards were the best candidates to be followed.

1.3.2 Who Would Be The Stakeholders and How It Can Be Demonstrated

It is important to understand who would be the people interested in Mission-Critical system development and deployment. The development and deployment of a Mission-Critical system would be interested in multiple stakeholders, such as customers, executives, chief engineers, legislation and so on. That means the level of demonstrating a Mission-Critical system can vary into three points of views.

First, is the verification point of view. Meaning that by using some numerical results will be sufficient for some of the stakeholders to understand what a Mission-Critical system is capable of. The second is the Systems Engineering point of view. This gets into a lower level than the aforementioned point of view, enough to understand the behaviour of the Mission-Critical system. The last one is the validation point of view. Meaning that a testbed simulation demonstrator may be required in real-time to observe how a Mission-Critical system will perform[1].

1.3.3 What Tools Shall be Used For This Research

Another concern in this research was, what would be the best tools to be used that can be low cost but very effective to achieve this research's objectives and propositions. There are various available tools for providing services for different applications. In this research, the

[1] For this research this is allocated as an option and as a future work

tools were used to assist in defining and developing Mission-Critical systems are widely used in the industry and defence.

Tools such as the Failure Mode and Effect Analysis (FMEA) and the Unified Modelling Language (UML) were the main tools used in this research. Those tools are very useful to define systematic techniques and consistently specify processes for many sectors in industry and defence. Additionally, the Node-RED by IBM has proved that can be used to develop and perform a real-time high-level simulation testbed.

1.4 Research Aims and Objectives

The work proposed in this research is to address the aforementioned challenges, questions and missing elements, as discussed and analysed earlier. By reviewing activities within the industry and defence for functional safety and mission assurance, this research proposes a layered approach to achieve Mission-Critical systems definition precisely.

1.4.1 Clearly Define Mission

A framework that aims the combination of definition, specification, threat analysis and mitigation process of mission and its critical elements efficiently and effectively. From this framework, the main objective is to enhance the data specification of mission-related systems and to describe the benefits, risks and mitigation values for Mission-Critical elements of each mission that will be deployed.

1.4.2 Assist to identify Mission and Mission-Critical systems

An approach that its aim is to provide a clear statement between any stakeholders such as engineers and suppliers to pre-defined attributes of Mission and Mission-Critical elements at the early stages of the V-Cycle development process. This approach's objective should be vital enough and specify the necessary information in a formal way such that, multidisciplinary partakers will effectively review and perform to achieve the full life-cycle and functionality of Mission-Critical systems and their critical elements.

1.4.3 Early De-Risk Demonstrator

Present a proof of concept case study that aims the demonstration on how this research's approach can be used to design an existing or non existing Mission-Critical system in military platforms and by extracting the Mission-Critical aspects of the bespoke system using qualitative and quantitive results. The main objective is to provide a fast early de-risking capability in the very early stages of mission-related systems that can be developed and understood by all the stakeholders of the V-Cycle development process.

The following figure, Figure 1-1 demonstrates the standard V-Cycle of a system, and Figure 1-2 depicts the proposal of this research.

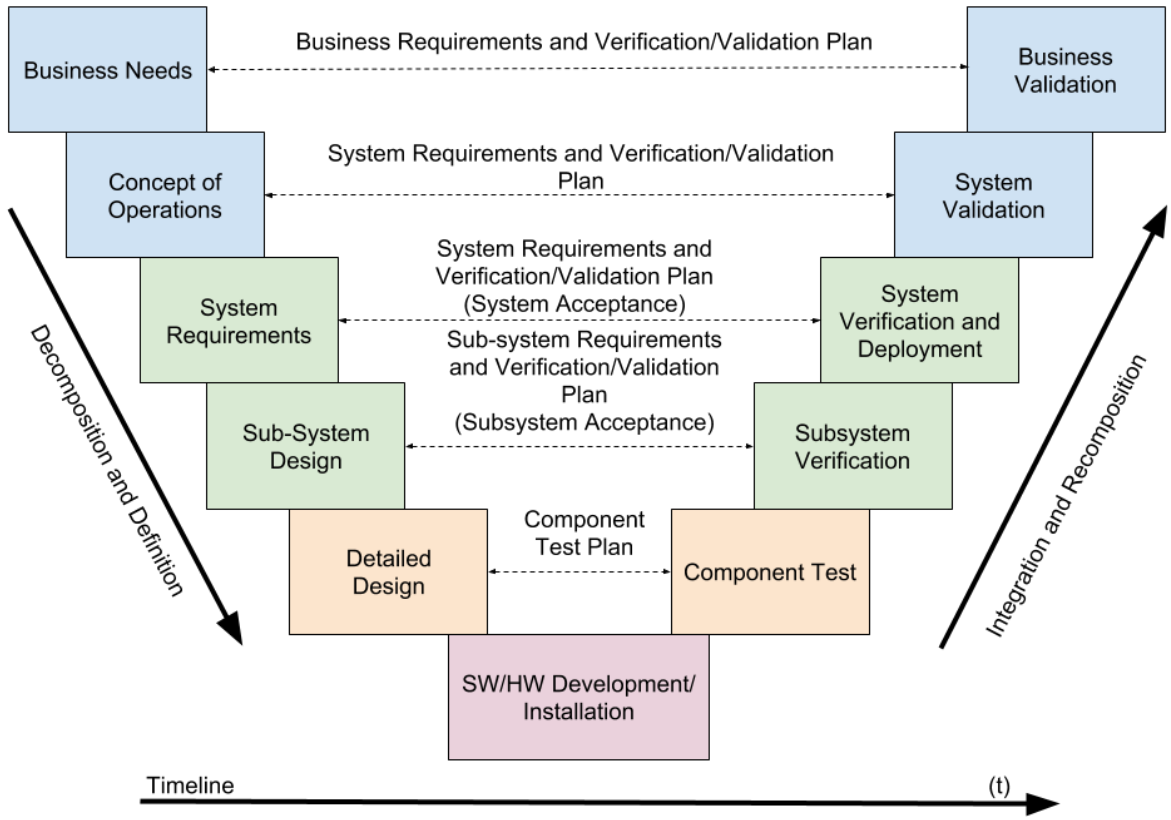

Figure 1-1 Standard V-Cycle

Figure 1-1, is the standard V-Cycle used during the development of a system or programme. The cycle follows a “U” shape process that it starts from the top left and ends up on the top right of the diagram. However, the challenges and research questions of this research can be addressed by the proposed approach. The proposed approach can be simply depicted in Figure 1-2 as shown below.

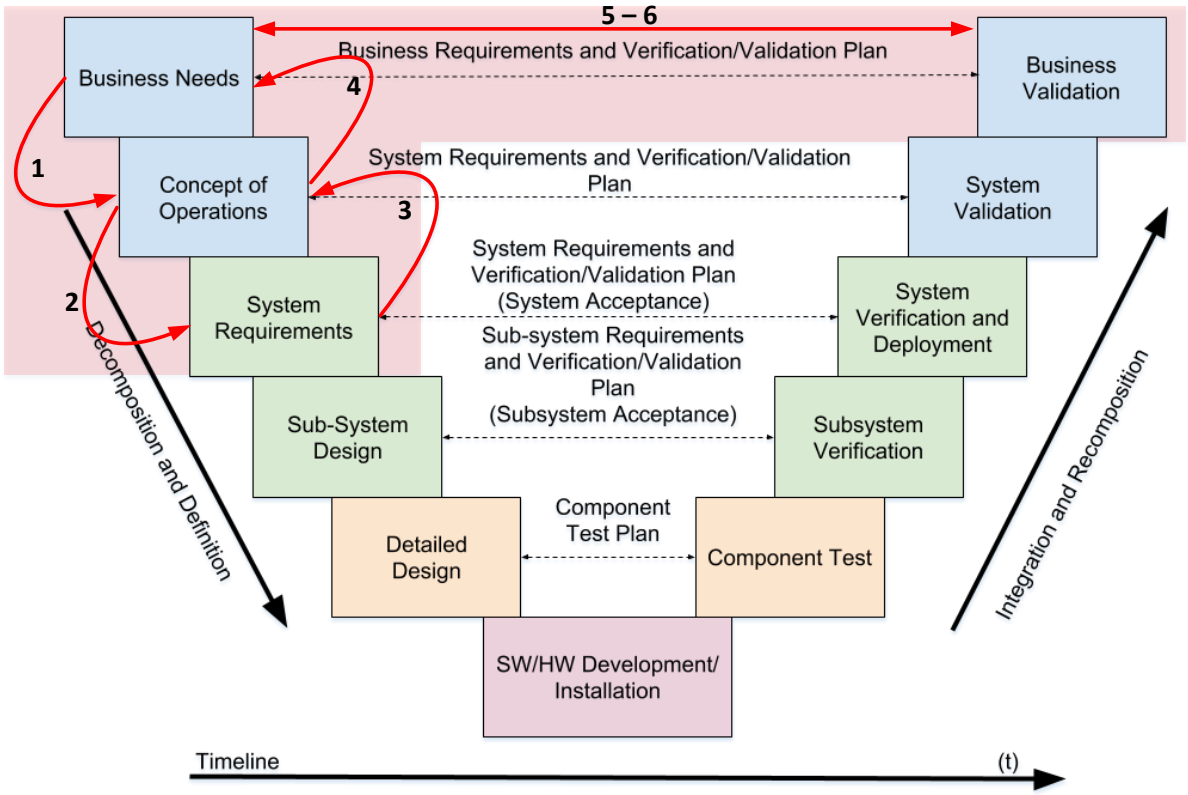

Figure 1-2 Research's Proposal

The research proposal follows the same line as explained in Figure 1-1, but the novelty is that the concept level is well defined, as shown in the red area, and to ensure that; the business gets what requested (steps 5 and 6); the case study is understood by multiple departments involved in the business(steps 1 and 4); and the engineers have a clear and well-defined set of requirements from the very early stage of the system development (steps 2 and 3). A more detailed analysis of steps 1 to 6 will be discussed further in this research. Ultimately, this proposal should satisfy the steps in an efficient and effective manner.

1.5 Thesis Layout

The contents of the thesis are structured such that the research challenges, research question and proposal are presented. The layout of the thesis is as followed,

Chapter 2 presents a historical review of the development and evolvement of automotive electronics in vehicles. It shows that automotive electronics have increased in number, capability, efficiency and effectiveness but in parallel complexity, such as system integration, has also increased. Approaches, such as the Data Model approach is relatively recent and is used to mitigate the complexity of system integration. This chapter presents gaps within the vehicle system integration subject area directly related to mission criticality. One of the main

issue identified, is the virtually non-existent considerations of using data models within critical systems design to define critical operational capabilities.

In Chapter 3, a literature review is conducted collecting fruitful definitions of various Mission-Critical systems. It has been identified that describing a Mission-Critical system is not an easy task; each Mission-Critical system within the literature is characterised differently upon the application's needs. A brief summary of this chapter concludes that critical systems mainly focus on safety or security for humans and environments, protection of data from various threats, whereas systems that focus on survivability prioritise the whole mission envelope. Based on these findings a recommended approach of describing a Mission-Critical system is discussed and presented. However, the key attribute of a Mission-Critical system is to be dependable, where this is inspired by a fundamental study of determining dependable and reliable computing systems.

Moreover, Chapter 3 is looking at currently deployed activities for the design of critical systems around the industry and defence. These activities have a massive content of literature and documentation used to achieve their purposes, i.e. functional-safety. The discussed activities are based on, the concepts of Systems Engineering (SE), the development of documentations, specifications and standards; hazard identification processes; a certification process that declares a system is capable of providing the required functional-safety in its environment; and lastly, how mature systems have to be in order provide the required intended purpose.

Next, a new framework approach on how a Mission-Critical system could be defined and developed, in terms of a standardisation process of mission-related E/E/PE systems, is presented in Chapter 4. Starting from, on how to declare a mission accurately and ending on how valid the procedures are to achieve the optimum design of a Mission-Critical system. The approach is mainly based on, user and system requirements; system's expectations, with calculations as a proof of concept; threat identification with classification and effect levels; a sophisticated procedure for defining and design Mission-Critical systems (in abstract level), along with calculations on how effective the Mission-Critical systems are against the identified threat(s) and how are affecting the mission system(s) and the overall mission. Lastly, a significant step of the proposed framework is to identify the steps' validity within the framework in order to achieve the optimum Mission-Critical system.

Furthermore, in Chapter 5 a review of the Data Model approach is discussed. This chapter explains how the data model is used to abstract and describe complex systems effectively and efficiently. Within the chapter, questions such as, "How the data model can describe Mission-Critical aspects in their deployment" came forth, including considerations on the importance of data, entity types and their inter-relationships. By analysing these questions, a set of

proposed procedures provided potential answers to each question. Additionally, Chapter 5 is looking at various electronic architectures that are currently employed in modern military platforms and E/E architectures that could potentially be used to accommodate the Chapter's 4 framework.

In Chapter 6, a case study demonstrated as a proof-of-concept to verify this research's intention. The Defensive Aid Suite (DAS) system is a well-known complex Mission-Critical system and is used in this case study to evaluate that a Mission-Critical system is challenging to design and develop. The case study combines four main elements associated with the capability of defining the mission, mission system, threat and Mission-Critical system. A brief introduction of these four main elements are, the mission is "survivability", the mission system is a "survivability system", the threat is an "anti-tank penetrator" and the Mission-Critical system is an "obscurant". By taking advantage of the recommendations and proposals of this research the stakeholders, system engineers and architects and any other discipline involve in the design and development of the system, should be able to describe the aforementioned elements with effective and efficient precision.

Lastly, Chapter 7, gives a critical discussion of the research presented in this thesis by distilling and summarising highlights of the major contributions and achievements. Furthermore, the lessons learned during the implementation of this work are analysed, providing basic proposed extensions, limitations and constraints to the current research.

Chapter 2 Background

2.1 Introduction

This chapter will provide a brief background introduction and a historical review of the development of automotive electronics, embedded systems and in-vehicle embedded automotive communication. The chapter will also describe how the first developed transistor revolutionised the automotive industry and how it is developed up to today. Additionally, this chapter will provide an introduction on how the embedded functions were performed from point-to-point stand-alone connections to today's technology.

Furthermore, the motivation of this research will be pointed out during the review of the data models that are used to describe architectures. Reading through this review, it can be realised that the new way of those systems communicate in IOA is by exchanging essential data and it has some Mission-Critical elements missing. However, with observation, it arises concerns such as, how important (Mission-Critical) the data is and how can it be dealt with when it is distributed in other systems with different critical criteria and missions.

Another concern, which is mainly applicable in the military domain, how these systems and sub-systems when exchanging data between them can be able to declare different functional capabilities based on a different mission, effectively and efficiently. Therefore, in this background chapter of this research, all the aforementioned points will be discussed and analysed.

2.2 History

Military technology is often described as the dark side of innovation. Back in time, the ancient Greeks used to believe in the Olympian gods. Hephaestus was amongst these gods. Hephaestus was the god of technology and for this, he was the only god to have been lame and misshapen. From that time, military technology has proved otherwise in which many inventors and innovators think positively about military technology [2].

Today, in the military domain, technology shapes the battlefield and battlefield shapes the technology. Meaning that technological innovations are to be extracted from the battlefield's demands. Almost every land, air and sea defence forces operate with the aid of technology innovations in electronics, and because of that other disciplines such as medicine, businesses and even for personal usage are benefiting.

2.3 The dawn of Vehicle Electronics (Vetronics)

The birth of the first developed transistor in 1947, inspired the automotive industry to introduce electronics in their products. It all started in the 1960s where a radio, an alternator (diode) and a voltage regulator to control the alternator [3] were used. The first standardised embedded system into a production vehicle was introduced in 1973. That was the Electronic Fuel Injection system (Bosch K-Jetronic) which was integrated into a number of automobiles brands such as the Porsche, BMW, Mercedes-Benz, Volkswagen and others.

During that time, Gordon Moore, the co-founder of Intel, predicted that the invention of the first transistor will be used extensively in the foreseeable future.

"*The number of transistors per square inch on an integrated circuit had doubled every year since the integrated circuit was invented"* [4].

This observation was stated in 1965, after the invention of the first developed transistor that was already being used in many other domains. In the automotive domain, the number of embedded systems has increased exponentially, as shown in Figure 2-1. This figure is produced from various related papers, [4], [5], [6] and [7].

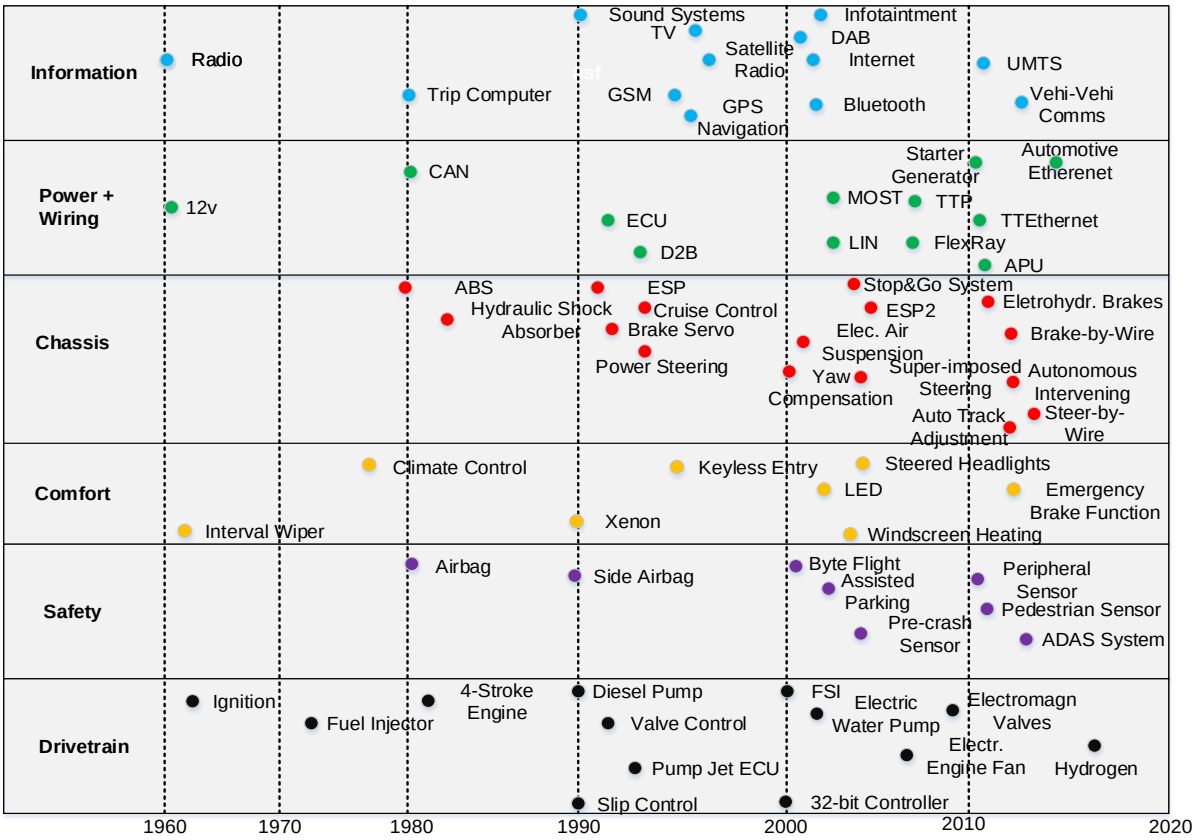

Figure 2-1 Milestones of Automotive Electronics Evolution

In the modern automotive domain, the embedded systems are referred to as Electronic Control Units (ECU). The ECUs are the components that mainly control the data from various systems and sub-systems. As depicted in Figure 2-1, a vehicle may be integrated with multiple systems resulting in the vehicle's electronic architecture to be complex. For example, the first embedded system in vehicles was responsible for one signal, today a modern vehicle has 70 ECUs with 2500 signals [5].

2.4 Automotive Electronics Systems and Sub-Systems

There are three main manufacturing categories involved during the development of a production vehicle; the Original Equipment Manufacturer (OEM), Tier 1 and Tier 2 suppliers. In the automotive industry, the OEM is the company that produces the final product for the consumer marketplace. For instance, the BMW Group is a successful OEM brand that manufactures luxury vehicles. In order to consider a vehicle luxurious, it must be comprised of, comfort, safety, innovational features, engine performance and include high-quality equipment certifications.

A first in-line direct supplier to the OEM is the Tier 1 company. It is the major supplier of parts for the OEM. For example, Robert Bosch GmbH is a Tier 1 company that manufactures automotive electronic systems. Automotive electronic systems such as the stability management system, the electronic safety system and autonomous driving assistance systems. A list of the top 100 global suppliers of OEM parts can be found in [8]. A Tier 2 company supplies a Tier 1 company and have no direct relationship to the OEM company. Tier 2 suppliers provide individual parts (electronic capacitor, plastic etc.) for the implementation and construction of a Tier 1 product.

Figure 2-2 A representation of various supplier systems for a luxury vehicle [9]

The electronics within military land vehicles are referred to as Vetronics (Vehicle Electronics). The electronic systems in the modern and future military vehicles provide capabilities beyond safety and comfort as described earlier in civilian vehicles. In military vehicles, the Vetronics systems provide capabilities such as communication, situational awareness, information intelligence, survivability and many more. A generic definition of a generic system is:

> *"The combination of interacting elements organised to achieve one or more stated purposes"* [10].

Vetronics systems are not stand alone but are fully integrated with other systems. A demonstration of these electronic systems integrated into a platform[2] is depicted in Figure 2-3. The design and management of such systems is referred to as Systems Engineering or System Architecture[3]. The overall operation of the Vetronics system(s) can be accomplished with the aid of sub-systems. A sub-system is the subset of elements of the system that contributes to the overall operation.

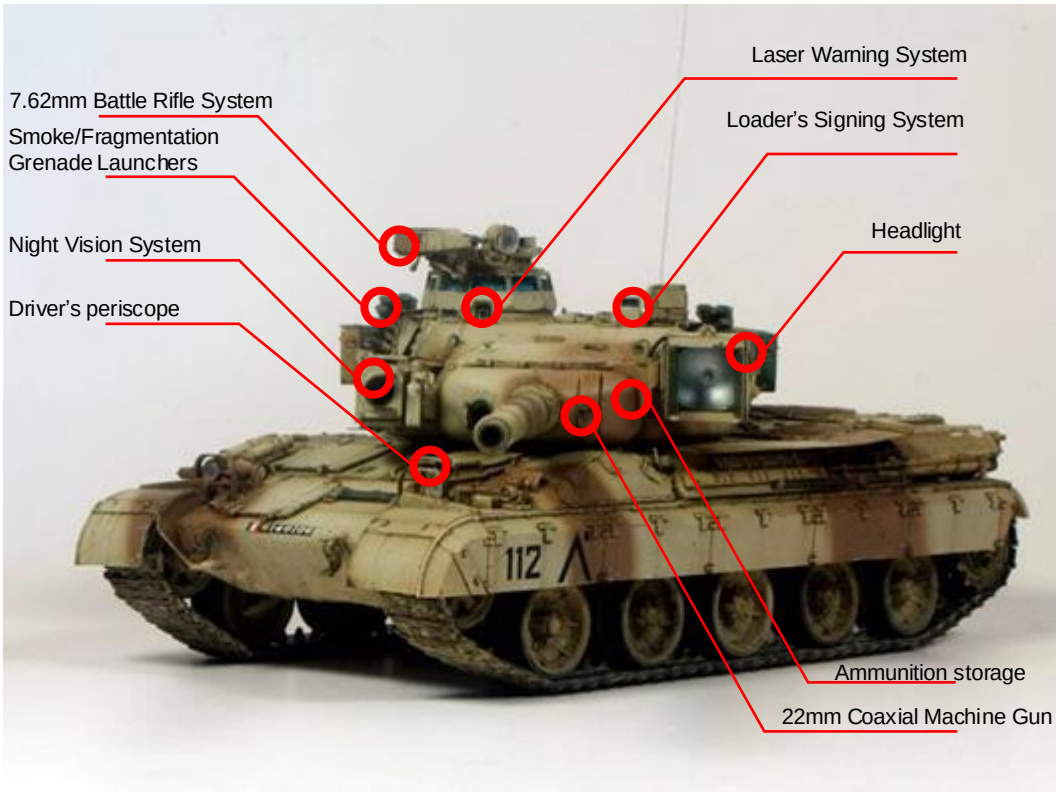

Figure 2-3 Illustration of a modern platform[4]

Within a platform, the integrated network, contains multiple connection points, systems and sub-systems. These are able to receive, store and send data across the electronic

[2] In this study, military vehicle(s) will also be referred as platform(s).
[3] These terms will be discussed in detail further in this study.
[4] AMX-30B2, a main battle tank designed by GIAT.

communication network. Connection points are known also as network nodes, Figure 2-4, and they are divided into three main segments.

- **Application Program** – Is the computer program designed to perform a group of coordinated functions.
- **Processing Capability** – Is the combination of machines, people, software (SW), hardware (HW), the sensor(s) and/or effector(s), that for a set of inputs produces a defined set of outputs. To achieve this, an operating system is installed for the support of application software and management of the resources of the application platform.
- **Communication Interface** – Is the serial interface that allows the transmission of the bit-transfer and exchange of physical and logical information of devices, the topology of the system and so on [11].

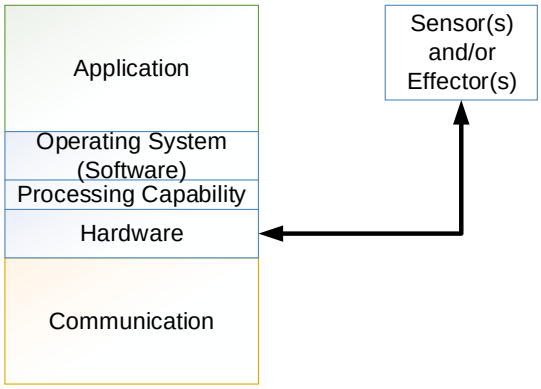

Figure 2-4 Network Node (Generic Sub-System)

Today, a platform consists of multiple integrated network nodes, as shown in Figure 2-5. All these network nodes are interacting with each other through a communication network. A communication network is capable of controlling one or more nodes over a logical or virtual network that are decoupled from the underlining network hardware. This is used to ensure that the network nodes[5] can efficiently be integrated and perform communication.

[5] Gateways are integrated to support system legacy and allow different existing on-board systems to provide and receive services.

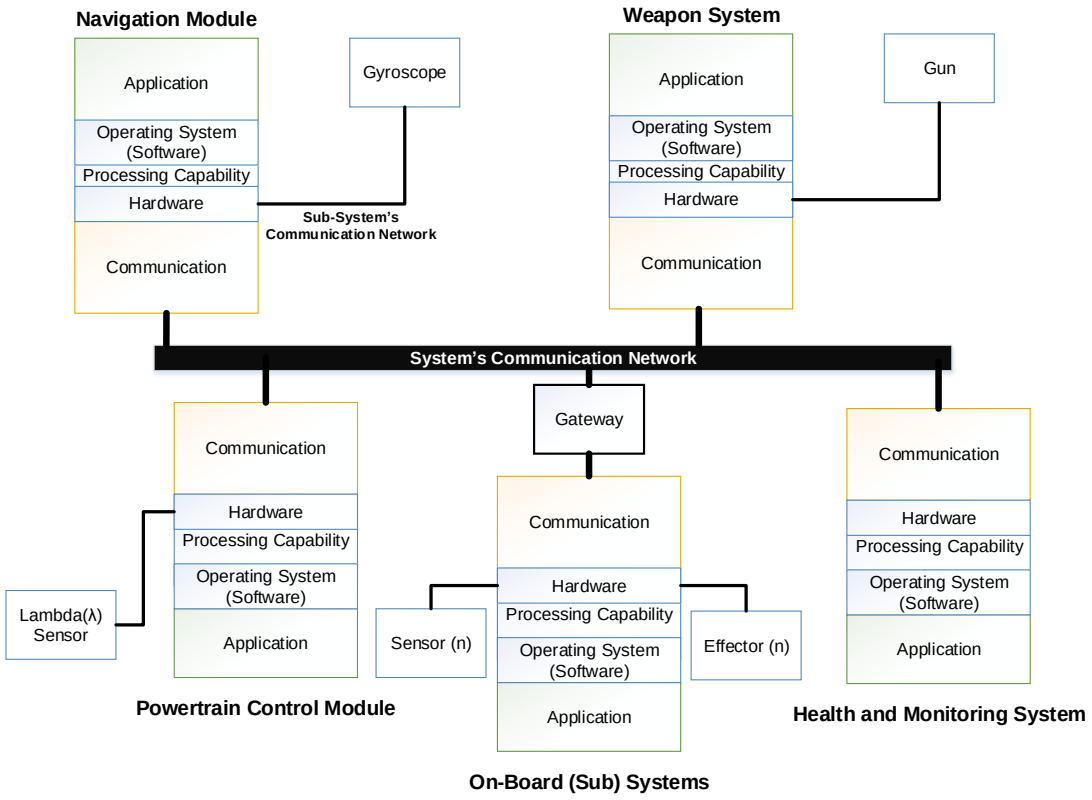

Figure 2-5 Sub-System to Sub-System Communication

2.5 Electronic Communication Networks

With the introduction of the first automotive system in the 1970s, the functions between sensors and actuators were implemented using Point-to-Point (P2P) links. P2P links were used for stand-alone ECUs, field devices and controllers[6], performing only a single function. The need for functions to be distributed over other ECUs and the need for information exchange among functions, quickly proven P2P links to be insufficient due to the constrained capabilities. Hence, an approach to replace P2P links with digital links was needed [12].

This is where the Fieldbus was born. The term Fieldbus is consisted of two other terms; Field and Bus. Field means generally to the apparatus installed in the operating location; and Bus, from the computer science, refers to the signal line to be connected and deliver messages to various devices with the same interface. The data transfer between systems on the Fieldbus was accomplished by a set of rules. The rules were: first, grant bus access and second, to synchronise multi-units on that bus. These sets of rules are called Communication Protocols or just Protocols. In 1998 a publication was released, describing a benefit of using Fieldbus in a vehicle,

[6] Field Devices such as sensors and actuators and controllers such as Programmable Logic Controller (PLC), Computer Numerical Control (CNC).

"…*the replacement of a wiring harness with LANs*[7] *in a four-door BMW reduced the weight by 15 kilogrammes*" [13].

However, in the 1980s the Controller Area Network (CAN) firstly appeared in the automotive market from the Tier 1 company Robert Bosch GmbH. Since then, the CAN network is available and functional until today. Moreover, Fieldbus was developed to such a degree, that each integrated Fieldbus provide complex services and functions. These services and functions were distinguished into different functional vehicle domains, to reflect the different features and constraints. Existing functional vehicle domains are:

- **Powertrain** – Engine control (mainly real-time control and safety of the platform's behaviour).
- **Chassis** – For controlling the chassis components according to the steering/braking solicitations and driving conditions, (X-by-Wire[8] technology is an example used for such applications).
- **Body** – The domain that usually does not require large bandwidth for its functions, i.e. mirrors, climate control, doors, wipers.
- **Telematics** – Unlikely the body domain, this domain requires large bandwidth for the exchange and transfer of large data. For example, Human Machine Interface (HMI) and Global Positioning Systems (GPS).

Moreover, the functional vehicle domains were broken down into sub-functions. These sub-functions had different objectives to accomplish with a pre-defined level of safety. Two main sub-functions were implemented using "event-triggered" and "time-triggered" network communications. The event-triggered protocol executes an action when a significant event occurred. For example, the horn is sounded when the horn button is pressed. This mechanism is very efficient in terms of bandwidth and processor power consumption. However, event-triggered communications have some drawbacks such as constraints on detection of nodes within the network.

The time-triggered protocol is the process that the data is transferred within a periodic time slot. The periodic time slot is pre-defined and is repeated infinitely, assuming that only when is online and function correctly. The pre-defined mechanism is called Time Division Multiple Access (TDMA) thus, the expected messages can be fully predictable and any missing message can be easily identified, [14]. Hence, this also makes the nodes that use a time-triggered protocol to be identifiable when are not functional. This communication protocol can

[7] Local Area Network (LAN).
[8] From the Avionics, X-by-Wire is the term that replaced most of the mechanical or hydraulic systems by electrical and electronic systems.

be inefficient in terms of flexibility and network utilisation. It uses pre-defined message scheduling (i.e. not at run-time) that cannot be compared with the asynchronous message mechanism because of the inefficient response time. Furthermore, when an additional node is integrated into a time-triggered protocol architecture, the overall time scheduling has to change, thus making it also inefficient, [15].

Today there are networks, such as the Time-Triggered CAN (TTCAN), Flexible Time-Triggered CAN (FTT-CAN), Flex Ray and Time-Triggered Ethernet (TTEthernet) that can cover both the aforementioned protocols and additionally, performing different functionalities. For further reading, reference [16] have all the necessary information needed.

The need for dependability and performance in these networks was difficult to accomplish. However, in 1994 the Society of Automotive Engineers (SAE), came across with a solution to mitigate this difficulty. The solution was the creation of four main classification networks that are depicted in Table 2-1.

Table 2-1 Network Classes, Speed, Application and Implementation

	Domain	**Data Rate**	**Networks**
Class A	Body	<10kbit/s	LIN and TTP/A
Class B	Powertrain - Chassis	10kbit/s – 125kbit/s	J1850 and Low-speed CAN
Class C	Powertrain - Chassis	125kbit/s – 1Mbit/s	High-speed CAN
Class D	Telematics	>1Mbit/s	MOST, TTP/C, Flex Ray and Automotive Ethernet

From that point forward, the automotive electronics technology has developed dramatically, Where: the systems engineers and system architects had to take into account some other constraints. The first constraint is, the functions within a vehicle, such as performance, safety and security, are based on software and are tightly coupled. For example, the Advanced Driving Assistance System (ADAS) is a system that is used to enhance safety and provides the vehicle's autonomy by using other systems on the vehicle. Once the system is developed, it is difficult to modify its functionalities and/or integrate new features, [17], therefore, a new life-cycle design will be needed. Second, various suppliers that contribute to the development of a vehicle make difficult the development process of a vehicle to achieved due to different

design interface approaches. In order to make the development process less complicated, two system capabilities should be achieved.

- A flexible electronic architecture that will allow the portability of the components from one system to another.
- The re-usage and the interoperability between the components [18].

The implementation of these system capabilities is accomplished using the middleware (MW) technology.

2.6 Middleware

In general, the middleware serves as a “glue or abstraction layer” between two separate and often already existing components. The communication within the middleware is performed using data, which is packed and unpacked by the application or program and/or service[9]. For any participant that uses a common interface to accommodate middleware technology, a common language is required. This is referred to as “common vocabulary” and can be recognised by all participants. A more specific definition of the common vocabulary in middleware technology is discussed further in Chapter 5.

The middleware’s “layer”, as depicted in Figure 2-6, enables the application to communicate with another application using the principles of the Application Programming Interface (API) by distributing the necessary data using the underlying network as a “Virtual Bus” and hence, abstracting the application from the physical network.

[9] Application or program and/or service will be considered only as application within this section.

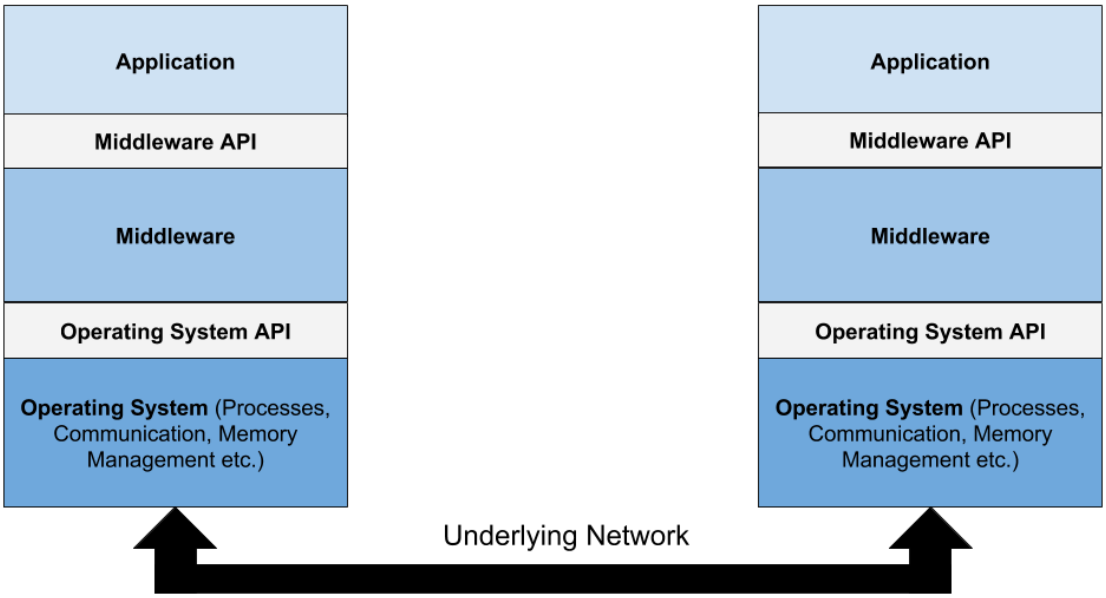

Figure 2-6 Middleware Layer

There are two types of middleware available today which are the Client/Server middleware and Publisher/Subscriber middleware. Client/Server middleware, Figure 2-7, requires a great effort in large E/E networks to accomplish communication between applications and are relatively slow.

