CHAIR OF PICO AND NANO SATELLITES, AND SATELLITE CONSTELLATIONS

TECHNICAL UNIVERSITY OF MUNICH

Master Thesis

A Systems Engineering Approach to the Development of a Virtual, Networked Laboratory Infrastructure for Full Electric Aircraft Power Train Research

Batuhan Kartopu

CHAIR OF PICO AND NANO SATELLITES, AND SATELLITE CONSTELLATIONS

TECHNICAL UNIVERSITY OF MUNICH

Master Thesis

A Systems Engineering Approach to the Development of a Virtual, Networked Laboratory Infrastructure for Full Electric Aircraft Power Train Research

Ein Systems Engineering Ansatz für die Entwicklung einer virtuellen, vernetzten Laborinfrastruktur für die Forschung an elektrischen Flugzeugantrieben

Author:	Batuhan Kartopu
Registration number:	03681862
Supervisor:	Prof. Dr. Alessandro Golkar
Advisor:	Dr. Marcel Stößel & M. Sc Jaspar Sindermann
Submission Date:	31.05.2023

Acknowledgments

I would like to express my deepest appreciation to my supervisor Dr. Marcel Stößel for his great supervision of my thesis as well as his trust in me to introduce the systems engineering approach to a great project.

I would also like to give special thanks to my other supervisors Prof. Dr. Alessandro Golkar and Jaspar Sindermann for their continuous support and knowledge. They made it possible to learn a lot of new things during the process.

In addition, I would like to thank Philipp Chrysalidis for taking time and sharing his knowledge based on a similar project.

Finally, I would like to thank my colleague Clemente Juan for his great work as part of the systems engineering team. As a great team member, he supported this work in every possible way and helped me to broaden my vision.

Abstract

This work aims to introduce the systems engineering approach to the development of a virtual laboratory network. A systems engineering team has been built and the process started with a stakeholder analysis. A stakeholder model was created to visualize the stakeholder structure of the project. After understanding the relevance of each stakeholder, the systems engineering team proceeded with stakeholder interviews to understand the needs, goals, and objectives of the stakeholders and derive the technical requirements of the virtual laboratory network. Using these requirements, ideas were developed for different system architecture options, these system designs were modeled, and the options were analyzed. The focus lied on the strengths and weaknesses of centralized and decentralized communication architectures as well as the two main transport layer communication protocols. The iteration of possible system designs led to the final approach, which combines the benefits of different philosophies. This approach uses two different communication layers for different purposes. A "Data storage and control layer" allows a centralized control of the test and reliable data storage of the whole experiment with the underlying transport layer protocol Transmission Control Protocol (TCP). Another "Real-Time Application Layer" uses User Datagram Protocol (UDP) as the underlying communication protocol and allows the systems to communicate peer-to-peer with real-time capable network characteristics. The increased efficiency of the development process with the systems engineering approach is also emphasized.

Kurzfassung

Ziel dieser Arbeit ist die Einführung des Systems-Engineering-Ansatzes für die Entwicklung eines virtuellen Labornetzes. Es wurde ein Systems-Engineering-Team gebildet und der Prozess begann mit einer Stakeholder-Analyse. Es wurde ein Stakeholder-Modell erstellt, um die Stakeholder-Struktur des Projekts zu visualisieren. Nachdem die Relevanz der einzelnen Stakeholder geklärt war, führte das Systems-Engineering-Team Stakeholder-Interviews durch, um die Bedürfnisse, Ziele und Vorgaben der Stakeholder zu verstehen und die technischen Anforderungen an das virtuelle Labornetzwerk abzuleiten. Anhand dieser Anforderungen wurden Ideen für verschiedene Optionen der Systemarchitektur entwickelt, diese Systemdesigns wurden modelliert und die Optionen wurden analysiert. Der Schwerpunkt lag dabei auf den Stärken und Schwächen zentraler und dezentraler Kommunikationsarchitekturen sowie der beiden wichtigsten Transportschicht-Kommunikationsprotokolle. Die Iteration möglicher Systementwürfe führte zu dem endgültigen Ansatz, der die Vorteile der verschiedenen Philosophien kombiniert. Bei diesem Ansatz werden zwei verschiedene Kommunikationsschichten für unterschiedliche Zwecke verwendet. Eine "Datenspeicher- und Steuerungsschichtërmöglicht eine zentralisierte Steuerung des Tests und eine zuverlässige Datenspeicherung des gesamten Experiments mit dem zugrunde liegenden Transportschichtprotokoll Transmission Control Protocol (TCP). Ein weiterer "Real-Time Application Layer"verwendet User Datagram Protocol (UDP) als zugrunde liegendes Kommunikationsprotokoll und ermöglicht den Systemen eine Peer-to-Peer-Kommunikation mit echtzeitfähigen Netzwerkeigenschaften. Die erhöhte Effizienz des Entwicklungsprozesses mit dem Systems-Engineering-Ansatz wird ebenfalls hervorgehoben.

Contents

List of Figures

List of Tables

Acronyms

API Application Programming Interface. 10, 32

BDD Block Definition Diagram. 8, 24–26, 31, 33

CANAerospace Controller Area Network Aerospace. 33

ELAPSED Electric Aircraft Propulsion - Safe, Efficient, Digitally Linked. viii, 1, 2, 13, 16, 18, 19, 28, 29, 41, 42

HiL Hardware in the Loop. 22

IBD Internal Block Diagram. 8, 38

INCOSE International Council on Systems Engineering. 5

IP Internet Protocol. 10

ISO International Organization for Standardization. 11

IT Information Technology. 29

LAN Local Area Network. 10, 11, 26, 33

MBSE Model Based Systems Engineering. 2, 4, 7, 8, 16, 21, 29–31, 37, 38

MIT Massachusetts Institute of Technology. 4

NASA National Aeronautics and Space Administration. 5, 6, 22

OSI Open System Interconnection. 11

PiL Pilot in the Loop. 22

PLM Product Life-Cycle Management. 15

SiL Software in the Loop. 22

SoS System of Systems. 2, 4, 8–10, 28, 29

SysML System Modeling Language. viii, 7, 8, 18, 21, 24, 38

TBD To be Determined. 22, 23

TCP Transmission Control Protocol. iii, iv, 10, 12, 33, 34, 36

UDP User Datagram Protocol. iii, iv, 12, 33, 34, 36

UML Unified Modeling Language. viii, 7

UniBwM University of Bundeswehr Munich. 1, 8

VLN Virtual Laboratory Network. 2

WAN Wide Area Network. 10, 11, 26, 33

WET Water Enhanced Turbofan. 1

1. Introduction

One of the biggest challenges in today's engineering world is to reduce the utilization of fossil fuels in various industries. Fossil energy sources are limited and scarce by nature and their byproducts are accelerating climate change by emitting high amounts of carbon consisting products into the atmosphere (Wuebbles & Jain 2001). Governments around the world are tending to limit the carbon footprint of different industries with individual regulations as one can see in the Paris Agreement (2015). Aerospace is one of the industries affected by these regulations and limitations.

The aerospace industry came up with multiple innovative approaches to combat above-mentioned problem. Focusing on improving the emission characteristics of the conventional aero-engines has led to concepts like Water Enhanced Turbofan (WET) engines (Pouzolz et al. 2021). Focusing on a completely different approach with zero carbon emission during flight leads to full electric power train concepts. The electrical energy needed for the rotation of the propeller can be stored in batteries or hydrogen fuel cells.

The University of Bundeswehr Munich (UniBwM) is one of the universities, which is hosting research and development projects about these new approaches to achieve sustainable aerospace. The project Electric Aircraft Propulsion - Safe, Efficient, Digitally Linked (ELAPSED) is one of these projects and divides into three sub-projects. Sub-project 1 aims to create a virtual laboratory network to be able to conduct tests with different engine components without being in the final assembly phase. This approach also eliminates the necessity of the development and testing of all components in the same location. Sub-project 2 concentrates on the development of a multi-phase electrical engine. Sub-project 3 focuses on new highly efficient propeller technologies. This master thesis is a part of sub-project 1 and depicts a systems engineering approach for the efficient development of above-mentioned virtual laboratory network.

Systems engineering is a field of engineering and entails various methodologies and approaches for the design, technical management, and operation management of interdisciplinary systems. These methods simplify and standardize the management of interfaces and ensures the fulfillment of system requirements (Hirshorn 2017). Methods of systems engineering are applied during the development of the virtual laboratory network as the compilation of various laboratories build a multidisciplinary complex system.

Maier (1998) defines the concept of a "System of Systems (SoS)" as an assembly of components that are themselves significantly complex enough that they may be regarded as systems and that are assembled into a larger system. As the virtual laboratory arises from multiple laboratories and subsystems, which are also functional without coming together to build an assembly, the virtual laboratory is considered a system of systems. Since these types of systems require high interoperability, this work will demonstrate the edges of systems engineering for fulfilling interoperability as well as synchronization requirements.

By designing the system architecture of a system of systems, there are some main considerations and decisions for the systems engineer to make. This master thesis centers on the choice between a centralized and decentralized architecture. Another important focal point is to find the correct communication protocol for each communication happening between the subsystems. The advantages and disadvantages of different communication protocols as well as system architectures will be analyzed and the decision will be reasoned. Model Based Systems Engineering (MBSE) plays a key role in this process and the contributions of MBSE to interface management will also be analyzed.

Chapter 2 presents the state of the art of fundamentals of this work. The concept of systems engineering is summarized with historical development and used methods. The significance of systems engineering and the possible improvements this approach brings are emphasized. Another focus point of the chapter is the concept of an SoS. Studies about system architecture concepts for such complex systems are analyzed and the role of communication is discussed. Different communication layers and protocols are also introduced in this chapter.

Chapter 3 starts with the details of the project ELAPSED. The motivations for the whole project and specifically sub-project 1 are mentioned as it is the focus of this work. Some problems in the development phase of products that can be solved with our Virtual Laboratory Network (VLN) concept are addressed. Also, the concept of digital twins and their significance for engineering applications are stressed. After that, the research questions that this work aims to answer are introduced. These research questions focus on the advantages of the systems engineering approach as well as the high interoperability of an SoS.