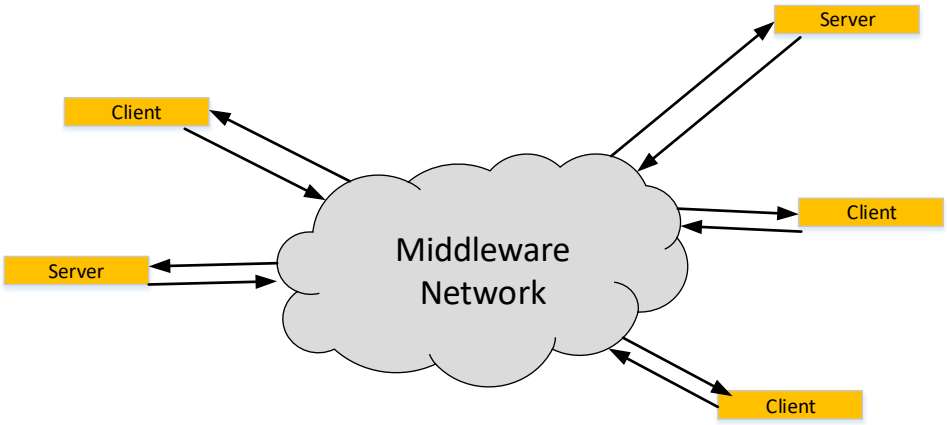

Figure 2-7 Client/Server Middleware

On the other hand, the communication between applications in the publisher/Subscriber middleware, Figure 2-8, is more efficient in terms of achieving initial functions.

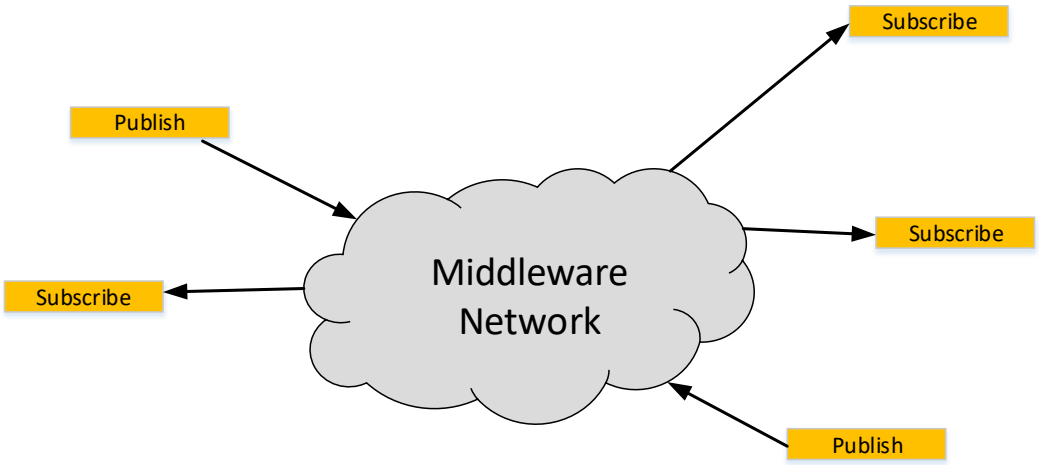

Figure 2-8 Publisher/Subscriber Middleware

The communication between applications in Client/Server middleware is requested by the Client application. The Client's request is typically a data set (information) and it must refer to the Server. The Server will search its database, find the requested data and send it back to the Client, ensuring first that is the right Client requested the update. Whereas in Publisher/Subscriber middleware, the Subscriber is able to get a data update without requesting it. Another interesting feature of the Publisher/Subscriber middleware is regardless of how many applications are subscribed to that data update, all will receive the update almost instantly, as long as they all share a common language. As a result, the E/E architecture of the platform is becoming loosely coupled and easily updated. A paper analysing the performance of this middleware is presented in [19] and for more information on middleware technology, in general, can be found in [20].

Defence procurement agencies realised the benefits of middleware technology and applied it to their military platforms design. Typically, military platforms are in service for many years, hence, updates are essential to quickly respond to new threats and scenarios. Standalone sub-systems on military platforms are using the traditional (pre-middleware era) approach for their design that results into stove-piped upgrades with integration conflicts, a proliferation of crew controls and displays, power conflicts, lack of exploitation of data and lack of standardisation across the fleet with training and maintenance issues. All these issues combined are resulting in a significantly higher cost of ownership through life.

Hence, defence procurement agencies proceeded to the Systems Engineering principles, to define a generic architecture that requires open[10] implementation standards and to support

[10] Open in electronic architectures means an architecture whose specifications are public.

cost-effective integration of sub-systems and that's the beginning of the Interoperable Open Architecture approach.

2.7 Interoperable Open Architecture (IOA) Approach

Today, new or upgraded military platforms are using an architecture that can enable modularity, flexibility, platform portability along with increased data transmission performance and efficient system-of-systems data interoperability, Figure 2-9. The Interoperable Open Architecture (IOA) is a System-of-Systems Architecture (SoSa) based open standards that deliver the ability of sub-systems and applications to perform a given task using a single set of rules, built and procured at different times. It is the ability to exchange services, (provide and accept) between systems, units, or forces, with each other that enables them to operate effectively. Simply, interoperability enables any integrator to connect multiple components developed by different parties and it represents a key objective for defence procurement agencies.

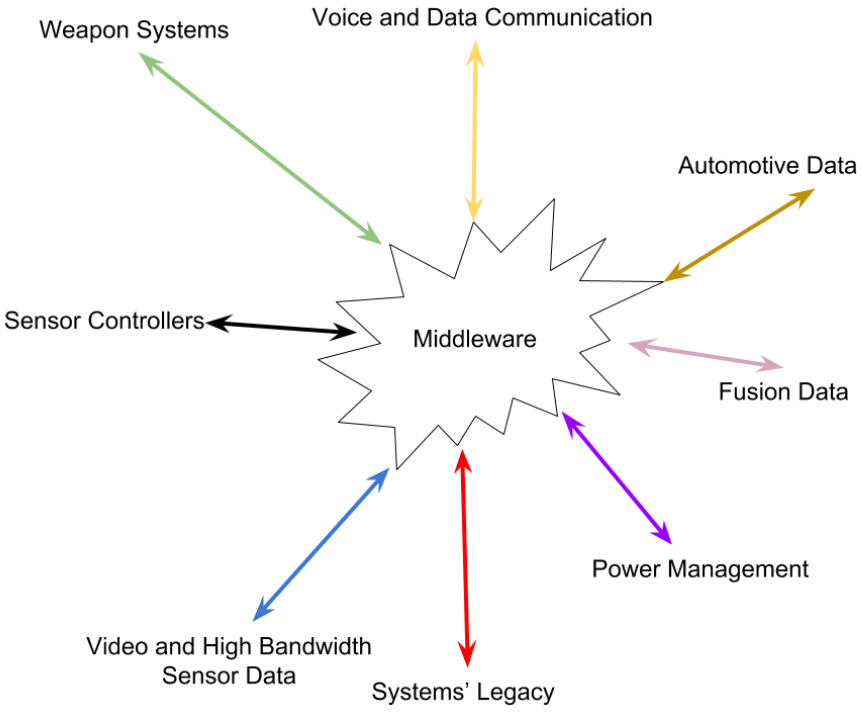

Figure 2-9 Middleware Technology Principle

Today within a modern military platform, land, naval and air forces, have adapted the principles of the IOA in their system design to speed up acquisition and upgrading alongside with reducing life-cycle costs. Below a list of significant international programmes in the area of architectures and standardisation using the IOA approach, is presented.

UK - Generic Vehicle Architecture (GVA)

The Generic Vehicle Architecture (GVA) is an approach taken by the UK Ministry of Defence (MOD) to the development of electronic and power architectures, for land military platforms. The approach is based on establishing systems engineering principles to define a generic architecture that requires open implementation standards, Def-Stan 23-009, to support cost-effective integration of sub-systems on land platforms, electronically, electrically and physically. Any equipment must be designed following the Land Data Model (LDM) which is a sub-system standardisation process [21].

North Atlantic Treaty Organisation (NATO) GVA (NGVA)

The NGVA is an approach to ensure interoperability among military land vehicles equipment. The NGVA follows a similar line to the GVA, incorporating new methods of verification and validation and to mature the NGVA Data Model concepts, focus and implementation [22].

US - Future Airborne Capability Environment (FACE)

The FACE approach is an aviation US government-industry software standard and business strategy for acquisition of affordable software systems. This approach promotes innovation and rapid integration of portable capabilities across global defence programmes. The main objective of this approach is to make military operations more robust, interoperable and secure using open standards [23].

US - Vehicle Integration for C4ISR/EW Interoperability (VICTORY)

VICTORY is a US Army vehicle's open standard for physical and logical interfaces between systems and C4ISR components. The VICTORY's architecture targets to provide a clear picture between user and developer. Throughout the usage of an open standard, the platforms can accept upgrades without a significant impact on the design [24].

Spain - SCOBA Program F-110 Design

A Spanish government programme, (marine) to enable three main capabilities:

- Develop sensor's integration on the mast.
- The incorporation of new capabilities in the SCOBA combat system.
- Control of the missile system to be installed in future frigates F-110[11].

[11] No source available. Source from, author's conference attendance, "Interoperable Open Architecture 2016", Kensington, London, 26-28 April 2016.

Germany - Multilateral Interoperability Programme (MIP)

MIP is an interoperability organisation established by national Command and Control Information Systems (C2IS) systems developers with a requirement to share relevant Command and Control (C2) information in a multinational/coalition environment. As a result of collaboration within the programme, MIP produces a set of specifications which, when implemented by the nations, provide the required interoperability capability. MIP provides a venue for system level interoperability testing of national MIP implementations as well as providing a forum for exchanging information relevant to national implementation and fielding plans to enable synchronisation [25].

Potential Activities

The IOA approach also attracted the point of interest in other international defence procurement agencies. These potential countries are interested to introduce the IOA principles into their platform's electronic architecture design, similarly to the previous programmes, such as the Layered Approach to Service Architecture for a Global Network Environment (LASAGNE) [26]. LASAGNE is an Australian Government Defence Science and Technology (DST) programme which is amongst the countries interested in IOA.

A compulsory procedure for the designing process of a system that uses IOA is the data that flows between other IOA systems must be common and be specified. Hence, a Data Model or System Data Dictionary is required.

2.8 Data Model

Data exchange and integration of sub-systems are fundamental for achieving the goals of the IOA. Commonality and coherence across data exchanges are needed to enable data interoperability. These models represent the data structure definitions and semantics for data interaction between other systems and sub-systems on a platform. Each platform or equipment deployment will implement a subset of the data model collection appropriate to its requirements. Nonetheless, the data models that these programmes describe have some missing elements. These missing elements are encapsulated into a single package and are regarding the “Mission-Criticality” essence in data modelling.

Nevertheless, the encapsulated missing elements are unfolded within the remaining content of this research.

2.9 Conclusion

In this background study, the chapter provided a brief introduction and a historical review of the development of automotive electronics, embedded systems and in-vehicle embedded automotive communication. The chapter analysed how the first developed transistor revolutionised the automotive industry and how has been developed until today.

The chapter gave also an introduction on how the electronic embedded functions were performed through point-to-point stand-alone connections to today's technology. In summary, today's technology is middleware[12], which is based on "glueing" applications and services of existing embedded systems in the vehicle using a common language and common data dictionary. Common language implementation has been described and also how the IOA approach exploited within various international defence programmes.

However, the motivation of this thesis extracted from the fact that the data models that these architectures describe have some missing elements. Reading through this chapter, it can be realised that the new way of those systems communicate in IOA is by exchanging essential data. This raises concerns such as, how important (Mission-Critical) the data is and how can it be dealt with when it is distributed in other systems with different critical criteria and missions. Another concern, which is mainly applicable in the military domain, how these systems and sub-systems when exchanging data between them can be able to declare different functional capabilities based on a different mission, effectively and efficiently.

The next chapter will provide, deeper analysis and discussion providing that Mission-Critical systems are described differently and can lead to misinterpretation and misconception of defining Mission-Critical systems. Furthermore, the next chapter will also provide a review of how systems are developed through their life-cycle using SE principles.

[12] More specific detailed middleware technology approaches will be discussed later in the contribution Chapter 5.

Chapter 3 Defining Mission-Critical Systems

3.1 Introduction

Among many critical electronic systems today, there is a system labelled as Mission-Critical and is considered to be one of the most significant systems integrated on the military platform. Despite that, safety-critical and security-critical systems are also very significant systems on platforms and are well documented and standardised (e.g. IEC 61508 and TRCA DO-178B) within the system design cycle.

However, in defence, systems that are critical to the success of a mission are not well defined within the literature nor are there any guidelines in defining criticality in their design or operational capabilities. When it comes to Vetronics, a Mission-Critical system is a system with much complexity and mixed criticality levels that are a part of the overall platform. A literature review is presented in this chapter to provide awareness that, defining a Mission-Critical system cannot be that easy. Also from this literature review, attributes that could potentially be used to generate guidelines for defining Mission-Critical systems are also presented.

Furthermore, a section of this chapter is looking at currently deployed activities for the design of critical systems around the industry and defence. The section is looking at the concept activities used in Systems Engineering (SE), for the development of documentations, specifications and standards; hazard identification along with recommended hazard mitigation procedures; a certification process that declares a system is capable of providing the required functional-safety in its environment; and how mature systems have to be in order provide the required intended purpose.

3.2 Mission-Critical Definitions

In general, the “mission” is the formal summary of the aims and values of an activity. The activity can be achieved with specific Mission-Critical elements. Those Mission-Critical elements are defined as vital to the activity. Meaning that a successful mission can be achieved when the correct Mission-Critical elements are applied. There are two possible ways to define Mission-Critical elements correctly.

A first possible way is when there is enough maturity on deciding what are the correct Mission-Critical elements. The maturity level must reach a level that is equally understood by all the involved disciplines. When this level is reached, the decision of selecting the correct Mission-Critical elements is likely to occur.

The second possible way is when there is an indication that provides an accurate prediction on how a mission will be performed using the correct Mission-Critical elements. To achieve this, a systematic approach is needed. The approach shall consider all the possible factors that are

involved and interact in the mission. Those factors are usually known thus the prediction can be easily estimated. However, there are also factors that are unknown. Those factors can be anything. Therefore, having unknown factors in missions the estimation is hard to be observed. The prediction can be gained by using three main questions; “What”, “Why” and “How”, (extracted from [27]):

What – Formal stated rules, limits and proscriptions.

Why – To provide specifications and guidance.

How – Set standards, mission statements and operational guidelines.

Once these questions are answered, in a mature manner, the Mission-Critical elements can be considered as a correct decision and can proceed to the development. By also answering these questions, other considerations can be taken which are outside the consideration boundary thus, it helps to rise the bar of the maturity and lowers the bar of the unknown factors.

Today, technology has been developed to such a degree that many missions can be successfully completed with the aid of electronic systems. These electronic systems are referred to as Mission-Critical systems. A generic definition of a Mission-Critical system is,

“A system that is essential to the survival of service, and whose failure or interruption significantly impacts the mission” [28].

A Mission-Critical system in land military platforms is composed of many discrete Vetronics sub-systems and components including sensors, actuators, effectors, processors and controller. Each of these sub-systems may contain further sub-systems and components including mechanical parts.

In Vetronics the mission can be designed, described and/or accomplished either in a simple or complex manner. A simple manner for a Vetronics system is when there are not many factors involved in the mission. It is also simple when a clear and easy step-by-step procedure is provided. For example, a Mission-Critical system has to transmit data from node A to node B. That can be described as a simple mission, since there is only one task to be completed.

However, in Vetronics, for data to be transmitted from one node to another, in reality, is much more complicated than the previous example. What makes the mission more complicated in Vetronics Mission-Critical systems is when more a refined definition is required, especially when multiple disciplines, such as safety, security, survivability, procurement and so on, are involved. For example, a more refined mission definition could be, how critical the transmitted data is from node A and how critical is for node B and so forth. Therefore, the desire for a more refined and detailed Mission-Critical system integration procedure is required. Assume each of

the aforementioned disciplines requires completing a specific goal on the same mission using the same system.

The safety will prioritise the safety of the people and environment; the security will prioritise the protection of data from various threats; the survivability will prioritise the whole mission envelope; procurement will prioritise the costs, and the bureaucracy will prioritise the political associations of the government. These and additionally the desire of a more detailed mission definition from different disciplines, make the missions more challenging to be accomplished. Unexceptionally, if there is not enough maturity or confidence on the applied Mission-Critical elements for a Mission-Critical system life-cycle. Below, there is a literature review generated from authors based on their own Mission-Critical system experiences and opinions. Their experiences and opinions are used and expanded as Mission-Critical systems' main requirements for this research.

3.2.1 Mission-Critical System's Requirements

In critical systems, **dependability**[13], **safety**, **security** and **real-time systems** attributes are common [29]. Below, there are some related works/publications, in which their main focus is on how a Mission-Critical system should be designed and/or be characterised.

The authors in [30], characterised Mission-Critical systems for their **extreme availability** and **Quality of Service**[14] (QoS) features they provide. These characteristics are often related to **time-critical** operations. Main issues of the Mission-Critical systems, are the **cyber-attacks** or any **anomalous behaviours**; thus, Mission-Critical systems must be **robust** and **resilient**. Their approach is to increase the **availability** capability of their Mission-Critical systems using **security-critical communication**.

Ciccozzi et. al. [28], expressed that a Mission-Critical system in the Internet of Things (IoT) must not only be **highly available**, **reliable**, **safe** and **secure** but also should be **scalable** and **serviceable**. Model-Driven Engineering (MDE)[15] approach is selected as a potential candidate to enhance the aforementioned Mission-Critical system's capabilities and to address threats, which are referred to as challenges within the paper.

A study with functional experience of a large scale Mission-Critical system (ERICA), used for emergency services in Finland is described in [31]. The Mission-Critical system for that project is responsible for providing emergency services in no less than 24h a day. That means the

[13] Anything **bold** within this section, is to give emphasise to the findings of the different terms for Mission-Critical systems.
[14] E.g. real-time responsiveness, jitter, error tolerance, bandwidth, redundancy and so forth.
[15] Some of the better known MDE initiatives is the model-driven architecture approach (MDA) by Object Management Group (OMG) and it will be analysed and discussed later in this study.

system shall be **highly available**. The authors also describe for such system in providing these emergency services must reflect also to **qualities**. According to the study, the definition of quality cannot be formed into one meaning but many. The authors suggested some key attributes of Mission-Critical systems which are referred to as qualities.

- **Functional quality of the system** – An understanding of customer needs and system's required features and behaviours.
- **Usability as regarded by real end users** – An unclear definition, yet, important to be understood by the end users.
- **The performance of the system** – A definition of how the system will perform under normal and heavy load operations. This cannot be easily defined due to the absence of proper testing and understanding of the system's attributes – early de-risking.
- **Reliability of the system** – Mission-Critical systems must be highly reliable, by serving different types of **real-time responsiveness**. Another critical mention, apart for Mission-Critical systems to be fully failure-proof, the system must also be **reconfigurable** in order to enable **availability** according to various critical services and applications.
- **Maintainability** – It is recommended that the system is built to accommodate upgrades. Downtime should be also avoided in Mission-Critical systems.
- **Scalability** – The involved parties of the service should be aware by the computing resource availability.
- **Traceability** – It is required for current and future users of Mission-Critical system development. This will allow any user to observe the system's real-time environmental responsiveness. Have the real-time environmental responsiveness observed the user should be able to detect faults and errors effectively.
- **Testability** – The system should autonomously test functionality under stress, long-lasting load and fuzzy tests.
- **Portability** – The system must be portable and adaptable on future trends, scenarios, threats and challenges. If the system does not possess portability and robustness capabilities, it will be then considered insufficient.
- **Supportability** – The system must be able to declare its configuration, how its elements can be rearranged, alongside with its behaviour and performance. With this configuration, the system should be able to be debugged autonomously.

Yet another Mission-Critical system, the Terrestrial Trunked Radio (TETRA) is a European standard for trunked radio systems that are specifically designed to serve emergency and safety network services. In March 2004 when the terrorist attacks took place in Madrid, the medical services operations director stated the following:

"*Our TETRA communication system played a critical role, unlike the cellular network which did not handle the situation due to a communication overload. It was clear to us that we needed a dedicated, secure private communication network in order to deal with life threatening situations. We are now pleased that we made the right decision in 2001 and chose TETRA.*" [32]

TETRA is divided into the following Mission-Critical segments:

- **True Interoperability** – An instant high-level communication between responders and organisations.
- **Critical Networks** - Networks with "**Always Availability**" lifeline, **Security** (encrypted data) and **Reliability** unlikely to the commercial networks for general public purposes.
- **Mission-Critical Data** – **The rapid access** to **vital information**.

It can be noticed that in the above-related articles and documents, the authors described each Mission-Critical systems, according to their experiences and needs. Thus, a Mission-Critical system can be any system that is designed for any task, hence, the definition of a Mission-Critical system is hard to define. In the following section, a study on how to define an electronic system is presented, and based on that study the core definitions for a Mission-Critical system will be stated.

3.2.2 Dependable System Taxonomy

The fruitful definitions and characteristics used for electronic systems was an issue that existed within Systems Engineering for years. The Technical Committee (TC56) standard suggested that the systems should be described and characterised as **dependable** [33]. The standard was developed in 1965 by the International Electrotechnical Commission (IEC) and was entitled: "Reliability of electronic components and equipment". The title then changed to reliability and maintainability of electronic components and equipment and afterwards, in 1989 the title changed to the dependability of electronic components and equipment, by Jean-Claude Laprie [34]. Laprie described a dependable system using the following taxonomy as depicted in Figure 3-1.

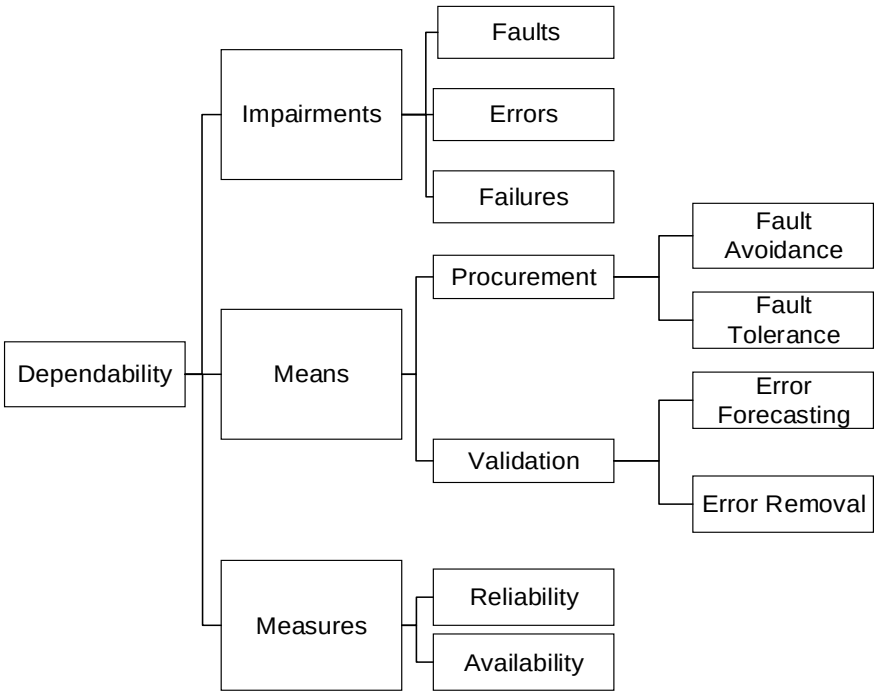

Figure 3-1 Laprie et. al. Fault-Tolerance

According to the paper produced by Laprie in [34], the terms are defined as,

Dependability – The quality of the delivery service such that reliance can justifiably be placed on this service.

- **Impairments** – Are the undesired, not unexpected, circumstances causing or resulting from un-dependability, whose definition is very simply derived from the definition of dependability: reliance cannot, or will not, be any more justifiably placed on the service.
    - **Faults** – Create one or several latent errors in the component where it occurs; physical faults can directly affect the physical layer components only, whereas human-made faults may be also affected.
    - **Errors** – May be stated into **a**) A latent error becomes effective once it is activated; **b**) an error may cycle between its latent and effective state; and **c**) an effective error may, and in general do, propagate from one component to another; by propagating an error creates other (new) errors.
    - **Failures** – Occur when an error affects the service delivered (as a response to the request(s) by the component).
- **Means** – Are the methods, tools and solutions enabling, **a**) to provide with the ability to deliver a service on which reliance can be placed and **b**) to reach confidence in this ability.

- **Procurement** – How to provide the system with the ability to deliver the specified service.
  - **Fault Avoidance** – How to prevent, by construction fault occurrence.
  - **Fault Tolerance** – How to provide, by redundancy service complying with the specification in spite of faults having occurred or occurring.
- **Validation** – How to reach confidence in the system's ability to deliver the specified service.
  - **Error Removal** – How to minimise, by verification, the presence of latent errors.
  - **Error Forecasting** – How to estimate by evaluation, the presence, the creation of the consequences of errors.

- **Measures** – Enables the service quality resulting from the impairments and the means opposing them to be appraised.
  - **Reliability** – A measure of the continuous service accomplishment (or, equivalently, of the time to failure) from a reference initial instant.
  - **Availability** – A measure of the service accomplishment with respect to the alternation of accomplishment and interruption.

Today, most of the previous terms have been replaced by security and safety, where safety pays attention in human lives, economies and environments and security pays attention in the integrity and loss of sensitive data through theft or accidental loss [35]–[37].

3.2.3 Military Vehicle System Integration (VSI): Standards and Guidelines

In the military domain, the Vehicle Systems Integration (VSI) is a programme sponsored by the UK's Ministry of Defence (MOD), that produces standards and technologies originated from the commercial domain and report on how they may be adapted in their platforms. In addition, the main focus of the VSI programme is at the open architecture, in which technologies and standards may be seamlessly applied and integrated to land platforms and on their applications [38].

According to the programme, the operational effectiveness is enhanced when the systems and sub-systems are fully interacted with each other. Within the VSI standard, the systems should be attributed with the following definitions,

- **Common, Layered Network Interface** – Based on the Open Systems Interconnection (OSI) model, systems should be able to accommodate and maintain flexibility in their architecture.
- **Modularity** – Referred as functional and equipment modularity;
    - **Functional Modularity** – The combination of functions based on specific modules and requirements.
    - **Equipment Modularity** – The combination of equipment to form a system based on specific instances and requirements.
- **Scaled Performance** – It is demanded that Vetronics functions are implemented such that to provide the necessary performance/cost trade-offs that satisfy requirements such as **scalability** and **upgradeability**.
- **Distributed Processing** – It is essential that two or more systems can be used to run a single application to satisfy requirements such as;
    - **Fault Avoidance**
        - Avoid single points of failure.
        - Promoting standalone operation of individual devices and whole system segments in reversionary modes.
        - Supporting graceful degradation.
        - Facilitating the reassignment of essential processing to surviving processors in the event of device failure.
    - **Network Efficiency** – Avoid communication bottlenecks into central system processors.
    - **Data and Resource sharing**;
        - A facility of peer to peer interactions.
        - A promotion of using a standardised message set between distributed devices.
    - **Flexibility** – Interaction between electronic systems and devices that are independent of any central controller, so that modifications can be made independently.
- **Real-Time Processing** – The capability of utilising network protocols that support both **hard** and **soft real-time** data transfers.
- **Network Protocol Overheads** – Minimising network protocol overheads is yet another system's requirements for land military platforms. In order to minimise network protocol overheads, **rapid start-up**, **fault recovery** and **on-line automated network management** activities should be considered.

- **Diagnostics** – Fault diagnostics are required, to detect and report network equipment errors, communication errors and local equipment failures to its controlling application software.
- **Fault Tolerance**
    - The accommodation of reversionary modes of operation,
        - The utilisation of protocols that support the dynamic re-allocation of platform resources to essential functions, based on priority i.e. graceful degradation.
        - Stand-alone function operation, i.e. platform functions shall use reversionary parameter values, modes of operation and safe states to continue operation in the even of system equipment failure.
        - Distributed processing.
    - Redundant resources where the criticality of the function is demanded,
        - The utilisation of network protocols that are based on masterless network operation.
        - The utilisation of protocols that support the dynamic redistribution of responsibilities and processes of vetronics functions in the event of equipment failure.
        - n-level network media redundancy where appropriate.
- **Architecture Partitioning –** It is demanded that the technology employed on a platform should match in both, cost and performance requirements to the function performed by the particular equipment. For example, the imposition of over specified network technology on platform installed equipment in order to satisfy the highest data communication performance requirement should be avoided.
- **Power** – This requirement is applicable for intelligent platform power management to selectively isolate power from low-priority equipment in favour of high priority as required to conserve power.

3.2.4 Mission-Critical System Taxonomy (Proposal)

In this thesis, a Mission-Critical system is described based mainly on Laprie's taxonomy. Additionally, more terms are included that are potentially better to reflect the technological evolution, business and military needs on a broader scope of applications, based on the concept of Mission-Critical systems. The Mission-Critical system for this thesis is as depicted in Figure 3-2.

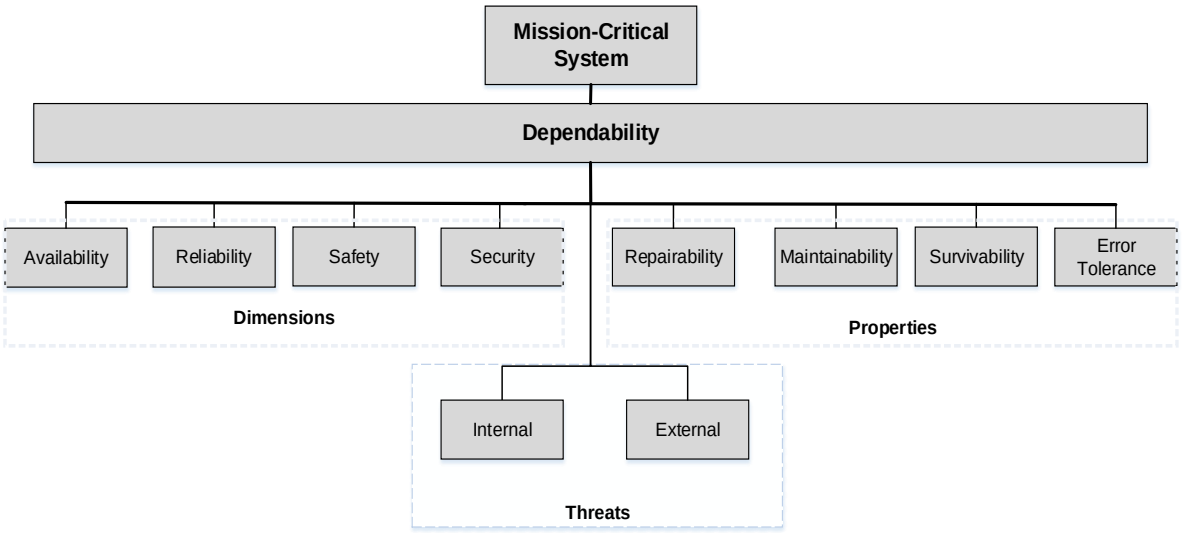

Figure 3-2 Mission-Critical System Taxonomy

Mission-Critical System - A system that is essential to the survival of service, and whose failure or interruption significantly impacts the mission. A Mission-Critical system should at least consist of the following proposed characteristics.

- **Dependability** - Trustworthiness of an E/E/PE system such that reliance can be justifiably placed on the service it delivers.
    - **Dimensions** – The maximum functional capability.
        - **Availability** – The ability of the system to deliver services when requested.
        - **Reliability** – The ability of the system to deliver services as specified.
        - **Safety** - The ability of the system to operate without catastrophic failure and any unreasonable risk.
        - **Security** - The ability of the system to protect itself against accidental or deliberate intrusions.
    - **Properties** – A kind of responsibility that is an inherent or distinctive characteristic or trait that manifests some aspect of an object's knowledge or behaviour.
        - **Repairability** – Reflects the extent to which the system can be repaired in the event of failure.
        - **Maintainability** - Reflects the extent to which the system can be adapted to new requirements.
        - **Survivability** - Reflects the extent to which the system can deliver services whilst under hostile attack.

        - **Error Tolerance** - Reflects the extent to which user input errors can be avoided and tolerated.
    - **Threats** – Means by which a system may be adversely affected.
        - **Internal** – Means by which as the system may be adversely affected internally such as fault, error, failure, as discussed earlier in Laprie's taxonomy.
        - **External** – Means by which as the system may be adversely affected by elements external to the vehicle, (enemy fire, impact damage, environmental conditions).

These are only some key attributes for a Mission-Critical system. However, the key attributes specified are not used to characterise a Mission-Critical system but to enhance its purpose. Consider the Main Battle Tank (MBT) that requires to travel from point X to point Y. Assume that the MBT is fully integrated with very expensive systems that are essential for the mission success, i.e. Mission-Critical systems. These systems could be used for surveillance or survivability. Despite that, a fuel pump is also integrated on the platform, but is neither expensive, in comparison with the other systems, nor is considered as a Mission-Critical system. However, while in the mission the platform runs out of fuel because the fuel pump system stopped operating before even reach point Y. That will impact the and probably will be attributed as a mission failure. In that case, from a "typical" system, the fuel pump system will be referred to as Mission-Critical. Moreover, the importance of this section is to prove that almost any system can be considered as Mission-Critical system and the Mission-Critical system taxonomy within this chapter can always be differentiated depending on the mission.

In Chapter 4, a framework is presented that aims to give a clearer definition of what and how a Mission-Critical system should be characterised and designed. Also, the aim of the framework is to avoid misrepresentation or any issues similar to what has been mentioned earlier. Furthermore, in Chapter 6 the framework is applied as a proof-of-concept of this study. Below, a section will discuss what activities are applied for designing related critical systems either in industry or in the military domain.

3.3 Mission-Critical System Development Approach

Designing a Mission-Critical system can be challenging. This is due to their complexity in defining a Mission-Critical system. As discussed earlier, a sophisticated approach, (ways of thinking), is needed to achieve the definition and design of a Mission-Critical system, efficiently and effectively. For this study, the approaches used to define and design such systems are; the Systems Engineering's principles that consist of the Systems Engineering Process such as the Failure Mode and Effects Analysis (FMEA) tool, Safety Integrity Level (SIL) and

Technology Readiness Level (TRL). A brief introduction and further analysis will be presented below.

3.3.1 Systems Engineering

"*Μὴ μου τοὺς κύκλους τάραττε*"[16]

These were the last words said from one of the first systems engineer in history. A direct translation of these words is: "Do not disturb my circles" and were stated by Archimedes of Syracuse. Archimedes's was an ancient Greek mathematician, astronomer, engineer, physicist and is considered to be the first systems engineer [39] who lived c. 287 BC – 212 BC. His inventions, innovations, thoughts and ideas were described using mathematical models and graphical representations. Using those as his tools, Archimedes was able to describe a complex system, easily, fast and with precision. His activities are what is known today; the Systems Engineering (SE).

Systems Engineering is a term that has been first developed by the Bell Telephone Laboratories in the 1940s [40]. That was essential when complex engineering projects were developed and when the system's properties identification and manipulation were also needed. This approach is interdisciplinary which enables the realisation of successful systems. A system could be considered successful when it satisfies the needs of the end user.

With this approach, a collection of technical, natural and social elements were combined in order to produce a common understanding from various disciplines. For instance, when a system is referred to as "open" the system can interact with other engineering environments. When the system is referred to as "closed", then the system can interact only with its environment. In some other engineering disciplines "open" might also mean "non-proprietary". In that case, SE attempt to avoid this misinterpretation by using pre-defined terminologies that are focused around an engineered system and also accommodating the relationship of other engineering disciplines in the system's life-cycle.

With this approach, studies proved that approximately up to 20% of an entire project is based on SE. The studies also proved that costs were reduced, (INCOSE Systems Engineering Center of Excellence) [41]. The SE approach can be used from modelling and simulating procedures, to verify and validate the theory and assumptions, up to the system's development and integration. SE is well used in safety engineering, for the early detection of possible risks and to produce functional safety requirements.

[16] A sentence in ancient Greek language.

3.3.2 Systems Engineering Process

Today, in systems engineering there are various tools available to assist the system's design and development. These tools are usually strategies, procedures and techniques. The tools are varying to graphical representations, simulation, database management, document production and many more. A unified approach for all the aforementioned tools is the Systems Engineering Process (SEP) [42].

SEP is a top-down approach that enables a comprehensive, iterative and recursive problem-solving solution for complex system design. It can translate requirements and needs into a process description, decision making information and enable the next level of system development. The process is constructed by the following steps,

Systems Engineering Process Inputs – Is the initial phase of the systems engineering process in which it consists of; the end-user needs, objectives, requirements and constraints. Inputs may be composed of the mission, environments, available technology base, measures of effectiveness and requirements upon "corporate knowledge".

Requirements Analysis – This is the first stage where the SE process inputs are analysed in order to extract the system's functional and performance requirements. In other words, how the system must be developed and be able to satisfy the end-user. The requirements must be understandable, unambiguous, comprehensive, complete and concise. The Requirement Analysis can be transformed into smaller segments for the clean collaboration of various life-cycle customers. These segments are categorised in several ways.

- **Customer Requirements** – These are the requirements extracted from the customer's needs. The specific requirements are corresponding to the expectations of the system's functionalities, mission objectives, environment and Measurement of Effectiveness (MoE).
- **Functional Requirements** – Activity description and how can be achieved in terms of tasks and actions.
- **Performance Requirements** – This is the point where the customer must specify requirements such as response time, processing work, utilisation of resources, bandwidth, data transmission time and so on. This will distinguish different functionalities with different criticality levels within the overall system for the success of the mission.
- **Design Requirements** – The necessary requirements needed for the system's execution. Requirements such as software, hardware elements for the build, code and so on.

- **Derived Requirements** – A generic top-level to low-level requirements transition transformation.
- **Allocated Requirements** – The allocation of the high-level requirement to a sub-allocated low-level requirement.

Once the above requirement actions are completed, a unified and an encapsulated picture resulted in terms of requirements analysis output. The extraction of the Requirements Analysis is followed by, the operational, functional and physical views.

- **Operational view** – Referred to how the system will satisfy the user's demanding and in which degree, (how well and into what circumstances).
- **Functional view** – Is exclusively referred to what the system must do in terms of input, outputs, states and transformation rules.
- **Physical view** – Focuses on how the system must be constructed in terms of physical interfaces between operators, requirements and technology environments.

Functional Analysis/Allocation – This is the phase where the high-level functions are transformed into a lower level. This is resulting in a detailed description of the system's functionality and performance. This phase is also known as the functional architecture of the system. Nevertheless, this step will provide a clearer definition into, "What the system has to do to make sure that the mission is not compromised". Whilst, the system can associate priorities and conflicts with other low-level functions.

Requirement Loop – The results from the functional and allocation performance are used for the better understanding and reconsideration of the requirement analysis. This phase can provide clear traceability from the Requirements Analysis to Functional Analysis/Allocation and vice versa.

Design Synthesis – It is also known as the physical architecture in which, it is the basic structure for generating specification and baselines of the system. This process is to make up and define the system in terms of physical and software elements.

Design Loop – The design loop allows reassessments on how the system performs during its "simulated mission". In this phase, a review of the functional performance, that are pre-defined in the Functional Analysis/Allocation step, can be verified and enhance the optimisation of the system.

Verification – The outcome of each SE application process is compared with its corresponding requirement. A set of methods for this phase must be followed; the methods

composed of examination, demonstration, analysis and testing. Also, a formal test and evaluation are critical for this phase.

System Analysis and Control – These are technical management activities to measure the progress, evaluate and choose alternatives, document data and decisions.

- **System Analysis** – The activity to satisfy technical requirements of the system's mission and offers a strict decision making of performance, functional and design requirements.
- **Control** – Is the activity that is composed of risk, configuration and data management for reviewing the Measurement of Effectiveness (MoE), Measurement of Performance (MoP) and Technical Performance Measurement (TPM) of the system. Where MoE is the collection of the reporting information regarding the accomplishments of the system's mission. MoP is the collection of the reporting information regarding how the system performed. TPM is the extraction of the MoP that is critical from a periodic review and control standpoint.

Systems Engineering Process Output – This is the final phase of the whole process, that in general, it is any data that specifies the configuration that is essential to develop systems.

- **Specifications** – Specifications, are the documents that fully describes a design element of its interfaces in terms of technical requirements (functional, performance, constraints, and design characteristics), and the qualification conditions and procedures for each requirement.

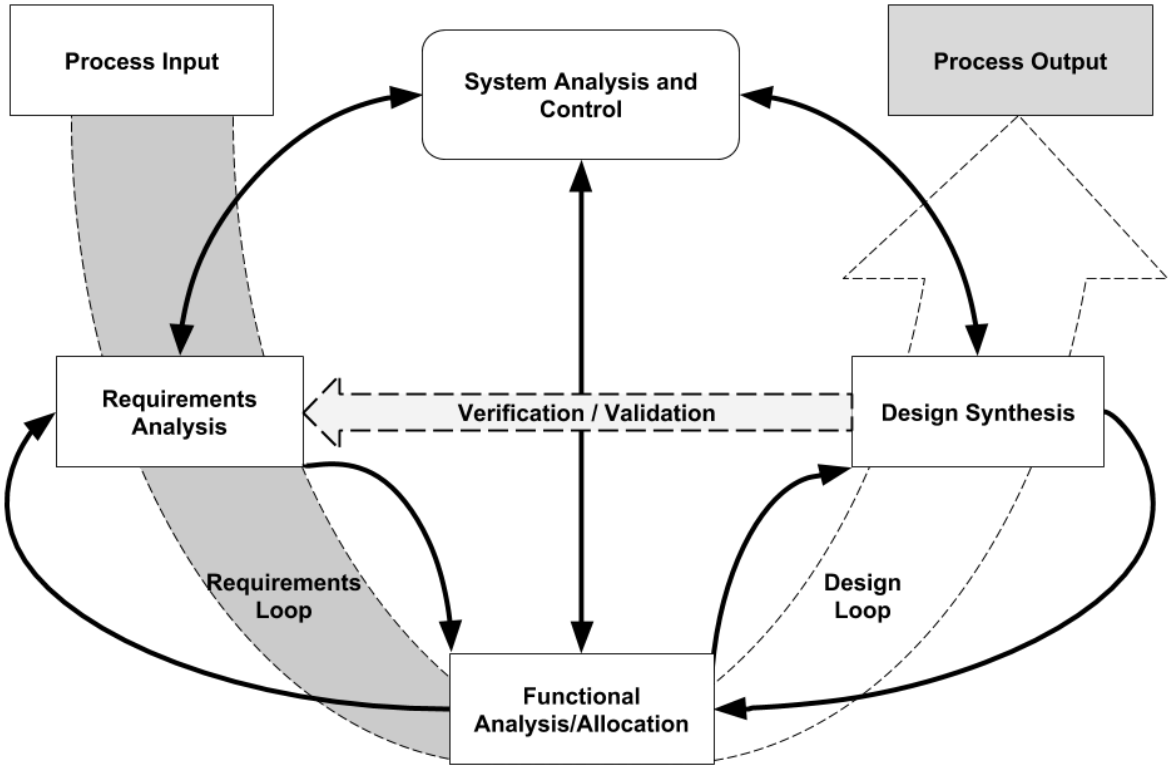

Figure 3-3 Systems Engineering Process

The main objective of the diagram, shown in Figure 3-3, is to conduct a verification from the lowest level to the actual implementation and to guarantee that system's efficiency and performance are acceptable within the expected and acceptable levels. In another point of view, Figure 3-3 depicts briefly how a generic system is developed from the concept phase to the development specification and in the end to the implementation and integration.

3.3.3 Standards

The rules, guidelines, practices and definitions of manufacturers, sellers, buyers, customers, trade associations, users, regulators, and so on, are encapsulated within documents, known as standards. These standards are the representation of any programme's needs, distilled from the aforementioned participants' experience. Standards establish engineering and technical limitations and applications for items, materials, processes, methods, designs and engineering practices. They are "cooperating knowledge" documents describing how to do some process or a description of a body of knowledge. Standards coming from many sources, reflecting the practices or knowledge base of the source.

Lastly, in a few words, standards are very powerful tools that could help to drive innovation and increase productivity. They can make activities more successful as well as people's everyday lives easier, safer, secure and healthier. Below, there are some standards for safety,

security, mission assurance and system's interoperability that are used across all over the industry as well as, in the military sector.

Safety

Safety standards are standards exclusively for the safety of people, environment and organisation. E/E/PE systems have been developed to perform functions for safety or for non-safety, such as quality. Either way, the safety concept is applicable across all industry areas. The industry used safety-related systems for many years in order to provide the required confidence that offers demanding risk reduction. Industry such as oil, gas, nuclear plants and machinery, relying on safety-related systems when a hazard occurs.

There are various safety standards which are well documented and applied in industry. Among them, the IEC 61508 is a vendor-agnostic engineering standard. All the IEC International Standards in the IEC 61508 series were developed by IEC SC (Subcommittee) 65 A: Industrial-process measurement, control and automation. Within the automotive industry, a related safety standard which is derived from the IEC 61508, is the ISO 26262. For further information proceed to [43].

Security

Defines the needs for achievement, realising, supporting and continually improving an information security management system within the context of the organisation. It also includes needs for the estimation and usage of information security risks fitted to the requirements of the programme. The needs specified in the standards, are abstracted such that all the programmes have the relevant level of knowledge, regardless of the type, size or nature. A UK MOD Security Standard, JSP 440 D Def Sy/6/3, comprise the three main elements to define security,

- **Confidentiality** – "*The restriction of information and other valuable assets to authorised individuals. (e.g. protection from espionage eavesdropping, leaks and computer hacking)*".
- **Integrity** – "*The maintenance of information systems of all kinds and physical assets in their complete and usable form (e.g. protection from unauthorised alteration to a computer programme)*".

- **Availability** – "*The permitting of continuous or timely access to information systems or physical assets by authorised users (e.g. protection from sabotage, malicious damage, theft, fire and flood)*"[17].

Mission Assurance

Mission Assurance is a full-cycle process for the support of defining and mitigating threats against missions. The approach used for Mission Assurance is the SE, risk management, quality and management principles. All the aforementioned approaches, aim at the success of the system's design, development, testing, deployment and operation. The main objective of Mission Assurance is to develop a condition of resilience that enhances the continuation of the programme's critical processes and protects its employees, assets, services and functions. Mission Assurance identifies risks in a uniform and systematic manner for the entire programme envelope.

Mission Assurance unifies multiple disciplines to contribute (project management, governance, system architecture, design, development, integration, testing and operations) and take advantage of their combined performance in use. For example, the US Department of Defense (DOD) 8500-series standard, uses three main mission assurance categories to define availability and integrity requirements. These main categories are assigned as Mission Assurance Category (MAC I, II and III),

- MAC I defines the control of data that is essential to the functional readiness or effectiveness of deployed or contingency forces.
- MAC II defines the control of data that is critical to the support of deployed and contingency forces.
- MAC III defines the control of data that is critical for day-to-day operations, but not directly related to the support of deployed or contingency forces.

For further information proceed to [44].

The combination of the Model-Based Engineering (MBSE) and Mission Assurance (MBMA) in NASA and in general in the defence is well established [45], [46], [47], [48]; where the MBSE focusing in a cost-effective capability of complex systems and the Mission Assurance ensures the system operates as is intended to.

[17] Adapted from "The Defence Manual of Security", Vol. 1,2 and 3, Issue 2"

Def-Stan 23-009

The defence standard 23-009 is a UK Ministry of Defence (MOD) standard and its main objective is to exploit the benefits of the open architecture approach to their platform's design and integration. Particular attention is paid to the platform's electronic and power infrastructure and the associated Human Machine Interface (HMI). The requirement of this standard is to advance the operational effectiveness across all the Defence Lines of Development (DLOD), integration's early de-risking and minimise the cost of ownership across the fleet, by maintaining and applying the appropriate standards and design constraints.

GVA approach is described and defined, throughout technical design rules and standards within Def-Stan 23-009. However, the descriptions and definitions may vary the design process due to the different requirements from platform to platform. Def-Stan 23-009 follows the principles of Verification and Validations (V&V) process, similar to SE process depicted in Figure 3-3. Sub-systems that consists of an electronic data and power infrastructure are integrated into a platform through the GVA. Along, with specific mechanical mountings, connectors and HMI requirements. This enables data interoperability of sub-system and crew stations and the rapid re-rolling and upgrading of vehicle platforms. Def-Stan 23-009 is divided into,

- Part 0 – The GVA approach.
- Part 1 – Infrastructure (Data and Power)
- Part 2 – Human Machine Interface (HMI)
- Part 3 – Health and Usage Monitoring (HUMS) (under development)
- Part 4 – Physical Interfaces
- Part 5 – GVA Data Model and Model Driven Architecture (LDM and MDA)
- Part 6 – Security (under development)
- Part 7 – Common Services (under development)
- Part 8 – Safety (under development)

3.3.4 Systematic Technique for System's Failure Analysis

Systems that are categorised as critical, are highly expected to deliver the intended quality and reliable service. Many land military platforms are relying on those critical systems and usually, a threat[18] is undesired. Normally, there are some threats that could be detected either before or during the system's development. If threats are identified during the development, it can result in significant programme's cost and delays to schedules. The challenging part of

[18] Threat definition is discussed in detail in Section 3.2.4.

the system's development is to identify any potential threat in the very early stages. An approach to achieve that is by the Failure Mode and Effects Analysis (FMEA).

Basic Analysis

FMEA is an approach that provides a qualitative and systematic step-by-step guidance, to assist systems engineers to observe any (possible) threat that may cause failures in products or processes. Additionally, it can assist to identify how a product or process might fail and also the effects of that failure. FMEA also assists to identify the potential causes of failures and the likelihood of failures detected before the occurrence. Applicable in many industries, FMEA is one of the most effective methods for examining possible reliability issues, enabling manufacturers to take quick actions and mitigate failures, early in the development cycle. The capability of observing hazards in the early stages of the development let system engineers design out failures and design in a reliable and safe way.

Finding Failure Modes

An initial step needed during the FMEA process is to determine the team or partakers. Partakers can be customers, suppliers, system owners, safety engineers and so on and should participate in order to address any potential threat against the system's development. The partakers should be able to identify all the attributes of the programme, components, systems, processes, functions etc., including all the possible failures that may downgrade the quality and reliability. It is also essential if the partakers are able to identify the effects and the potential causes of the defined threat. As shown in Table 3-1 and Table 3-2 [49], a simple example is used to understand how the FMEA works. The team, in this case, is analysing a tyre component of a vehicle.

Table 3-1 FMEA Process 1a [49]

Function or Process Step	**Failure Type**	**Potential Impact**	**Severity**	**Potential Causes**	**Occurrence**	**Detection Mode**	**Detection**	**RPN**
Briefly, outline function, step or item analysed.	*Describe what has gone wrong.*	*What is the impact on the key output variables or internal requirements?*	*How severe is the effect on the customer?*	*What causes the key input to go wrong?*	*How frequently is this likely to occur?*	*What are the existing controls that either prevent the failure from occurring or detect it?*	*How easy is it to detect?*	*Risk priority number?*
Tyre function supports the weight of the car, traction, comfort, etc.	Flat tyre.	Stops car journey, driver and passengers stranded.	10	Puncture	2	Tyre checks before the journey. While driving, steering pulls to one side, excess noise.	6	120

Table 3-2 FMEA Process 1b [49]

Recommended Actions	**Responsibility**	**Target Date**	**Action Taken**	**Severity**	**Occurrence**	**Detection**	**RPN**
What are the actions for reducing the occurrence of the cause or improving the detection?	*Who is responsible for the recommended action?*	*What is the target date for the recommended action?*	*What are the actions implements? Now recalculate the RPN to see if the action has reduced the risk*				
Carry spare tyre and appropriate tools to change the tire.	Car owner.	From immediate effect.	A spare tyre and appropriate tools permanently carried in the trunk.	10	2	4	80

Criteria for Analysis

The FMEA process uses three main categories to estimate an issue:

- Severity
- Occurrence
- Detection.

Partakers should arrange and agree between 1 and 10 (1 = low and 10 = high) levels of the severity, occurrence and detection level for each of the failure modes. Despite this, FMEA is a qualitative process, so it is critical to use data (if available) to qualify the decisions of the teams. A further explanation of the ratings is shown in Table 3-3.

Table 3-3 Categories for estimating issues [49]

	Description	**Low Number**	**High Number**
Severity	Severity encompasses what is important to the industry, company or customers (e.g. safety standards, environment, legal, production continuity, scrap, loss of business, damage reputation)	Low Impact	High Impact
Occurrence	Ranks the probability of a failure occurring during the expected lifetime of the product or service	Not likely to occur	Inevitable
Detection	Ranks the probability of the problem being detected and acted upon before it has happened	Very likely to be detected	Not likely to be detected

Once the severity, occurrence and detection levels are ranked for each failure mode, the team will be able to estimate a Risk Priority Number (RPN). The RPN can be calculated:

$$RPN = Severity \times Occurrence \times Detection \quad \text{Equation 1}$$

Where:

RPN: Risk Priority Number

Setting Priorities

When all the failure modes are analysed, the team should re-arrange the FMEA to levels of failures in descending RPN order. With this approach corrective actions can be organised effectively.

RPN levels cannot be definitively due to many reasons; “Different purposes, different disciplines, different priorities”. For example, the safety party will prioritise safety, the security party prioritises security, the survivability party prioritises survivability and so on.

3.3.5 Safety Integrity Levels

Nowadays in industry, most of the processes are implemented with the aid of electronic systems, as discussed and analysed earlier. These systems have replaced a big degree of manpower therefore, these systems are designed to serve specific task(s). In some cases, undesired faults may arise during the process and as a result, these faults may lead to accidents, environmental hazards and even deaths. However, in response to that issue, a measurement was taken to reduce the severity of the faults and/or to prevent them arise. These measurements are known as the “Functional Safety” functions [50]. Functional safety is known as,

“The safety that controls systems in order to reduce potential risks to the desired level.” [50]

There is an increased demand for functional safety within industries, slowly implemented and standardised (IEC 61511, IEC 61508, ISO26262 and ANSI/ISA 84). The functional safety depends on the correct functioning of the logic solver, sensors and elements, Safety Instrumented Systems (SIS) to accomplish a desired risk reduction level.

When predetermined conditions are violated, SIS can prevent or mitigate potentially hazardous events[19]. Each of these instruments consists of one or more Safety Instrumented Function (SIF). The functional execution is performed through, the sensor(s), logic solver(s) and an actuator(s); and every function has its own Safety Integrity Level (SIL). Depending on the process, each SIL levels can vary or can be the same. Ideally, functions should have the same SIL levels in every system within their environment.

Safety Integrity Level (SIL) is defined as a relative level of risk-reduction provided by a safety function, to specify a target of risk reduction. In a simple manner, SIL is a measurement of

[19] For example, Emergency Shutdown Systems (ESS) in oil and gas industry.

performance required for safety instrumented function in terms of probability of failure on demand (PFD).

Table 3-4 Safety Integrity Levels[20] [51]

Safety Integrity Level	Risk Reduction Factor	Probability of Failure on Demand
SIL 4	100,000 to 10,000	10^{-5} to 10^{-4}
SIL 3	10,000 to 1,000	10^{-4} to 10^{-3}
SIL 2	1,000 to 100	10^{-3} to 10^{-2}
SIL 1	100 to 10	10^{-2} to 10^{-1}

The high value of SIL indicates greater associated safety level and the lower level indicates the probability that the system will fail to operate appropriately. In parallel, higher SIL level means higher the costs, maintenance and complexity of the bespoke system.

Assigning SIL level to a system, a risk analysis must be conducted where the risk is associated with specific hazards. The risk extracted from the analysis should be protected by the SIF. The Process Hazard Analysis is the procedure, often the first step, to determine the SIF and the tolerable risk level. If SIF's risk is higher than the tolerable risk, then this must be addressed through risk reduction SIF; that depends on the customer. There are various methods to assign SIL levels to systems, as shown in the following tables.

Table 3-5 Categories of the likelihood of failure

Category	Definition
Frequent	Many times
Probable	Several times
Occasional	Once
Remote	Unlikely
Improbable	Very Unlikely
Incredible	Cannot believe that it could occur

[20] Figures in the tables are examples based on the cited reference.

Table 3-6 Consequence categories

Category	Definition
Catastrophic	Complete failure
Critical	Impacts but not a complete failure
Marginal	Major issues
Negligible	Minor issues

Table 3-7 Risk class matrix [52]

Likelihood	Consequence			
	Catastrophic	***Critical***	***Marginal***	***Negligible***
Frequent	Class 1	1	1	2
Probable	1	1	2	3
Occasional	1	2	3	3
Remote	2	3	3	4
Improbable	3	3	4	4
Incredible	4	4	4	Class 4

The classification of the consequences is as follow:

Class 1: Unacceptable in any circumstance.

Class 2: Undesirable; tolerable only if risk reduction is impracticable or if the costs are grossly disproportionate to the improvement gained.

Class 3: Tolerable if the cost of risk reduction would exceed the improvement.

Class 4: Acceptable as it stands, though it may need to be monitored [53].

3.3.6 Technology Readiness Levels

The management in making decisions regarding the development and the transition of the technology is enabled by the Technology Readiness Levels (TRL) principle. The TRL approach is used for a system to satisfy the theory into a more practical evaluation. This approach is followed by 9 levels.

TRL1: Basic principles observed and reported – Transition from scientific research into technology's basic properties. Critical attributes for the system and architectures using descriptive tools.

TRL2: Technology concept and/or application – Research application level. Assumptions and scientific principles are united to the specified concept. Attributes of the activity can be described using analytical tools for the simulation and analysis of the application.

TRL3: Analytical and experimental critical functions and/or characteristics proof of concept – A demonstration of technical feasibility is taking place for the proof of concept validation. Active Research and Development (R&D) is launched with fruitful analysis and experimental studies.

TRL4: Component/sub-system validation in a laboratory environment – Test and implementation of standalone prototyping. Integration and collaboration of technological elements.

TRL5: System/sub-system/component validation in a relevant environment – Test and implementation of the prototyped concept in some realistic simulated environments. This aims the target of the implemented environment and interface.

TRL6: System/sub-system model or prototyping demonstration in a relevant end-to-end environment – Implementations of the prototyping concept into a full-scale realistic problem beyond the TRL5.

TRL7: System prototyping demonstration in an operational environment – A demonstration of the prototype system into an operational environment.

TRL8: Actual system completed and "mission qualified" through test and demonstration in an operational environment – Final system development process that is fully integrated with functional software and hardware systems. With the Verification and Validation (V&V) process accomplished, user, training and maintenance documentations completed.

TRL9: Actual system "mission proven" through successful mission operations – Final form of the system that is integrated and operated in an actual operational environment.

The usage of the TRL method will provide a common understanding of the technology status of the Mission-Critical system. A more analysed description of the TRL levels can be found in NASA's white paper in [54].

3.4 Conclusions and Future Work

In this chapter, it has been identified that a Mission-Critical system cannot be specified or narrowed down into a single element that easily. This results when many participants are involved in the development of an of Mission-Critical system. Most of the participants have a different point of views and as aforementioned earlier in this chapter, each discipline prioritises their own needs. The safety prioritises the safety of people and environment; the security prioritises the protection of data from various threats; survivability prioritises the whole mission envelope; the procurement prioritises the costs and the bureaucracy prioritises the political associations of the government.

However, in this chapter, only a few Mission-Critical system's characteristics are discussed, whilst there are more other Mission-Critical systems existing in different areas, (medicine, banks and so on).

In the future, more mission-related and critical-related standards and specifications will be reviewed in order to increase the knowledge how a Mission-Critical system should be characterised. With this increase, the systems engineers and architects should be able to understand better the requirements needed for a Mission-Critical system that could avoid any unreasonable risks that downgrade the system's dependability. Lastly, when more information is gained, it is more likely the designed Mission-Critical system be able to enhance mission success.

Lastly, in this chapter, various activities, approaches and methodologies for designing critical systems for different applications were presented and analysed. According to this chapter, this research will adopt significant parts from these activities, approaches and methodologies, in order to construct a novel unified framework applicable to mission-related systems with the association of the Interoperable Open Architecture (IOA) approach.

Chapter 4 Framework for Designing Vetronics Mission-Related Systems

4.1 Introduction

Systems consisting of electrical and/or electronic components have been applicable for many years to execute different services in most application sectors. Programmable electronic systems, also known as computer-based systems, are being in action in all application sectors to execute various generic services and increasingly to perform services beyond generic, such as safety, security, survivability and so on. If programmable electronic system technology is to efficiently make full use of it, it is critical that those partakers for making decisions have adequate guidance on system's service aspects on which to make these decisions.

However, the decision that describes Mission-Critical systems can be challenging, in particular when many disciplines are involved in the development. In [55], there is an introduction of decision-making methods that could enhance the picture of describing Mission-Critical systems.

"*Decision making is the study of identifying and choosing alternatives based on the values and preferences of the decision maker. Making a decision implies that there are alternative choices to be considered and is such a case we want not only to identify as many of these alternatives as possible but to choose one that best fits our goals, objectives, desires, values and so on"* [55].

Therefore, the following section presents the different areas involved in making decisions to describe and implement missions using pre-defined terms.

4.2 Framework of techniques and measures for the design of mission-related systems

4.2.1 Mission – M[n][21]

Aim: To define the formal summary of the aims and values of an activity.

Description: The singular objective, task or purpose of an activity. To identify the primary objective.

[21] For each step within the framework, will be assigned with an abbreviation according to its initials. For example, **M**ission will be assigned as **M**[n]; [n] is the natural number of each requirement. A further explanation on how each requirement adopts its own natural number will be provided later in this chapter.

4.2.2 Mission System – MS[n]

Aim: To define the system that is essential for mission M[n].

Description: An organised, purposeful structure that consists of interrelated and interdependent elements (Electrical, Electronic and Electronic Programmable systems, components, entities, factors, members, parts, functions, etc.). These elements continually influence one another (directly or indirectly) to maintain their activity and the existence of the system, in order to achieve the mission of the system and therefore, the overall mission M[n].

*Note**:** To increase the mission's success capability, other systems may be also introduced within this part, either critical or non-critical, that are involved in the entire environment.*

4.2.3 System Analysis – SA[n]

Aim: To define a statement of the reason that the specific mission system MS[n] exists and considered in the mission M[n].

Description: Describe the reason for which the system is selected. Describe the capabilities and the functions of the mission system MS[n].

*Note: It is an advantage if the user also describes how the system operates and what does it need to operate. For example, hardware, software, operating system, sub-systems, power etc., then converted into UML domains, packages, entities, data attributes and relationships that could be useful for the next step. This can also be considered as Requirements Analysis as defined in Systems Engineering, Section 3.3.1. Furthermore, following the ISO 26262 standard this step of the framework can be also considered as the "**Item Definition**" found in Part 3, Clause 3-5 of the standard, giving details to the item's boundaries.*

4.2.4 Data Model – Mission System[n]

Aim: To model Mission System MS[n] using data modelling procedures[22].

Description: A graphical and textual representation of analysis that identifies the data needed by the programme to describe mission system's MS[n], mission, functions, goals, objectives and strategies and to manage and rate the programme.

Note: A data model identifies the domains, packages, entities, data attributes and relationships with other data and provides the conceptual view of the data and the relationships among data, key style.

4.2.5 Benefits – B[n]

Aim: To define any advantage gained from the selected mission system MS[n].

Description: Any distinctive attribute, characteristic, feature or aspect possessed by the mission system MS[n], that gives an individual, entity or any other thing a more favourable opportunity for mission success.

[22] In Chapter 5, an introduction and a further discussion on how to develop a data model is presented.

Note: In this procedure, qualities, capabilities, features and functions can be introduced as part of the selected mission system.

4.2.6 Effectiveness Level – EL_B[n]

Aim: To define the effectiveness level of the described benefit B[n].

Description: The possible degree to which the MS[n] will successfully produce the desired result.

4.2.7 Threat – T[n]

Aim: To identify a situation in which there is an actual or potential negative affect in mission M[n] and in the mission system's MS[n] intended purpose.

Description: Describe the situations that can jeopardise the process of the mission M[n] and the mission system MS[n]. All causes of failure, random hardware failure or systematic failure, that lead to an unwanted state should be included.

*Note: Failure Type – It is also important to consider how a benefit can be possibly transformed into a negative impact. Or describe what has gone wrong if the system is already existing and needs improvements. Moreover, in this step up to Section 4.2.15, following the ISO 26262 can be considered as the "**Initiation of the Safety Lifecycle**" and "**Hazard Analysis and Risk Assessment**" Part 3, Clause 3-6 and 3-7.*

4.2.8 Threatening System – TS[n]

Aim: To give the potential reason for the threat to occur.

Description: What are the key inputs to cause the threat?

4.2.9 Occurrences – O[n]

Aim: Identify the frequency of the cause.

Description: How frequently it is likely the Threatening System TS[n] to cause the threat T[n]?

4.2.10 Potential Impact – PI[n]

Aim: Describe the impact on the key output variables or internal requirements.

Description: Identify the key demotions that influence the mission M[n], mission system's MS[n] and its benefits B[n].

4.2.11 Severity – SE[n]

Aim: To categorise the severity of the potential impact PI[n].

Description: The degree of the potential impact P[n] that the threat T[n] and TS[n], is causing against the mission M[n], Mission System MS[n] and its benefits B[n].

4.2.12 Threat Classification

Aim: To classify the threat.

Description: The threat will be classified when severity and occurrence are identified. A further explanation will be provided in Section 4.3.

Note: This can be used if the threat is modelled using the Data Modelling approach in order to represent the risk of the threat in different missions.

4.2.13 Detection Mode – DM[n]

Aim: To address the detection of the actions and causes of the threatening event.

Description: What is the existing control that either prevents the failure from occurring or being detected?

4.2.14 Detection – D[n]

Aim: To describe the state of the detection.

Description: What is the detection degree?

4.2.15 Threat Level – TL_T[n]

Aim: Evaluate the threat's T[n] affect level.

Description: To evaluate the affected level of the defined T[n] threat and recalculate the effectiveness level of the defined benefit EL_B[n].

4.2.16 Data Model - Threat

Aim: To model threat T[n] using data modelling procedures.

Description: A graphical and textual representation of analysis that identifies the data needed by the programme to describe threat's T[n], cause, occurrence, potential impact, severity, and classification. To construct a data model of the threat identified including its properties and features.

Note: In this step, the threat can be constructed using the data modelling's approach. This could be useful in the future when different mission systems identified the same threat.

4.2.17 Mission-Critical System – MCS[n]

Aim: To declare the element needed for the threat T[n] to be dealt with or to be prevented.

Description: What are the actions for reducing the occurrence of the cause. What are the actions for improving detection? What are the actions for dealing with the failure? The selected element will be considered as Mission-Critical System MCS[n].

*Note: In this step, up to Section 4.2.22 can be considered as Part 3, Clause 3-8 of the ISO 26262 standard, where this is specifying the "**Functional Safety Concept**" requirements.*

4.2.18 Mission-Critical Function – MCF[n]

Aim: To describe the actions required.

Description: This is the analysis needed of how the Mission-Critical system MCS[n] must operate and eventually provide the recommended actions needed to deal with the identified threat T[n].

4.2.19 Responsibility – R[n]

Aim: To identify the key controller credited for the Mission-Critical Function MCF[n].

Description: The element that is responsible to execute the recommended action identified in MCF[n].

Note: If human factors are involved in this procedure, personnel training should be conducted.

4.2.20 Target Date – TD[n]

Aim: To assign the time for the MCF[n] to be executed.

Description: Identify the target date for the recommended action to occur and be completed. Referring about Mission-Critical systems, a range of Real-Time responsiveness levels such as Non-Real-Time, Real-Time, Soft Real-Time and Hard Real-Time, will not only be used to assign the response time needed for the action to be executed, but also the time constraints of the action to be executed.

Note: Time cannot be specified until is agreed. This part is important in how the electronic architecture of the systems should be constructed using specific communication networks, hardware and software satisfying the Real-Time responsiveness levels.

4.2.21 Data Model – Mission-Critical System

Aim : To model Mission-Critical System MCS[n] using data modelling procedures.

Description: A graphical and textual representation of analysis that identifies the data needed by the programme to describe Mission Critical system's MCS[n], Mission-Critical function, responsibility and target date. To design the data or requirements that will enhance the Mission-Critical system's functionalities; Mainly to reflect or prevent the threat to occur.

Note: In this part of the framework, the author attempts to exploit the data model approach and generate data specification that will be useful for programmes consisting of multidisciplinary departments.

4.2.22 Action Taken – AT[n]

Aim: To describe in detail the methods and approaches for the implementation of the MCS[n] and its data model.

Description: What were the actions taken in order to achieve the description of the MCS[n], MCF[n], R[n], TD[n] and data model.

Note: This is yet another critical part of the framework because this step can be used as a requirement(s) generator. When the system is developed in the early stages, the candidate Tier 1 supplier, if applicable, can use this step and strictly follow it on how the system shall be developed and behave. With this approach, Tier 1 suppliers and the "OEM" shall avoid time driven instances of misunderstanding.

4.2.23 Mitigation Process – MP[n]

Aim: To indicate the Technology Readiness Level of the Mission-Critical System MCS[n].

Description: A method that will estimate the maturity and give a written assurance that the Mission-Critical system MCS[n] conforms to specified mission requirement during the acquisition process. To indicate approved confidence of the system for the specific or related missions TRLs will be assigned.

4.2.24 Mission Integrity Level – MIL(n)

Aim: Assign the MIL to the system.

Description: MIL, are levels assigned to Mission-Critical systems to indicate their integrity on different missions and how will it impact the mission in case of a failure. In other words, the likelihood of the mission-related system satisfactorily performing the required mission functions under all the stated conditions within a stated period of time.

4.2.25 Source – SO[n]

Aim: All of the participants who contributed or involved in the framework must be cited and referenced.

Description: All the partakers from different disciplines must be assigned to each framework's step. For example, the person or the team that addresses the potential impacts must be declared. This will be useful afterwards, to ensure that requirements, proposals and so on, are valid.

Note: The citations and references shall not be only the framework's active participants but also any source from academic research, technical reports, white papers, conference presentation, etc.

4.3 Framework's usage

This section presents the process of how the framework can be used. A detailed, step-by-step explanation is conducted to provide the necessary definitions on how to use the framework.

4.3.1 User's Requirements and Definitions

"**Mission – M[n]**", is the first step and is to identify the mission. Once the mission is defined, a system that is essential to the mission is considered as "**Mission System – MS[n]**", to potentially ensure the mission's success; this step can also consider other systems that are also involved or integrated on the platform or are used in the mission M[n]. These systems can be independent, critical or even non-critical, despite their application. The reason for this consideration is to help participants to discuss, observe, study and think about anything that could potentially benefit or jeopardise the defined mission M[n].

4.3.2 System's Requirements

Next, it is essential to analyse the system, "**System Analysis – SA[n]**", of the defined mission system MS[n]. For more benefits, this step can also describe how precisely the mission system operates and what are the elements needed to achieve its operation. The participant should

have the freedom to go into a deep analysis for more promising results but this is depending on the disciplines involved and the programme's procurement. Unfortunately, human's knowledge does not meet boundaries thus, it can be infinite and as a result, this will be converted into an extreme disadvantage for this purpose.

Next is the data modelling procedure, "**Data Model – Mission System[n]**", of the defined MS[n] and its analysation process SA[n]. The data model will be the graphical representation of the MS[n] and it's SA[n] in a UML notation.

4.3.3 System's Expectation

Once the previous steps are completed, the user can proceed to the next step, which is the benefits identification, "**Benefits B[n]**", of the mission system MS[n] for the mission M[n]. This is where the qualities, capabilities, features, functions and so on, of the mission system should be declared. This step can also use definitions as described in Section 3.1. However, each declared benefit must be assigned with a percentage value, that indicates the probability of the defined MS[n] system to be successful on the mission M[n]. This can be accomplished in the "**Effectiveness Level – EL_B[n]**", (Equation 2), a procedure where threat analysis and mitigation process takes place to indicate the effectiveness level of the defined mission system's MS[n] benefit B[n].

$$\mathrm{EL\_B[n]\_T[n]} = B[n] - TL\_T[n] \qquad \text{Equation 2}$$

Where:

EL_B[n]_T[n]: The effectiveness level of the defined benefit with the identified threat.

B[n]: The defined benefit.

TL_T[n]: The threat's level value of the identified threat.

$$EL\_B[n]\_MP[n] = MP[n]\_MCS[n] * EL\_B[n]\_T[n] \qquad \text{Equation 3}$$

Where:

EL_B[n]_MP[n]: The effectiveness level of the defined benefit with the defined mitigation process.

MP[n]_MCS[n]: The mitigation's process value (Compliance Levels - Table 4-8) of the Mission-Critical System MCS[n].

EL_B[n]_T[n]: The effectiveness level of the defined benefit with the identified threat.

4.3.4 Threat Analysis

Once the benefit(s) B[n] is defined, the user should proceed with threat assessment procedure, "**Threat** – **T[n]**", to ensure that the defined benefits are valid and applicable to the mission system MS[n] and the mission M[n]. The T[n] step, is the process that identifies any possible circumstances that may affect in any way the defined B[n] and its effectiveness level. This process may also focus on the mission system's failure mechanisms[23]. A risk analysis process must be followed, which determines a detailed analysis of the threat. This procedure follows the same line as the Failure Mode Effects Analysis (FMEA) but slightly modified, closely related to this research's approach.

Next, the "**Threatening System – TS[n]**" step is used to analyse the key input for the threat caused. It can be also considered as the "Threats" of the mission system's MS[n] benefits B[n]; "Threats" are defined in Section 3.2.4. The "**Occurrence – O[n]"** is the step that declares how frequently the threat is likely to occur. The user should have the freedom to choose from between the options shown in Table 4-1.

Table 4-1 Categories of the likelihood of failure – O[n]

Category	**Definition**	**Occurrence Level**
Frequent	Many times	O3
Probable	Several times	
Occasional	Once	O2
Remote	Unlikely	
Improbable	Very Unlikely	O1
Incredible	Cannot believe that it could occur	

The "**Potential Impact – PI[n]**" is the procedure that describes the resulted actions when the specific threat T[n] occurs against the mission M[n], mission system MS[n] and mission system's benefit B[n]. The "**Severity – SE[n]**" step declares the severity level of the potential impact. The user would have the freedom to choose between the options available in Table 4-2.

[23] I.e. System's internal malfunction errors or external impacts.

Table 4-2 Consequence categories – SE[n]

Category	Definition	Severity Level
Catastrophic	Complete mission failure	S3
Critical	Impacts mission but not complete failure	S2
Marginal	Major mission issues	S1
Negligible	Minor mission issues	S0

By defining the occurrence and severity of the threat T[n], the next step is to classify it, "**Threat Classification**". The classification level can be obtained from Table 4-3.

Table 4-3 Risk Classification Matrices

Occurence	Severity			
	Catastrophic	*Critical*	*Marginal*	*Negligible*
Frequent	Class 1	1	1	2
Probable	1	1	2	3
Occasional	1	2	3	3
Remote	2	3	3	4
Improbable	3	3	4	4
Incredible	4	4	4	Class 4

Where:

Class 1: Unacceptable in any circumstance and it will be a great negative impact on the mission system's effectiveness EL_B[n] and therefore, to the overall mission M[n].

Class 2: Undesirable, tolerable only if risk reduction is impracticable or if the costs are grossly disproportionate to the improvement gained.

Class 3: Tolerable if the cost of risk reduction would exceed the improvement.

Class 4: Acceptable as it stands, though it may need to be monitored.

The reason for classifying the threat in this research is to declare the relationship between the defined threat T[n] and the affected element. This is discussed further in Chapter 6, in a more detailed analysis.