The methodology used throughout this work is introduced in Chapter 4. The systems engineering approach is detailed step by step. The process starts with the stakeholder analysis and a stakeholder model. With the help of the stakeholder interviews, the technical requirements are derived and the requirement definition process is finalized with an iterative approach. With the help of these requirements, the systems engineer focuses on different system design approaches and models different system architecture options.

Chapter 5 analyzes our work and compares the systems engineering approach with the traditional development process. Different system design perspectives and communication protocols are presented. An argumentation for the final system architecture is also provided.

The whole work is summarized in Chapter 6 and the conclusion is presented. Possible next steps are also presented in the last chapter and an outlook for the future is given.

2. State of Art

To understand the current state of the research, technologies, and approaches to the topic of interest, the existing literature is reviewed in this chapter. The reader will understand what Systems Engineering is, why it is needed in engineering projects, how it might be applied as well as methods like MBSE. The second part of this chapter will focus on the concept of SoS. The reader will get an idea about the concept, different system architectures, and state of art communication protocols.

2.1. Systems Engineering

Systems engineering is an emerging field that gains importance in the engineering world. For the reader to be familiar with the term systems engineering, this section will start with the history and the reader will understand the needs that created this field.

2.1.1. History

Schlager (1956) assumes, that the term systems engineering was first used in The Bell Laboratories in the 1940s. The need in this specific area is explained by the complexity of a national telephone network at a time when other industries were not dealing with that complexity level. The increasing complexity of systems created a need for an interdisciplinary approach to engineering projects. Realization of the fact, that problem-free operation of individual components does not mean a problem-free operation of the whole system, made systems engineering relevant (Schlager 1956).

Hall (1962) traces the first academic teachings of systems engineering back to the lectures of G. W. Gilman at the Massachusetts Institute of Technology (MIT) (Buede & Miller 2016). The adoption of this holistic approach and top-down way of thinking by other fields accelerated the development of the systems engineering field. Figure 2.1 shows the acceleration of scientific research and publications with a histogram. The horizontal axis is divided into 3 time intervals, which represent the introduction, exploration, and revolution of systems engineering (Hossain et al. 2020). The revolution interval begins with the founding of the non-profit organization International Council on Systems Engineering (INCOSE). As one can see in the histogram, publications with a systems engineering background increased drastically in recent years. Technological advances allow engineers to design more complex and interdisciplinary systems. This trend leads to the requirement of interoperability and more efficient interface management. Hence the demand for systems engineering experiences a significant increase.

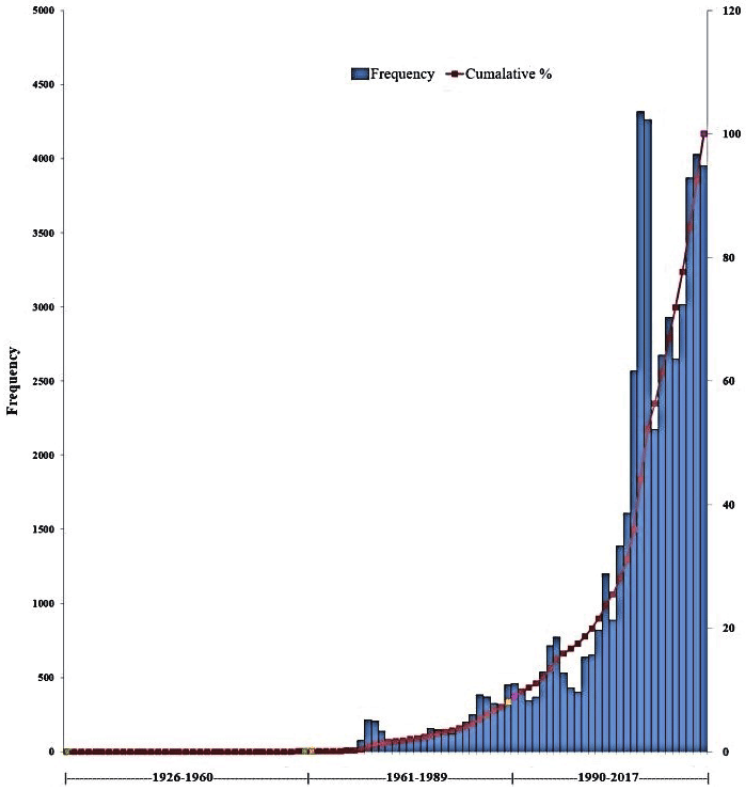

Figure 2.1.: Frequency of publications with regards to systems engineering (Hossain et al. 2020)

2.1.2. Methods

In this subsection, the typical workflow and methods of the systems engineering approach are introduced. The details of these methods, steps, and how we implemented these in our project will be represented in Chapter 4. As it is aimed to apply the systems engineering practice of the National Aeronautics and Space Administration (NASA) in this work, the focus will lie on the NASA Systems Engineering Handbook of Hirshorn (2017). This book divides the system design process into four main steps:

1. Stakeholder Expectations Definition
2. Technical Requirements Definition
3. Logical Decomposition
4. Design Solution Definition

The stakeholder expectations definition process helps the systems engineers to identify the stakeholders and understand what they want to achieve with the product. The stakeholders are typically customers as well as the design and development team of the end product. These stakeholders are interviewed by the systems engineer and asked questions intending to recognize their needs, goals, and objectives of them (Hirshorn 2017).

Trainor & Parnell (2007) offers 3 different ways to execute a stakeholder analysis. These are (1) stakeholder interviews, (2) focus groups, and (3) surveys. Stakeholder interviews come forward if the systems engineer wants to get information from each stakeholder without the influence of other stakeholders. However, conducting separate interviews with each stakeholder is the most time-consuming method.

Focus groups may offer different kinds of information to the systems engineers as the format allows discussion between the stakeholders. These discussions may help the systems engineer to identify significant problems, nevertheless domination of some participants besides conflict of interest between some stakeholders may prevent achieving useful details and facts (Trainor & Parnell 2007).

Surveys are especially useful for collecting quantitative data from stakeholders. Although it is a fast way to collect data and information, the systems engineer is not available for questions and clarification. The questions have to be formulated very clearly and the systems engineer has to avoid preparing surveys that are longer than necessary (Trainor & Parnell 2007).

After analyzing the results of the stakeholder analysis, the systems engineer derives the technical requirements of the system. These requirements have to be worked over with the technical team so that the requirements are validated. The validated requirements have to be formulated unambiguously (Hirshorn 2017).

After the analysis of these requirements, the systems engineer develops design solution, which would fulfill these requirements. This process includes developing different ideas for possible design solutions and analyzing these solutions to identify the most feasible design solution. The systems engineer should verify that the final design solution fulfills the defined requirements (Hirshorn 2017).

2.1.3. Model Based Systems Engineering

As mentioned in previous sections, the systems engineering field aims to tackle problems arising with complexity and improve interoperability of systems and subsystems so they can perform the required function. Digitization and emergence of computer-aided design tools also affected the systems engineering tasks, leading to a transition from the document-based approach to the model-based approach. As the systems engineer deals with the stakeholder needs, requirements, system design, and test-related information, the document based approach collects all relevant information in many different documents. This aspect reduces traceability and consistency between above-mentioned aspects. In contrast, the model-based approach offers harmony and high traceability (Friedenthal et al. 2014).

There are many modeling languages available for MBSE applications however this work will focus on System Modeling Language (SysML). SysML is an extension of Unified Modeling Language (UML) for systems engineering applications and used widely in the aerospace industry (Batarseh & McGinnis 2012). An overview of UML, as well as SysML diagram types can be found in Figure 2.2 (*SysML FAQ: What are the SysML diagram types?* n.d.).

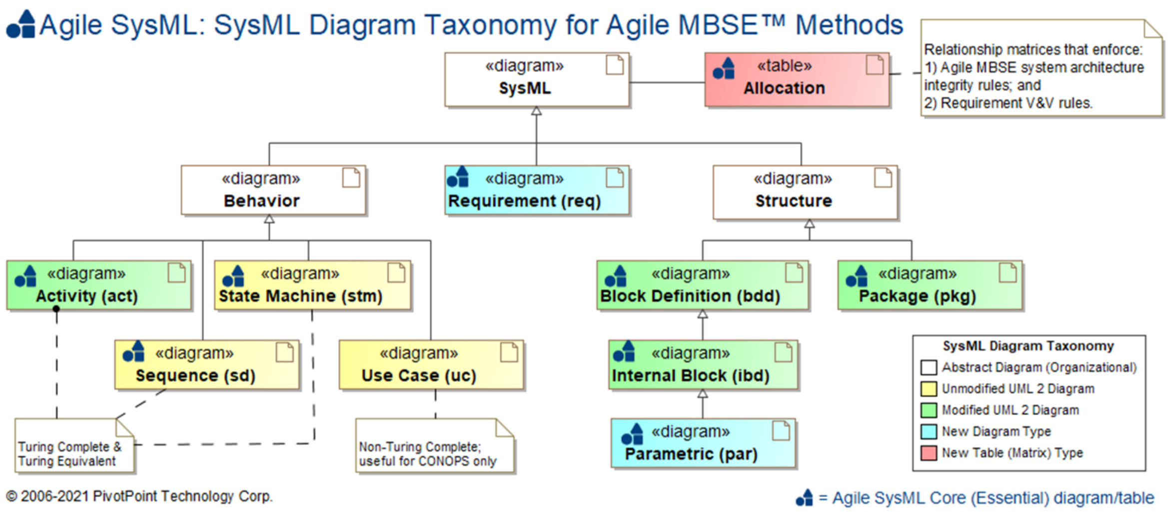

Figure 2.2.: Overview of UML and SysML diagram types (*SysML FAQ: What are the SysML diagram types?* n.d.)

SysML offers different types of diagrams to satisfy the needs of systems engineers. The four main types of diagrams are listed here:

- Structure Diagrams
- Parametric Diagram
- Behaviour Diagrams
- Requirements Diagram

The system structure can be illustrated with a Block Definition Diagram (BDD) and an Internal Block Diagram (IBD). The hierarchical structure of systems or subsystems is represented with BDDs. The internal structure of these systems can be described further with IBDs. Parametric diagrams show relevant mathematical rules and constraints for the parameters of system blocks. Use case, activity, state machine, and sequence diagrams are behavior diagrams. Use case diagrams deliver high-level functionalities of systems. Activity diagrams represent the control mechanisms and flow of data. The collaboration of systems can be represented with sequence diagrams. Different states of a system and changes of states are shown with state machine diagrams (Hause et al. 2006).