Moving on to the next step of the threat assessment, the "**Detection Mode – DM[n]**" is used to analyse ways, methods or approaches to detect the threat T[n]. This can be an advantage

for personnel training if platforms are not fully autonomous and human factors are involved in the mission M[n]. The "**Detection – D[n]**" procedure defines the degree to which the threat T[n] can be detected. The user would have the option to choose between the different levels as depicted in Table 4-4.

Table 4-4 Detection Levels – D[n]

Difficulty Level	Definition	Detection Level
No effort	Very likely to be detected	D1
Very Easy	With almost no effort	
Easy	Without great effort	D2
Normal	Conforming to a standard	
Hard	With a great deal of effort	D3
Very Hard	Not likely to be detected	

The last step of the risk analysis process is to calculate the affected level of the pre-defined threat "**Threat Level – TL_T[n]**". It is important to calculate the affected value of the threat T[n] against the mission system MS[n], its benefits B[n] and eventually the overall mission M[n] in order to give a clear definition of its degree and criticality. Before explaining the process of how to calculate the TL_T[n], firstly, it is necessary to explain what is the theory behind it.

Threat Evaluation and Estimation

First, it is important to note that the requirements Occurrence – O[n], Severity – SE[n], Detection – D[n] carry their own percentage value. In order to evaluate the criticality of the identified threat T[n], the aforementioned requirements must be assigned with values as depicted in Table 4-5. The values are indicative (assumption), therefore, the maximum value a threat T[n] can be roughly 99.9% and the lowest, 0%. These values will indicate how much the threat can threaten the mission in terms of percentage probability.

Table 4-5 Threat's Elements and their Values

Occurrence	Occurrence (%)	Severity	Severity (%)	Detection	Detection (%)
Frequent	25	**Catastrophic**	≤50	**Very Hard**	25
Probable	20	**Critical**	~33.4[3.s.f]	**Hard**	20
Occasional	15	**Marginal**	~16.7[3.s.f]	**Normal**	15
Remote	10	**Negligible**	0	**Easy**	10
Improbable	5			**Very Easy**	5
Incredible	0			**No Effort**	0

Therefore, the threat level of the identified threat T[n], TL_T[n] can be calculated using the following expression, Equation 4,

$$TL\_T[n] = O[n] + SE[n] + D[n] \qquad \text{Equation 4}$$

Where:

TL_T[n]: The threat level of the identified threat.

O[n]: The occurrence value of the O[n]

SE[n]: The severity value of the SE[n]

D[n]: The detection value of the D[n]

As stated earlier, each step of the framework will be assigned with abbreviations extracted from the initial(s) of each step. For example, the step **M**ission has a notation **M**[n] where "[n]" represents the natural number of each requirement. For instance, if one mission is identified within the framework, then the mission will be assigned as M[1]. Unless, requirements have sub-requirements; for example, if mission M[1] has a sub-mission requirement then, it can be assigned as M[1][1] and so on. The additional sub-requirement takes a real number starting from "1" right after the previous core requirement. A representation of this is shown in Table 4-6.

Table 4-6 Requirements Sequence

Req	1st Sub	n Sub
	Req[1][1]	Req[1]…i…[1]
Req[1]	…	…
	Req[1][n]	Req[1]…i…[n]
	…	…
	Req[n][1]	Req[n]…i…[1]
Req[n]	…	…
	Req[n][n]	Req[n]…i…[n]

Where:

Req: The core requirement.

1st Sub: The first sub_core_requirement.

n Sub: Indicates the last sub_requirement.

n: Represents the real number.

i: Represents the sequential number of sub-requirements.

An approach on how to calculate the average value of two or more requirements of the same degree and yet not limited to have an overall value of multiple requirements when accumulated, Equation 5 can be used.

$$Req\_core = \sum_{i=1}^{n} \frac{Req[i]}{n} \qquad \text{Equation 5}$$

Where:

Req_core: The overall average value of the core requirement.

i: Lower limit number of requirement.

n: Upper limit number of requirement.

Once the mathematics behind are clear and properly applied, it is now possible to evaluate the threat level of the identified threat T[n]. However, whatever the calculated value is from the TL_T[n], it must be carried back to the effectiveness level procedure and calculate the effectiveness level of the defined benefit using the identified threat EL_B[n]_T[n]. To calculate the EL_B[n]_T[n] the expressions, Equation 4, Equation 5, (if more than one) and Equation 2 must be fulfilled.

Lastly, when the threat analysis is completed a data model of the threat T[n], "**Data Model** – **Threat**" shall be constructed. The purpose of this data model is to provide a graphical representation of the threat in order to make clear definitions between the stakeholder, systems engineer and system architect. Another purpose of this, the threat T[n] data model can be re-used for different applications (missions M[n], mission systems MS[n] and Mission-Critical systems MCS[n]), when threats are similar or closely related.

4.3.5 Threat's Risk Reduction Process

When threat T[n] is identified, analysed and estimate its effect, the user can proceed to the next part of the framework. The next part is to countermeasure the identified threat T[n]. The main objective of this part is to minimise the affected level in a level that ideally is considered

"Negligible". That means, the defined benefit B[n], needs to gain back its expected value, which is close to 99.9% and be considered that the mission system MS[n] will successfully contribute its full potential to the mission M[n].

To achieve that, firstly the user must identify a system that is essential to the survival of the mission system MS[n], hence, "**Mission-Critical System – MCS[n]**". When the user decides an appropriate MCS[n], it is important to provide the recommended actions needed to reflect the threat. The purpose of the recommended actions is to reduce the occurrence of the event; to improve the detection and to eliminate or prevent any severe result. These actions can be described in the "**Mission-Critical Function – MCF[n]**" procedure.

To identify the key controller credited for the Mission-Critical Function MCF[n], this step is referred to as "**Responsibility – R[n]**". It is the procedure that assigns the key factor responsible for the action to be executed or to be controlled. R[n] have no restrictions on the factors considered responsible for the MCF[n]. They can be either for autonomous systems, semi-autonomous systems and/or human factors.

Next step is the "**Target Date – TD[n]**" procedure. TD[n] can be challenging and requires sophisticated decisions to provide clear time definitions as possible. The main purpose of this step is to declare when the MCF[n] must be executed and what is the time constraint of the execution. It is crucial to agree, pre-define or specify, the time and constraints right from the beginning of the programme. The user may use Table 4-7 to decide the time of execution and deadline. With this approach, it could potentially benefit systems engineers and architects to decide the appropriate electronic architecture environment for the specific function MCF[n] to be executed and therefore, how the MCS[n] shall be developed. The user would have to select from between the options as depicted in Table 4-7.

Table 4-7 Real-Time Responsiveness Levels – TD[n]

Level	Definition
Hard Real-Time	A responsiveness level to respond to an event with a specified time constraint and must be executed immediately. All the deadlines must be accomplished within that time constraint
Soft Real-Time	This responsiveness level is important but no matter of complete failure. Must be executed in any of the upcoming event included in the release. The acceptable frequency of missed deadlines is dictated by the design
Real-Time	A responsiveness level that uses human-related responsiveness time during a process or event. May be fixed in the future release, not necessarily in the next release
Non-Real-Time	A responsiveness level that does not require time constraints. Potentially may not get fixed, but can be a candidate for future releases

The last step for this part of the framework is to estimate the maturity level of the recommended MCF[n] and prove whether is capable of dealing with the defined T[n] threat or not. The specific step is referred to as the "**Mitigation Process** – **MP[n]**". Using the content in Section 3.3.6, the MCS[n] can be estimated how mature is and be able to deal with the defined threat T[n] during the acquisition process. Table 4-8 indicates the TRL[n] levels alongside with their value. For example, if the MCS[n] is in theory then, "**MCS[n]** = **TRL[1]**". And with the aid of the expression, Equation 6, the MCS[n] has also its own percentage value to evaluate the mitigation process MP[n].

Table 4-8 Technology Readiness Level

TRL[n]	Compliance Levels	Mitigation Achievement
TRL[1]	1	0%
TRL[2]	0.875	12.5%
TRL[3]	0.75	25%
TRL[4]	0.625	37.5%
TRL[5]	0.5	50%
TRL[6]	0.375	62.5%
TRL[7]	0.25	75%
TRL[8]	0.125	87.5%
TRL[9]	0	100%

$$MP[n]\_MCS[n] = TRL[n] \quad \text{Equation 6}$$

Where:

MP[n]_MCS[n]: The value of the mitigation process.

TRL[n]: The value of the Mission-Critical system with an assigned Technology Readiness Level.

Lastly, when the MP[n] is estimated, it has to be included in the expression, Equation 3, to re-calculate the effectiveness level EL_B[n], of the benefit B[n], of the mission system MS[n] for the mission M[n] using the MCS[n] against the defined threat T[n]. With this approach, participants such as stakeholders and engineers should have a clear picture of how the mission M[n] requirement would have the probability to be successful or not.

4.3.6 Mission Integrity

An important aspect of the framework is to assign mission integrity levels, "**MIL(n)**" to the system. MIL are levels assigned to Mission-Critical systems MCS[n] to indicate their integrity on different missions and how they would impact the mission in case of a failure. Table 4-9 shows the different integrity levels, with their RRF and PFD, (the figures are adapted from

[52]). Similarly to the IEC standard, this procedure sets out targeted failure rates and extract MCS[n]'s MIL(n). Meaning that it defines the maximum number of times that a Mission-Critical system MCS[n] built to a particular integrity level, would be expected to fail in a given period of time.

Table 4-9 Mission Integrity Levels

Mission Integrity Level	Risk Reduction Factor (RRF)	Probability of Failure on Demand (PFD)
MIL 4	100,000 to 10,000	10^{-5} to 10^{-4}
MIL 3	10,000 to 1,000	10^{-4} to 10^{-3}
MIL 2	1,000 to 100	10^{-3} to 10^{-2}
MIL 1	100 to 10	10^{-2} to 10^{-1}

However, in this research, there is not a significant activity to assign MILs to MCS[n]s, hence, this is considered as future work of this framework based on current safety-related SIL level activities i.e. IEC 61508 and ASIL levels as per ISO 26262 (Table 4-10).

Table 4-10 MIL levels based on the Automotive Safety Integrity Levels (ISO 26262)

		O1	O2	O3
S0	D1	**QM**[24]	**QM**	**QM**
	D2	**QM**	**QM**	**QM**
	D3	**QM**	**QM**	**MIL1**
S1	D1	**QM**	**QM**	**QM**
	D2	**QM**	**QM**	**MIL1**
	D3	**QM**	**MIL1**	**MIL2**
S2	D1	**QM**	**QM**	**MIL1**
	D2	**QM**	**MIL1**	**MIL2**
	D3	**MIL1**	**MIL2**	**MIL3**
S3	D1	**QM**	**MIL1**	**MIL2**
	D2	**MIL1**	**MIL2**	**MIL3**
	D3	**MIL2**	**MIL3**	**MIL4**

4.3.7 Citation

This final step of the framework, "**Source – SO[n]**", in one hand is an independent step from the framework, yet on the other is very critical. Each and every step within the framework shall have an input reference source. For example, "The participant or the team recommended the MCS[n] as appropriate"; from this simple consideration, the definition would potentially help the framework user to be more confident with the decision making of the specific assigned requirement, according to the defined participant or team. In other words, the user would be able to judge how valid the information is. Also, it would be useful when further information is

[24] QM means Quality Management as per ISO 26262.

needed to understand the need for the assigned requirement. An example is presented later in this research as a case study in Chapter 6. A graphical representation of the framework's procedure and the sequence is depicted in Figure 4-1.

A brief summary of the framework presented in Figure 4-1:

Phase 1 – Identify the mission

Phase 2 – Item Definition (Benefit and Effectiveness Level)

Phase 3 – Threat Analysis and Risk Assessment

Phase 4a – Threat Estimation against Effectiveness Level

Phase 4b – Mission Functional Concept

Phase 5 – Mission-Critical System Estimation against Threat

Phase 6 – Re-analyse Mission System

Phase 7 – Mission Performance Estimation

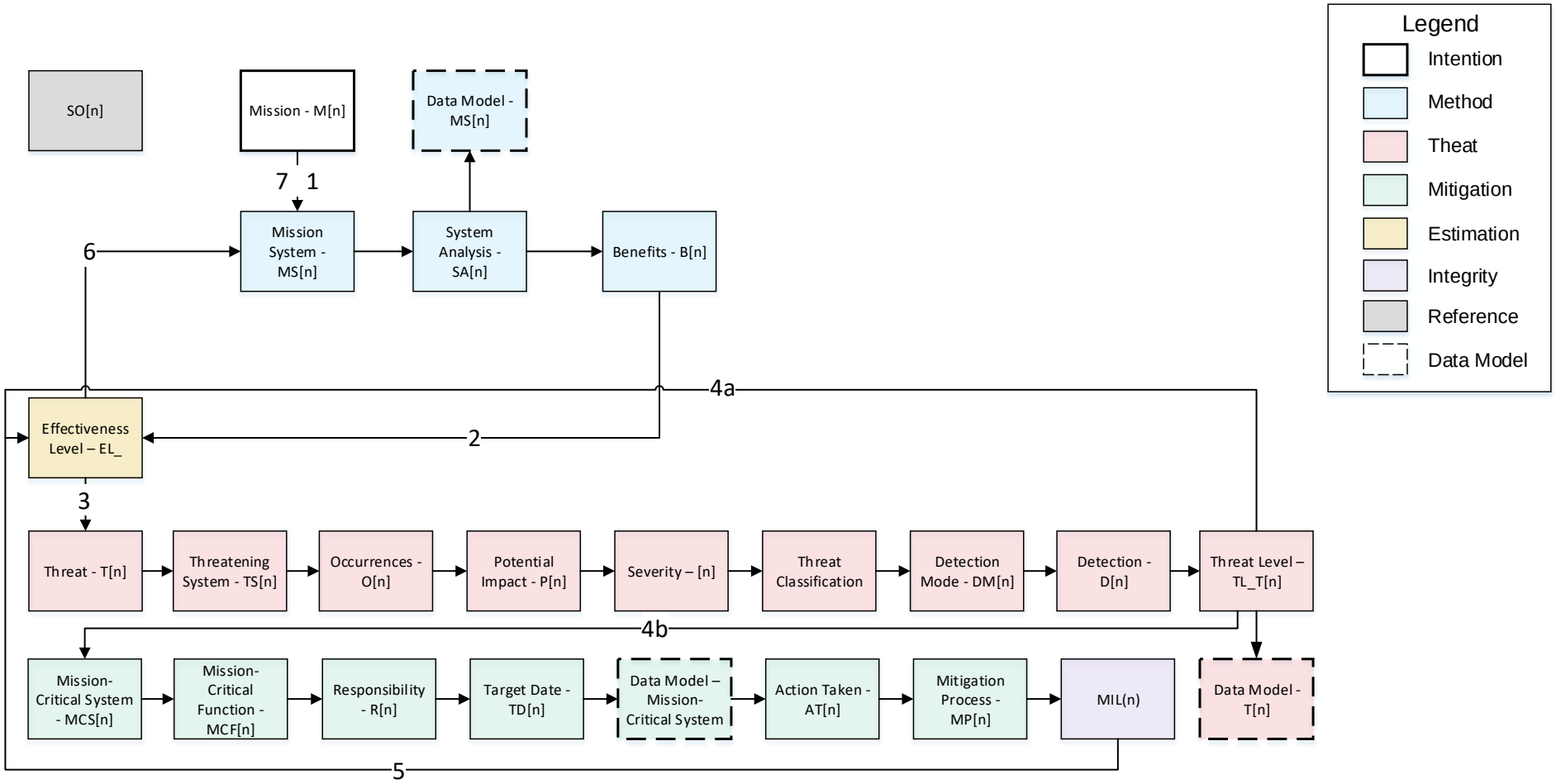

Figure 4-1 The framework for designing Mission-Critical and mission-related systems

4.4 The Context of the Framework

The framework depicted in Figure 4-1, it is the refinement of this research's proposal discussed in the introduction and shown in Figure 1-2. Using the framework, the user can estimate how would the systems involved in the mission performed when finally the system is developed. Also, the framework can be used to de-risk any integration issues that the Mission-Critical systems may face during the development.

Another major context of this framework is by following the V-Cycle process proposed in Section 1.4 Research Proposition, the framework can be easily applied to ensure that; the business gets what requested (steps 5 and 6); the case study is understood by multiple departments involved in the business(steps 1 and 4); and the engineers have a clear and well-defined set of requirements from the very early stage of the system development (steps 2 and 3).

4.5 Conclusion and Future Work

In this chapter, the proposed framework that could be applicable for designing Mission-Critical Systems and mission-related systems is presented. By collecting various approaches from around the industry and mostly from safety-oriented, the framework could potentially be used to design and standardise systems for any application (missions).

The framework can be useful for systematically define mission(s) M[n] and what system(s) MS[n] are critical for the specific mission. The effectiveness level indicates how effective the system MS[n] is for the mission M[n] and the degree to which it will be successful in producing the desired result. The calculated evaluation represents the percentage range of the system MS[n], as the probability for mission M[n] success.

From the framework, the user would benefit to identify threats T[n] with different identification levels in a simple and effective way. The methodology for addressing the threat, it would benefit stakeholders as well as the engineers if these step-by-step procedures are followed as explained earlier. Therefore, in this framework, the step that indicates the maturity of the program along with the input reference of each step, confidence can be increased; hence, mission success capability could be also increased into the desired result. The integrity level indicates in what degree the MCS[n] system can be capable to deal with the mission following threat analysis and recommended actions to reflect the potential threat.

Summing up, this framework's general purposes are,

1. Enhance the mission's success capability.
2. Enhance Mission-Critical system's dependability – Improve Effectiveness Level.
3. Provide the necessary information to all the participants, to easily develop a Mission-Critical system using an interoperable environment.
4. Provide the necessary information to decide what is the appropriate electronic architecture for the Mission-Critical system.
5. Addressing threats and provide traceability.
6. (Potentially) Provide mission integrity levels for each mission function[25].

In the future, the framework will be developed to a more refined degree, similarly to the industry. To achieve that, different approaches that enhance the desired result of the system will be added. Approaches such as, Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Layer of Protection Analysis (LOPA), Hazard and Operability Study (HAZOP), Independent Layer of Protection (IPL), Safety Requirements Specifications (SRS), Functional Safety Assessment (FSA) and the tools need for the elaboration of the concept to an actual testbed and afterwards, the migration from the testbed to a more realistic testbed demonstrator.

[25] This will be considered and expanded as a future work of this framework.

Chapter 5 Model Driven Architecture for Mission-Critical Systems

5.1 Introduction

Data Modelling is the act of exploring data-oriented structures as an abstract model, [56]. The data model can vary depending on the purpose. Some data models are used for high-level conceptual models and some others for physical models. Data modelling follows similar principles of class modelling. In comparison with class modelling, data modelling identifies entity types rather than classes. Entity types have their own assigned data attributes in the same way as classes have their own assigned attributes and operations. The associations between entities are also in common with the associations between classes; such as relationships, inheritance, composition and aggregation, are all used in data modelling. However, the difference between data modelling and class modelling is that data modelling is only concentrated in data rather than the exploration of behaviour and data domain. Hence, data modelling only explores data issues. When the focus is only on data, then the data modellers have the advantage to get the data "right" [26]. There are three basic approaches to data modelling.

5.2 Data Model Approach

Conceptual Data Model (CDM) – CDM can be also referred to as domain models. These models are typically applied to investigate domain concepts with project stakeholders. It is often used as a part of defining high-level requirements to estimate the determined attempts and investigation of the high-level structures and concepts of the programme.

[26] This will be the main focus of this chapter.

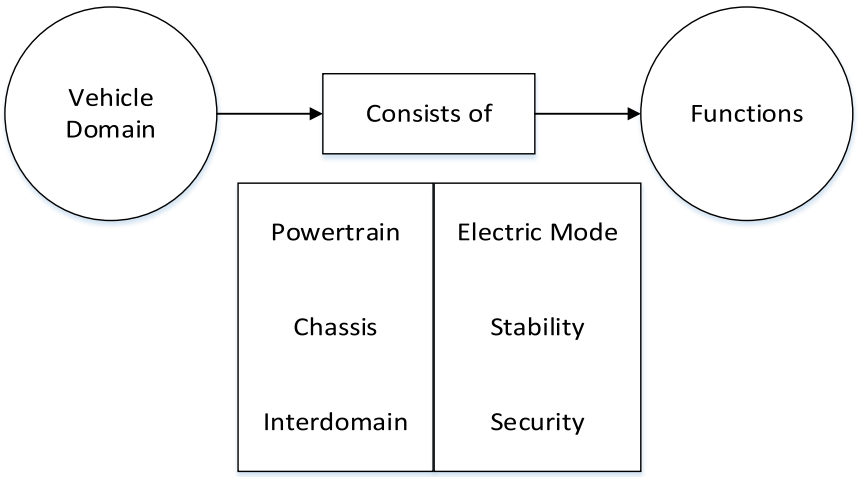

Figure 5-1 Conceptual Data Model Example

Logical Data Model (LDM) – These models are to investigate the domain concept and their relationships. LDMs are the logical representation of entity types, or simply as entity types. The entities and the relationship between entities are described through data attributes, Figure 5-2.

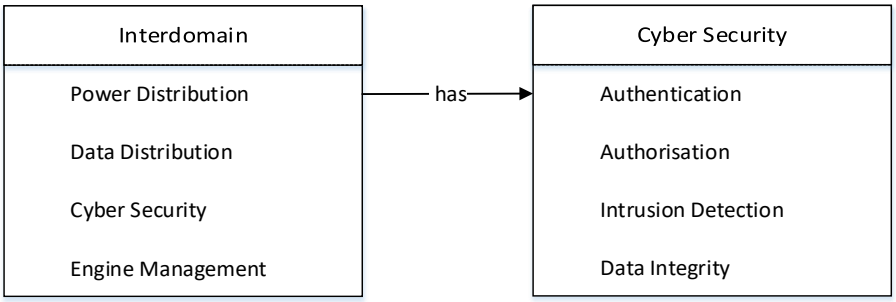

Figure 5-2 Logical Data Model Example

Physical Data Model (PDM) – PDM are used to design the internal schema of a database, representing the data tables, the data columns of those tables and the relationship between the tables, Figure 5-3.

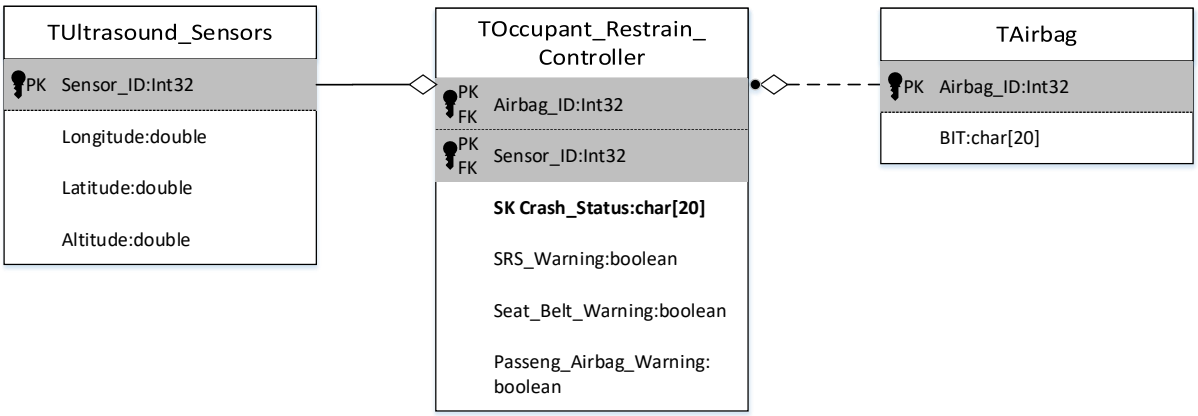

Figure 5-3 Physical Data Model Example

As depicted in Figure 5-1, the notation follows a very simple and straight forward definition that can be adapted swiftly by all the programme's participants. However, this notation can quickly become large and as a result, this could evolve into a disadvantage if the system becomes more complex. As can be seen in Figure 5-2 and Figure 5-3 they are relatively similar but what makes the difference, is the PDM provides more technical details. LDM's target is to make clearer and easier the investigation of the domain concept between the data modellers and the stakeholders. On the other hand, PDMs are targeting to provide a more detailed investigation of the database design that could reflect the programme's data base standard. For example in the PDM example, the entity association includes keys, (Primary Key (PK) and Foreign Key (FK)) to keep their relationships clear and easy to understand. Another, characteristic that differs from the LDM, the PDM also provides programme's database naming standard where an appended abbreviation of the entity name is attached to each column name. Furthermore, PDM is also pointing out data types for each entity type's data attributes.

5.3 Data Model Notations

The design, formalisation and documentation of data model's structures appears to be a really important tool especially, on how easily can be constructed. Many people have been interested in this approach and today data modelling has been developed into multiple and different notations. In this section, the most commonly used notations are presented starting from the pioneer of data modelling notation.

5.3.1 Chen's Notation

Peter Chen firstly introduced the entity/relationship modelling in 1976 [57]. His idea revolutionised the representation of data, which is still being currently used until today. Chen's idea was to relieve the complexity of describing data using the data modelling approach. That has been created in the late 1980s. The notation, however, could not support all the

subsequent techniques and by the time it became insufficient to use. An improved version of Chen's data modelling approach is object modelling.

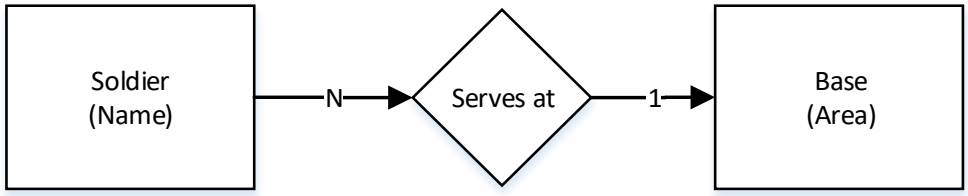

Figure 5-4 Chen's Notation

5.3.2 Information Engineering

The Information Engineering (IE) notation can be straight forward to understand and is well equipped for high-level logical and enterprise data modelling features. The demotion of this notation is the lack of supporting the identification of data attributes of an entity type. The assumption is that the attributes would be constructed with another graphical representation or simply defined in the guideline documents [58].

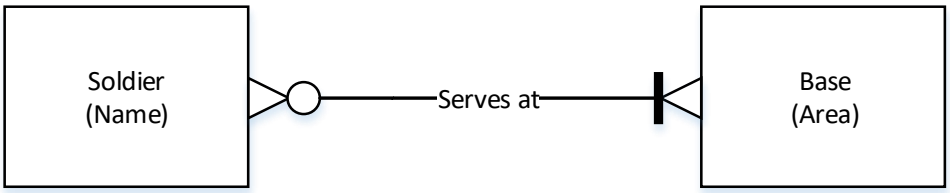

Figure 5-5 Information Engineering Notation Diagram

5.3.3 Barker Notation

The Barker notation is one of the most popular notation in the data modelling world. It is maintained by Oracle and is well equipped for all types of data modelling notations. The approach of sub-typing could become clucky with hierarchies that go several levels deep [59].

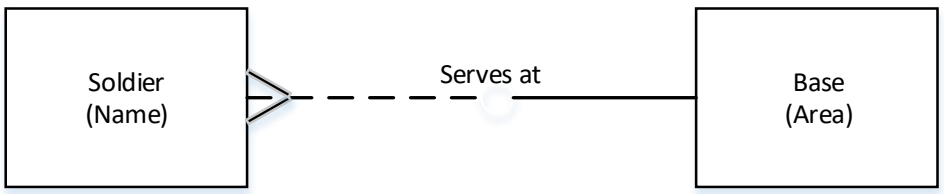

Figure 5-6 Barker Notation Diagram

5.3.4 IDEF1X

The Integration DEFinition (IDEF1X) is unexceptionally a misunderstood notation. This notation was initially created for physical models but accidentally applied for logical models.

When this mistake came forth, not only abandoned by the Department of Defense (DoD), who was the official user but also by everyone else [60]. Therefore, this notation has not been developed any further due to the accident occurred.

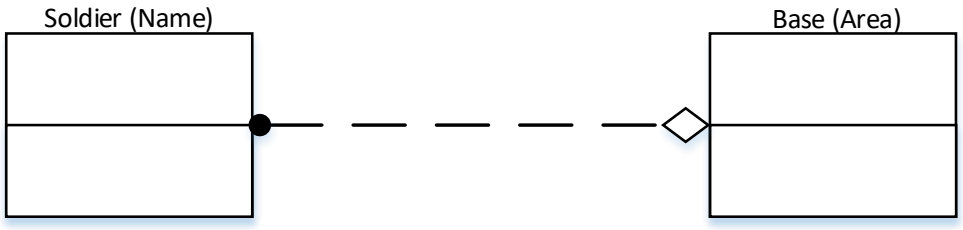

Figure 5-7 IDEF1X Notation Diagram

5.3.5 Unified Modelling Language

The Object Management Group (OMG) in 1997 developed the Unified Modelling Language (UML) to support modelling object-oriented systems and applications. This notation has been developed and constructed based on three notations, Bouch's Object Oriented Design (OOD), Rumbaugh's Object Modelling Technique (OMT) and Jacobson's Object-Oriented Software Technique (OOST) [61]. By combining these notations, the UML notation unifies their strengths into a single package to be the optimum notation for data modelling [62]. Several suggestions on data model profiling for UML appeared thus the Object Management Group (OMG) in December 2005 announced a request for proposals for data-oriented models and since then, this notation is growing and spreading exponentially[27].

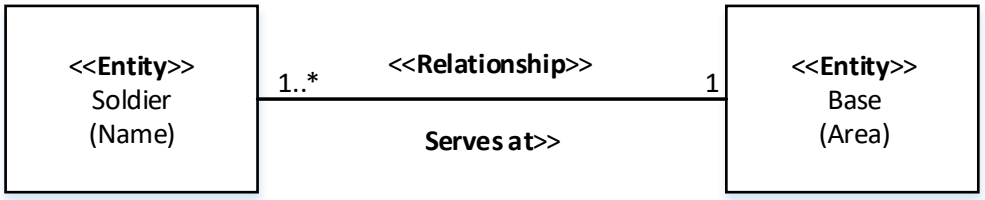

Figure 5-8 UML Notation Diagram

[27] *Note: One of the suggestions of improving the UML notation is the Systems Modelling Language (SysML). SysML consists of extra features that are useful for the system's design and development. Features such as "Requirement Diagram", "Behaviour Diagram", "Structure Diagram" and "Parametric Diagram".*

5.3.6 Extensible Mark-up Language (XML)

XML is touted as an external format for representing data. XML schema features both name and structural types, with a structure based on tree grammars. This notation is considered to be the perfect choice for small and intuitive schemas but because of this, the notation's visualisation makes it harder for a complex data to produce [63]. Contrariwise the previous data modelling notations, the XML notation is in the form of a human-machine script and do not possesses modelling diagrams.

5.4 Data Modelling Procedures

Data modelling not only is used to provide a clear reading of data models but also to enhance with efficiency of the programme's development. This can be achieved when systems engineers grasp the fundamentals of data modelling. This section goes through an introduction on the data model's procedures and in parallel, this section will identify gaps in these data models along with suggestions addressing these identified gaps.

5.4.1 Entity Types

Entity types are to describe the high-level structure of the data. It is for any component in a system that requires an explicit representation in the model. Entities possess attributes denoting to specific properties.

5.4.2 Attributes

Attributes are the named characteristic or property of a design entity. They provide a statement about the entity. Attributes can be one as well as many for describing the entity types. Attributes should be presented in a simple and comprehensive form in order to provide an easy and efficient grasp of the user. With this way, the development and maintenance efforts of describing models are more significant.

5.4.3 Data Naming Conventions

Standards and guidelines in data modelling are essential tools and should be included in the programme, provided by the administrators. Naming conventions in standards and guidelines shall be used in both logical and physical data models. For logical data models, the naming conventions should be human-readable friendly, meaning that people with any sort of discipline or background should be able to understand them easily and fast. For the physical data models, the naming conventions can be more technical oriented, similarly to the example in Figure 5-2 and Figure 5-3 used specifically for the engineers.

5.4.4 Relationships

Entities types have their own relationship with other entities types, similarly to the real world. For instance, the example in Figure 5-8, is used to represent different data model notations, “Soldier <<**Serves at**>> Base”, where soldier and base (military base) have their own relationship indicating their names and roles at the same time. Relationships, in an application development environment, are relatively similar to “associations” between objects. Relationships in complex programmes are likely to pollute the environment during the development process with redundant information. To avoid this, it is highly recommended to pay particular attention during this development phase.

It is essential to declare cardinality values between entities types but, that can be optional. Cardinality values are the specification of how many instances of a first entity type may or must exist for each instance of the second entity type. Also, how many instances of a second entity type may or must exist for each instance of the first entity type. For each direction of a relationship, the cardinality values are constrained. In UML for example, the cardinality values are represented as, one to one (1..1), many to one (*..1) or many to many (*..*). Using the same example depicted in Figure 5-8, “Soldier” can be from one to many (1..*) but soldier(s) belong into only one (1) base. It is a property of the entity type that specifies if the value is mandatory or optional.

In relationships there are other elements or rather another way to specify the relationship between the entity types and are as follows in the diagram, Figure 5-9:

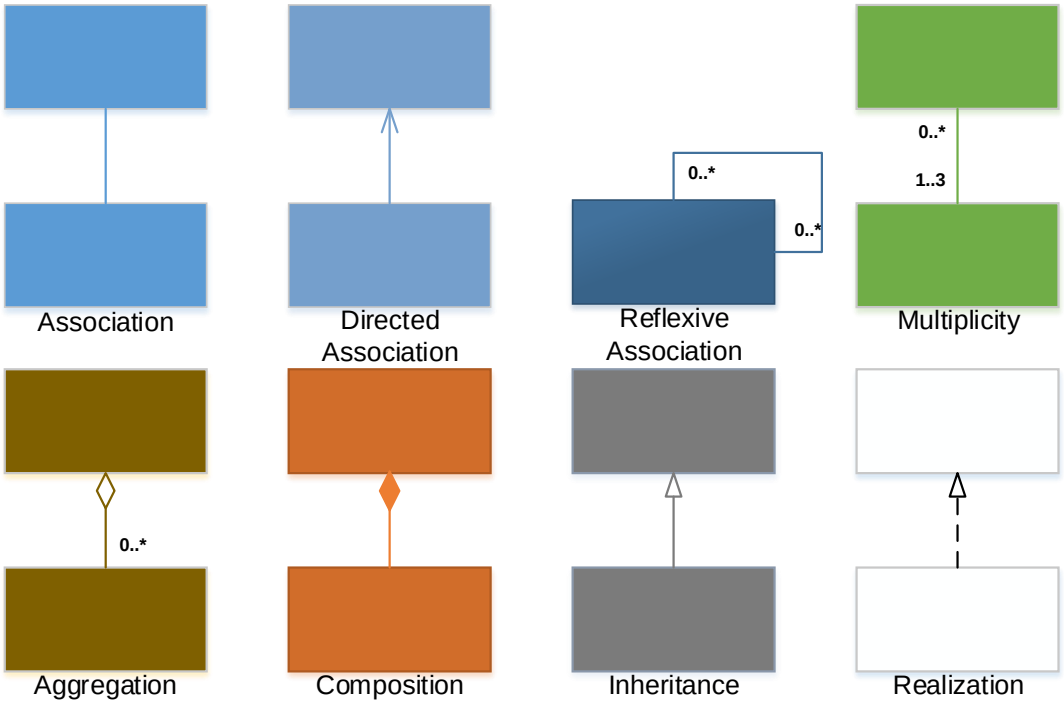

Figure 5-9 Relationships

- Association – The structural relationship between two model elements that show objects.
- Directed Association – Is a structural relationship between two entity types that can navigate objects.
- Reflexive Association – It represents the entity type that has one or more functions or responsibilities.
- Multiplicity – The active logical association when the cardinality of a class in relation to another.
- Aggregation – Depicts a classifier as a part of, or as subordinate to, another classifier.
- Composition – An aggregation forms strong ownership and a coincident lifetime as part of the whole.
- Inheritance/Generalisation – The relationship between the entity type and sub-entity types (sub-class of an entity type) whereby, the sub-entity types have all the properties, operations and associations of the entity type.
- Realisation – The implementation of the functionality, defined in one entity type to another.

5.4.5 Data Model Patterns

Data model patterns are critical when a data structure is constructed. These patterns are used in describing in great detail the entity types and their relationships. It is also used to separate the representation of fixed and variable data.

5.4.6 Keys

To assign keys to entity types, two basic methods are used. The first method is the "**N**atural **K**ey" that can be assigned to one or more existing data attributes and is unique in the overall programme or environment. The second method is the "**S**urrogate **K**ey" that can be assigned to data attributes within the same entity type table like the one in the natural key. Although the surrogate key might not have a direct meaning within the entity type table, it can provide an additional "flavour" that the natural key cannot provide, see Figure 5-3, <<**SK Crash_Status>>**. The intention of the "Primary Key <<**PK**>>", is to provide a unique characteristic of a specific data attribute, whilst, the "Surrogate Key <<**SK**>>", is to provide a characteristic that is indirectly related with the table, but could be useful for other purposes. Both keys can be used equally, but for a slightly different purpose.

Using the example in Figure 5-3, it can be noticed that the entity type table, <<**TOccupant_Restrain_Controller**>>, uses attributes that supposedly considered as "unique". In that case, if a unique attribute is used in a different entity type table, it is then considered to as "**F**oreign **K**ey", <<**FK**>>.

Using keys to indicate the relationships between entity types and data attributes, can be a powerful method that could provide the ability to easily discover and understand the purpose and importance of each data.

5.5 Mission-Critical Data Model in Data Model

By reaching this section, it can be noticed that data modelling is an important and powerful tool. Since the first introduction of Chen's notation, the notation is still in used but, in a form to better reflect today's needs. Despite that, the data models that these notations describe have some missing elements, the gaps aforementioned earlier. These missing elements are the Mission-Criticality elements of the data attributes; entity types; and relationships. In this section, the methods used to achieve Mission-Criticality in the data models will be demonstrated. However, it is important first to understand the reason for Mission-Criticality in the data models.

5.5.1 The reason of Mission-Critical Data Model in Data Models

A common phrase used today for the development of a platform that follows the IOA architecture's principles is,

"*Get the right data, at the right place, at the right time*". [64]–[68]

That is the inspiration behind the "Mission-Critical Data Model" proposal; acting as an addition to the data modelling approach as discussed in Section 4.2.21.

Figure 5-10**,** represents an automotive system, the Occupant Restraint Controller (ORC) system, in a UML notation, constructed from various examples in [69], [70]. As can be seen, there are three entity types, with their data attributes, data types and relationships. The following sections are discussing the proposed additional elements in this UML model example to address the "Mission Criticality" gaps.

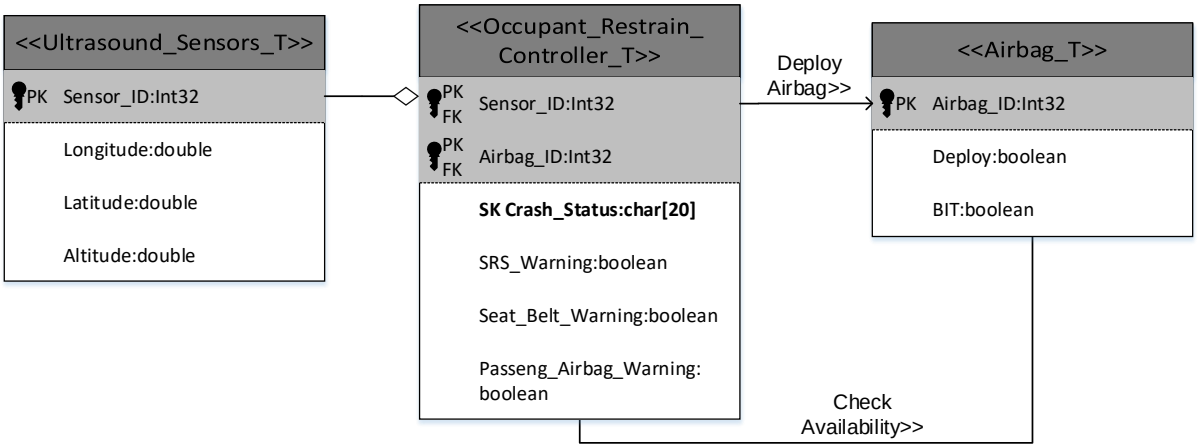

Figure 5-10 Passive Safety System – UML Notation[28]

Mission-Critical Data Attributes

Starting from the data attributes, the data attributes are used to describe the physical or abstract property of an entity type. To enhance the definition of the available data is by adding critical attributes along with their influence they may have on other entity types, systems, sub-systems and eventually to the overall mission. Data attributes can be distributed and used in many other entity type tables, which means they can be also used in different systems or sub-systems or re-used in different applications (missions). Assuming that there are two different scenarios on how the data attributes within the <<**Airbag_T**>> entity type can be used.

[28] The data model is simple and could be in more detailed for time driven purposes.

- **BIT:boolean** – Build-in-Test (BIT) is a mechanism that allows the airbag system, in that case <<**Airbag_T**>>, to test itself or to be tested if the system is ”healthy”. Reason being for this data attribute is to enhance reliability capabilities of the system. Assuming that the BIT provides negative feedback but cannot declare what the negative feedback means. Negative feedback may mean a risk for the vehicle’s occupants thus, is critical. Also, negative feedback may mean to just raise warning information for the driver that the system is in a failure mode, thus, the negative feedback may not be that critical.

  Using these examples, the data attribute cannot declare its criticality by its own and when it comes to IOA platforms, the platform will not be able to distinguish the importance of the negative feedback thus, the mission might be aborted without having any reasonable risk of compromising the mission.

- **Deploy:boolean** – This data attribute is considered as the initialisation or the command from the system to deploy the airbags. Either during or pre-collision, this data attribute is extremely critical (safety-critical). For this reason, this data, within the <<**Aribag_T**>> – entity is more critical than the aforementioned data attribute in terms of minimizing the risk of the impact. But on the other hand, if BIT:boolean fails to declare any type of failure on the airbags, the deploy data attribute will be meaningless. Therefore, all of the data attributes can be formed as Mission-Critical data attributes depending on who is using it and for what reason (mission).

These two scenarios prove how a data attribute can be assigned as Mission-Critical or non-Mission-Critical. The same can be applied for the entire entity types. A possible way to address this is by adding elements assigning each data attribute, or entity type or relationship as described in Section 4.3. A more detailed description and application are analysed further in Chapter 6.

Mission Exchange Information

The second reason on “why Mission-Critical Data Model” is important, is when exchanging mission information across the networked entity types. This can be accomplished by exchanging information such as Mission-Critical system’s effectiveness and integrity levels; threat affects levels, and real-time responsiveness levels within data the attribute(s) or entity type(s) or relationship(s). For example, in Safety-Critical systems, the Safety Integrity Levels (SIL) are used to assign the level of risk-reduction provided by a safety function[29]. Following

[29] See Section 3.3.5 Safety Integrity Level.

this research's framework, the "Mission-Critical Data Model" procedure is used as the data model responsible to reduce the risk of mission failure.

However, this can be exploited and used not only to indicate the Mission-Critical system's or mission function's integrity and effectiveness but also to indicate the integrity of the specific data attribute(s) or entity type(s). Moreover, the exchange from sub-system to another sub-system could have the knowledge and the freedom to decide whether the data (function) is critical or not at run time. This could improve performance in network's bandwidth, Central Processing Unit (CPU) usage, Random Access Memory (RAM) usage, memory capacity etc. but this is out of the scope of this thesis.

Mission-Critical Relationships

Another important aspect of the "Mission-Critical Data Model", is the criticality between relationships. As aforementioned earlier in Section 5.4.4, relationships are used to indicate the relationship between entity types. However, relationships can be also used to indicate the "Real-Time Responsiveness" degree. Based on the same example that is depicted in Figure 5-10, the entity types have two relationships, <<**Deploy Airbag**>> and <<**Check Availability**>>. Consider that the ORC system, is responsible for checking airbag's status using the BIT data attributes, as well as, deploy the airbag in the event of an accident.

Nevertheless, the additional element in this example is, within what real-time responsiveness the <<**Deploy Airbag**>> and the <<**Check Availability**>> relationships should occur. It is not clear whether the data is important, or what priority and criticality is and when must be executed in which time constraint. There is not an indication declaring the difference from each other in terms of criticality. These are very abstract concerns when two or more entity types have two or more relationships. Using the Table 4-7, this could be potentially useful for systems engineers and architects to decide the importance of the entity type and their relationships; or as an optimum option to achieve it effectively and efficiently (e.g. the clear definition of User and System Requirements - Communications networks, electronic processors and controllers, etc.).

These were some considerations of using "Mission-Critical Data Model" in data modelling procedures. However, there is an approach that could potentially help to swiftly achieve the data model process, discussed in Sections 5.2, "Mission-Critical Data Model" and that is the MDA approach.

5.6 Model Driven Architecture

Model Driven Architecture (MDA) is an approach that is used in the systems engineering domain to improve product development and delivery. The approach was initially launched in 2001 by the Object Management Group (OMG) to support software and system development throughout model-driven engineering. The main objective of the MDA is to provide a set of specifications for the system's functionality and behaviour. These specifications are expressed in models. Instead of writing the code manually, the MDA approach with the help of a data modelling tool, it is possible to regenerate automatically an application code fast. Additionally, this approach reduces the implementation and integration risks when an activity is designed at a very early stage. MDA is an approach that is currently used for the development of IOA systems. The MDA approach is designed to achieve the followings objectives.

Technology obsolescence – The easy integration of new implementation infrastructures and the support of the existing designs.

Portability – The rapid migration of an existing functionality into new environments and platforms as dictated by the programme.

Productivity – Helps system architects and developers to pay more attention to the core logic of the system instead of time-consuming or tedious development tasks.

Quality – Uncertainties are formally separated from this approach along with the consistency and reliability of the model.

Integration – The integration of legacy and/or external systems can be implemented by this approach.

Maintenance – Provides simplified maintenance tasks for testers and analysts, as well as, the direct access to the specifications of the system.

Testing and Simulation – Requirements and infrastructures are directly validated and tested. This can be achieved during the development phase before the development if fully completed.

An illustration of the MDA process is presented in Figure 5-11. Initially, the system requirements can be defined and specified into the Platform Independent Model (PIM) model. By using standards and specifications, the model can be constructed in a formal way; the Unified Modelling Language (UML) can be used to accomplish that. The objective of the PIM model is to specify data, operations, functions and modes of the system independently from the platform it may be integrated. In order to organise and standardise the data, as well as

facilitating a long-term improvement in interoperability and upgradability within the model, the PIM model shall be used.

A Platform Specific Model (PSM) contains elements of the specific software platform. PSM can be generated from the PIM model either manually or automatically when the appropriate tools are used. For example, the UK MOD has developed a tool for their the Def Stan 23-009 standard that translates their PIM models to PSM models automatically, (GVA Translator) [71]. Moreover, with this translation, the PSM model is able to embed the chosen software architecture strategy which refines the PIM model based on the specifications.

The last step of the MDA approach is the Platform Specific Implementation (PSI). PSI embeds the chosen middleware technology and is able to explain the usage of the specific platform using more refined information. Each electronic component is developed to satisfy one or a more specific task(s) thus, the components can only generate or receive a set of specific data for their operation. Therefore, when these electronic elements are integrated into an IOA architecture, the broadcasted data cannot be ensured if is critical or not. This is due to the nature of the MDA approach which is following the similar line as the UML approach for describing their models. Thus, considering the “Mission-Critical Data Model” approach, criticality between electronic elements in platforms can be distinguished from the very beginning of the programme development. Additionally, when using MDA’s principles and achieving its objectives, the development of the Mission-Critical data model can be accomplished swiftly.

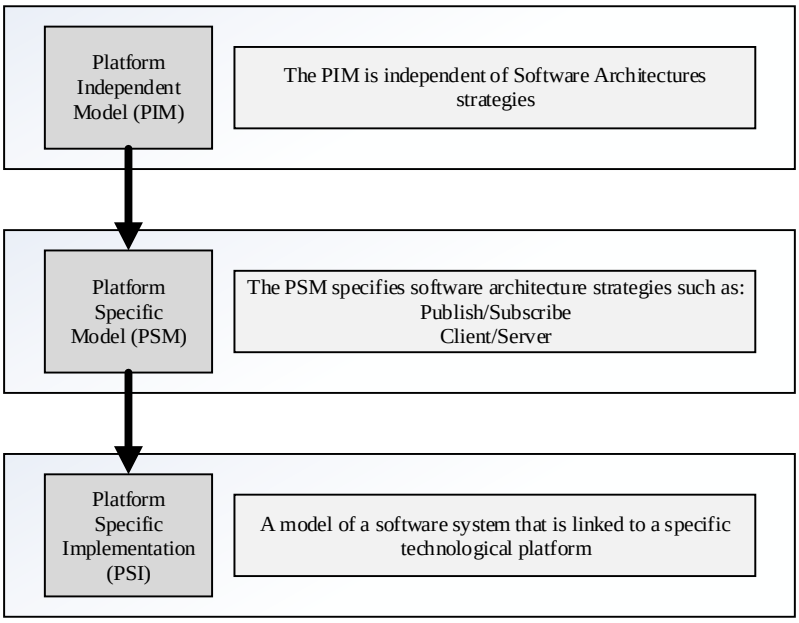

Figure 5-11 Model Driven Architecture Approach

Nevertheless, the MDA is founded upon the combination of abstraction and automation. Abstraction refers to the model construction which is very simple yet very powerful that enables the observation of the system's architectural requirements without the prejudice of the implemented technology. The top-level architectural requirements are defined in terms of use cases and domains by representing the required capabilities and components of the system. The system engineers and architects themselves are responsible to define classes, attributes, relationships, operations and states encapsulated within each domain. All these in an abstract and platform independently. The use of the automation eliminates the need to maintain derived artefacts, such as the design model, message definitions and documentation.

5.7 Proposed Mission-Critical Electronic Architecture and Electronics Instrumentation

5.7.1 Introduction

This section presents a proposed IOA architecture structure for Mission-Critical applications, systems and platforms. The main objective of this architecture is to enable the reader to have a better understanding of how this research's framework can be used in the development of an IOA E/E architecture. The architecture follows the proposed Mission-Critical system's taxonomy that is described in Chapter 5. The structure of the architecture also uses paradigms from other critical related systems which are going to be discussed further.

This proposed architecture aims to fuse systems and software modelling and simulation capabilities; modular open system architectures; and device integration techniques into a single package, so that to enable rapid design, development, verification, certifications and deployment of interoperable, platform portable and manoeuvre embedded mission criticality.

5.7.2 Background

The ANSI/IEEE 1471-2000 standard which is a superseded IEEE standard for describing architectures defines architecture as,

"*The fundamental organisation of a system, embodied in its components, their relationships to each other and the environment and the principles governing it's design and evolution"* [72].

In Vetronics and generally in Information Technology (IT), architecture is the organisational structure of systems, networks, data, functions and technologies. In Vetronics and specifically in the UK MoD platforms, the architectural construction and orchestration are specified in the

Def-Stan 23-009 specification. An existing platform that uses Def Stan's specification, is the fully GVA compliant platform, Foxhound [73].

A Vetronics architecture mainly consists of,

- Sensor and instrumentation data acquisition.
- System and sub-system internal data/information communication.
- Sub-system, functions, platform power and sensors/effectors control.
- Platform power distribution and management.

There are three main architectures are satisfying a Vetronics architecture for command and control platforms. These E/E architecture are described in the following sections which are extracted from [74].

5.7.2.1 Common Platform Architecture

First described architecture is the Common Platform architecture. This architecture is in the simplest form of an electronic architecture, as shown in Figure 5-12. A single data network is used to receive and transmit data across all the sub-systems within the platform. In this architecture, the sub-systems can be either connected to the network via a single node or through a shared node that connects multiple sub-systems.

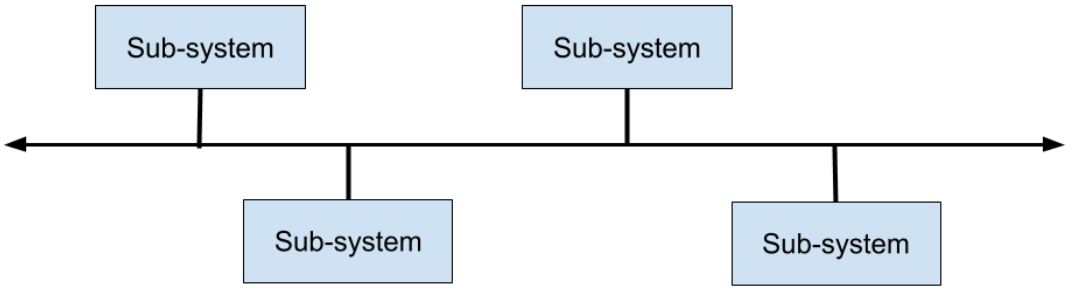

Figure 5-12 Common Platform Architecture

The complexity of communication and integration of sub-systems is low due to the single communication network used for all the sub-systems. Therefore, the commonality between sub-systems is easy to implement. Despite this, the network technology used for this architecture is able to support both system capabilities from different industries as well as supporting longevity technologies. Mission-Criticality in this architecture is an issue. Systems that are Mission-Critical and non-Mission-Critical are able to share information using a common communication network, therefore, criticality in data, for example, the data integrity (security) of the data, in this architecture may be exposed resulting in a major impact of the entire platform also to the mission.

5.7.2.2 Data Type Architecture

The data type architecture is specified on sub-systems that share high and low data exchange bandwidths. In this architecture, the technologies of digital image sub-systems are differentiated into high bandwidth and low bandwidth sub-systems. Each aforementioned sub-systems can be implemented separately, depending on the required technology.

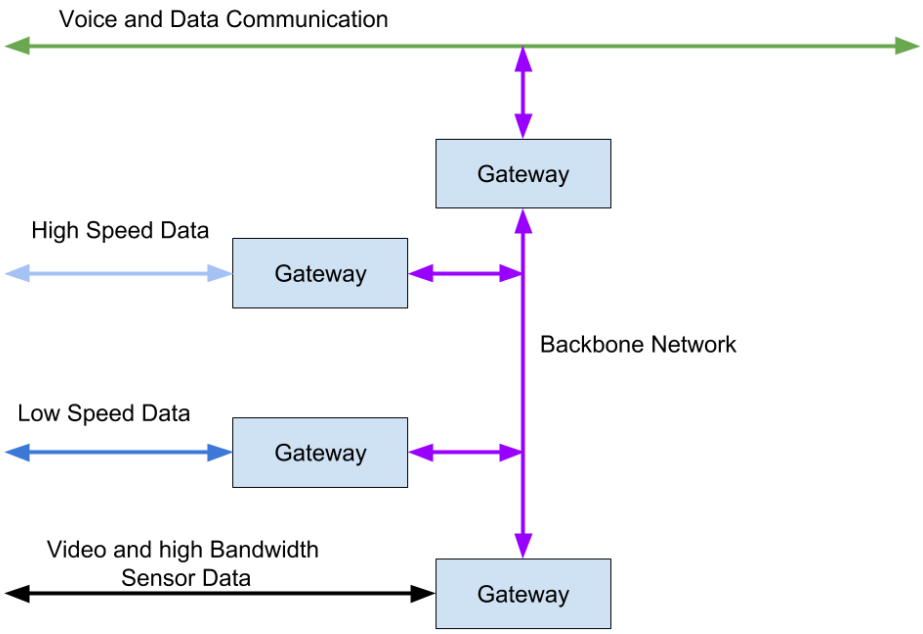

Figure 5-13 Data Type Architecture

However, when the data type determines the organisation of the architecture the costs of the specification are relatively low. Therefore, using multiple technologies within a common architecture would require multiple gateways and bridges to support the communication between the sub-systems. When multiple gateways and bridges are integrated into architecture, Mission-Critical and non-Mission-Critical components are difficult to identify and the Mission-Critical functions could potentially be compromised.

5.7.2.3 Functional Architecture

Functional architecture is different from the previous architectures due to its features and capabilities that have. In this architecture, the organisation of sub-systems and the technology used, are defined by the functions.

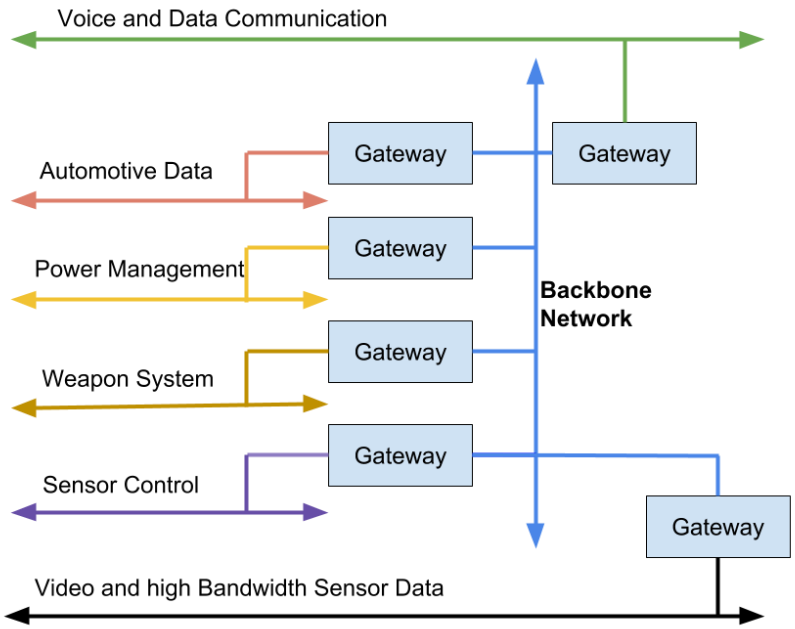

Figure 5-14 Functional Architecture

Mission-Criticality between functions on this architecture can be distinguished between Mission-Critical and non-Mission-Critical components. Therefore, the costs and the effort needed for the implementation and testing can be reduced, based on the different Mission-Critical Integrity (MIL) levels. Although this architecture can distinguish Mission-Critical components from non-Mission-Critical components, it requires multiple network technologies. As a result, the need for gateways and bridges for data exchange can be significant.

5.7.3 Proposed Mission-Oriented Architecture

Based on the three aforementioned architectures and the current trends in the communication patterns (discussed in Chapter 2), an appropriate architecture for a Mission-Critical systems could be considered as the one depicted in Figure 5-15. To support the interpretation of data and functions, various elements are included in this proposed architecture. The nodes in circles, demonstrating the production or collection of the “raw data”, to or from either applications or services, systems, items[30], components or devices. On the other hand, the nodes in squares are processing the “raw data” into Mission-Mission critical data, following similar line as described in Section 5.5.1, the reason of the Mission-Critical Data Model in Data Models.

[30] From the ISO 26262, systems or array of systems.

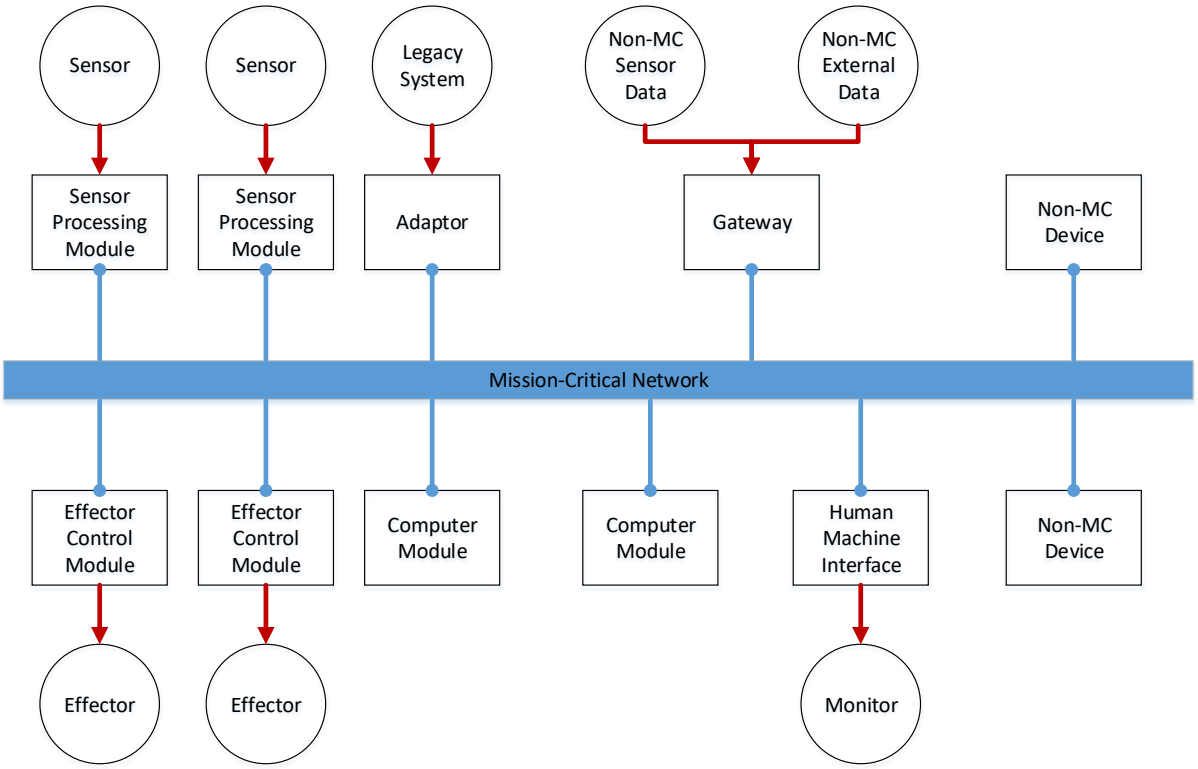

Figure 5-15 Mission-Critical Oriented Top Level Architecture

The breakdown of this architecture is modular and has a distributed design. All the communications utilise the Mission-Critical network that is not dedicated only to Mission-Critical related systems but it can be also used for other critical systems. Furthermore, the system uses a dedicated Mission Data Model with strict component allocation to the blue buses representing the Mission-Critical network inside the network environment. The red buses that can be accessible from outside the network environment, (i.e. other platforms). The breakdown of the architecture is discussed in the following sections.

5.7.3.1 Sensor

> "*The portion of a channel that responds to changes in a plan variable or condition and converts the measured process variable into an electric, optic or pneumatic signal*" [75].

Sensor represents all the candidate sensors of the mission-related system. This module is for detecting and responding to any incoming threat[31]. The signal from the sensor(s), in most cases, is raw data collected and modified into a format that is essential for the mission. The sensor should be directly connected to the Sensor Processing Module (SPM) for the exploitation of the sensor's raw data.

[31] See Threat definition in Section **Error! Reference source not found.**

5.7.3.2 Sensor Processing Module (SPM)

"*A main system processor unit that executes operating system code and manages system resources.*" [76].

The SPM is the module that is capable of receiving the raw data from the sensor(s). The SPM converts and enhances the data into a more usable format that can exploit depending on the mission. The connection between the sensor and the SPM and can be either analogue or digital depending on the sensor and probably not secured. However, the information is then transmitted through a Mission-Critical network and the data is as defined in the Mission-Critical Data Model[32]. The SPM acts as a gateway to the network protecting it from threats[33]. With this approach, it should be offering an easy configuration, essential to the mission.

The SPM, therefore, receives the data/information from the candidate sensor and extra qualities can be added. The user has the freedom to customise the data and modify it into the corresponding real-time environment, criticality level and Mission-Critical Integrity Level. The SPM should also be able to communicate directly with the Effector Control Module (ECM) in order to increase the system's "dimension" and "properties" qualities.

5.7.3.3 Computation Module (CM)

"*A functional programmable unit that consists of one or more associated processing units and peripheral equipment, that is controlled by internally stored programs and that can perform substantial computation, including numerous arithmetic or logic operations, without human intervention.*" [77].

The CM is collecting and processing data from, SPM(s), ECM(s), another Computer(s) or any other type of source as shown in Figure 5-15. The CM's responsibility is to decide the best action for each information. The action might be either required human factor's interaction or it can be fully automated. The modular design allows the system designer to use a single or multiple CM that may deal with the same or different types of threats at the same time, providing fault tolerance and distributed design.

The CM's purpose is to communicate with the appropriate ECM(s), depending on the type of action decided. When there is no CM available, the SPM(s) and ECM(s) is responsible for countermeasure threats themselves.

[32] A further explanation will be provided in ChapterChapter 6.
[33] Discussed in Chapter **Error! Reference source not found.**.

5.7.3.4 Effector Control Module (ECM)

"*A device used to control in a predetermined manner the electric power delivered to the apparatus to which it is connected.*" [78].

The ECM is responsible for controlling effectors according to the information received either from the CM or from the SPM. This module has many commonalities in data and functionality as in the SPM. Additionally, it offers re-configuration options for the effector to achieve modularity and dynamic configuration.

5.7.3.5 Effector

"*A transducer that accepts a data sample or samples*[34] *and converts them into an action.*" [79].

The effector is the representation of the candidate effectors used in Mission-Critical systems. This module is constructed upon available effectors needed to be installed on the platform. Each of the candidate effectors carries specific attributes thus, the module is using message specifications, common to the overall architecture.

5.7.3.6 Gateway

"*A device connecting two computer systems that usually use different protocols, or to connect two independent networks.*" [80].

The gateway makes available any relevant data from the non-Mission-Critical system(s) or devices on the Mission-Critical environment network to allow the CM and the other modules to process their data. This data can be from on-board sensors e.g. speedometer, wind speed/direction or off-board from other vehicles and ally forces. The gateway should ensure, that this data is active and it protects the network from excessive use of available bandwidth.

5.7.3.7 Adaptor

"*A device or series of devices designed to provide a compatible connection between the test subject and the test equipment.*" [81].

The adaptor can utilise existing on-board Mission-Critical systems by providing a bi-directional communication to any existing Mission-Critical system. It allows the collection of sensor data from existing Mission-Critical systems and the control of any available effector on Mission-Critical systems. The adaptor also adheres to the Mission-Critical Data Model, as the other

[34] Can be an analogue or digital signal.

Mission-Critical system's components do. It is important to design the appropriate configuration to ensure that there are no bandwidth related issues.

5.7.3.8 Non MC Sensor Data and Non MC External Data

Non Mission-Critical Sensor Data

A data collection from a non Mission-Critical sensor.

Non Mission-Critical External Data

A data collection from a non Mission-Critical system/equipment/components etc.

This is useful for collecting information from other systems or sub-systems integrated on the platform. It can be also possible that the platform is able to receive information from other platforms externally.

5.7.3.9 Non-Mission-Critical (MC) Device

"*An independent test resource; a test resource may be either manually or automatically controlled.*" [82].

On platforms, it is possible that different devices are integrated and these devices should be categorised as non-Mission-Critical device. With this approach, the systems engineers and architects should pay particular attention on how the data from such devices should be controlled and/or handled.

5.7.3.10 Human Machine Interface (HMI)

A user interface that is defined as:

"*Includes keyboards, displays, keypads, touch screens and similar devices to allow human interaction with a system.*" [83].

This part involves the user operating the platform such as driving or giving permissions to systems for actions. Either way, when electronics and operators are combined, it is essential that an HMI is required.

5.7.3.11 Monitor

"*A software tool or hardware device that operates concurrently with a system or component and supervises, records, analyses or verifies the operation of the system of component.*" [84].

A monitor can be used for tracking the distributed data within mission-related electronic architectures, for various reasons; operation, maintenance etc.

5.8 Conclusion and Future Work

In this chapter, the approach used for developing conceptual systems following IOA principles is discussed. In this chapter, the reader should be able to understand how important the data model processes are in the designing and development of such systems.

However, it has been identified that current data modelling procedures have some missing elements in their approach. These missing elements are mostly Mission-Critical oriented elements or could be also mentioned as "Mission-Critical flavour". Hence, not only the missing Mission-Critical elements are identified, but also proposed recommendations are presented to enhance data model architecture beyond their identified capabilities.

It is important for systems possessing these Mission-Critical elements in their design and operation. System engineers and architects could have more precise definitions when a mission-related system is developed in their Mission-Critical system design and development life-cycle. Moreover, the confidence of achieving successful systems in real environments could be also increased with the recommendation addressing the missing gaps in current data models. This should lead to an overall dependable Mission-Critical system for achieving successful missions.

In the future, the proposal for adding Mission-Criticality flavour in the data models can be analysed and applied in the SysML notation. SysML is designed explicitly for systems, therefore, it can potentially provide a more precise definition and guidance for the development of Mission-Critical and mission-related systems. Furthermore, that could be useful for better understand and offer a clearer picture between all participants involved in the Mission-Critical systems development.

Also in this chapter, a review of how an IOA Mission-Critical electronic architecture consisting of E/E/PE system should be constructed is presented. Constructing an electronic architecture for platforms using such systems requires sophisticated decision makings due to the cost and time consuming when changes are required. With the proposed Mission-Critical architecture

there is a potential that, even though some components needed to be integrated or removed, the data and functions will be still effective for the mission.

This chapter provides awareness of the concepts and the need for such Mission-Critical oriented architectures. However, there is still more discussion needed for these architectures. For example, communication networks, hardware, software, operating system etc. Therefore, as future work, this chapter will be looking at other Mission-Critical system's electronic architectures and expand/modify this proposed Mission-Critical architecture. This will be extracting requirements for an optimum Mission-Critical oriented architecture, that will be useful to achieve types of different missions.

A testbed could also be constructed as a proof of concept provides,

- Mission's rapid prototyping requirements.
- Mission-Critical functions interpretation and criticality exchange.
- Early de-risk capabilities, in the early stages of the mission-related system's development.
- The use of affordable tools and components to achieve Mission-Critical capabilities.

The migration of high-level implementations to a low level, effectively and efficiently.

Chapter 6 Mission-Critical System Use Case: Defence Aid Suite (DAS) System

6.1 Introduction

This chapter represents the implementation of this research's approach, as a proof-of-concept, using an existing Mission-Critical system. The scope of this chapter is to analyse a system which is considered as Mission-Critical system and prove that regardless the criticality of the system, different definitions, approaches and implementations can vary, depending on the application, capabilities and so on. Also, an early de-risking estimation is presented in this chapter, indicating and addressing the risks of the system's development and integration; and to observe the system's application performance including different environment and threat scenarios. For this case study, the scenario will be composed by a mission, mission system, a threat and a Mission-Critical system. The mission is "survivability", mission system is a "survivability system – the Defensive Aid Suite (DAS)", a threat is a "30mm Gun Armour-Piercing Discarding Sabot (APDS)" and the Mission-Critical system is a "smoke grenade system". This case study is using two vehicles to represent two different scenarios. The two different vehicles are classified as, passive armoured vehicle (**Case 1**) and light armoured vehicle (**Case 2**). Using those two different vehicles, in armour structure, the mission can be changed even though when the items[35], threats, E/E architectures and so on, are the same. With this approach the recommendations, proposals and Mission-Critical concept requirements can be identified by the stakeholders, systems engineers and architects, efficiently and effectively with a precision at the very early stages of the systems' life-cycle.

6.2 Background

Today, there are "countless" E/E/PE systems developed to achieve pre-defined tasks (missions). In the military platforms, these systems are known as Mission-Critical systems. One of those Mission-Critical systems is the Defence Aid Suite (DAS) system. DAS system is known as a survivability system and is used to improve the overall battlefield effectiveness and survivability of the platform and/or the crew against threats[36]. This system is composed of various sensors, computers, effectors, Human Machine Interface (HMI), algorithms and so on. There are two main functional types of DAS systems which are, an autonomous DAS and semi-autonomous DAS systems. Both functional types are able to detect, classify, provide effective warnings to the crew and counter measurements when threats are detected.

[35] From the ISO 26262, systems or array of systems.
[36] See Chapter **Error! Reference source not found.**, Section **Error! Reference source not found.**, page 60, "Mission-Critical System Taxonomy – Threats".

However, in general, the DAS systems are classified into three main categories, Soft-Kill, Hard-Kill and other effective sensor and countermeasure actions,

- Soft-Kill – Avoids incoming threat(s) by using counter-measures in order to defeat the threat instead of counter-attacking the projectile.
- Hard-Kill – Prevents a hit or reduce its penetration capabilities by directly impacting on the incoming threat.
- Effective Sensor and Countermeasures Action – Avoids the impact by applying pre and post firing human/vehicle innervations.

6.3 DAS System Basic Operation

To describe the operation of the DAS system, an example system that is utilising both soft-kill and the hard-kill mode is considered. The hybrid system described below can be applied as a representative example of many but not for all DAS systems. For this example, the vehicle is attacked by a kinetic energy projectile. When the missile is approaching, it is being detected and tracked by an on-board vehicle sensor. The soft-kill countermeasures of DAS (in Figure 6-1) include an Infrared jammer that has been developed specifically against optically guided missiles. The jammer deceives the guidance system of the missile by affecting its control system thus causing it to miss the target. In conjunction, the hard-kill system continues tracking the trajectory of the threat while countermeasures such as fragmentation launcher systems mounted on the vehicle may be activated into standby mode.

If the soft-kill system does not successfully deceive the threat and the missile continues to approach the vehicle, the hard-kill countermeasure will be launched using the tracking signal to ensure an accurate trajectory. The control unit is responsible for deciding which of the countermeasure modules should be launched and at what range the missile needs to be at for effective deployment. By launching a spread of fragments, the threat can be either destroyed or deflected. Further operational information can be found in, [85]–[87].

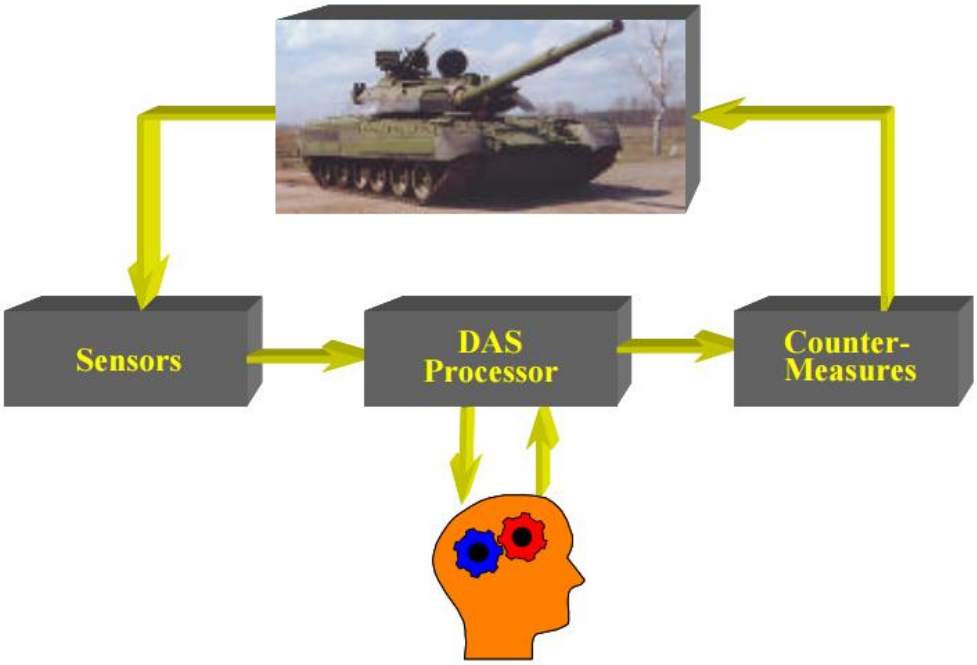

Figure 6-1 DAS System Concept [88]

6.3.1 DAS On-Board Items and Architectures

The DAS system, displayed in Figure 6-1, consists of different on-board items[37] and architectures to enhance mission capabilities, depending on the type of vehicle, operation type and location. All of the elements in the figure are integrated on the platform and are able to exchange information. DAS uses sensors to detected threats as depicted in Figure 6-2. The DAS processor is the element for computing how the threat can be dealt or mitigated and the effectors, Figure 6-3, are used as countermeasure actuators to physically deal with the detected threat.