The requirement diagrams are not only documentations of requirements but they serve as a foundation for model-based requirements management applications. With these models, the systems engineer can link systems blocks with the requirements they are fulfilling. Test cases used for verification of the requirements can also be traced with these diagrams. The systems engineer can also represent the hierarchy of the requirements and possible derivations in these requirement diagrams (Hause et al. 2006).

There are many SysML-based tools available for MBSE applications. As the UniBwM offers Astah SysML Version 8.0.0 for free within the software center of the university, Astah is used for our MBSE work. A use case diagram was used for our stakeholder overview and several BDDs were used for different system architecture concepts. These diagrams will be presented in Chapter 4 and possible extensions of this work with other SysML diagram types will be mentioned in Chapter 7.

2.2. Concept: System of Systems

In this section, the concept of a System of Systems (SoS) is analyzed. The focus lies on understanding the difference between a system and SoS. What constitutes a system? What is the difference between the subsystems of a system and systems of an SoS? Why is it important to understand this difference and how does it affect the design and implementation process? The literature is reviewed to answer these questions in this section.

As mentioned in section 2.1.1, the complexity of systems is rising with growing technological opportunities. This leads to a high number of systems assembled with components that are complex systems themselves. An assembly of different components is called a system when it produces a function that is not performed by the components individually. Maier (1998) introduces two criteria to call a system a system of systems:

1. The components can produce functions on their own and may continue to perform these functions if they are disassembled from the joint system.
2. The components are not managed globally by the joint system but they are managed to fulfill their own function.

Maier (1998) also mentions that some systems may fulfill these criteria although they are not accepted as an SoS. The term is broadly used for distinguishing distributed large systems which also fulfill above mentioned criteria from smaller less complex systems. Eisner et al. (1993) and Shenhar (1995) use the term SoS also to describe systems that consists of complex systems that are geographically distributed.

Another analysis of the difference between systems and SoS is done by Boardman & Sauser (2006). Some characteristics of systems are chosen and analyzed to identify an SoS. These characteristics are:

- Autonomy
- Belonging
- Connectivity
- Diversity
- Emergence

Subsystems of a system do not work autonomously, whereas the systems in an SoS can work autonomously. Subsystems belong to the system and they do not have individual functions without the system. Systems in an SoS do not necessarily belong to the whole system but they can achieve additional functions working together. A network of systems provides high connectivity in an SoS, whilst a system requires high connectivity of smaller parts but low connectivity of major subsystems. Diversity is high in an SoS because it consists of complex systems of different types for the achievement of a common goal. As an SoS needs to reach a broad functionality, the emergence aspect must be enriched. For a simpler system, being foreseeable comes forward (Boardman & Sauser 2006).

Designing a system architecture for such complex systems of systems requires new approaches. New fundamentals for system architecture designs are presented in the next subsection.

2.2.1. System Architecture

Maier (1998) makes a significant statement about the architecture of an SoS. Accordingly, the architecture of an SoS is not the physical architecture of the systems but the communication between the systems. This aspect of an SoS is very crucial and will be relevant to the system architecture models of this work. Systems that are located in the same laboratory can maintain the communication over physical busses whereas systems in different location communicate over a Local Area Network (LAN)/Wide Area Network (WAN) connection. This work concentrates on the LAN/WAN connection between the systems from different locations. The suitable hardware for the physical communication is decided by the sub-teams for their systems.

When different system architecture options are analyzed and communication is considered as the foundation of the architecture, an important question arises: Is it better to use a centralized or a decentralized architecture? As the communication between the laboratories is going to be maintained over the internet, we can analyze centralized and decentralized architecture options using Transmission Control Protocol (TCP)/Internet Protocol (IP). Maier (1998) mentions the package-based nature of TCP/IP, which makes different data packages within a piece of information independent from each other. All nodes can decide the routing route of these packages. This feature of TCP/IP allows a decentralized system architecture with independent nodes possible. However centralized architectures also offer many advantages.

A centralized approach includes a central node that manages the communication of the system. This approach brings a small set of Application Programming Interface (API)s so that they can acquire and provide data. All of the entities can use these APIs to make further developments. This approach also helps with the interoperability aspect as all of the systems can use just one adapter to interact with the central API. This aspect also reduces complexity. A centralized system controls access homogeneously, whereas a decentralized architecture allows heterogeneous access control. The data government of a centralized approach can be simpler but less flexible than the decentralized approach (Roman et al. 2013). It is also mentioned, that these two approaches can be combined.

2.2.2. Communication Protocols

As it is aimed to connect laboratories virtually and conduct tests simultaneously, one has to understand how machines communicate with each other over the LAN and WAN. International Organization for Standardization (ISO) introduced the Open System Interconnection (OSI) model for the representation of different layers of communication. Each layer has its specific set of function so that the services provided by the lower layer is used and enhanced. These layers decompose the whole communication network into smaller parts for efficient management and reduced complexity. They also enable standardization of each level of communication which prevents chaos for developers and users (Kumar et al. 2014). The seven layers are listed here:

1. Physical Layer
2. Data Link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer

The physical layer represents the hardware used for maintaining the connection, therefore physical and electrical attributes of the communication are defined in this layer. The data link layer creates a direct link between the communication partners. If the communication is not direct and multiple nodes are involved, higher layers have to provide end-to-end communication. The network layer is responsible for the organization of the data. The destination address and path is determined in this layer. End-to-end communication between the applications on different devices is provided in the transport layer. Transport-level protocols establish the connection, keep it running, and terminate the connection in the end. It is also the layer, that application developers and programmers interact with. Communication between two partners happens as long as a "session" is active. The activation, as well as deactivation of the session and its properties, are managed in the session layer. The presentation layer transforms the data from the network to the application layer, so that this collection of data may be shown to the end user. Finally, the user can see and control the communication in the application layer (Kumar et al. 2014).

As the transport layer is the layer of end-to-end communication and is decisive for throughput, delay, real-time capability, and reliability, the focus will lie more on the transport layer. The two main protocols used in the transport layer are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP provides a reliable data transfer, however, the increase in reliability is achieved with higher delays. UDP offers less delay due to the lack of flow control mechanisms which ensure high reliability for TCP (Kumar & Rai 2012).

TCP is considered a "connection-oriented" protocol. The application layer gets a reliable byte stream from the transport layer with the usage of TCP. The congestion control mechanism of TCP reduces the transmission to avoid overloading the network (Rahmani et al. 2008). The reliability of TCP is achieved with a mechanism based on acknowledgment. The source keeps the data that has been sent in a "sliding window" and does not remove the data immediately after sending it. The receiver gets the data and sends an acknowledgment packet to the source with the received data. The source removes the data after receiving the acknowledgment packet. This mechanism ensures that all the information is received by the receiver in the correct order and that the receiver has enough space for new information to avoid a congestion (Georg 2006).

UDP on the other hand does not guarantee delivery of messages or messages being received in the same order as the source sent them. It is not connection-oriented, the source publishes the information and has no mechanism to make sure the information is received. Therefore there is also no congestion control mechanism as TCP has. However, the lack of a control mechanism makes lower delays possible and it is used in time-sensitive applications (Kumar & Rai 2012).

A similar distributed testing architecture was researched by Martinen et al. (2017). The communication layer ensures a standardized communication of different modules and consists of two different standard protocols. The real-time communication is provided by a UDP-based real-time bus whereas the control commands use a TCP-based standard protocol.

3. Project Goal and Research Questions

This chapter is an overview of the project goal and research questions, which are aimed to be answered with this work. These research questions also represent the gap between what has already been researched and what we aim to understand and analyze. The first section will show the reader the details and motivation of the project. The first section will concentrate on sub-project 1 as this work is a part of sub-project 1. The second section will concentrate on the research questions and their relevance. The first two research questions intellectualize the benefits of systems engineering methods and approaches to the engineering project. Remaining two research questions focus on the effects of system architecture and communication protocols on the interoperability of a system of systems.

3.1. Project ELAPSED - Motivation

Due to the reasons mentioned in the introduction, the aerospace industry tries to revolutionize the basic principles of aircraft engines and reduce carbon emissions. Consequently, aircraft engine types like electrical/hybrid engines with low or zero carbon emission characteristics become the focal point of research. This trend is also researched by Roland Berger partner Thomson (2020) and depicted in the Figure 3.1. Project ELAPSED also aims to investigate the options in this field and develop electric-driven power trains within the framework of sub-project 2. It is aimed to develop a power train with 80 kW power output and a multi-phase electric motor. This motor can be powered by conventional batteries as well as a fuel cell.

3.1.1. Sub-Project 1 - Virtual Laboratory Network

Another important trend that is researched by project ELAPSED is a virtual networking of laboratories and test benches. The laboratories used for the testing and development of the electrical engine for the sub-project 2 will be connected virtually to conduct tests with all the components of the engine.

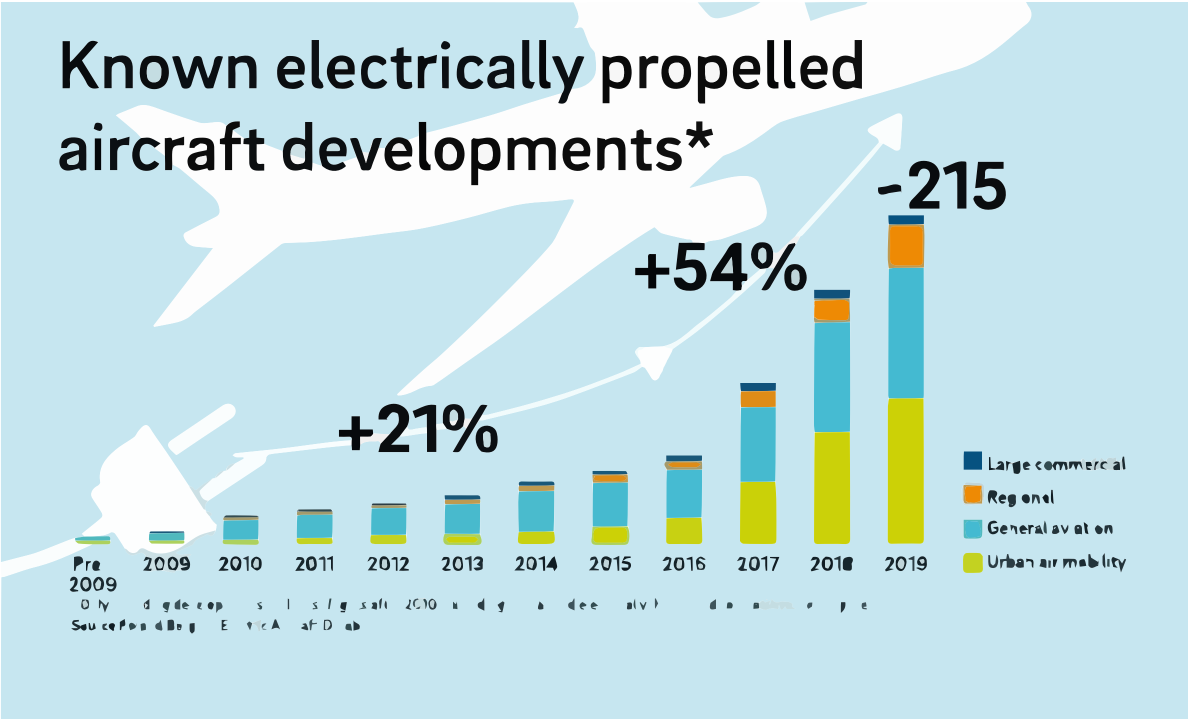

Figure 3.1.: Trend of known electrically propelled aircraft developments (Thomson 2020)

Why is connecting different laboratories and creating a virtual laboratory network important? An aircraft engine is a complex system with many complex components. These components are not designed and developed by the same engineers, hence the development team is a multi-disciplinary team and divides into sub-teams. Each team conducts tests in its own laboratory during various design phases and develops the components iteratively with the help of these tests. For the end product to fulfill its function, all of these components should not only deliver function on their own but also should be able to work together in harmony. The conservative approach to test the cooperation of these components is to develop each of these parts individually and integrate them in the late phases of the project in a new test environment designed for the whole system. This approach leads to two major problems:

1. The project team needs to design a laboratory for testing the whole system and move the components from their own laboratory to the engine test facility. Extra development efforts as well as logistics between testing facilities create extra costs and take extra time.

2. As the project team can not conduct tests with all the components before they finish the development and finish the integration, possible problems can not be identified until

the end of the development. This will lead to a necessity to modify the components after the end of their development phase, therefore to extra costs and extra time.

With the virtually connected laboratories approach, these problems cease to exist. The sub-teams can conduct tests with other components without having to move their components from their laboratories. These tests can also be conducted during development and they do not have to wait until the later stages of development and production.

3.1.2. Concept of Digital Twins

Another important goal of sub-project 1 is to develop digital twins of each component. Therefore it makes sense to start this subsection with the question: What constitutes a digital twin? A digital twin is fundamentally an extension of a model with data to perform realistic simulations and portray an accurate digital representation of a physical component (Tao et al. 2022). Static and theoretical models are used to simulate the behavior of systems in early development stages and thus help with optimization, verification, and validation activities. But today's technology allows to collect huge amounts of data during the run-time of the products and use these data to extend the simulation model. Hence the number of academic and industrial works about digital twins see an acceleration. (Liu et al. 2021)

Tchana de Tchana et al. (2019) specify the automation of the data flow as the distinctive feature of a digital twin. A conventional digital model features manual data flow from the physical entity to the digital entity and vice versa. A "digital shadow" is considered an intermediate step in the development of a digital twin. The data flow from the physical entity to the digital entity is automated and the data flow from the digital entity to the physical entity is still manual. In the next step, data flow in both directions is automated and the digital entity can be called a digital twin. This process is visualized in Figure 3.2.

Another benefit of a digital twin is that the representation of the used hardware continues to be accurate over time, even when the physical entity changes. A conventional model represents one state of the physical entity whereas a digital twin includes behavior changes over time. Therefore it makes even more sense to use a digital twin, if the product does not behave identically over time (Wright & Davidson 2020).

Considering the long life cycles of aerospace products compared to other industries, an efficient Product Life-Cycle Management (PLM) gains importance for aerospace products. A digital twin provides an iterative process, which provides feedback for the engineers throughout the entire life-cycle of the product (Li et al. 2022).

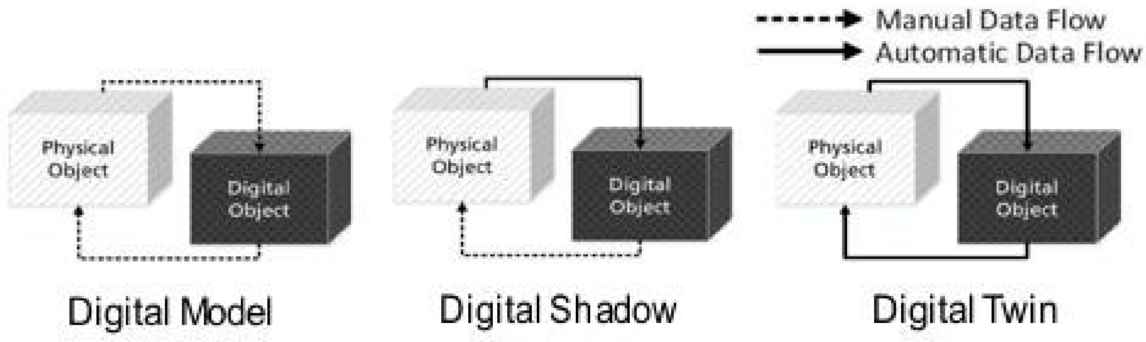

Figure 3.2.: Evolution of a digital twin (Tchana de Tchana et al. 2019)

3.2. Research Questions

The research questions are presented in this section.

3.2.1. How Does the Systems Engineering Approach Effect the Project Progress Rate?

The product development phase may look different even for identical products, depending on the development approach. The efficient approach depends on various criteria and desired system requirements. The project management team has to decide which approach to use after analyzing stakeholder needs and available resources. The project manager of ELAPSED Sub-Project 1 decided to deploy a systems engineering team for stakeholder analysis, requirements management, system architecture definition, and interface management purposes. With this research question, the edges of this approach compared to the traditional approach where the engineers of each subsystem have to complete these tasks will be analyzed.

3.2.2. What are the Benefits of Model Based Systems Engineering During System Architecture Studies?

Designing and combining complex systems bring up new challenges. Growing complexity and size make it harder to manage the interfaces between systems and subsystems. This research question will analyze how system architecture studies can benefit from MBSE.

3.2.3. How Does Centralized and Decentralized Architecture Concepts Effect the Interoperability in a System of Systems?

Using the model based approach also helps to visualize the system architecture and identify critical points of different approaches. One of the most significant decisions regarding the system architecture is to use a centralized or a decentralized architecture. With this research question, the benefits and drawbacks of both architectural philosophies as well as our decision process will be analyzed.

3.2.4. Which Transport Layer Communication Protocol Should be Used for the Virtual Laboratory Network?

As mentioned in section 2.2, the system architecture of a system of systems is defined by communication. The laboratory network consists of various systems. Each of these systems has its communication partners and each communication is happening with different frequencies and data sizes. With this research question, different communication protocols for these individual communications and the way we decide on the best option, using the systems engineering approach will be analyzed.

4. Methodology

The project manager decided that ELAPSED sub-project 1 can benefit from a systems engineering approach. This work has been initiated with this regard. The first line of work was to build a systems engineering team and define the goal of this team. This work should represent the introduction of systems engineering methods to the project and assist with the creation of a virtual laboratory network. It has been decided to use the methods mentioned in section 2.1.2 and manage the interfaces between different laboratories.

4.1. Stakeholder Analysis

The first work package for the systems engineering team was to identify the stakeholders involved in the project. For this purpose, a SysML stakeholder model was created and relevant stakeholders and their relations were represented. This analysis made it possible for the systems engineering team to understand the involvement level of each stakeholder and their roles for the project ELAPSED.

The stakeholder model has two different layers. The first layer represents every individual and institute involved in the project and gives an overview of the relevance of the project. It also makes the hierarchical structure of the project clear. This first layer can be noticed in Figure 4.1. A bigger version of the figure can be found in the Appendix A.

The second layer analyzes the project team structure and shows the technically involved stakeholders of the project. These stakeholders are the relevant stakeholders for the systems engineer to understand the needs, goals, and objectives of each subsystem and to derive technical requirements from them. As most of the subsystems also own their test benches, this layer also gives the systems engineer an idea of the test benches, which have to be connected virtually at the end of the project. This second layer is depicted in Figure 4.2. The bigger version of the figure can be found in Appendix A.

After the model was created and the relevant stakeholders of the project were understood, it has been decided to proceed with the analysis. As the aim was to introduce the concept of systems engineering to the project group individually and acknowledge the understanding of