Below, a list of DAS system's components, sensors, effectors and architectures, is presented. The components extracted from various other DAS systems used in defence and avionics [89], [90], [91], [92], [93] and [94].

[37] Item is a system or an array of systems in order to implements a specific function at platform level.

DAS Sensors

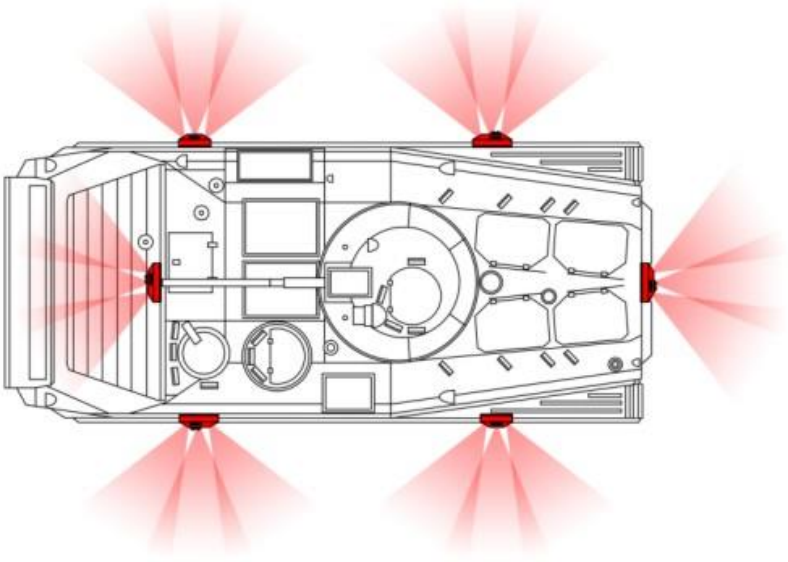

Figure 6-2 Vetronics Sensors

Radar Warning Receiver (RWR)

A Radar Warning Receiver is capable of detecting, identifying and managing Radio Frequency (RF) signals transmitted by radar systems. It also generates visual and audio cues to the crew by managing interfaces to other systems.

Optical Systems

Survivability can be increased with the aid of optical sensitivity systems. Infrared range contrast between threat background and dimensions are the main elements of WFOV and NFOV optical systems. These optical systems can provide an essential detection of high-temperature mixtures of water, particulates and carbon dioxide. The detection is made from threats using combustion from their propellant.

Laser Warner Receivers (LWR)

These receivers are able to detect, classify, analyse and locate laser wavelength, modulation and angle of arrival. LWRs can be from various laser systems such as laser beam riding missiles, laser designators, command links, range finders and electro-optic countermeasures.

Missile Approaching Warner (MAW)

Missile Approaching Warner is another essential component used in DAS systems. With almost zero false alarm this warner can detect all aspects of projectile launches with the aid

of active sensors/radars. In comparison with the ultraviolet/infrared sensors, the MAW is more effective in threat detection due to the possession of having capabilities beyond detecting only the booster plume and the launch of the projectile. The MAW warner uses a central computer to analyse any detected threat and provide either a manual interaction through visualisation in the cockpit or automatically countermeasures.

Radio Frequency (RF) Radar

RF radars are capable of detecting electromagnetic energy and compute the frequency of the radiation event, the angle of arrival and the signal strength from the operational mode of the radar.

DAS Effectors

Figure 6-3 Vetronics Effectors

Noise Jammer

Noise jammers are used against any potential laser eavesdroppers. Noise jammers are capable of interrupting network communication between the network nodes. It is transmitting electromagnetic waves to the corresponding receiver by minimising the Signal to Noise Ratio (SNR). The distorted signal will lead to the platform's cease to exist in the operator's display.

Decoys

There is no guarantee that all jammers have the full effect of survivability thus, that is the reason platforms are using the Towed Radar Decoys (TRDs). The operational function of the decoy it is usually happening using a kevlar cable with a fibre optic link. When a decoy is

deployed a trail is created near the platform in order to attract radar-guided missiles and improve survivability.

However, there are also different types of countermeasures decoys which are non-towed. The non-towed decoys are generic expendable dispensers which are using sophisticated electronics and lithium thermal powered batteries to lure radar-guided missiles from their aimed target.

Obscurants

Passive smoke grenades are the essential elements of obscuration which are using metal-flake and chaff providing hemispherical extended with laser dazzling. Also, the high-performance obscurant grenade is used in platforms (Australian Army) based on red phosphorus pyrotechnic composition. When initialised the combustion of the composition creates IR emissions, light or heat, (phosphoric acid) that derived from the chemical reaction in air.

Countermeasures Dispensers

In association with the MAWs the most commonly countermeasures used are the countermeasures dispensers using chaff and flares. These dispensers are cheaper than complex self-protection systems. The dispensers consist of either a plastic coated in metal or finely sliced metal foil, for example, aluminium, to distort the signals length for an interception radar frequencies. The chaff can create large clouds that lead the platform to “cease from existence” or it can also provide small elements used as decoys as a target.

Countermeasures Flares

Flares are used to confuse heat-seeking missiles or replace the aimed target. The operation lasts for between 2-4 seconds after the flares are ejected.

Infrared Countermeasures

Infrared countermeasures are used to lose track on a locked target with a signal interference of thermal imagers and seekers.

RF Jammers

The RF jammer is able to destroy signatures or give faulty targets by re-emitting the signal from a hostile radar.

DAS Systems and Architectures

Leonardo's Praetorian Defensive Aid Sub-Systems (DASS)

The Eurofighter Typhoon aircraft uses an effective electronic self-protection system that is employed in modern air warfare. The system is provided exclusively to Eurofighter Typhoon with the association of the EADS (Germany), BAE Systems (UK), Elettronica (Italy) and Indra (Spain). The system can fully autonomously analyse all threats, such as air-air and air-ground threats, in which it works essentially in two steps. Firstly, it locates, analyses and provides warnings, any potential threats. And secondly, it passes automatically defensive countermeasures. The platform uses high-speed conventional buses, (MIL-STD-1553 and MIL-STD-1760) and fibre-optic data-buses (STANAG 3910) which usage of rapid developments in computing can be provided.

Terma AN/ALQ-213(V)

The AN/ALQ-213(V) system has been developed by the TERMA A/S company as an electronic warfare management unit solution for military Aircraft Survivability Equipment (ASE) suites. It is currently used in more than 2000 aircraft platforms in which 25 platforms are different in 15 different countries. Its main objective is to provide easy integration of any ASE sensor or effector system. Any typical application of the system can be integrated into different ASE sensors and effectors through one or more MIL-STD-1553B buses in combination with discrete and serial interfaces.

ELIX-IR

From Thales group, the ELIX-IR system is the next generation threat warning system providing enhanced mission survivability. The system is most commonly used on platforms such as large aircraft, helicopters and unmanned aerial vehicles. The platform is equipped with 4-6 sensors, a central processor, dedicated display and integral data logging and optimal full image recorder. The used interfaces for the platform are MIL-STD 1553B, RS-422/429 and Ethernet. It is also compatible with the proposed NATO Interface Standard[38] and Modular Open Architecture (ITAR). Below there are the technical description/capability of the system.

Enhanced Platform Survivability against

- Guided Missiles
    - MANPADS, SAMS
    - IR, Laser and Radar Guided

[38] Refer to NATO Standardisation Agreement (STANAG) for further information [106].

- Hostile Fire
    - RPG and other unguided missiles
    - Multi-calibre Guns
    - Aircraft Equipment
- 4 – 6 Sensors
- ½ ATR CPU or processing card supply
- Dedicated display or custom interface

Interfaces

- MIL-STD 1553B
- Ethernet
- RS-422
- Various INU/IGI

6.3.2 DAS Modelling and Functional Simulation Platform

This section presents a modelling and functional simulation platform design of the DAS system. The design is based on the proposed DAS architecture depicted in Figure 5-15 that potentially be able to accommodate the framework of this research. The implementation of DAS design is composed of different nodes. The nodes are threat nodes for providing a threat to the DAS sensors with random coordinates to compute speed; DAS sensors that will distinguish the threat type and provide events to the DAS computer; a DAS computer that calculates and deals with the threats; and an effector node for the elimination and misdirection of the threat.

The DAS design is implemented using the Node-RED tool [95], which is a very powerful tool for controlling data flows from input and output modules. The Node-RED tool uses an event-processing method for its specific modules and can implement the DAS architecture by analysing and tracking streams of data.

The programming language that the Node-RED uses, for constructing function modules, is in JavaScript or HTML. The data between the function modules are wrapped in JavaScript Object Notation (JSON). The data wrapper is in a format that is humanly readable for better and easier usage of controlling data flow. Each DAS node is using the Message Queue Telemetry Transport (MQTT) communication protocol to communicate with other DAS nodes. Platform design is based on a publisher-subscriber communications and the MQTT protocol is the best candidate achieving that. Node-RED is using a Graphical User Interface (GUI) module block and by drag, drop and wiring the modules within the workspace the integration between DAS nodes is very easy.

A simple construction of the DAS system, including external threats, using the Node-RED tool is presented in Figure 6-4.

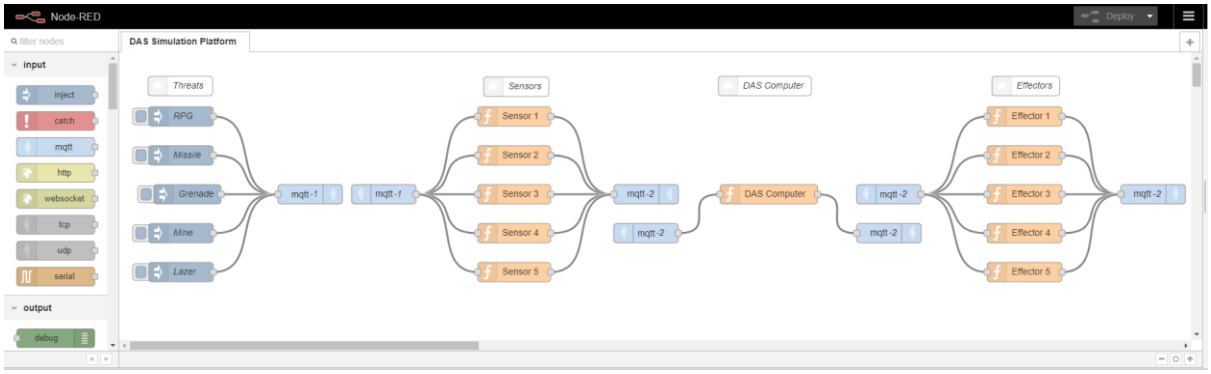

Figure 6-4 Modelling and Functional Simulation Platform Design of DAS

The modules listed in blue colour are the threat nodes. The threat nodes are threat event inputs. When a threat event is triggered (injected), a specific threat is simulated. Each threat node provides data that is essential for the simulation, for example, ID and coordinates (raw data). The output of the threat node is fed into the sensor node, through an MQTT publisher (mqtt-1). The sensor node is triggered only when a specific threat is simulated. The sensor node is acting as a subscriber and as publisher. Furthermore, the specific sensor publishes (mqtt-2) a message to the DAS computer for calculations. The message from the sensor combines the threat's data (raw data) and adds additional data attributes as (if) specified in a message specification of the overall DAS system design.

The DAS computer node receives the complete message from the corresponding sensor and calculates the distance and speed of the specific threat. In the event of the DAS computer is unavailable, the message from the sensor is directly fed into the corresponding effector. However, when the specific counter measurements are made, the candidate effector will deal with the detected threat. Once the threat is dealt with, the effector will provide an indication to either the vehicle crew or to the system of the threat status. Below, a bit more technical specification of the modelling and functional simulation platform design of the DAS system is presented in Figure 6-5. Also, the following content of this section describes the nodes used in the simulation.

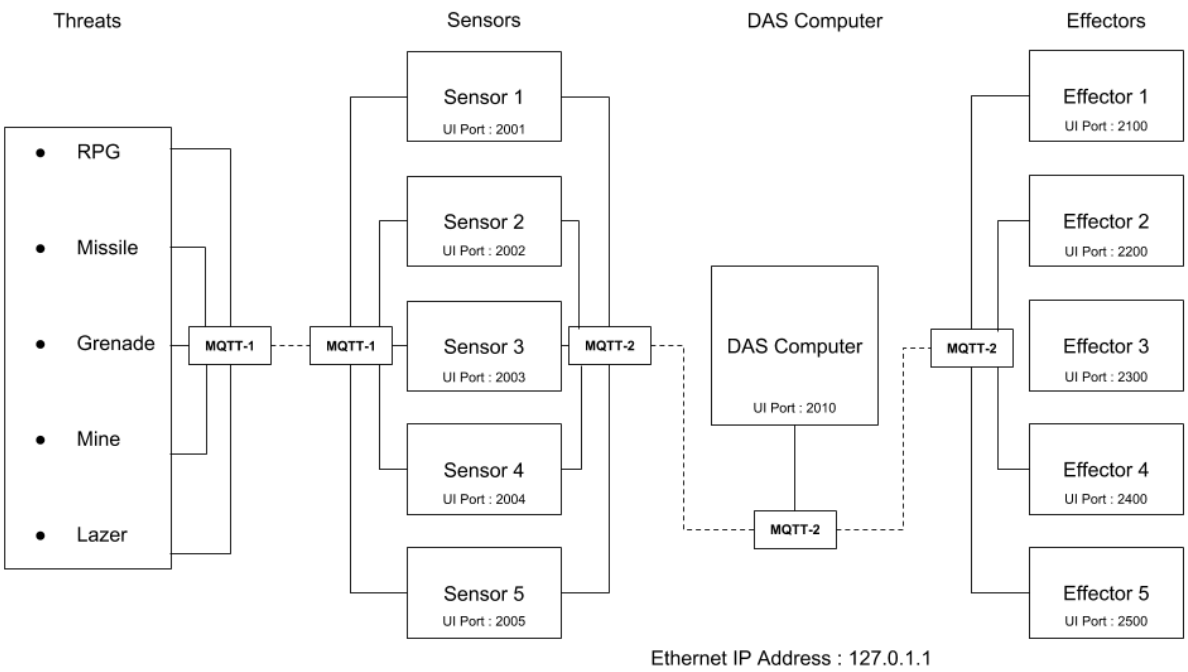

Figure 6-5 Generic DAS Architecture

Threat Node

The threat node is designed to inject threats to the vehicle and initialises the coordinates of the incoming threat using three-dimensional space parameters and values. Each threat has a unique identification number such that the DAS computer is able to process multiple threats and report accordingly with IDs and status to the crew or to the rest of the platform. The implementation of this node is shown in Figure 6-6.

Figure 6-6 Threat Node

The above figure depicts the threat node implementation where all the threats initialised by using the injection module with specific message payload (threat attributes). The message payload changes each time, by injecting a different threat message and a unique identification number can be seen in Figure 6-7 on the debug display. The information is passed to mqtt-1, which is communicating with the equivalent mqtt module in the sensor node.

Figure 6-7 Threat Node Output (Example)

Sensor Node

The sensor node, Figure 6-8, is aware of the incoming threats and must submit the information to the DAS computer. Each of the sensors is initialised when the specific threat and threat type is detected. For instance, a Rocket-Propelled Grenade (RPG) threat initialises the RPG sensor and from the sensor information is transmitted to the DAS computer for processing.

The sensor node only processes a threat with a specific ID as can be seen in Figure 6-9.

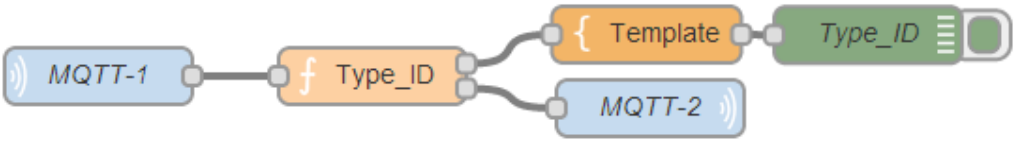

Figure 6-8 Sensor Node

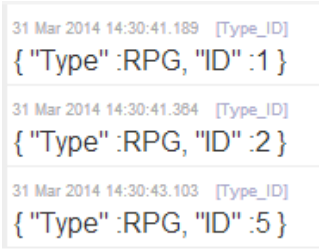

Figure 6-9 Sensor Node Output (Example)

The complete message is passed to the mqtt-2 which is forwarded to the DAS computer for the speed and position calculation.

DAS Computer Node

The DAS Computer receives the data from the sensors and acts as a server within the system. The DAS configuration must estimate the threat's speed and active position of the incoming threat. The communication between the DASC and the sensor node uses the same mqtt module as shown in Figure 6-10. By configuring the DASC node for the incoming threat provides data to the effectors in order to destroy or deflect the threat, Figure 6-11.

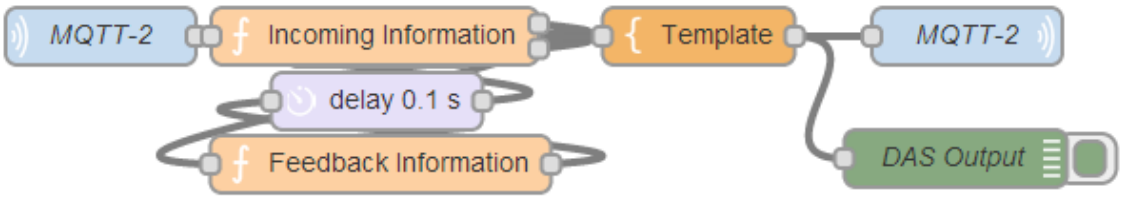

Figure 6-10 DAS Computer Node

Figure 6-11 DAS Computer Node Output (Example)

Effectors Node

The effectors, Figure 6-12 must be constructed in order to deal with the incoming threat. Given the specific threat type the corresponding effector acts and gives a result back to the DASC as well as to the crew that the effector succeeded in its objective.

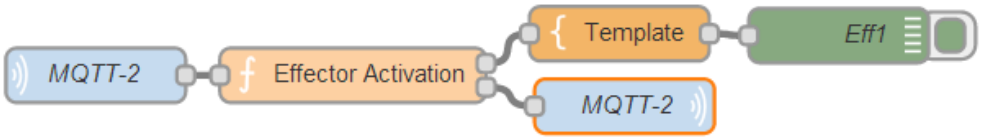

Figure 6-12 Effector Node

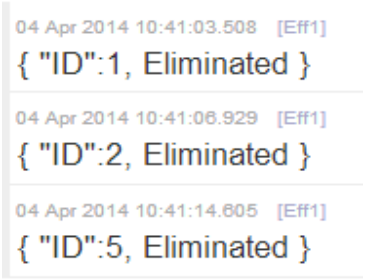

Figure 6-13 Effector Node Output (Example)

As shown in the example, Figure 6-13, three of the specific threats with corresponding threat ID's have been eliminated when the effector has been activated. The message shown above feeds back to the DASC to declare that the specific ID threat has been dealt with.

6.4 Survivability

Survivability, like many other terms[39], can be described differently depending on where it is applicable to, (**What is the Mission?**). Survivability in this research is defined as,

"*The extent to which the system can deliver services whilst under hostile attack".* [96]

The definition is used to describe a part of a generic Mission-Critical system, within a system's taxonomy property. However, survivability is formed otherwise in this use case scenario. A clear overview of survivability used for military vehicles is depicted in Figure 6-14, which is also known as the "Survivability Onion". Each layer is categorised to levels of survivability or opportunities that a platform has to mitigate the effects of a threat.

[39] See Chapter **Error! Reference source not found.**, Section **Error! Reference source not found.**, page 60, "Mission-Critical System Taxonomy".

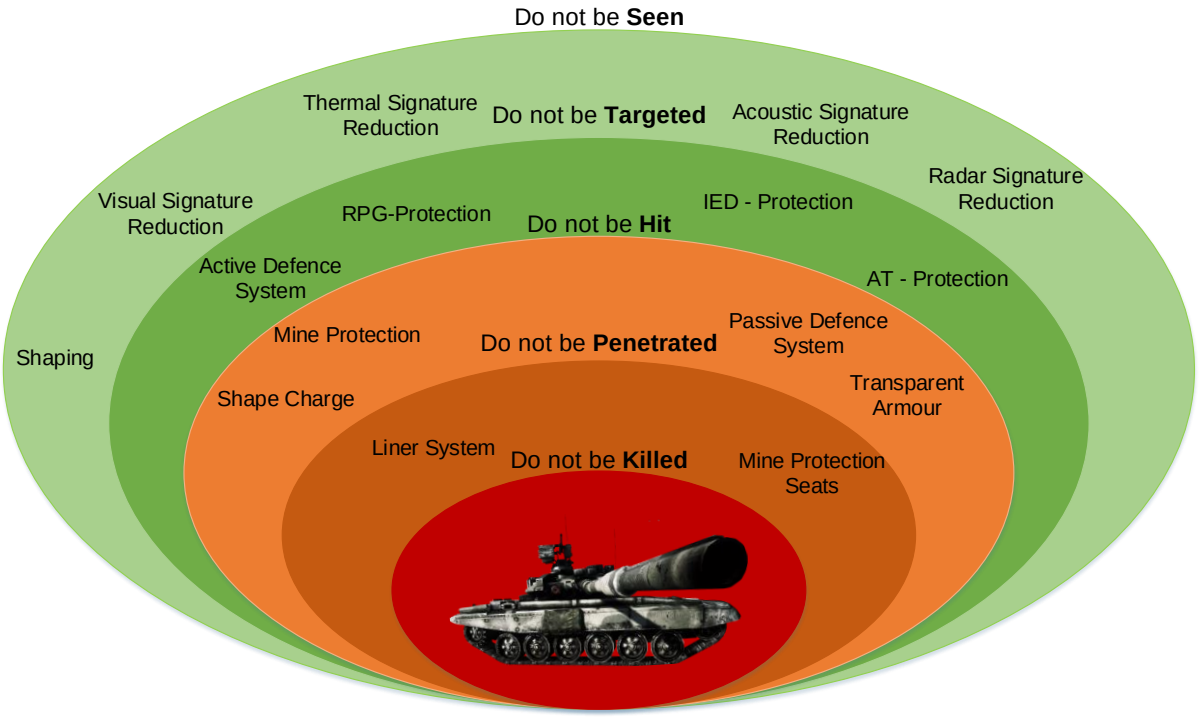

Figure 6-14 Layers of Survivability [97],[98]

The sequence of the survivability onion is as follows; Do not be seen. If you are seen, do not be targeted. If you are targeted, do not be hit. If you are hit, do not be penetrated. And if you are penetrated, do not be killed. A more refined definition of the sequence is,

Do not be seen – The vehicle shall not be spotted. As an alternative to this layer, the vehicle can hide, camouflaged, turn off the engine at a "mission safe state"[40] area or any other actions in order to "do not be spotted".

Do not be targeted – The vehicle has partially seen by the threat and is attempting to acquire for engagement. In this layer, the vehicle may use options as depicted in Figure 6-14.

Do not be hit – The vehicle now has been seen from the threat and is engaging. In this layer, the vehicle can proceed to the options displayed in that layer.

Do not be penetrated – The threat acquired a visual contact with the vehicle, threat engaged and hit the vehicle. The vehicle can choose between this layer's options.

Do not be killed – Finally, the threat has a clear visualisation of the vehicle, hit and penetrated the vehicle, thus, the threat now is critically affecting the vehicle and the crew.

[40] A state in which potential threats and operational risks are minimised, similar definition of "Safe State" in safety functional IEC standard [107].

However, survivability may be and is defined in many ways depending on the mission. The next section deploys the proposed framework as part of defining mission, mission system, threats, Mission-Critical system's, opportunities, enhancements, specifications, designs, architectures, development, rapid prototyping, early de-risking tests, integrity levels, evaluation of threat's affect, system's effectiveness level, maturity using number theory, in an effective and efficient manner. At the end of this chapter, an overall Mission-Critical Data Model, qualitative and quantitative results will be presented.

6.5 Case Study – Introduction

By collecting all the information stated earlier in this chapter, the framework is applied using very simple examples in order to avoid complexity and be more comprehensible to the reader. In this case study, the two scenarios (Case 1 and Case 2) are examined in a simple manner, but sufficiently enough to prove the concept and the importance of this study. The mission in both cases studies are exactly the same and is as follows,

"*The survivability of the vehicle travelling from (X_a, Y_a, Z_a) coordinates to (X_b, Y_b, Z_b) coordinates*".

Using the above mission which is for both scenarios, the framework is constructed upon a real DAS system using the elements and attributes found in [99].

Table 6-1 Sensors, Camera and Threat Attributes

Anti-Armour Threats	***Threat, Calibre***	M-712, LSAH, 155mm	RPG-7, 80mm	Gun, 30mm, APDS
IR WFOV[41]	***Distance, [m]***	400	470	5480
IR NFOV[42]	***Distance, [m]***	3600	4200	340
LI/RG Camera	***Threat, [Pixels]***	1.3	42 x 42	0.8
	Target, [Pixels]	25x20	234 x 187	118 x 60
Threat Variables	***Dimensions, [m]***	0.155 dia.	0.18 dia.	2.1 dia.
	Range, [m]	14000	500	2000
	Velocity, [m/s]	255	255	815

Table 6-1 is adapted from the aforementioned reference and from this point forward, the framework is applied[43].

6.5.1 Mission

The core mission in this case study is “Survivability”. According to the “Survivability Onion”, survivability can be expressed in different categories. For this case study, the “Do not be hit” category will be applied as part of the mission. Therefore, the mission is “Survivability – Do not be hit”. Hence, the mission’s requirements can be defined as:

Mission: **M[1]** – Survivability.

- **M[1][1]** – Do not be hit.

Consider – M[1] is the core mission and M[1][1] is a property of the core mission M[1].

6.5.2 Mission System

Since the mission is survivability, a survivability system will be the first candidate mission system to be examined. For this case study, the Defence Aid Suite (DAS) system is the

[41] Infrared Wide Field of View
[42] Infrared Narrow Field of View
[43] For an easy understanding of the following steps, the reader should follow Figure 4-1.

candidate mission system and it will be a Soft-Kill DAS system, defined earlier Section 6.3. Hence, the Mission System step is categorised as:

Mission System: **MS[1]** – Survivability system.

- **MS[1][1]** – Defence Aid Suite (DAS) system.
    - **MS[1][1][1]** – Soft-Kill system.

6.5.3 System Analysis

The purpose of the SA[1] should be extracted from the MS[1] and it should be oriented into mission effectiveness manner[44] [100].

System Analysis: **SA[1]** – "*Is the system that has the ability to resume functioning without evidence of degradation following temporary exposure to an adverse environment. This implies that the system performance will degrade during exposure to the environment, but the system will not experience any damage, that will prevent it from operating when the adverse effects are removed or reduced below allowable susceptibility levels*." [101]

- **SA[1][1]** – Protect the platform against threat(s).
    - **SA[1][1][1]** – Detect threat(s).
        - **SA[1][1][1][1]** – e.g. DAS Sensors, (i.e. Section 5.7.3.1).
    - **SA[1][1][2] –** Compute countermeasurement(s).
        - **SA[1][1][2][1]** – e.g. DAS Computer(s), (i.e. Section 5.7.3.3).
    - **SA[1][1][3]** – Counter threat(s).
        - **SA[1][1][3][1] –** e.g. DAS Effectors, (i.e. Section 5.7.3.5).

6.5.4 Data Model – Mission System

By identifying the purpose of the system, a data model can be constructed, or if already exists then the data model of the existing system must be depicted in this step. A very simple DAS system demonstration is shown in Figure 6-15[45]. The DAS system's data model will be consisting entities of DAS - sensors, computer and effectors. The domains, packages, entity types, data attributes, data types and relationships for this section will be categorised as:

[44] The probability that a system is available to initiate its mission and will complete its mission when initiated. (USAF/LGMM, 1994)
[45] A Data Model for a DAS system can be developed further, using the DAS Elements shown in Section 6.3.1.

<<Domain>>

- **<<Package>>**
  - **<<Relationship>>**
    - **<<Entity Type>>**
      - **Data Attribute**

Moreover, it is not necessary for this case study to explain further the purpose of the selected Mission system, in order to avoid complexity. Assume that is sufficient enough[46].

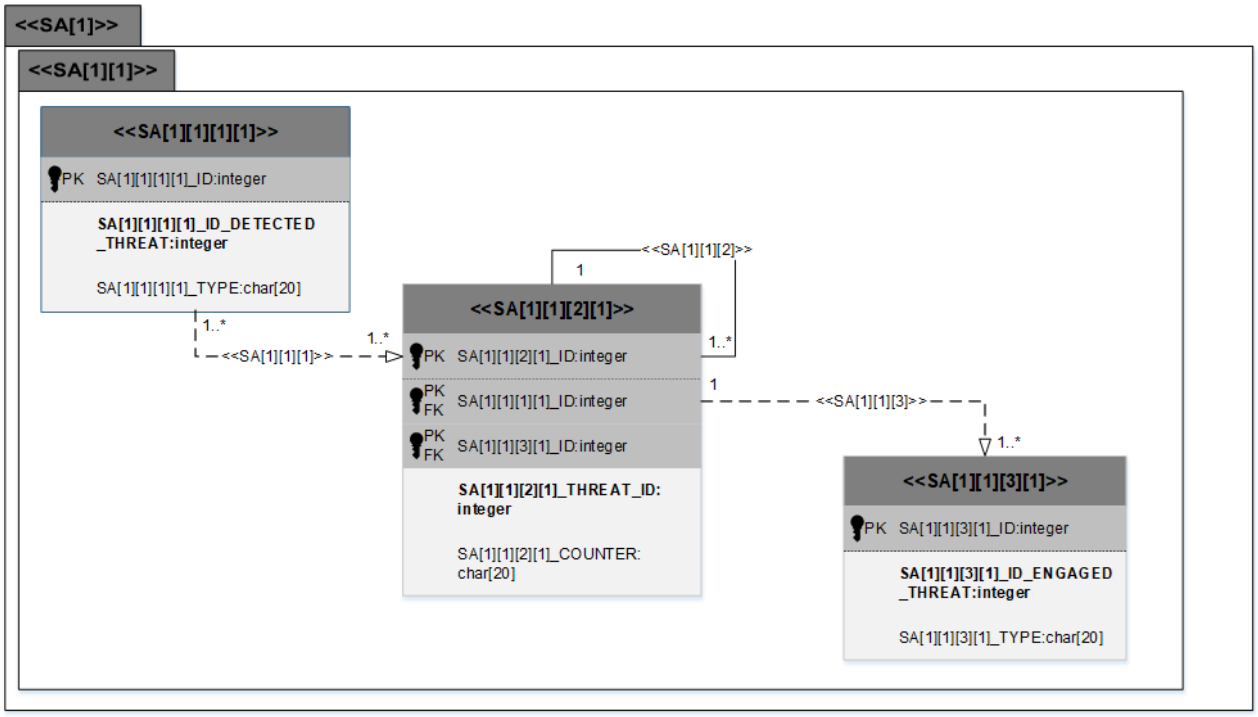

Figure 6-15 SA[1] in UML

<<SA[1]>> – ***Survivability System definition, Section 6.5.3*.

- **<<SA[1][1]>>** – Protect the platform against threat(s).
  - **<<SA[1][1][1]>>** – One or more DAS sensor(s) detect threat(s) and send data to DAS computer(s) for counter measurements.
    - **<<SA[1][1][1][1]>>** – DAS Sensors entity type, Section 5.7.3.1.
      - **PK SA[1][1][1][1]_ID** – The ID number of sensor, assigned as Primary Key.
      - **SA[1][1][1][1]_ID_DETECTED_THREAT**: The ID number of the detected threat. Assigned as Surrogate Key (in bold). For example, is required by the Health and Usage Monitoring

[46] Parts of the UML representation adapted from the Generic Vehicle Architecture (GVA), Land Data Model (LDM), [17].

System (HUMS) system or to indicate the “raw data”, XYZ axis coordinates, of the sensor.

- **SA[1][1][1][1]_TYPE**: This data attribute is to define the sensor type (e.g. IR NFOV, IR WFOV sensors or cameras).

- **<<SA[1][1][2]>> –** DAS computer(s) is responsible to receive data from the DAS Sensor(s), compute countermeasure(s) and send a command to DAS Effector(s) to deal with the detected threat.
  - **<<SA[1][1][2][1]>>** – DAS Computer(s) entity type, Section 5.7.3.3.
    - **PK SA[1][1][2][1]_ID** – The ID number of DAS computer, assigned as Primary Key.
    - **PK/FK SA[1][1][1][1]_ID** – The ID number of the initialised DAS Sensor(s). Assigned to as Primary Key and Foreign Key.
    - **PK/FK SA[1][1][3][1]_ID** – The ID number of the DAS Effector(s), commanded to engage and countermeasure the detected threat. Assigned to as Primary Key and Foreign Key.
    - **SA[1][1][2][1]_THREAT_ID**: The ID of all the detected threats during the mission. Assigned as Surrogate Key, (in bold) and can be useful for other purposes. For example, is required by the Health and Usage Monitoring System (HUMS) system or used to prioritise threat countermeasures.
    - **SA[1][1][2][1]_COUNTER**: The countermeasure algorithm decided by the computer.
- **<<SA[1][1][3]>>** – One or more DAS Effector(s) to be engaged and deal the detected threat.
  - **<<SA[1][1][3][1]>>** – DAS Effector(s) entity type, i.e. Section 5.7.3.5.
    - **PK SA[1][1][3][1]_ID** – The ID number of effector, assigned as Primary Key.
    - **SA[1][1][3][1]_ID_ENGAGED_THREAT**: The ID number of the engaged threat. Assigned as Surrogate Key, (in bold). For example, is required by the Health and Usage Monitoring System (HUMS) system or to count the availability of the effector (e.g. number of flares).
    - **SA[1][1][3][1]_TYPE**: This data attribute is to define the effector type (e.g. smoke grenades and countermeasures flares).

6.5.5 Benefits

Selecting the MS[1] system, hypothetically is to improve the overall mission effectiveness by protecting the platform against threats. Assume that B[n] is the only identified benefit from the MS[1] system and the expected value of that benefit is 99.9%[47].

Benefits: **B[1]** – Improves overall mission effectiveness.

6.5.6 Effectiveness Level

The Effectiveness Level step will collect all the "Benefits" B[n] requirement(s) extracted in Section 6.5.5 and then, a threat analysis shall examine each benefit in order to identify any possible "Theat" T[n] that could downgrade the specific benefit. In this use case scenario, the T[1] is a 30mm ADPS has been identified as a threat.

The effectiveness level of the system, MS[1], for the mission M[1] including the "Threat", EL_B[n]_T[n] (Equation 2) is calculated to be:

Case 1: Passive Armour Vehicle

EL_B[1]_T[1]: 64.9%

Meaning that in Case 1 according to the defined threat T[1], the possibility of the Mission System MS[1] to support the mission M[1] to succeed, is 64.9%.

Case 2: Light Armour Vehicle

EL_B[1]_T[1]: 24.9%

And for Case 2, according to the same defined threat T[1], the possibility of the Mission system MS[1] to support the mission M[1] to succeed, is 25% which is relatively low.

The effectiveness level of the system with the "Mitigation Process" EL_B[n]_MP[n], Equation 3, is calculated to be:

Case 1: Passive Armour Vehicle

EL_B[1]_MP[1]_MCS[1]: 64.9%

Case 2: Light Armour Vehicle

EL_B[1]_MP[1]_MCS[1]: 53%

[47] It must be noted here that all the values and calculations will be only demonstrated in this section. Section 6.6, will present the steps on how these values are derived.

The effectiveness level of B[1], in both cases, should reach TRL1 level as part of their mitigation process MP[1]. This is due to the Mission-Critical System MCS[1] is developed only in theory. However, in order to understand the importance and effect of the mitigation process, the author selected Case 1 to reach only TRL1 level and Case 2 to reach TRL4. Further discussion will be provided in Section 6.6, in order to understand the meaning of the mitigation process.

6.5.7 Threat

Once, the benefit B[1] is defined, a threat analysis and risk assessment should take place. The identified threat for this case study is a 30mm ADPS. This threat is a type of kinetic energy projectile fire and is developed such that the armoured vehicles are penetrated and damage the overall vehicle[48].

Threat: **T[1]** – 30mm ADPS.

However, this potential threat shall be analysed using the following steps.

6.5.8 Threatening System

The above threat is used in an ADPS gun, thus, the "Threatening System" is:

Threatening System: **TS[1]** – 30mm ADPS Gun.

6.5.9 Occurrence

Assuming that the occurrence of the potential cause TS[1] can be occasional in both cases. Therefore, the occurrence's category selected from Table 4-1 is:

Occurrence: **O[1]** – Occasional

Using Table 4-5 the O[1] is 15%.

6.5.10 Potential Impact

The potential impact of the threat T[1], is to penetrate the vehicle, hence:

Potential Impact: **PI[1]** – Penetrate the vehicle (in both cases).

6.5.11 Severity

The severity in both cases is different. The difference between these two vehicles is that the one is a heavy armoured vehicle and the other is a light armoured. The heavy armoured

[48] [102] has further details on the effect of the specific threat.

vehicle is relatively heavier than the light armoured vehicle due to its body structure. The heavy armoured vehicle uses a thick armour that can be less susceptible to their attacks. On the other hand, the light armoured vehicle is better in tactical mobility compared to the heavy armoured vehicle but less resistant to anti-tank threats. Therefore, the severity of both cases is as follow:

Case 1: Passive Armour Vehicle

Considering that the threat does not significantly impact the mission or the vehicle, thus the severity of the specified threat can be considered as negligible. Using Table 4-2 and Table 4-5, the severity for Case 1 is:

Severity: **SE[1]** – Negligible

Using Table 4-5 the SE[1] is 0%.