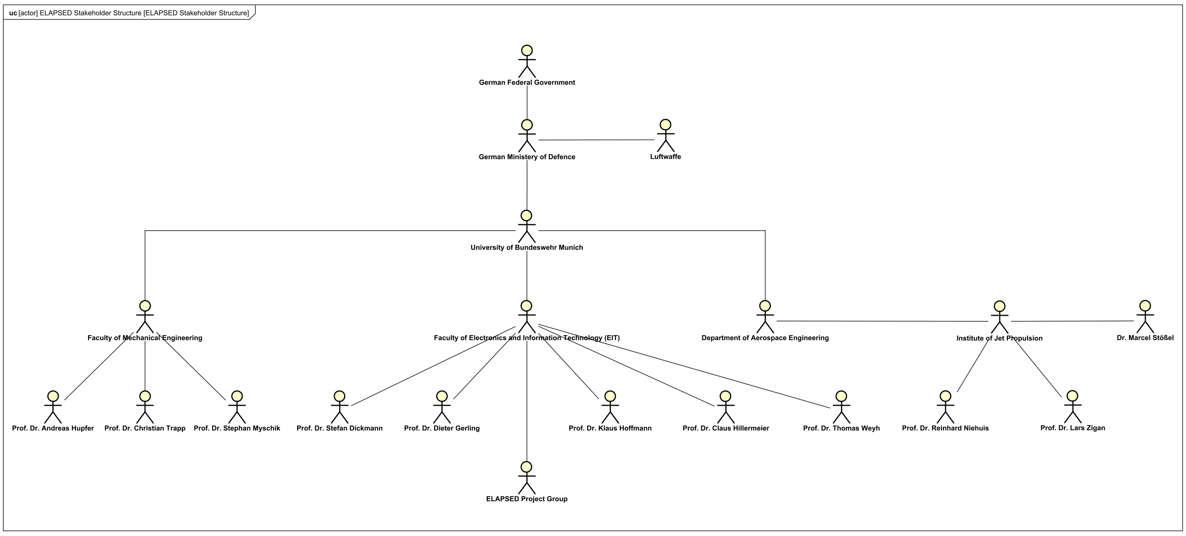

Figure 4.1.: Stakeholder Model of project ELAPSED - Layer 1

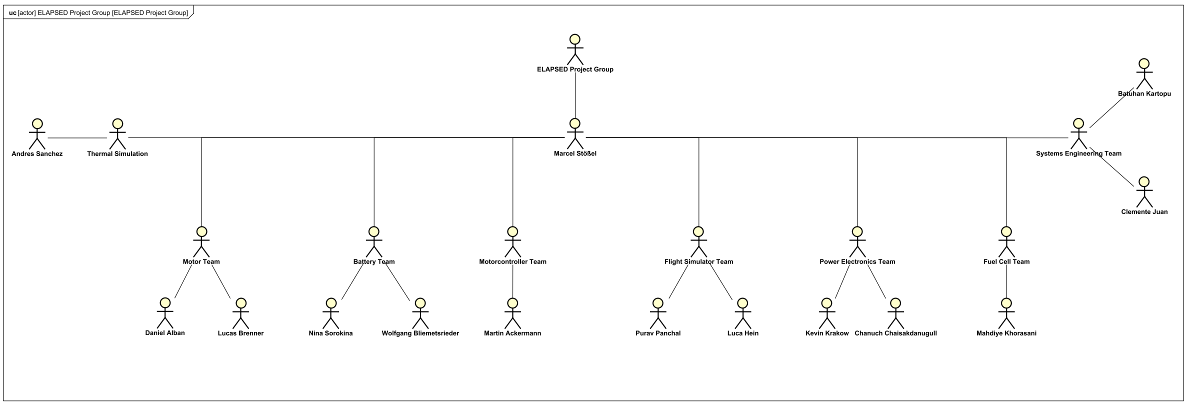

Figure 4.2.: Stakeholder Model of project ELAPSED - Layer 2

the project of each subsystem, it has been decided to analyze the stakeholder needs, goals, and objectives with stakeholder interviews. Although it is the most time-consuming method among the analysis methods introduced in section 2.1.2, the advantages of active face-to-face communication and avoidance of influence between subsystems weighted higher. The small size of the group also minimized the negative time-consuming effects of the stakeholder interviews method.

Conducting stakeholder interviews is an iterative process. The systems engineering team prepares the questions for the interviews and the first round enables the interpretation of relevant stakeholders' perspectives. The questions aim to extract the goals, precise stakeholder structure, current status, needs, objectives, and time plan of each subsystem. These questions can be found in Appendix B. The first question aims to derive the understanding of the stakeholders regarding the overall goal of the sub-project 1. The second question allows the systems engineering team to identify each actively involved stakeholder for the sub-project 1. Involved components of the virtual laboratory network can be identified with the third question. Following three questions focus on the development of digital twins. The systems engineering team aims to emphasize the goal of developing digital twins that are compatible with the testing environments with these questions. The formulation of these questions allows the systems engineering team to indicate the desired functionalities of the digital twins. Therefore these questions achieve a bidirectional information exchange between the systems engineering team and the stakeholders. The remaining questions provide information for technical requirements and organizational aspects of the project. The answers of each stakeholder were documented and shared with the project manager to keep the management updated about the project's progress and current status.

After the first round of stakeholder interviews, it has been concluded that the team is not familiar with the concept of digital twins. Digital twins play a key role to achieve virtual testing and eliminate the necessity of real components being able to test each time. The difference between digital twins and simulation models is explained in section 3.1.2. The systems engineering team made these differences clear and emphasized the importance of digital twins for the project. All of the important results of the stakeholder interviews are listed below:

- The stakeholders do not share the same understanding of the goal and the end product of sub-project 1. All of the teams are clearly more focused on the sub-project 2. Some of the team members are not even aware of the sub-project 1.
- All relevant stakeholders were made clear and have been added to the stakeholder model.
- The stakeholders are not familiar with creating and using digital twins. None of the teams has started to work for a digital twin and none of the teams were planning to extend the digital model to develop a digital twin of the physical entity.
- The stakeholders do not have a plan for managing the interfaces between different subsystems, they are focused on their own models and components.
- The stakeholders made their expectations of the virtual laboratory network and its

mode of operation clear.

- The stakeholders do not expect to end sub-project 1 and maintain the connection of the laboratories before the end of this master thesis.

These results allowed the systems engineering team to identify various false comprehensions of relevant stakeholders and communicate the desired information. All of the missing information was provided to the teams and a common understanding of the project goal was defined. The interviews made it possible for the teams to express their vision for the virtual laboratory network. Discussions of these ideas and deriving a common vision for the virtual laboratory network allowed the systems engineering team to derive the technical requirements of the system and move on with the technical requirements definition.

4.2. Technical Requirements Definition

Although it was intended to use MBSE for the whole systems engineering process, the SysML tool Astah was not available during the requirements definition phase due to bureaucratic reasons. Instead, Polarion was used, which was developed by Siemens for life-cycle management tasks including requirements management.

The systems engineering team analyzes the results of stakeholder interviews as well as the project proposal to define technical requirements. The project proposal provides the desired state of the end product whereas the stakeholder interviews outline the desired states of various development phases and subsystem-specific requirements.

After the analysis, it has been decided to define major groups of requirements for a sound operation of the virtual laboratory network and use these groups as sub-categories for the requirements. The following fields were chosen as critical:

- Connectivity and Communication Requirements
- Security Requirements
- Synchronization and Data Management Requirements
- Remote Access and Control Requirements
- Simulation Requirements
- Usability Requirements

- Scalability Requirements

Connectivity and communication requirements define the manner of communication regarding protocols, the maximum allowed latency, and real-time operation. Security requirements ensure protection against unauthorized access, role definition, and different ways of user access. Synchronization and data management requirements determine the synchronization rate of the test benches as well as allowed miss ticks, data storage, data handling capabilities, and hybrid operation with real components and digital twins. Remote access and control requirements secure access over the internet and set the interface and user count. The virtual laboratory network's ability to contain Hardware in the Loop (HiL), Software in the Loop (SiL), and Pilot in the Loop (PiL) elements to simulate the real-world scenarios are secured by the simulation requirements. Usability requirements ensure user-friendliness, sufficient documentation, and uncomplicated troubleshooting. Finally, a modular architecture philosophy is determined by the scalability requirements. A list of all these requirements can be found in Appendix C. Quantitative values like the latency of the communication, levels of user access, missed ticks, data storage period, user count were established within the requirements.

Besides the content of these requirements, the way of the formulation was also done according to NASA guidelines for "good requirements". It is important to note that the requirements are grammatically correct, unambiguous, measurable, verifiable, testable, positively stated "shall" statements.

4.3. Logical Decomposition

After the creation of the first draft of the technical requirements, it has been decided to create a team within the team and update each other weekly to proceed with the requirements and exchange ideas about possible system architectures. This team would concentrate on sub-project 1 and organize weekly meetings. As the simulation team members have an overview of all the subsystems and their models, it has been decided to build this team with the systems engineering team and the simulation team.

The first draft of the requirements is done by the systems engineering team and does not include quantitative values. Quantitative values were replaced with "To be Determined (TBD)". The above-mentioned team reviewed the requirements and analyzed meaningful values for the TBD values. These numbers are determined based on the success criteria of the project. Therefore the project proposal was reviewed by the team and possible options were compared with each other. The project management was also consulted. This process is iterative and goes on until the requirements are finalized and all TBD values are determined.

After finalizing the requirements definition and logical decomposition, it has been decided to start generating design solutions and keep discussing them with this team within the team. This process will be analyzed in the next section.

4.4. Design Solution Definition

To be able to brainstorm about possible system architectures and their comparison with each other, the systems engineer needs to understand the requirements that the system has to fulfill. After the team was built and finalized the requirements, the team started to think about possible design solutions.

As the aim of the project is creating a virtual laboratory network, possible system architecture options differ in the way of communication and data management of the components. The realization of this fact led to research on status quo communication protocols and architecture options. This research is summarized in section 2.2.

Understanding and listing technical requirements is not enough to generate ideas, as the systems engineer also needs to understand what each subsystem does and how they communicate with each other. As discussed in section 2.2.2, there are different communication protocols suitable for various communication types. All of the communications happening in the virtual laboratory network have to be analyzed and matched with the corresponding communication protocol. Hence this phase includes all the teams and subsystems involved in the project.

A Microsoft Excel document was prepared for collecting all the necessary information from the subsystems. Each subsystem has its own sheet with a table for all communication partners and the specifics of the communication. A sample of this table is given in Table 4.1:

	Inputs						
	Communication Partner	**Signal Name**	**Unit**	**Signal Direction**	**Data Exchange Rate Required (Hz)**	**Data Exchange Rate Desired (Hz)**	**Data Type**
	Subsystem A	Information Alfa	V	input	10 Hz	20 Hz	integer
Subsystem X	Subsystem B	Information Beta	I	input	20 Hz	20 Hz	double
	Outputs						
	Communication Partner	**Signal Name**	**Unit**	**Signal Direction**	**Data Exchange Rate Required (Hz)**	**Data Exchange Rate Desired (Hz)**	**Data Type**
	Subsystem C	Information Gamma	I/s	output	200 Hz	400 Hz	booelan
	Subsystem D	Information Delta	W	output	10 Hz	20 Hz	integer

Table 4.1.: Sample table for collecting communication information of all subsystems

Using the entries of the table, one can calculate the required data throughput and see if the communication protocols can handle the required throughput. As it is aimed to run real-time experiments with our virtual laboratory network, it is essential for the communication protocol to supply the required throughput. Nevertheless, it has to be taken into account that the throughput capacity is not a single criterion to identify the suitable communication protocol.