Case 2: Light Armour Vehicle

Considering that this is a light armoured vehicle the threat can significantly impact the mission or the vehicle, thus, the severity of this threat can be catastrophic. Using Table 4-2 and Table 4-5, the severity for Case 2 is:

Severity: **SE[1]** – Catastrophic%

Using Table 4-5 the SE[1] is 50%.

Despite this, it can be clearly seen that one single element for the same mission can be categorised differently, either negligible or catastrophic. This is resulting in T[1] requirement to cause minor mission issues to a complete mission failure.

6.5.12 Threat Classification

This is the step where the threat T[1] shall be classified. The threat T[1] can be categorised into two classifications.

Case 1: Passive Armour Vehicle

T[1]: Negligible – Occasional: "***Class 3***"

- **Class 3**: Tolerable if the cost of risk reduction would exceed the improvement.

Case 2: Light Armour Vehicle

T[1]: Catastrophic – Occasional: "***Class 1***"

- **Class 1**: Unacceptable in any circumstance and it will have a great negative impact on the mission system's effectiveness and therefore, to the overall mission.

When threat T[n] is defined, it can be therefore used for different missions, or data model domains, thus, the user will be able to re-use the threat depending on the application (mission). With this approach, the reusability of threats, hazards or risks can be easily adapted, modified and applied for different mission scenarios. This can be achieved following the next steps.

6.5.13 Detection Mode

As aforementioned earlier, both vehicles using a DAS system. According to Table 6-1 both of the vehicles using IR sensors, therefore:

Detection Mode: **DM[1]** – DAS sensors.

6.5.14 Detection

In order to exploit the available sensors from Table 6-1, assume that the heavy armoured vehicle uses the IR NFOV sensor and the light armoured vehicle uses the IR WFOV sensor. Using the Table 6-1 it can be seen that the IR NFOV is able to detect the 30mm APDS from a distance of 340 metres and with the threat's velocity makes it "hard"[49] to detect. The IR WFOV can detect the 30mm APDS from a distance of 5480 metres, where is relatively "easy" to detect due to the sensor detection capability. However, the detection in both cases are categorised as such:

Case 1: Passive Armour Vehicle

Detection: **D[1]** – Hard | **SO[1], SO[2]**[50]

Using Table 4-5 the D[1] is 20%.

Case 2: Light Armour Vehicle

Detection: **D[1]** – Easy

Using Table 4-5 the D[1] is 10%.

[49] Considering that the DAS system has to detect, process and countermeasure (using mechanical components) within a very short amount of time.
[50] Note: The purpose of SO[1] and SO[2] will be discussed further in Section 0.

6.5.15 Threat Level

Once the threat is analysed and classified, the next step is to calculate the affect level of the threat against the predefined benefit. The affect level in both case is calculated to be:

Case 1: Passive Armour Vehicle

TL_T[1] : [(O[n]:15%)+(SE[n]:0%)+(D[n]:20%) = 35%

Case 2: Light Armour Vehicle

TL_T[1] : [(O[n]:15%)+(SE[n]:50%)+(D[n]:10%) = 75%

When this step is completed, the effectiveness level of the predefined benefit with the affect level of identified threat can be calculated, see Section 6.5.6, EL_B[1]_T[1].

6.5.16 Data Model - Threat

Based on the threat analysis, a threat data model can be constructed as depicted in Figure 6-16.

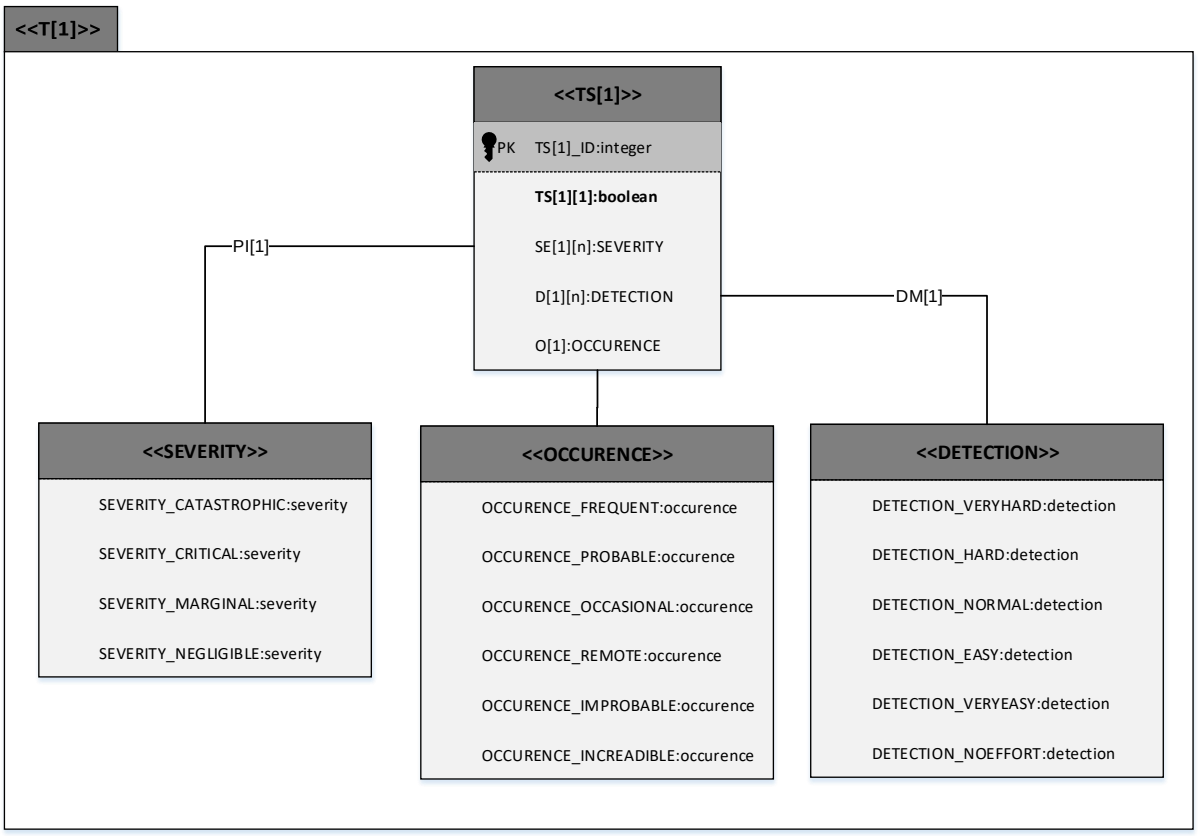

Figure 6-16 Data Model – Threat

<<T[1]>> – 30mm ADPS.

- **<<PI[1]>>** – Penetrate vehicle.
- **<<DM[n]>>** – DAS Sensor.
  - **<<TS[1]>>** – 30mm ADPS Gun.
    - **PK S[1]** – TS[1]_ID, set as Primary Key. Used as the identification number of the entity.
    - **TS[1][1]** – Threat detected, true or false. Assigned as Surrogate Key (in bold). For example, to enumerate the number of detections that can be used as “raw data” the DAS sensor to DAS computer.
    - **SE[1]** – The severity, see Section 6.5.11.
    - **D[1]** – The detection, see Section 6.5.14.
    - **O[1]** – The occurrence, see Section 6.5.9.

6.5.17 Mission-Critical System

Analysing and reviewing the T[1] threat, in which in Case 1 is 35% and in Case 2 is 75% of a possible effect on the mission M[1], for that reason a Mission-Critical System is required in

order to reflect the identified threat. Below there are the candidate systems that can be considered as Mission-Critical system for this use case study.

Mission-Critical System – **MCS[1]**: Obscurants

Mission-Critical System – **MCS[2]**: Smoke Grenades

Mission-Critical System – **MCS[3]**: Flares

To simplify the remaining procedure of the framework, only the MCS[1] will be analysed and developed. The rest of the Mission-Critical Systems will only appear in the Mitigation Process, Section 6.6, as an indication of using different mitigation approaches with different TRL levels.

Nevertheless, in general, Obscurants MCS[1] are particles suspended in the air that block or attenuate a portion(s) of the electromagnetic spectrum[51]. The development of the Mission-Critical system can be as follows.

6.5.18 Mission-Critical Function

The recommended action needed from the defined Mission-Critical system MCS[n] to reflect the threat T[n].

Mission Function – **MCF[1]**: Upon threat detection, initialise smoke grenades.

6.5.19 Responsibility

For this function to be executed, it must be commanded by the DAS system, hence:

Responsibility – R[1]: DAS computer SA[1][1][2][1].

6.5.20 Target Date

This step of the framework cannot be defined unless time is agreed prior to the development, as discussed earlier in Chapter 4, Section 4.3.5. However, some of the specified times in Table 6-2 is the author's suggestion (real-time and non-real-time). Hard Real-Time has been assigned as 0.42 seconds based on the IR NFOV detection distance and to the velocity of the 30mm APDS threat. Soft Real-Time is assigned as 6.72 seconds based on the IR WFOV detection distance and to the velocity of the 30mm APDS threat. Therefore, the target dates depicted in Table 6-2 part of is the author's suggestion and can be used as an example to understand the meaning of the time ranges. The remaining times are calculated from the sensor, camera and threat attribute Table 6-1. The table's intention is to specify also, that this

[51] See Appendices chapter, Section Effectors for more information

is the time needed from the moment the treat is detected until the threat is dealt with. Therefore, System Architects shall decide how to achieve the timings in Table 6-2 using appropriate tools.

Table 6-2 Real-Time Responsiveness Values

Hard Real-Time	0.42 sec
Soft Real-Time	6.72 sec
Real-Time	1 min
Non-Real-Time	1 hr

For both cases, the Target Date is selected to be as:

Case 1: Passive Armour Vehicle

Target Date – TD[1]: Hard Real-Time.

Due to the limited time constrained of the DAS sensor.

Case 2: Light Armour Vehicle

Target Date – TD[1]: Soft Real-Time.

Due to the available time, the DAS sensor can provide until the that has a direct impact on the platform.

6.5.21 Data Model – Mission-Critical System

The implementation of the steps in Sections 6.5.17, 6.5.18, 6.5.19 and 6.5.20, a Mission-Critical Data Model (MCDM) can be now constructed. The MCDM's development is to mitigate the threat identified T[1], where it can potentially improve the Section's 6.5.15 content, "Effectiveness Level EL_B[n]_T[n]". Furthermore, MCDM's main objective is to allow Systems Engineers and Architects to simulate swiftly the Mission-Critical System behaviours almost effortlessly[52].

[52] This can be achieved using Model Driver Architecture (MDA) approach and tools for rapid development of Object Oriented architectures, such the Node-RED [95].

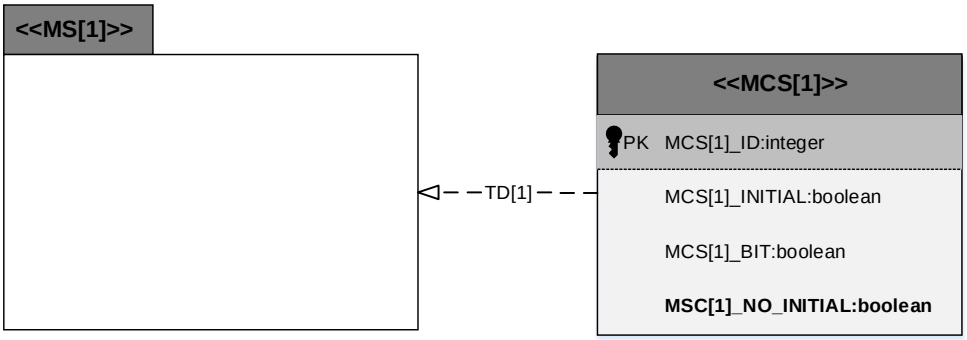

Figure 6-17 Mission-Critical System Data Model

- **<<MS[1]>>** – Survivability system.
  - **<<TD[1]** – Hard Real-Time (Case 1) / Soft Real-Time (Case 2).
    - **<<MCS[1]>>** – Obscurants.
      - **<<MCS[1]_ID>>** – The ID number of the system, assigned as Primary Key.
      - **<<MCS[1]_INITIAL>>** – The signal that initialise the effector(s) to be deployed.
      - **<<MCS[1]_BIT>>** - Build-in-Test indicating MCS[1]'s health status.
      - **<<MCS[1]_NO_INITIAL>>** – The number of times the MCS[1] initialised. Assigned as Surrogate Key (in bold). For example, it is required to count the number of MCS[1] initialisation as well as, the number of remaining smoke grenades.

6.5.22 Action Taken

In this step, it is to associate the relationships between the Mission-Critical System MCS[n], Mission-Critical Function MCF[n], Responsibility R[n] and the Target Date TD[n] requirements, but in more detail. The Action Taken AT[n] requirements can be also considered as the mission functional concept. For this use case, the Action Taken requirements will be following only an abstract definition, in which, it will be only on what was defined in the previous steps MCS[1] – TD[1].

Action Taken – AT[1]: Upon threat detection, initialise obscurants.

- Obscuration **MCS[1]**. Upon detection, initialise smoke granades, **MCF[1]**. DAS system is responsible for the execution of the Mission-Critical function **MCF[1]** in Hard/Soft Real Time responsiveness level **TD[1]**.

6.5.23 Mitigation Process

According to the TRL level definition, Chapter 3, Section 3.3.6, in this case, study the MCS[1] only reached the theoretical stage which is the equivalent to the TRL[1] level. That means, the threat T[1] still has the same effect. Using Equation 6 and Table 4-8, this is resulted as,

Case 1: Passive Armour Vehicle

Mitigation Process – MP[1]_MCS[1]: TRL[1] or;

MP[1]_MCS[1]: 1 or;

MP[1]_MCS[1]: 0%

Case 2: Light Armour Vehicle

Mitigation Process – MP[1]_MCS[1]: TRL[4] or;

MP[1]_MCS[1]: 0.625 or;

MP[1]_MCS[1]: 37.5%

Once this step is completed, the effectiveness level of the predefined benefit with the mitigation process level should be calculated in Section 6.5.6, EL_B[1]_MP[1].

6.5.24 Mission Integrity Level

Mission Integrity Level (MIL) are levels assigned to Mission-Critical systems MCS[n] to indicate their integrity and risk classification on missions. In other words, the likelihood of the mission-related system satisfactorily performing the required mission functions under all the stated conditions within a stated period of time.

For this use case scenario, the TRL[1] level was intentionally accomplished, thus, MIL step is left as incomplete and will be considered as future work for this research. Although the occurrence, severity and detection are defined and by using Table 4-10, the MIL levels can be estimated. An estimation of the proposed integrity levels for each Mission-Critical system are as followed:

Case 1: Passive Armour Vehicle

MCS[1] : QM – Quality Management

Case 2: Light Armour Vehicle

MCS[1]: MIL2 – Mission Integrity Level 2

6.5.25 Source

All of the information sources used for this framework's application, were online sources, books, magazines, author's personal experience and so on. However, if stakeholders require an ultimate Mission-Critical system for a specific mission, then, more sources are needed. For example, the detection D[1], is defined as "*Hard*" followed by SO[1] and SO[2]. As described in Section 4.2.25, SO[n] is used to indicate the participants who contributed or involved or consulted in the framework. For this case, SO[1] is defined by the author of this thesis. The author is a Functional Safety Engineer, thus, the information of that source might not be valid and/or precise.

Therefore a related paper was published [102], discussing how difficult the 30mm APDS is to be detected using more described definitions. The source is from the Defence Research and Development of Canada and it can be used as more valid, compared to Functional Safety Engineering discipline. For that reason, the more information this framework has, the more successful the mission could be. Nevertheless, for this example two SO[n] sources can be assigned,

Source: **SO[1]** – Source from a Functional Safety Engineer.

Source: **SO[2]** – Defence Research and Development Canada.

At this point, the framework's steps are completed and ready for the next stages of the development. The next section will provide clearer indications of how a mission can be developed and how successful it can be by using qualitative and quantitative results.

6.6 Early De-Risking Results

In this section, the results extracted from the framework, are used as part of the mission's early de-risking process. The significance of the results is to provide the primary functional mission concept in the very early stages of the mission, mission system and Mission-Critical system. Moreover, these results also indicate, the probability of the mission to be successful or not, using the threat's section results and the mitigation process with the level of technology readiness.

Starting from the generic mission which is defined in Section 6.5.1 as,

"*The survivability of the vehicle travelling from (X_a, Y_a, Z_a) coordinates to (X_b, Y_b, Z_b) coordinates*".

In Case 1, heavy armour, the overall mission M[1] it is likely to be 64.9% successful, Figure 6-18.

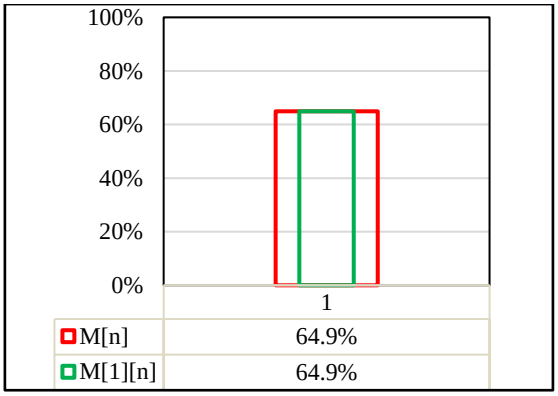

Figure 6-18 Case 1 – M[1] and M[1][1] Success Estimation

In Case 2, light armour, the overall mission M[1] it is less likely to be successful with 53% chances of succeeding, Figure 6-19.

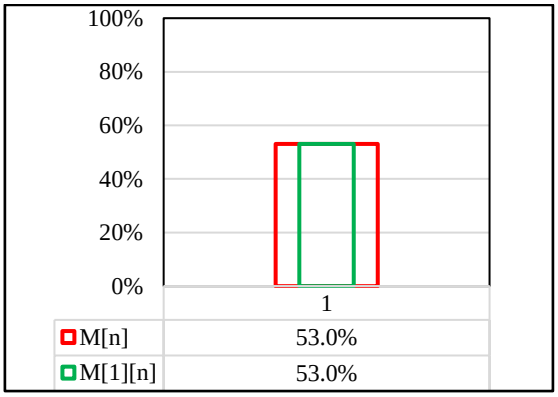

Figure 6-19 Case 2 – M[1] and M[1][1] Success Estimation

However, when the mission M[1] assigned in the first place, a candidate Mission System MS[1] was selected to be the most appropriate system. Since M[1] was survivability, a DAS system, which a survivability system, was a good example to be applied. Therefore, the DAS system was categorised as,

Mission System: **MS[1]** – Survivability system.

- **MS[1][1]** – Defence Aid Suite (DAS) system.
    - **MS[1][1][1]** – Soft-Kill system.

When the DAS system was initially selected as the most appropriate system for this mission, M[1], it was expected to be fully capable of supporting the mission to be successful. Therefore, the following figures, Figure 6-20 and Figure 6-21, represent the benefit B[1] of the MS[1] of

the expected benefit and the actual benefit, after the threat analysis and risk assessment, as well as, the mitigation process[53].

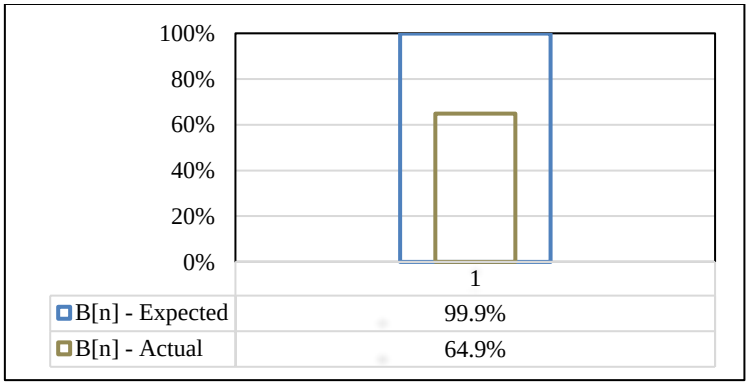

Figure 6-20 Case 1 – Benefit of Mission System Expected and Actual

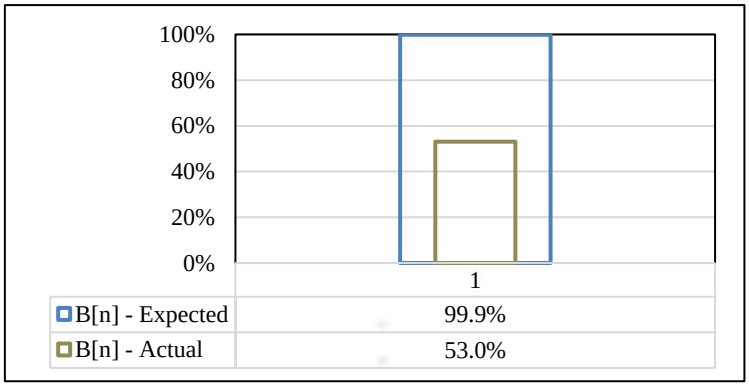

Figure 6-21 Case 2 – Benefit of Mission System Expected and Actual

Moreover, the reason being for the actual benefit B[1] to be less than the expected is attributed from the threat identified and its mitigation process depicted in Figure 6-22 for Case 1 and Figure 6-23 for Case 2.

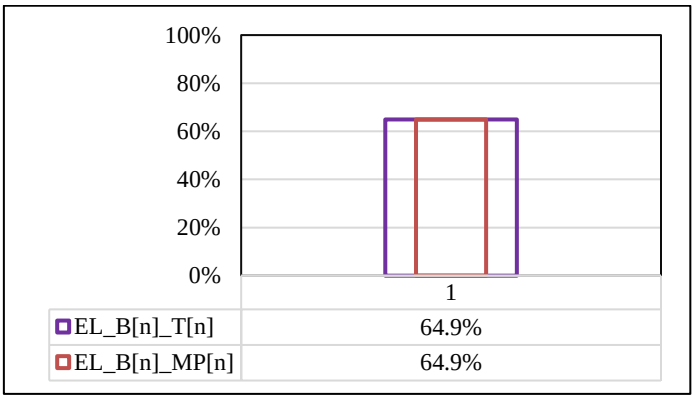

Figure 6-22 Case 1 – Effectiveness Level of Threat and Mitigation Process

[53] These will be presented further in this section.

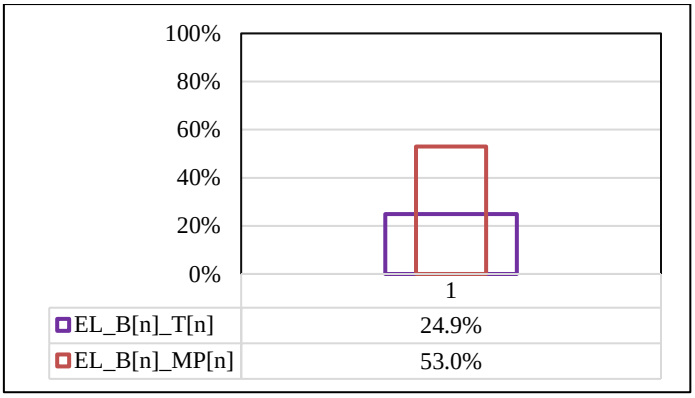

Figure 6-23 Case 2 – Effectiveness Level of Threat and Mitigation Process

For Case 1, since the Mission-Critical System MCS[1] reached only TRL1 there is no major mitigation for the threat T[1]. On the other hand, in Case 2, the MCS[1] has reached TRL4 thus the threat is minimised to 24.9% as shown in Figure 6-23 where the effectiveness level of the B[1] benefit with the threat T[1] was 53%, Section 6.5.15. However, this is how the actual benefit and the mission success was extracted for both cases.

Using Figure 4-1, the threat T[1] and the Threating System TS[1] should be analysed further. It can be noticed that in both cases, the threat T[1] have a great impact.

Threat: **T[1]** – 30mm ADPS.

Threatening System: **TS[1]** – ADPS System.

For Case 1, Figure 6-24, threat T[1] estimated to be,

TL_T[1] : [(O[n]:15%)+(SE[n]:0%)+(D[n]:20%) = 35%

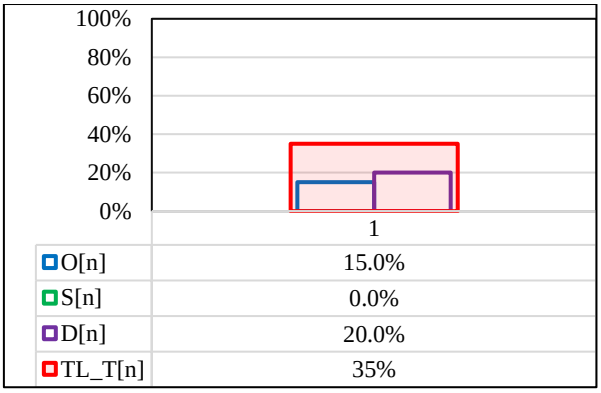

Figure 6-24 Case 1 – Threat Level (TL_T[1] = O[1]+SE[1]+D[1])

For Case 2, Figure 6-25 which is a Light Armour Vehicle, the same threat has a greater impact on the mission and is estimated to be,

TL_T[1] : [(O[n]:15%)+(SE[n]:50%)+(D[n]:10%) = 75%

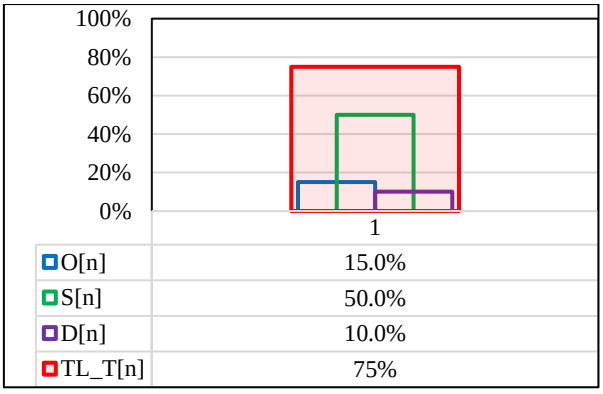

Figure 6-25 Case 2 – Threat Level (TL_T[1] = O[1]+SE[1]+D[1])

Based on Figure 6-24 and Figure 6-25, it is easy to observe how the same threat can affect the mission. For this purpose, the mitigation process shall be able to reduce this effect.

According to Section 6.5.17, three different Mission-Critical systems were the candidates for the mitigation process.

Mission-Critical System – **MCS[1]**: Obscurants

Mission-Critical System – **MCS[2]**: Smoke Grenades

Mission-Critical System – **MCS[3]**: Flares

For demonstration purposes only, Figure 6-26 and Figure 6-27 depict how different Mission-Critical systems perform on the same threat by achieving different TRL levels.

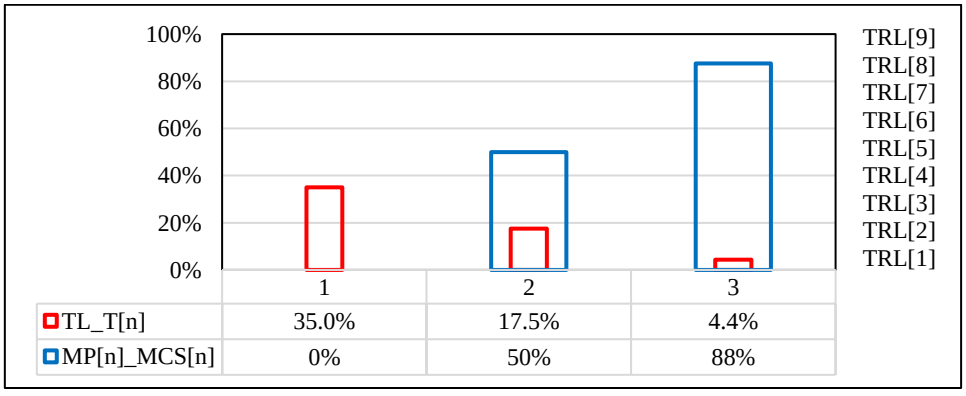

Figure 6-26 Case 1 – Mitigation Process

Where: MCS[1] reached TRL1 (0%) thus, the threat T[1] still has the same affect (35%), MCS[2] reached TRL5 (50%) thus, threat T[1] is downgraded to 17.5% and finally, MCS[3] reached TRL8 and that is 88% of the mitigation process and the threat T[1] reduced to 4.4%.

On the other hand, Case 2 had different results and are as follows.

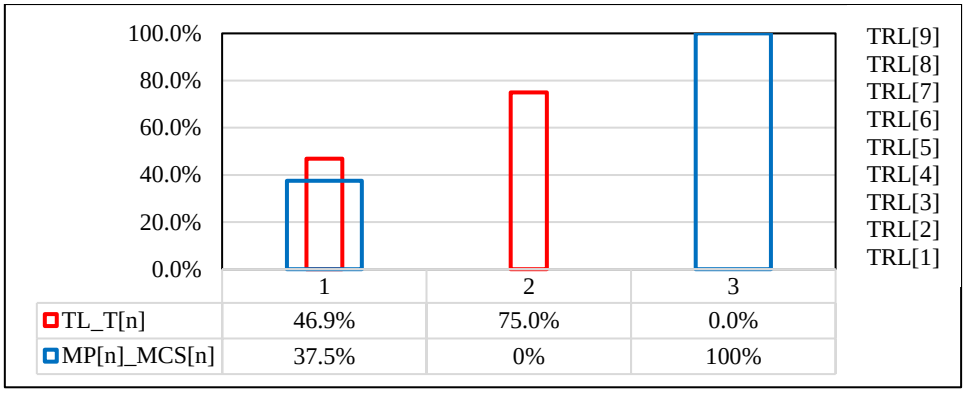

Figure 6-27 Case 2 – Mitigation Process

As mentioned in Section 6.5.23, MCS[1] reached TRL4 and that is 37.5% of the mitigation process. As can be seen in Figure 6-27, MCS[1] is not satisfactory enough for the mission thus, further developments will be required. However, MCS[2] is only in theory thus, TRL1 is reached and that is 0% of the mitigation process and finally, MCS[3] which is TRL9, is the best candidate Mission-Critical System for mission M[1]. Meaning that, if MCS[3] is agreed to be developed and integrated on the platform, the mission could be successful with 99.9% chances.

Finally, reaching to this point of the framework, stakeholders, systems engineers, system architects, suppliers, researchers and anyone wishes to contribute in the development of a Mission-Critical System, should be able to have a brief estimation of how a mission will perform. Using these results and additionally, following the principles of developing a Mission-Critical Data Model to such systems, integration, development and operational risks will be reduced significantly.

6.7 Conclusions and Future Work

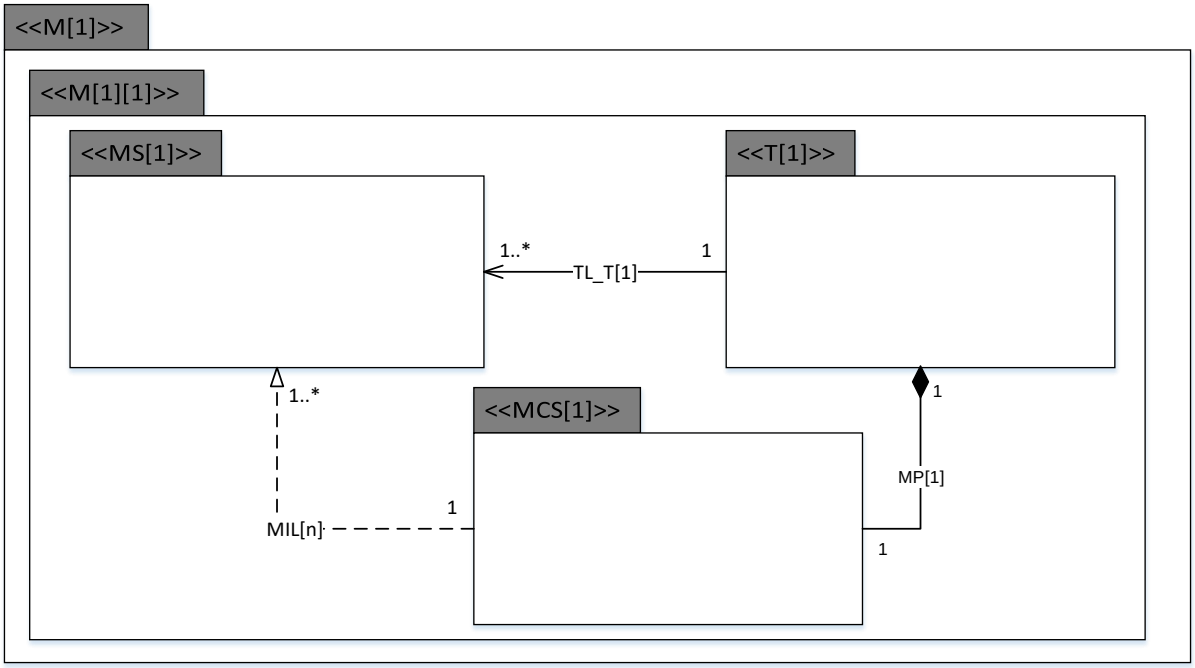

Figure 6-28 Mission - M[1] Data Model

<<M[1]>>: Survivability.

<<M[1][1]>>: Do not be hit.

- **<<MS[1]>>**: Survivability system.
- **<<T[1]>>**: 30mm ADPS.
    - **<<TL_T[1]>>**: 35% (Case 1) and 75% (Case 2)
- **<<MCS[1]>>**: Obscurants.
    - **MP[1]**: TRL1 (Case 1) and TRL4 (Case 2)
    - **MIL[n]**: MIL(n) (Is considered for future work)

This is the result of the framework, which is depicted in Figure 6-28, indicating the abstraction relationship and definition of the Mission – M[1] and [1][1], the system essential for the mission – MS[1], including its integrity level – MIL[n], the potential threat – T[1] alongside with the threatening level TL_T[1]. The Mission-Critical system MCS[1] that will potentially assist the mission and prevent the threat to negatively impact the mission MP[1] is also specified.

The framework intentionally used a very basic case study in order to present in the simplest way the novel contributions of this research. The framework aims to assist and encourage stakeholders, researchers, suppliers, systems engineers and architects to use a unified approach in order to design a Mission-Critical System effectively and efficiently. Furthermore, if Mission-Critical Data Modelling approach is used, the platforms using the IOA approach for

their architectures can share criticality levels between their data exchange so that the platform will autonomously be able to act accordingly on specific missions or sub-missions.

From the results, in Section 6.6, anyone who's participating in the system development will have a clearer picture on how the mission will perform using examples, of mission systems, threats and Mission-Critical systems. Once, the estimation is completed in the concept phase of the system, the system should be able also to estimate how it will perform when a specific threat is detected and so on.

In the future, the current state of this framework has many missing elements. These missing elements are oriented towards the SysML language approach and are shown in Figure 6-29.

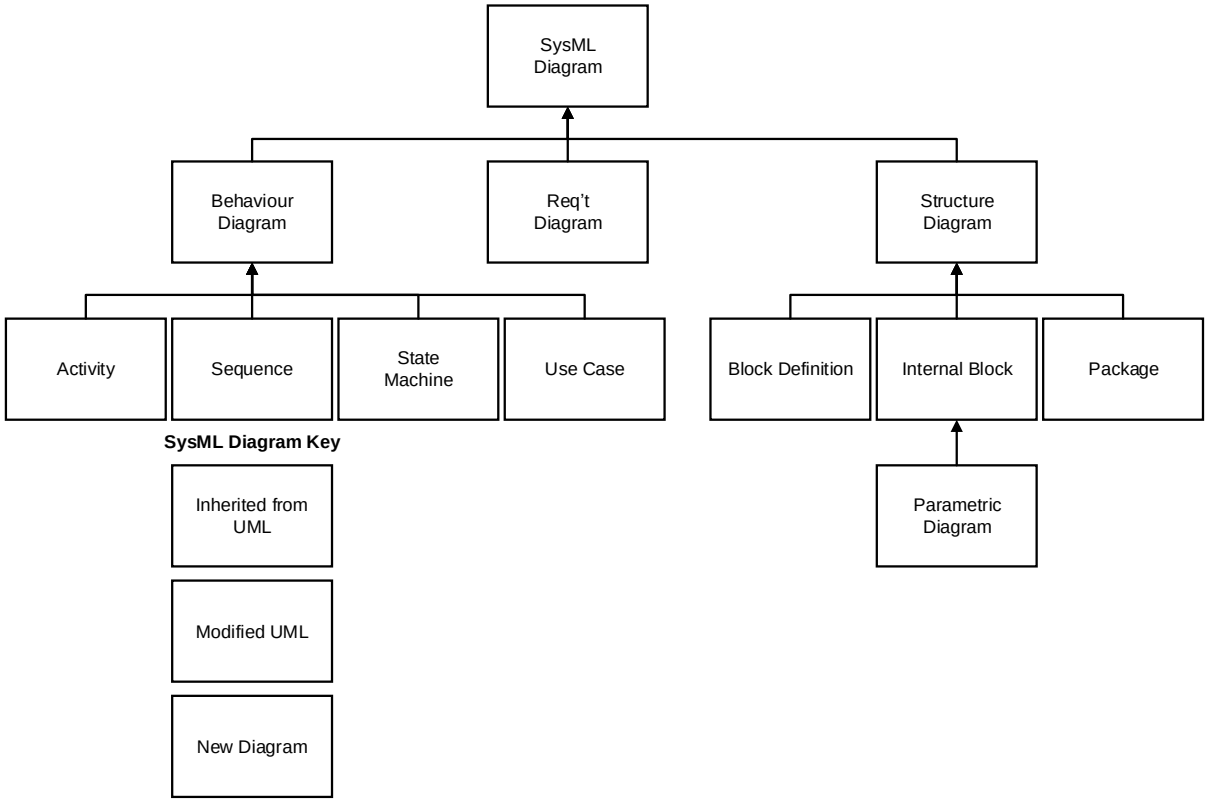

Figure 6-29 SysML Approach Diagram

SysML Diagram – Represent a model element.

- **Behaviour Diagram** – Represents function-based behaviours.
  - **Activity** – Specifies transformation of inputs to outputs through a controlled sequence of actions.
  - **Sequence** – Provides a "dynamic" view of ordering interactions and message flow.
  - **State Machine** – Used to represent the life cycle of a block.

    - **Use Case** – Provide means for describing basic functionality in terms of usages/goals of the system by actors.
- **Requirement (Req't) Diagram** – The <<requirement>> stereotype represents a text based requirement.
- **Structure Diagram** –To provide a common appearance between other structures and define any cross-referencing between structures.
    - **Block Definition** – Provides a unifying concept to describe the structure of an element or system.
    - **Internal Block** – Describes the internal structure of a block in terms of its properties and connectors.
        - **Parametric Diagram** – Used to express constraints between value property.
    - **Package** – Is used to organise the model.

Using the SysML approach for this framework it will be more likely the framework to be more sophisticated and therefore, mission-related systems more "dependable".

Lastly, as future work of this chapter, the Node-RED tool can be used and be able to reach TRL4 levels of the Mission-Critical system development. Using these tools, the engineers will be able to understand how a message specification can be improved by applying additional features which are discussed in the proposed taxonomy of the Mission-Critical system. Consider that, the threat node is producing the "raw data" and with the usage remaining elements of the modelling and functional simulation platform design of the DAS system, the DAS system will be able to apply functions beyond dealing with the threats. It can be used for other platforms, soldiers, headquarters and so on if all of them share the same message specification in their data exchange.

Chapter 7 General Conclusions

7.1 Conclusion

As stated earlier at the thesis, defining a Mission-Critical system can be challenging. It has been identified that defining a Mission-Critical system is not that simple. This has been proven by reviewing various related published papers. These papers are describing Mission-Critical systems upon their needs. These needs are typically the expectations of such systems. From these papers, some of the descriptions were similar to what a dependable system should be characterised. For a system to be dependable, it has to consist of capabilities such as properties and dimensions described in Laprie's system taxonomy. However, many standards have been produced to enhance the process of the system life-cycle to be dependable. Dependable could vary upon application, for example, safety focuses on safety-related capabilities thus the ISO 26262 and so on. Despite this, within the defence, there are not related standards focusing on Mission-Critical systems. When it comes to Vetronics systems usually the electronic systems integrated on land military platforms are considered as Mission-Critical. Apart from that, this thesis proved that anything could be considered as Mission-Critical. An example was given proving that a fuel tank system can be equally considered as Mission-Critical system similarly to an expensive surveillance system.

Moreover, this thesis presented a framework that could potentially assist people who are involved in the development of a Mission-Critical system following what is currently available today. It has been mentioned that developers of Mission-Critical systems, were facing issues of system integration, thus the middleware technology and the data model approach. However, these data models that describe the architecture of Mission-Critical systems were lack of data exploitation and the absence of the criticality between data, entities and relationships. Therefore, the presented framework of this thesis covers what is currently unavailable in defining a Mission-Critical system and additionally, how critical is each data within the data model is. This has been accomplished when mathematical evaluations introduced in the framework.

The extracted qualitative and quantitative results from the framework using a real Mission-Critical system, it is possible to de-risk the system's development from the concept phase. This is critical for developers to be aware, not only how the system should be developed but also how it could perform during application. With this estimation of the Mission-Critical system's performance could safe costs such time. The case study showed that a mission can vary from application to application depends on what elements are used. It has been presented that one threat can vary in level due to the mission scenario. A single threat has

been estimated to be effective on different platforms, even when each platform was using different Mission-Critical systems in terms of capabilities. Therefore, it can be costly to use high-end products for specific applications without a sophisticated early de-risk procedures from the very early stages of the system's development. Hence, this framework's approach is to gather all the participants of the system development (e.g. stakeholders, systems engineers and architects, researchers, suppliers and so on) and attempt to develop it into its full potential by having commonality between them as well as within the E/E/PE systems and architecture.

Finally, a major lesson learnt from this research is that describing a Mission-Critical system requires a lot of effort but when there is a close interaction between participants and by having an approach that everybody can understand, it is possible that missions could be more successful. Besides that, if that interaction between participants embedded in these systems the overall platform will have its own capability of defining criticality and decide whether is mission will be successful or not. In other words, if the overall platform possesses an algorithm similarly to the qualitative and quantitative results, the platform could potentially estimate the possibility of the mission's success. This could be achieved when the platform can observe the detected threat's affect level, the availability of mission and Mission-Critical systems and their benefits and mitigation processes. Nevertheless, this is after the concept phase of the Mission-Critical system and can be considered as future work and recommendations followed by the next section of this thesis's conclusions. A more analytical conclusions are provided below covering all the sections disccussed in the introduciton of this thesis.

7.2 Conclusion – Research Challenges

This part of the conclusion focuses on the challenges discussed in Chapter 1. The challenges of this research were mainly concerns about defining and developing the Mission-Critical system.

7.2.1 To Specify Mission-Critical Systems Effectively and Efficiently

This challenge, Section 1.2.1, pointed out that a Mission-Critical system can be any system that is involved in the mission. This challenge has been also defined and analysed in Section 3.1 leading to a conclusion that a Mission-Critical system is considered Mission-Critical based on the aims and objectives of the primary task. However, in this research, a Mission-Critical system can be defined by using and following the proposed framework described in Section 4.3. Additionally, in Section 6.6, an estimation of how a mission will be performed using specific Mission-Critical elements has been demonstrated at the very early stages of the system development or programme.

7.2.2 To Describe Mission-Criticality Between Interoperable Systems

This challenge discussed the concerns of the systems that are integrated into interoperable open architectures and that share their data across the network. The main concern was when sharing data across a network that is interoperable how the data can also declare its criticality in various systems and applications. This challenge has been addressed in Section 4.3.6 and demonstrated in Section 6.5.24. When a message contains Mission-Critical data, the approach of this challenge was to include also a Mission Integrity Level (MIL) data so that other systems will perform accordingly.

7.2.3 To Estimate Platform's Mission Success/Impact Prior the Design and Development Phase

It has been defined that if a system must make changes either in software or hardware during the development it can be costly. Usually, this is due to the stakeholders which are not sharing clear definitions and not satisfied with the results of the already developed system. However, in this research, this challenge has been addressed in Section 4.3.3 and demonstrated in Section 6.6. With this approach, the estimation of success and impact of the platform's mission can be estimated prior to the design and development phase.

7.2.4 To De-Risk the Integration Process of Mission-Critical Systems

This challenge has been raised when the principles of the Interoperable Open Architecture (IOA) have been defined and explained. IOA is an architecture that integrates the various system on it meaning that the integration of a Mission-Critical system can be challenging. As well as, when a programme is developed multiple stakeholders with different backgrounds are interacting together thus this is also challenging.

If there is no consistency during the development of a programme or a Mission-Critical system the integration process will be flawed which this results in additional costs. However, this challenge has been addressed in the framework during the data modelling process as depicted in Figure 4-1 and has been analysed further in Chapter 5. It has been also demonstrated in Section 6.5 when the data model was developed.

7.3 Conclution – Research Questions

This part of the conclusion focuses on the research questions discussed in Chapter 1. The research questions were asked prior and during this research. Below are the conclusions of this research questions.

7.3.1 What Would be the Best Standard to Follow and Standardise Mission-Critical Systems

This research question came forth during the research done to define Mission-Critical systems. As has been mentioned earlier in this research, there are various well-defined standards that are for functional safety, cyber-security and mission assurance. All of those standards are created to satisfy safety, security and missions. However, the question was, what you be the best standard to follow as a reference standard of this research. The answer was to understand the objectives of those standards and then unify and generalise them into one objective so that a mission can be applicable missions. The closest related and most applicable standards were the ones for functional safety but in safety, a mission will be aborted if a hazard is identified. For that, Section 4.3.5 has discussed how a system shall conceptually perform in case of an identified threat (internally or externally to the platform) and in Section 6.5.23 how a requirement shall be formed.

7.3.2 Who Would Be The Stakeholders and How It Can Be Demonstrated

A question has been asked on to who would be the people interested in Mission-Critical system development and deployment. It has been estimated that various stakeholders will be interested in different aspects of the development and deployment hence, the demonstration focused on three main stakeholders. First were the stakeholders interested in how the system will perform using numerical results. From Section 6.6 the qualitative and quantitative results indicate how precisely a mission will perform with all the involved elements, threats, systems and Mission-Critical systems in the concept level and in business level.

However, another estimation of who would be the other stakeholders were the stakeholders interested in the engineering side of the development and deployment of Mission-Critical systems. Typically, engineers are not persuaded by just number results hence, by using graphical representations such as the UML, the engineers can understand whether the system is capable of achieving its mission or not. UML has proved to be a useful tool to demonstrate the behaviour of Mission-Critical systems so that when the V-Cycle proceeds to the next steps, the engineers will be confident enough to develop a Mission-Critical system with consistency.

7.3.3 What Tools Shall be Used for this Research

Another question of this research was, what are the tools needed and to be used as proof of the concept. The tools used in this research were mainly tools to achieve the aims and objectives of this research. The main tool of this research was the Failure Mode and Effect Analysis tool. FMEA is a tool that was the best candidate tool for this research that could assist define the mission, identified threats and design mitigation process of the identified threat.

Despite that, FMEA on the other was lacking being interpreted into the interoperable open architecture world which is currently evolving in land military platforms. With this missing element, the data modelling approach and the mathematics are combined with the FMEA and form this research's framework. Other tools such as the Unified Modelling Language and the testbed's simulator were used but in a not high degree.

At the end of this section, it must be noted that the ultimate aim of this research has been fulfilled by answering the research challenges and questions that raised pre, during and post of this research.

7.4 Conclusion – Reseach Aims and Objectives

Finally, this section of the conclusion is covering the aims and objectives discussed in the introduction chapter of this thesis. Further, are the main aims and objectives' conclusions.

7.4.1 Clearly Define Mission

The aim was successfully succeeded by the creation of a framework that can combine definition, specification, threat analysis and mitigation process of mission and its critical elements efficiently and effectively. As discussed in Chapter 4 and proved in Chapter 6, the framework is capable of combining the aforementioned elements. . Also, the main objective has successfully achieved that the framework could enhance the data specification of mission-related systems and to describe the benefits, risks and mitigation values for Mission-Critical elements of each mission that will be deployed.

7.4.2 Assist to identify Mission and Mission-Critical systems

This thesis has successfully provided an approach that ultimately provides a clear statement between to any stakeholders such as engineers and suppliers to pre-defined attributes of Mission and Mission-Critical elements at the early stages of the V-Cycle development process. The approach that this research is able to provide the necessary information into a way such that, multidisciplinary partakers will effectively review and perform to achieve the full life-cycle and functionality of Mission-Critical systems and their critical elements. This has been accomplished by reviewing what were the best methods and tools such as the UML and FMEA discussed in Chapter 5 and Chapter 3 respectively, then combined and formed in Chapter 4 and finally deployed and proved in Chapter 6.

7.4.3 Early De-Risk Demonstrator

The final aim and objective of this research have been also accomplished by the presentation and the proof of concept of the case study in Chapter 6. The aim of the demonstration has

presented how this research's approach can be used to design an existing or non existing Mission-Critical system in military platforms and by extracting the Mission-Critical aspects of the bespoke system using qualitative and quantitive results. A fast early de-risking capability in the very early stages of mission-related systems has been also presented in Chapter 6 that can be developed and understood by all the stakeholders of the V-Cycle development process. An early de-risking demonstrator as intially presented in Figure 1-2 of this reseach, Section 6.6 and Figure 6-28 can represent the very early stages of the mission and its elements to various stakeholders, efficiently and effectively.

7.5 Future Work and Recommendations

There is always a potential of improving a concept or idea, hence, the rest of this section will focus on recommendations that could potentially be used to improve each chapter's objective.

Starting from the framework, additional methods for the development and deployment of a Mission-Critical or mission-related system can be used, similarly to the industry and military (safety-critical systems). These additions could potentially be able to enhance the desired result of such systems. Methods such as Layer Protection Analysis (LOPA), Hazard and Operability Study (HAZOP), Independent Layer of Protection (IPL), Safety Requirements Specifications (SRS), Functional Safety Assessment (FSA), Fault Tree Analysis (FTA) and the tools needed for the elaboration of the concept to an actual testbed. Afterwards, the migration from that testbed to a more elaborated testbed demonstrator.

Mission Integrity Levels (MIL) within the framework are assumed to be assigned to Mission-Critical systems to indicate their integrity on different missions and how will impact the mission in case of a failure. In other words, the likelihood of the mission-related system satisfactorily performing the required mission functions under all the stated conditions within a stated period of time. However, in this research, there is not an essential activity to assign MILs to MCS[n]s, hence, this is considered as future work. The work will be looking into more detail at safety-related activities, such as the IEC 61508 and ISO 26262 and (A)SIL levels.

A literature review was conducted in this thesis, in order to derive and collect enough information for Mission-Critical systems. The chapter recommended that there is a need for more mission-related and critical-related systems to be reviewed in order to increase the knowledge of how a Mission-Critical system should be characterised. With this work, systems engineers and architects will be more aware of Mission-Critical system's characteristics such as dependability, properties, dimensions and threats. Lastly, when more information is gained from the work, it is more likely Mission-Critical systems will be able to enhance the mission successfully.

An electronic architecture and electronics instrumentation for Mission-Critical and mission-related systems were proposed. The proposal was discussed and provided, concepts of constructing Mission-Critical oriented architectures and the orchestration of Mission-Critical or non-Mission-Critical electronic components. However, it has been mentioned that there is still more discussion needed for these architectures. For example, communication networks, hardware, software, operating system etc. Therefore, the future work of this, will be looking at other Mission-Critical system's electronic architectures to expand and modify this proposed Mission-Critical oriented architecture and electronics instrumentation. This will be extracting requirements for an optimum Mission-Critical oriented architecture, that will be useful to achieve types of different missions.

A testbed will be also be constructed as a TRL4 proof of concept, providing:

- Mission's rapid prototyping requirements.
- Mission-Critical functions interpretation and criticality exchange.
- Early de-risk capabilities, in the early stages of the mission-related system's development.
- The use of affordable tools and components to achieve Mission-Critical capabilities.
- The migration of high-level implementations to a low level, effectively and efficiently.

The data model and data model notations have been also discussed earlier. The data model notation used for this research was the UML notation. As was mentioned earlier, an improved version of the UML notation is the Systems Modelling Language (SysML). SysML consists of extra features that are useful for the system's design and development. Features such as "Requirement Diagram", "Behaviour Diagram", "Structure Diagram" and "Parametric Diagram". Since SysML has these extra features, it can potentially provide a more precise definition and guidance for the development of Mission-Critical and mission-related systems. Furthermore, that could be useful for better understand and offer a clearer picture between participants and E/E/PE systems.

With the aid of the MDA approach, it could be useful for approaches needed to accomplish a rapid prototyping testing, early de-risking development and permitting software functionality and operation of a Mission-Critical system using low-cost components systematically. Also, using the MDA could achieve a transition from a low-cost functional testbed to a more elaborated Mission-Critical performance verification testbed using this research's approach, if the appropriate tools are used for this migration.

7.6 Limitations and Constraints

Despite the fact that this research's main objective was the potential of designing Mission-Critical and mission-related systems successfully, efficiently and effectively, some limitations, drawbacks, constraints or concerns raised. Below, some of the identified limitations and constraints of this research are discussed.

"*Limitations – The acts of controlling and especially reducing the process*" [103]

"*Constraints – Conditions and/or resource requirement limitations affecting the process"* [104]

Limitations can be described as, the restriction beyond which movement or activity in any component of a system does not occur. And constraints can be described as, the restriction on the natural degrees of freedom of a system; the number of constraints is the difference between the number of natural degrees of freedom and the number of actual degrees of freedom.

"Because human's knowledge does not meet boundaries - it can be infinite and as a result, this will be converted into an extreme disadvantage for this purpose" - Author

The above expression extracted on the phase where the Mission System MS[n] should analyse its purpose. It is reasonable to say that everything in this world has its own purpose. When the purpose is described, is upon the entity that is exclusively focusing on resolving existential questions about theories or comments. Likewise, some philosophers asked some questions about cosmology, that are still not answered up to this moment. Therefore, when the mission, mission system, threat, Mission-Critical system are described, there is always a potential that there is something unknown or undescribed within these entities, hence, the 99.9% value for each B[n]. Moreover, if a user requires for the mission to be ultimately successful many considerations are needed and as a result, this will be converted as a threat against procurement costs and/or time.

TRL levels were described and discussed where can be applied within this research's framework. However, there are some TRL levels, such as the TRL9, is defined to as, "Actual system "mission proven" through successful mission operations". Therefore, in this research TRL9 which is the last level of the readiness levels cannot be reached, thus, this is converted into a limit for this research.

"*A Mission-Critical system cannot be specified or narrowed down into a single element that easily*". *- Author*

In defining Mission-Critical systems and their taxonomy, can be challenging. As discussed earlier in this study's, in the literature review of defining Mission-Critical system's characteristics, each defined Mission-Critical system is described upon the system's purpose and user's needs. Hence, the expression,

"The safety prioritises the safety of people and environment; the security prioritises the protection of data from various threats; survivability prioritises the whole mission envelope; the procurement prioritises the costs and the bureaucracy prioritises the political associations of the government." - Author

Another part of the "Limitations and Constraints" is what would the Mission-Critical system do in terms of "decision making", to ensure that the core mission is not compromised. Meaning that, if a Mission-Critical system is designed within the survivability discipline, that means safety is compromised and therefore, should system developer responds to negligence liability or product liability[54]? What if, the system is designed for battlefield applications only? These concerns can be converted into a Mission-Critical system's design, limitations and constraints.

Another, unmentioned concern within this study is based on Murphy's law and is as follows,

"Anything that can go wrong, will go wrong" [105]

For this research, this quote will be translated into, the possibility of the mission to fail. Meaning that if a threat cannot be dealt with, using mission elements that cannot cope with the specific threat, then the overall mission will fail, regardless. Therefore, this concern could be out of limits for this research's objectives.

[54] Consumer Protection Act (CPA).

References

[1] K. Fowler, “Mission-Critical and Safety-Critical Systems Handbook: Design and Development for Embedded Applications,” *Newnes*, pp. 1–82, 2009.

[2] A. Roland, “War and Technology: A Very Short Introduction,*" Oxford University Press*, 2016

[3] A. Chong, “Driving Asia: As Automotive Electronic Transforms a Region,” *Infineon Technologies Asia Pacific Limited*, pp. 13-27, 2011

[4] W. Fleming, “Forty-Year Review of Automotive Electronics: A Unique Source of Historical Information on Automotive Electronics,” *IEEE Vehicular Technology Magazine*, vol. 10, no. 3, pp. 80–90, 2015.

[5] A. Albert, “Comparison of Event-Triggered and Time-Triggered Concepts with Regards to Distributed Control Systems,” *Embedded World Conference*, pp. 235-252, 2004

[6] I. Knight, A. Eaton and D. Whitehead, “The reliability of electronically controlled systems on vehicles,” *Unpublished Project Report - The International Motor Vehicle Inspection Committee (CITA) Working Group 7*, 2001

[7] N. Navet, Y. Song, F. Simonot and C. Wilwert, “Trends in Automotive Communication Systems”, in *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1204-1223, 2005

[8] "Suppliment to Automotive News,” *Crain Comminications Inc., 2013,* [Online]. Available: https://www.autonews.com/assets/PDF/CA89220617.pdf, [Accessed: 17-Mar-2017]

[9] “Automotive News, Top Suppliers,” *Crain Comminications Inc., 2016*, [Online]. Available: https://www.autonews.com/assets/PDF/CA100044612.pdf, [Accessed: 17-Mar-2017]

[10] “Adoption of ISO/IEC 15288:2002 Systems Engineering-System Life Cycle Processes,” in *IEEE Std 15288-2004 (Adoption of ISO/IEC Std 15288:2002),* pp. 0_1-67, 2005.

[11] IEEE Power Engineering Society, “IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation,” *IEEE Std 1646,* 2004, pp. 0_1-24, 2005.

[12] M. Felser, “The Fieldbus Standards: History and Structures,” *University of Applied Science Berne, Hochschule für Techik und Architektur, Bern Morgartenstrasse 2c, 3014 Bern*.

[13] G. Leen and D. Heffernan, “Expanding automotive electronic systems,” IEE *Computer and Control Engineering, (Long. Beach. Calif).*, vol. 35, no. 1, pp. 88–93, 2002.

[14] B. Andersson, N. Pereira, and E. Tovar, “Analyzing TDMA with slot skipping,” *26th IEEE International Real-time Systems Symosium (RTSS'05)*, Miami, FL, pp. 10-24, 2005.

[15] J. J. Scarlett and R. W. Brennan, “Re-evaluating *Event-Triggered and Time-Triggered Systems,” 2006 IEEE Conference on Emerging Technologies and Factory Automation*, Prague, pp. 655-*661*, 2006

[16] T. Nolte, H. Hansson and L. L. Bello, "Automotive communications-past, current and future," *2005 IEEE Conference on Emerging Technologies and Factory Automation*, Catania, pp. 8-992, 2005

[17] United Kingdom, Ministry of Defence, Gateway “UK MOD Defence Gateway,” Online Service, [Online]. Available: https://sts.defencegateway.mod.uk/register.aspx, [Accessed: 23-Jan-2016]

[18] “The UK MOD Generic Vehicle Architecture: A Compelling Case for Interoperable Open Architecture.” [Online]. Available: https://info.rti.com/hubfs/Collateral 2017/Whitepapers/UK_Mod_Generic_Vehicle_Architecture_50010.pdf?t=1485376711 661.

[19] M. A. Mastouri and S. Hasnaoui, “Performance of a Publish/Subscribe Middleware for the RealTime Distributed Control systems,” *IJCSNS International Journal of Computer Science and Network Security, Vol.7 No.1*, 2007

[20] Object Management Group, “Data Distribution Service for Real-Time Systems Specification,”, *Object Management Group*, *Objective Interface Systems, Inc., Real-Time Innovations Inc., Thales*, Open Specification, 2001.

[21] Government of United Kingdom, “Defence Gateway (DGW).”, *United Kingdom - Ministry of Defence* [Online], Available: https://www.gov.uk/guidance/defence-gateway, [Accessed: 23-Jun-2015]

[22] North Atlantic Treaty Organisation, “STANAG 4754, NATO Generic Systems Architecture (NGVA) for Land Systems,.” Edition 1, Ratification Draft 1,” NATO Standardization Office (NSO), 2016.

[23] “Future Airborne Capability Enviroment (FACE),” *Open Group*, United States of America, Department of Defence*,* Web-Based Open Standard

[24] “Vehicular Integration for C4ISR/EW Interoperability (VICTORY),” Web-Based Open Standard

[25] “Multilateral Interoperability Programme (MIP),” *North Atlantic Treaty Organisation (NATO)*, Web-Based Open Standard

[26] “Layered Approach to Service Architectures for a Global Network Environment (LASAGNE),” *Australian Government, Department of Defence*, Science and Technology, Web-Based Open Standard

[27] R. A. Simons, "Levers of Control: How Managers Use Innovtive Control Systems to Drive Strategic Renewal," *Harvard Business School Press*, 1995.

[28] F. Ciccozzi, I. Crnkovic, D. Di Ruscio, I. Malavolta, P. Pelliccione, and R. Spalazzese, “Model-Driven Engineering for Mission-Critical IoT Systems,” *IEEE Software*, Vol. 34, no. 1, pp. 46–53, 2017.

[29] J. Rushby, “Critical system properties: Survey and Taxonomy,” in *Reliability Engineering and System Safety*, Vol. 43, no. 2, pp. 189–219, 1994

[30] D. Dasgupta and M. Cavarlo, “Designing Resilient Mission Critical Systems,” *Secure and Cyber Architecture Conference,* 2010.

[31] A. Koski and T. Mikkonen, “On the Windy Road to Become a Service Provider: Reflections from Designing a Mission Critical Information System Provided as a Service,” in *Proceedings International Conference on Information Systems Engineering, ICISE 2016*, pp. 51–56, 2016

[32] Motorola Inc., “Ensuring Resilience and Availability in a TETRA System Ensuring Resilience and Availability in a TETRA System Key Steps to Help Ensure Critical,”

Position Paper - Survey, 2008

[33] International Electrotechnical Commision, “Dependability Standards and Supporting Standards,” *IEC TC-56 Standard*, 2015.

[34] J. Laprie, “Dependable Computing and Fault Tolerance: Concepts and Terminology,” *TwentyFifth International Symposium Fault Tolerant Computers 1995 Highlights from TwentyFive Years*, pp. 2–11, 1995.

[35] A. Deshpande, O. Obi, E. Stipidis and P. Charchalakis, "Security in integrated vetronics: Applying elliptic curve digital signature algorithm to a safety-critical network protocol-TTP/C," *7th IET International Conference on System Safety, incorporating the Cyber Security Conference 2012*, Edinburgh, pp. 1-5, 2012

[36] O. Obi, A. Deshpande, E. Stipidis and P. Charchalakis, "Intrusion tolerant system for integrated vetronics survivability strategy," *8th IET International System Safety Conference incorporating the Cyber Security Conference 2013*, Cardiff, pp. 1-6, 2013

[37] A. Deshpande, O. Obi, E. Stipidis and P. Charchalakis, "Integrated vetronics survivability: Requirements for vetronics survivability strategies," *6th IET International Conference on System Safety 2011*, Birmingham, pp. 1-6, 2011

[38] R. M. Connor, “Vetronics Standards and Guidelines, QINETIQ/EMEA/TS/CR0702540 Issue 3,” *Vehicle System Integration,* Web-Based Open Standard

[39] S. Kapurch, “NASA Systems Engineering Challenges” *Presentation*, [Online]. Available: http://www.dtic.mil/ndia/2011/system/TuesdayKapurch.pdf.

[40] K. J. Schlager, "Systems engineering-key to modern development," in *IRE Transactions on Engineering Management*, vol. EM-3, no. 3, pp. 64-66, July 1956.

[41] E. C. Honour, “Understanding the Value of Systems Engineering”, *INCOSE International Symposium*, Online Library, 2014

[42] Defence Acquisition University, "Systems Engineering Fundamentals,” Defence Technical Information Center, pp. 31-73, 2001.

[43] Internationl Electrotechnical Commision, “Functional safety and IEC 61508: A Basic Guide,” *IEC International Electrotechnical Commission*, pp. 3-5, 2004.

[44] P. Campbell, "Department of Defense Instruction 8500.2 “Information Assurance (IA) Implementation:” A retrospective," *2012 IEEE International Carnahan Conference on Security Technology (ICCST)*, Boston, MA, pp. 187-194, 2012

[45] J. Evans, S. Cornford, and M. S. Feather, “Model based mission assurance: NASA’s assurance future,” in *Proceedings - Annual Reliability and Maintainability Symposium*, 2016.

[46] C. Williams, J. Ibbotson, J. Lockerbie, and K. Attwood, “Mission Assurance through Requirements Traceability,” in *Proceedings - IEEE Military Communications Conference MILCOM*, pp. 1645–1650, 2014

[47] K. Jabbour and S. Muccio, “The Science of Mission Assurance,” *Journal of Strategic Security*, vol. 4, no. 2, pp. 61–74, 2011

[48] R. Cressent, P. David, V. Idasiak and F. Kratz, "Dependability Analysis Activities Merged with System Engineering, a Real Case Study Feedback," *Le Centre pour la Communication Scientifique Directe,* Troyes, France, 2011 France.”

[49] G. Forest, “Quick Guide to Failure Mode and Effects Analysis,” *Six Sigma*, Web-Based Open Access

[50] Internationl Standards Organization, “ISO 26262. Road vehicles – Functional safety.” 2011.

[51] D. J. Smith and K. G. L. Simpson,"Safety Critical Systems Handbook: A Straightforward Guide to Functional Safety, IEC 61508 (2010 Edition) and Related Standards," *Elsevier Ltd*. 2011

[52] E. M. Marszal and E. W. Scharpf "Safety Integrity Level Selection: Systematic Methods Including Layer Protection Analysis", *Instrumentation, Systems and Automation Society, 2002*

[53] N. Storey, "Safety-Critical Computer Systems,*" Prentice Hall, 1996*

[54] B. Sauser, J. Ramirez-Marquez, D. Verma and R. Gove "From TRL to SRL: The Concept of Systems Readiness Levels," *Conference on Systems Engineering Research*, Los Angeles, CA, pp. 126, 2006

[55] J. Fülöp, “Introduction to decision making methods,” *Laboratory of Operations Research and Decision Systems, Computer and Automation Institute,* Hungarian Acedemy of Sciences, 2004.

[56] C. Raistrick, “Land Data Model Methodology Description,” *Generic Vehicle Architecture (GVA),* 2016

[57] C. Coronel, S. Morris "Database Systems: Design Implementation Management," *Course Technology,* Edition 11, 2014

[58] D. C. Hay, “A Comparison of Data Modeling Techniques,” *Essential Strategies Inc.,* 1999

[59] B. Schmidt and D. Warren, "Data Modeling For Information Professionals," *Prentice Hall*, 1998

[60] P. Merson, “Data Model as an Architectural View,” *Software Engineering Institute,* 2009.

[61] S. W. Ambler, “Discipline Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise" *IBM Press, 2012*

[62] G. Booch, J. Rumbaugh, I. J. Booch, "The Unified Modeling Language User Guide (Object Technology Series)," *Addison-Wesley Professional*, 2005

[63] N. Routledge, L. Bird, A. Goodchild, “UML and XML Schema,” *ADC '02 Proceedings of the 13th Australasian Database Conference, pp. 157-166,* 2002

[64] P. Pendle, "Right Data, Right Place, Right Time Storage tiering for the DB2 Database Administrator"" *IBM Corporation, 2010*

[65] J. Collin, "The Right Data in the Right Place at the Right Time," *Northwest Analytics,* [Online Presentation] Available on: https://www.nwasoft.com/resources/webinars/right-data-right-place-right-time

[66] S. Schneider, "Middlware: The Last Roadblock to Distributed Systems Development," *Embedded Computing Design Resource Guide,* 2006

[67] X. Chen, L. Rao, X. Liu, H. Li and X. Wang, "Right time in right place: Taming workload balancing oscillations in internet data center cost management," *International Green Computing Conference*, Dallas, TX, pp. 1-10, 2014

[68] M. Mazouzi, S. Hasnaoui and M. Abid, "Challenges and solutions in configuring, rapid developing and deploying of a QoS-enabled component middleware," *2008 3rd International Design and Test Workshop*, Monastir, pp. 221-224, 2008

[69] Infineon, "Driving the Future of Automotive Electronics Automotive Application Guide," *Infineon Automotive,* 2017

[70] M. Fowler, "UML Distilled: A Brief Guide to the Standard Object Modeling Language," *Addison-Wesley Professional* 3rd Edition, 2003

[71] K. Smith and M. Ollerton, “Generic Vehicle Architecture – DDS at the Core.” [Online Presentation], Available: https://www.slideshare.net/RealTimeInnovations/generic-vehicle-architecture-dds-at-the-core.

[72] International Organisation for Standardisation “IEEE Guide Adoption of ISO/IEC 24748-1:2010 Systems and Software Engineering Life Cycle Management Part 1: Guide for Life Cycle Management,” *IEEE Std 24748-1-2011*, pp. 1–96, 2011.

[73] Think Defence, “Generic Vehicle Architecture,” [Web Based Open Access], 2011, Available on: https://www.thinkdefence.co.uk/from-scimitar-to-fres-to-ajax/generic-vehicle-architecture

[74] UK MOD, “Generic Vehicle Architecture (GVA),” *Ministry of Defence*, Defence Standard 23-09 *Issue 1*, no. 1, pp. 1–54, 2010.

[75] IEEE, “IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations,” 2009.

[76] M. S. Committee, “IEEE Standard for Communicating Among Processors and Peripherals Using Shared Memory (Direct Memory Access — DMA),” 1993.

[77] H. Ri, “IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations,” *PE - IEEE Power and Energy Society*, 2017

[78] Marine Transportation Committee of the IEEE Industry Applications Society, “IEEE Recommended Practice for Electric Installations on Shipboard,” *Electronics*, vol. 2001, 2001.

[79] IEEE Instrumentation and Measurement Society, “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats,” *IEEE Std 1451.5-2007*, pp. C1-236, 2007.

[80] “IEEE Application Guide for Distributed Digital Control and Monitoring for Power Plants,” in *IEEE Std 1046-1991,* 1991

[81] “IEEE Trial-Use Standard for a Broad Based Environment for Test (ABBET), Overview and Architecture,” *IEEE Std 1226-1998*, 1999.

[82] “IEEE Standard for Test Equipment Description Language (TEDL),” *SASB/SCCC20 - Test and Diagnosis for Electronic Systems*, 1997.

[83] “IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Network Capable Application Processor (NCAP) Information Model,” *IEEE Std 1451.1-1999*, p. 341, 2000.

[84] “IEEE Standard Glossary of Software Engineering Terminology,” *Office*, vol. 121990, no. 1, p. 1, 1990.

[85] P. Fournier, “Assessing the effectiveness of Defensive Aid Suite technology using a field trial and modelling and simulation,” *Defence R&D Canada-Valcartier 2459 Pie-XI*, 2002.

[86] SAAB, “IDAS/CIDAS range of Integrated Defensive Aids Suites,” *Business Area Electronic Defence Systems*, White Paper, 2015

[87] P. M. Zanker, “Integration of Defensive Aids,” *Advances in Vehicle Systems Concept and Integration,* pp. 26–28, 2000.

[88] P. Fournier, “Assessing the effectiveness of Defensive Aid Suite technology using a field trial and modelling and simulation,” *Defence R&D Canada* - Valcartier, 2002.

[89] American National Standard Dictionary of Electromagnetic Compatibility (EMC) including Electromagnetic Environmental Effects (E3) - Redline," in *ANSI C63.14-2014 (Revision of ANSI C63.14-2009) - Redline* , vol., no., pp.1-152, Dec. 5 2014

[90] SAAB, “ SAAB Grintek Defence”, *SAAB Grintek Defence Catalogue Electronic Warfare Solutions*, 2016

[91] K. Cole, “Introduction to Radar Warning Receivers,” *Robins AFB*, 2009.

[92] S. Prasad and D. J. Thuente, "Jamming attacks in 802.11g — A cognitive radio based approach," *2011 - MILCOM 2011 Military Communications Conference*, Baltimore, MD, pp. 1219-1224, 2011

[93] J. L. Rapanotti, “Developing Soft-Kill Capability for Light Armoured Vehicles Through Battlefield Simulations,” *Defence R&D Canada – Valcartier TM 2003-276, pp. 9*, 2007.

[94] K. Smit, A. Lee and M. Burridge, “Infrared and Visual Smoke Countermeasures for Army,” *Defence Science and Technology Group,* 2008.

[95] Nick O’Leary, “Node Red,” *QCon*, 2014. [Online]. Available: https://www.infoq.com/presentations/ibm-node-red.

[96] “American National Standard Dictionary of Electromagnetic Compatibility (EMC) including Electromagnetic Environmental Effects (E3) - Redline,” *ANSI C63.14-2014 (Revision ANSI C63.14-2009) - Redline*, vol. 2014, pp. 1–152, 2014.

[97] P. Syverson, "Onion routing for resistance to traffic analysis," *Proceedings DARPA Information Survivability Conference and Exposition*, pp. 108-110 vol.2, 2003

[98] Jane's Defence Industry and Markets Intelligence Centre, “Active protection: US and UK vehicle defence projects kick off,” HIS Markit, 2001.

[99] J. Rapanotti, A. Demontigny, A. Cantin, and M. Palmarini, “Developing Vehicle Survivability on a Virtual Battlefield Survivability DAS layer,” *Defence R&D Canada – Valcartier,* pp. 12–14, 2001.

[100] F. C. Gentner, P. S. Best and P. H. Cunningham, "Measure of effectiveness (MOE) taxonomy for assessing human performance in aeronautical systems," *Proceedings of*

the IEEE 1997 National Aerospace and Electronics Conference. NAECON 1997, Dayton, OH, 1997, pp. 29-36 vol.1.

[101] A. Standards, C. Committee, E. Compatibility, and N. Standards, “American National Standard Dictionary of Electromagnetic Compatibility (EMC) including Electromagnetic Environmental Effects (E3) - Redline,” *ANSI C63.14-2014 (Revision ANSI C63.14-2009) - Redline*, vol. 2014, pp. 1–152, 2014.

[102] J. L. Rapanotti, “Vehicle DAS considerations for the Iron Gorget threats,” *Defence R&D Canada – Valcartier,* 2007.

[103] B. Bosanquet, J. Dewey, W. James, G. E. Moore, and C. S. Peirce, “Cambridge Dictionary,” *Philosophy*, pp. 138–138, 1994.

[104] “IEEE Std 1209-1992: IEEE Recommended Practice for the Evaluation and Selection of CASE Tools,” pp. 1–31, 1993.

[105] A. Bloch, “Murphy's Law: The 26th Anniversary Edition” *Tarcher Perigee; Subsequent edition,* 2012.

[106] “NATO International Standardisation”, *Defence Standardisation Program Journal*, 2004 [Online]. Available: http://www.nato.int/cps/en/natohq/topics_69269.htm.

[107] F. Redmill, “An introduction to the safety standard IEC 61508,” *Hazard Prevision*, Vol. 35, no. 1, pp. 20–25, 1999.