After all of the subsystems filled up the above-mentioned table, it was necessary to double-check if the registered data is consistent for all subsystems. If Subsystem A mentions an output directed to Subsystem B, this data should also be registered as input for Subsystem B. This double-check was done while modeling the system architecture in a SysML BDD.

Blocks were created for each subsystem of the power train and its laboratories. For each presence of communication between two entities a directed association was created and named after all the data that should be sent. Before a communication was included in the model, it is double-checked that an input of a subsystem is also registered as an output for the communication partner. Every communication included in the model was also marked in the Excel list. This way inconsistencies in the lists were identified, subsystems were contacted to clarify them, and communications and our model were validated.

As it is aimed to find the optimal system architecture, the virtual laboratory network systems and their communications for each scenario were modeled. These scenarios are:

- Centralized Architecture
- Decentralized Architecture
- Mixed Approach
- 2 Layer Approach

Following subsections show the details of these different scenarios and corresponding models. An analysis of different architecture models and the decision process will be explained in Chapter 5.

4.4.1. Centralized Architecture

The centralized architecture is an extreme scenario. In this architecture model all the components send every bit of data to a central server. The server saves the data in a central database and forwards the message to the receiver. There is no direct communication between any subsystems or laboratories. The BDD of the centralized architecture is depicted in Figure 4.3. A bigger version can be found in Appendix D.1. To reduce complexity, the communication details are not included in this architecture model. Each communication follows the same logic and runs centralized, therefore they are irrelevant to the model.

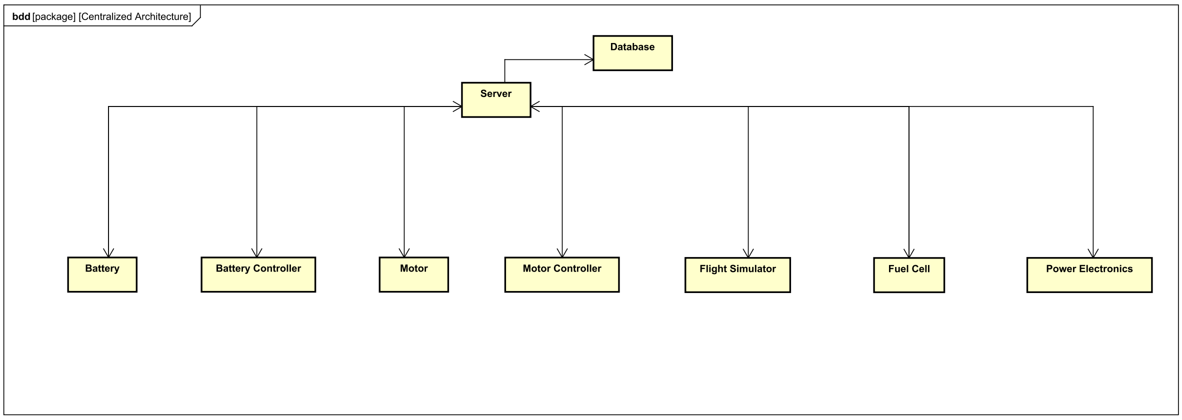

Figure 4.3.: Block Definition Diagram of the Centralized Architecture

4.4.2. Decentralized Architecture

The decentralized architecture is also an extreme scenario. There is no central server used in this architecture and all of the subsystems communicate directly with the communication partners. Associations colored blue are wireless connections whereas the red-colored communications represent a physical connection. These colors and reasoning of these choices will be analyzed in the next chapter. Figure 4.4 portrays the BDD of this decentralized architecture. A bigger version can be found in Appendix D.2.

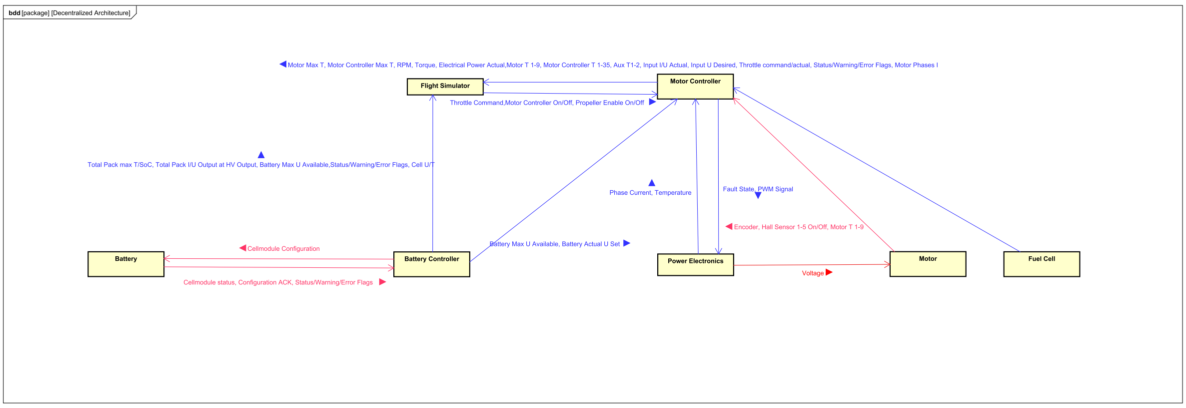

Figure 4.4.: Block Definition Diagram of the Decentralized Architecture

4.4.3. Mixed Approach

After presenting both extreme scenarios with complete centralization and decentralization, a mixed approach was introduced. A central server is connected to each component and its laboratories. Some components communicate directly with each other and some communications are directed over the central server. These two types of communications also differ in the underlying communication protocol and these protocols and reasoning for different protocols will be analyzed in the next chapter. The central server saves all the communication in a central database. The BDD of this approach is illustrated in Figure 4.5. A bigger version of the BDD can be found in Appendix D.3. The red-coloured associations are physical connections and the blue ones are wireless communications. The black-coloured associations represent the LAN/WAN communications running through the central server.

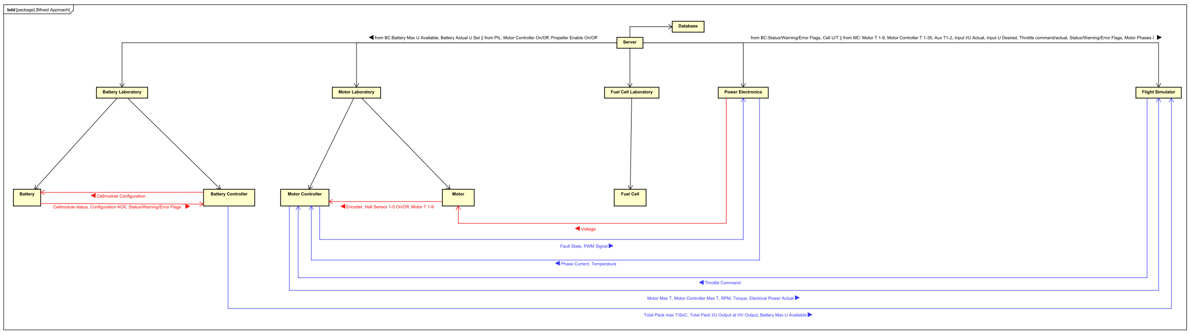

Figure 4.5.: Block Definition Diagram of the Mixed Approach

4.4.4. 2 Layer Approach

The 2 Layer Approach is the destination point of the iteration and represents the final system design. The communication runs both decentralized between each communication partner and each communication data is sent to a central server, where they are saved in a central database. The centralized layer is used for central commands of the experiment and data storage and the decentralized layer is used for the real-time communication between the components. The BDD of this approach can be seen in figure 4.6 and a bigger version can be found in appendix D.4. The red coloured associations are physical connections and blue coloured ones are wireless real-time communications. All of this data are also sent to the server with a different communication protocol and the components receive central commands from the server. This layer is represented in black color.

The framework of this work ends with the modeling of different system architecture options

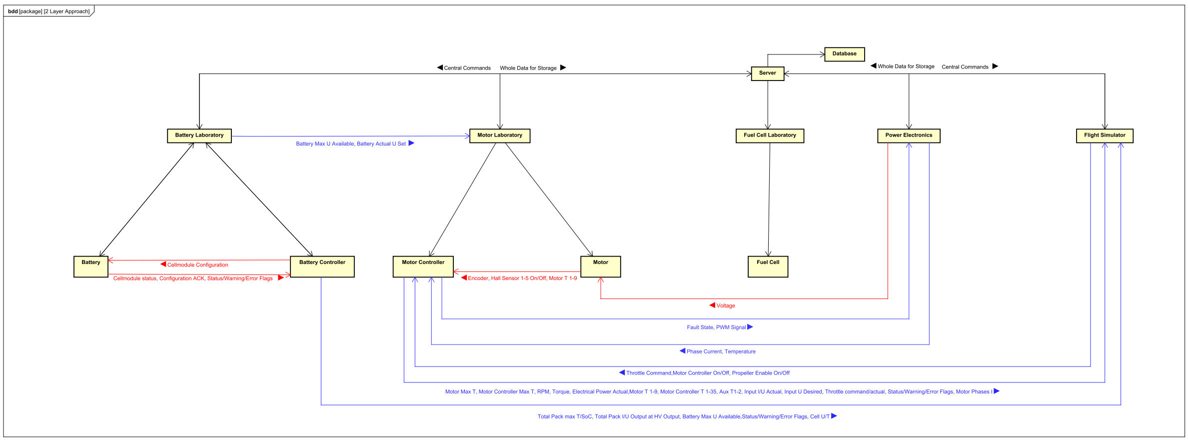

Figure 4.6.: Block Definition Diagram of the 2 Layer Approach

and an analysis of these architectures. The project continues with the next steps including implementing the communication protocols and testing them to validate our approach. The analysis in this work will not be based on testing results as it was not possible to proceed with the implementation during the thesis. The presented architecture options as well as our whole approach are analyzed in the next chapter.

5. Analysis

In this chapter, the whole process is analyzed and the effects of the systems engineering approach on the project ELAPSED is investigated. The research questions that we listed in Chapter 3 will also be answered.

5.1. Why Systems Engineering?

The first research question focuses on the effects of the systems engineering approach. The systems engineering approach is compared with the traditional approach and possible benefits of systems engineering in a project, where multiple complex systems are involved are emphasized. The sub-project 1 aims to maintain a virtual connection of different test benches and engine components to conduct distributed testing. This connection involves a real-time communication of each involved system and should be designed during the development of these components.

In a traditional development approach, all systems concentrate on their components and models. The project manager has an overview of the whole system and is responsible for the management. However, the project manager is also responsible for risk management, budget planning, resource management, the definition of non-technical requirements, and bureaucracy. It might be feasible for a project manager to be responsible for these tasks and also be involved in the technical management of the system. However, with the increasing complexity of the system, the technical management activities of the system constitute a "full-time job". As mentioned in section 2.1, systems engineering becomes more relevant each day with the increasing complexity of the systems.

The aim to create a virtual laboratory network leads to the requirement of high interoperability between different complex systems. Looking back at the criteria listed for the concept of SoS in section 2.2, one might argue that the virtual laboratory network is a system of systems due to the following reasons:

- One can conduct tests of each product in their own laboratory, without needing the connection to the other laboratories.

- The laboratories are managed individually, even when it is aimed to run tests with other components. For a global test with all of the components involved, the connection must be maintained and each laboratory has to fulfill its own function.
- The laboratories are geographically distributed. The battery laboratory and the motor laboratory are on the same campus, but not in the same building. The flight simulator is on another campus approx. 10 km away.
- The development of an electrical/hybrid engine is multidisciplinary. The diversity among the components is significantly high. Aerospace engineers, electrical engineers, and chemical engineers for the fuel cell option are working together on the project ELAPSED.
- Distributed testing is an emerging technology and became feasible recently thanks to the developments in the field of Information Technology (IT).

As the virtual laboratory network can be classified as an SoS, one can also understand the necessity of the systems engineering approach for the project. It is not feasible for the project to run without systems engineering. The virtual laboratory network concept runs based on the communication of involved systems and communication of such different systems require an intensive interface management process. To be able to manage the interfaces, the project needs to employ engineers, who have an overview of all involved systems and holistically approach the project. This requirement led to the systems engineering team and this work.

The stakeholder analysis and the representation of the stakeholder overview with MBSE showed uses in many ways. As it was aimed to get an overview of the systems and their current status, the stakeholder analysis helped with the understanding of relevant stakeholders. This way the stakeholder interviews could be planned and the relevant information for the approach is derived. The modeling of stakeholders visualized the hierarchical structure of the project and team structures and also gave an idea about the components and systems involved. It is also useful as documentation for new stakeholders and project managers.

As the process moved on with the stakeholder interviews, the systems engineers got to know each system, its functions, their structure, teams, and the specific fields of focus of each involved stakeholder. The stakeholders were asked about the goal of the project to make sure that the stakeholders have a uniform understanding of the common goal. The systems engineering team came to a conclusion that a uniform understanding of the goal among stakeholders was not the case. Therefore the comprehension of the team was communicated with the stakeholders. This comprehension is formulated based on the project proposal and includes the functionality of the virtual laboratory network as well as the desire to create digital twins of the systems. Clarification of the goal made it possible for the stakeholders to express their first ideas about the technical requirements. These ideas were worked over with the team mentioned in Chapter 4 and finalized. At the end of the interview process, systems engineers were able to create a first draft of the requirements, which set the foundation for the system design. Also, the results of the stakeholder interviews and the current status of the project were communicated to the project management. This approach reduces the number of communications that the project manager has to maintain significantly and allows the management to focus on non-technical management tasks. In the case of sub-project 1, the number of meetings that the project manager has to attend for being up to date decreases from 7 to 1.

Although the technical requirements are defined based on the stakeholder interviews, the definition and documentation of them are done by the systems engineering team. This distribution of tasks reduces the workload of the engineers and allows them to focus on their system. It also lowers the risk of the dominance of one systems standpoint in the general requirement definition as the requirements are defined by systems engineers.

Another significant benefit of the systems engineering work was to document and model each communication that should occur during the operation of the virtual laboratory network. As mentioned in Chapter 4, a list was prepared for each system to document the inputs and outputs of their system. After the teams filled up the list, systems engineers went over the information and took care of inconsistencies. This work package allowed complete documentation of each communication between involved systems and allowed the modeling of the communication using MBSE. Without a holistic systems engineering approach, each sub-team would need to define every communication with each communication partner to achieve consistent documentation.

To conclude this section one can say the systems engineering approach improved the progress of the project significantly. With systems engineering methods, the workload of management and sub-teams was reduced, the understanding of the goal and the end product by the involved stakeholders was uniformed, cooperation and collaboration was improved, documentation for stakeholders, requirements, and system architecture was provided, and the efficiency of the development phase was improved. These improvements were observed by all the stakeholders in a project where the stakeholders were not familiar with systems engineering methods.

5.2. Use of Model Based Systems Engineering

As mentioned in the previous section, MBSE was used for modeling the stakeholder structure and various system architecture alternatives in this work. Modeling the stakeholder allowed the systems engineers to provide sufficient documentation for the stakeholder structure of the project. The hierarchical structure of the project as well as the team structure were also visualized. This model can be used by all stakeholders involved in the project to identify relevant stakeholders and understand whom they should communicate with. Our model helped with the planning of the stakeholder interviews and provided model-based documentation of stakeholders for the project.

As we mentioned in Chapter 4, a requirement model could not be created using MBSE tools and Polarion was used for the documentation of requirements. The necessity of using MBSE for the requirements is emphasized in the outlook, as it allows the systems engineers to associate the requirements with responsible stakeholders and system blocks, which would fulfill these requirements. This will be another significant advantage of stakeholder modeling as it allows traceability of requirements to the responsible stakeholder.

BDDs were used for modeling the system architecture. Different components and laboratories are represented as system blocks and the communications between these systems are represented with directed associations. As different ideas and approaches for possible system architectures were developed, these BDDs were used for the visualization of these different approaches. This makes it possible to have a visual representation of each alternative and facilitate discussions with other team members using the visualization of the architecture.

These models will be used as a foundation for all teams to understand the desired state of interfaces and design their systems accordingly. As each team will be working with the same model, it reduces the risk of misunderstandings and inconsistencies. Thereby the model-based approach shows higher efficiency compared to the traditional approach.

5.3. Centralized vs. Decentralized Architecture

As presented in Chapter 4, four different system architectures were proposed. The first approach represents complete centralization whereas the second approach represents complete decentralization. The last two approaches use a mixture of both centralized and decentralized elements and differ in the usage of communication protocols and communication architecture.

A centralized architecture offers a significant advantage for data storage and documentation. The communication data is sent to a central server from the source and it is stored in a central database. The server forwards the message to the communication partner and the traffic is managed globally from this server. This approach allows the storage of the whole experiment data in the central database. An engineer, who wants to evaluate the whole experiment has to log in to the server API and can reach the data efficiently.

The centralization also ensures easier management of the nodes and the experiment. Test engineers can control the experiment using one computer and also supervise the test with an uncomplicated system. The communication protocol of each system is unified and the central management system helps with standardization. The standard rules of communication with the server may also increase scalability, as the engineers would know how to implement the communication according to the server and new systems that should be included in the network would have a clear guideline.

However, centralization brings strong data management requirements for the central server. If one considers a single communication between two communication partners, one data package would be sent and received twice in the centralized approach. A decentralized communication would allow the delivery of the data package from the source to the receiver in one iteration. Gathering all the information in a central node would require high bandwidth and may lead to congestion.

If the central node faces an error and stops working, the experiment can not continue and the error must be fixed. In this case, the centralized approach may experience a "single point of failure". The decentralized approach offers more flexibility in this regard, as the communication can continue among other nodes if one node faces an error and can not operate correctly.

As one can see from this analysis, both centralized and decentralized approaches have their strengths and weaknesses in different critical fields of operation. A mixture of both elements was proposed and our system was designed with both centralized and decentralized architecture in the last two system architectures presented in Chapter 4. In the next section, these two approaches will be analyzed and different approaches regarding communication protocols will be explored.

5.4. Different Architecture Options

This section starts with the third architecture option that is called the "Mixed Approach". In this approach, all the participant systems are connected to a central server, however the communication is not maintained exclusively through the central server. The communications were divided in two groups and it was decided to use different transport layer protocols for these groups to investigate different benefits of different communication protocols. The first group of communication uses a TCP transport layer and runs centralized through the central server. This communication is colored black in the corresponding BDD and can be seen in Figure D.3. The blue-colored associations are representing a UDP-based decentralized communication and the red-colored ones are physical Controller Area Network Aerospace (CANAerospace) busses.

The communications between the motor controller and motor as well as between the battery controller and battery are physical connections, as these components are always in the same location. This approach reduces the wireless network traffic in our virtual laboratory network. The decision to use CANAerospace for these communications was made by the teams for their systems. The communication of voltage value between the power electronics/inverter and the motor was not possible to maintain with any wireless approach, as the required frequency of information exchange is too high ($\approx 500.000 Datapoints/Second$) for a wireless connection. Therefore a physical connection of these elements will also be used for this communication, nevertheless, other communications between power electronics and the motor controller can be realized with a wireless UDP connection.

The reason to divide LAN/WAN communication in to two different groups is to use the benefits of both mentioned communication protocols for the most suitable communication data. In section 2.2.2, the research about the two main transport layer protocols were summarized. TCP provides reliable data transfer thanks to the acknowledgement-based mechanism. UDP on the other hand can send data packages with less delay due to the lack of control mechanisms like TCP. The first approach during the system design definition was calculating the maximum possible throughput with TCP and UDP and comparing them with the required throughput values for each communication. However, it is not possible to calculate the maximum throughput of both communication protocols without precise information about network characteristics and package sizes. Therefore two different regions of desired frequency (Datapoints/Second) in the communication requirements provided by the sub-teams were identified. It has been decided to use TCP for low-frequency data ($\approx 10 - 20 Datapoints/Second$) and UDP for higher frequency data ($\approx 100 - 200 Datapoints/Second$).

This way the reliability aspect of TCP can be used for communications in slower regions and no congestion and less delay aspect of UDP can be used for communications in the faster regions.

The last stage of the iteration and final system architecture design is called the "2 Layer Approach". In this approach, the whole communication runs:

1. Decentralized between the communication partners with the UDP transport layer for the real-time application
2. Centralized to a central server with the TCP transport layer for data storage and central commands from the server to the participant systems

The 2 Layer Approach is represented in Figure D.4. The black-colored associations represent the transmission of all communications to the central server and the central commands to each system. The blue-colored associations are all wireless communications and the red-colored ones are physical connections.

With this approach, most of the benefits of different system designs are combined. A UDP-based decentralized communication of components and laboratories offers the fastest communication option for each communication. This is crucial for the virtual laboratory network as it is aimed to run real-time tests with components in different locations. UDP comes forward in this aircraft layer real-time data exchange due to its higher capacity to fulfill speed requirements compared to TCP.

For the data storage and central command layer, TCP and centralization come forward as better options. With the central server, the experiment can be supervised and controlled by the test engineers more efficiently. Using TCP-based communication for this layer ensures accurate documentation of the whole experiment and assurance of central commands reaching the systems to control the experiment. This layer does not have high-speed requirements as the whole communication is happening faster in the real-time application layer.

To conclude, two different layers with different requirements and use cases were created. This 2 Layer Approach allowed the usage of the benefits of centralization and the TCP transport layer protocol for experiment control and data storage while using the benefits of decentralization and UDP transport layer protocol for real-time operation of our virtual laboratory network.

6. Conclusion

With this work, a completely new development approach was brought to sub-project 1. The sub-project 1 aims to maintain a connection between several test benches and components of an aircraft. This work aims to increase the efficiency of the development of this network and find the best system design solution that will fulfill the requirements.

An electrical aircraft engine consists of several components designed and developed by multiple disciplines. Each of these systems has its own test environment and different characteristics. These systems are complex systems by themselves and we aim to create a network, where these complex systems communicate with each other to achieve a distributed testing of the whole engine. Maintaining communication between already complex systems bring new challenges and to overcome these challenges the holistic systems engineering approach was introduced.

The process started with a stakeholder analysis and the benefits of the model-based approach were used to visualize the stakeholder structure of the project. Stakeholder interviews were conducted with each relevant stakeholder to extract useful information from the stakeholders. This method increased the efficiency of the organizational process, gave the systems engineers familiarity with all involved systems, and helped the systems engineers to derive the technical requirements of the virtual laboratory network.

The technical requirements were defined in multiple iterations and the systems engineering team started to think about possible design solutions, which will fulfill the requirements. To be able to model the different approaches and hold discussions with the model, lists were created for each system to define the interfaces. As the system architecture is the way of communication of the components in the network, these interfaces are the desired communications between the participants of the virtual laboratory network. The definition of the interfaces allowed the systems engineers to proceed with modeling the architecture options.

The different architecture concepts differ in the degree of centralization and underlying communication protocol. The benefits and drawbacks of each approach were analyzed and it has been decided to design a 2 layer architecture, which can combine the benefits of different methods in one system architecture. A TCP-based central command and data storage layer ensures the reliable storage of all experiment data in a central server and gives the test engineers to control the test globally. Another UDP-based real-time application layer allows a real-time capable decentralized communication between the components to achieve real-time distributed testing.

However, this master's thesis can only cover a limited part of the systems engineering approach and the project continues with the implementation phase. The discussion of used methods, possible next steps and the author's recommendations are presented in the next chapter.

7. Discussion and Outlook

In this chapter the ability of this work to answer the research questions as well as the efficiency of used methods will be discussed. An outlook for future works will also be given.

7.1. Discussion

The first research question investigates the effect of the systems engineering approach on the project progress. This research question was one of the main focus points of this work, as the systems engineering methods were introduced for the first time in the project group. Improved understanding of the team structure with the stakeholder analysis, reduced workload of the project manager and involved engineers throughout the whole development process, increased efficiency of the requirements management process and efficient interface management with the help of modeling and interface management made it possible to answer the first research question with high utility. It is also worth mentioning, that the question is formulated broadly and a further analysis is possible. As it is not possible to run the same project without the systems engineering approach, a comparison with quantitative metrics between the systems engineering approach and the traditional approach is not possible.

The second research question explores the benefits of MBSE during the system architecture studies. As mentioned in Chapter 4, at the time the technical requirements of the virtual laboratory network were defined, the MBSE tool Astah was not available and Polarion was used instead. To use all of the advantages MBSE has to offer, the requirements should also be included in the MBSE model of the virtual laboratory network. This way the systems engineers can have the opportunity to associate requirements with system blocks that fulfill these requirements. Knowing which requirement should be fulfilled by which system block simplifies the verification process of the requirements. Therefore this research question could not be answered with perfect utility.

The third research question focuses on the centralized and decentralized architecture options. This research question was answered with high utility, as the benefits of different approaches were discussed in the 5 and both methods were combined in this work. However this comparison could not be done quantitatively as the project was not in the implementation phase. A mathematical analysis of quantitative metrics during the implementation phase would lead to more scientific results.

The fourth research question analyzes different system architectures and compares the transport layer protocols with each other. This research question was answered with the demonstration of architecture models. The efficiency of answering this research question comes with the visualization of architectures with the help of MBSE. It is also with mentioning that this research question also might be answered more scientifically, if the project could progress into implementation phase and experiments could be run with different architecture options.

7.2. Outlook

Creating an MBSE model of a system does not necessarily mean the systems engineer should create each type of SysML diagram for a complete model. However, there are SysML diagram types that would help further for a sound operation of the virtual laboratory network. Different SysML diagram types and their use cases were mentioned in section 2.1.2. The internal structure of the laboratories as well as the subsystems of the engine can be further modeled using IBDs. As the focus of this work lied on the communication aspect of the system architecture for the whole network, IBDs can help to understand the physical architecture of the systems involved. Moreover, behavior diagrams would help the developers to understand the desired concept of operation. They can also be used by the users to understand the behavior of the system. Activity diagrams would clarify the necessary actions for the desired output whilst state machine diagrams would depict different possible states of the virtual laboratory network and the components.

The job of the systems engineering team is not completed with the definition of a design solution. A project benefits from the systems engineering approach from the initiation phase until the retirement of the end product. The systems engineering team in the project should continue to manage the interfaces of the virtual laboratory network by extending the MBSE model and verifying the requirements. After finalizing the design, the systems engineer should plan the implementation, develop test cases and run these tests. Providing sufficient documentation like user guides is also an important part of the systems engineers job. Finally, the systems engineers should monitor the system performance during the operation and work continuously to improve the performance.

As the project is still in the development phase, the systems engineering team was not able to test different system architectures and communication protocol options with the systems. Instead, the system architecture was designed based on theoretical knowledge. The implementation of the communication infrastructure is still in progress. After the communication protocols are successfully integrated into the models, the communication should be tested and compared with the theoretical results. This way the systems engineers can decide to proceed with the current system architecture or develop new ideas in case of deviating results.

A. Stakeholder Model

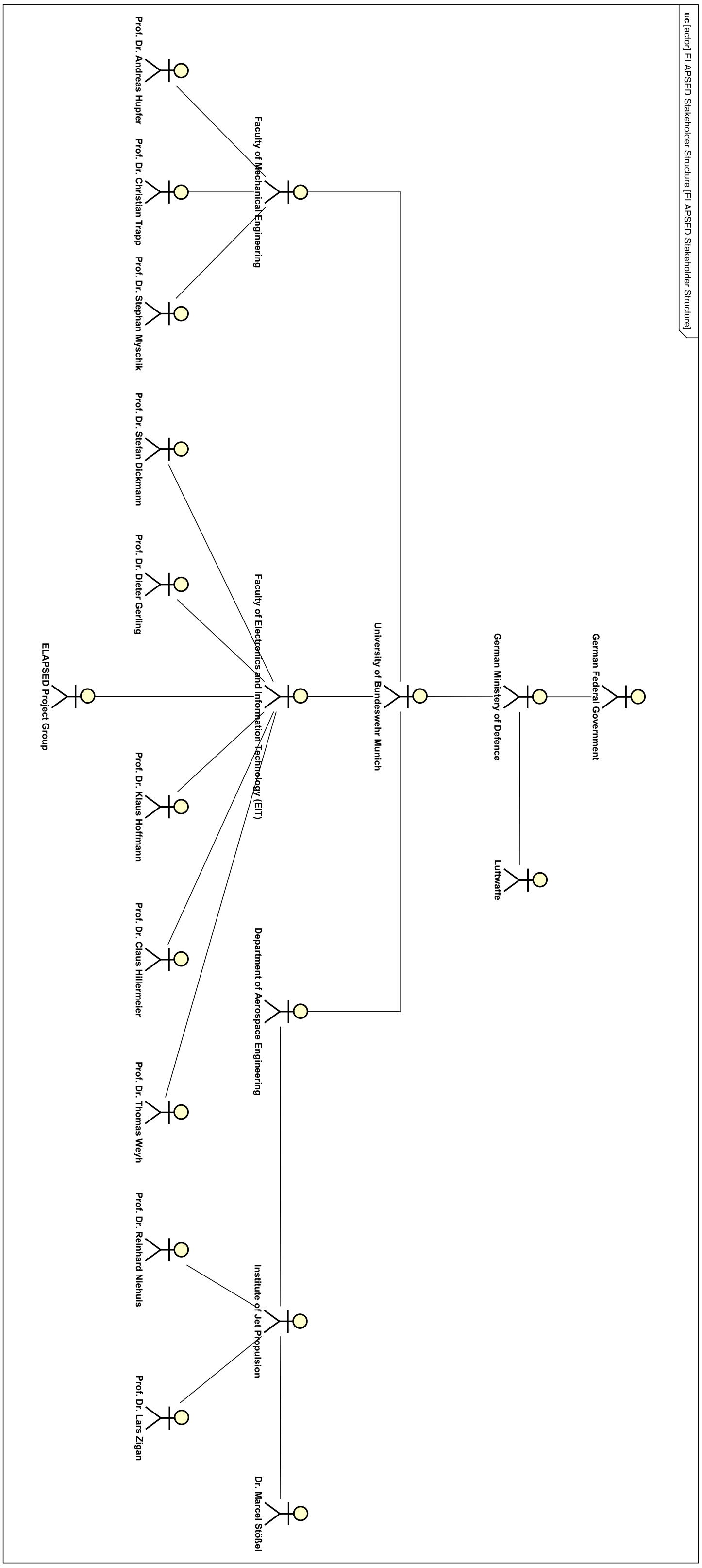

Figure A.1.: Stakeholder Model of project ELAPSED - Layer 1 - Bigger Version

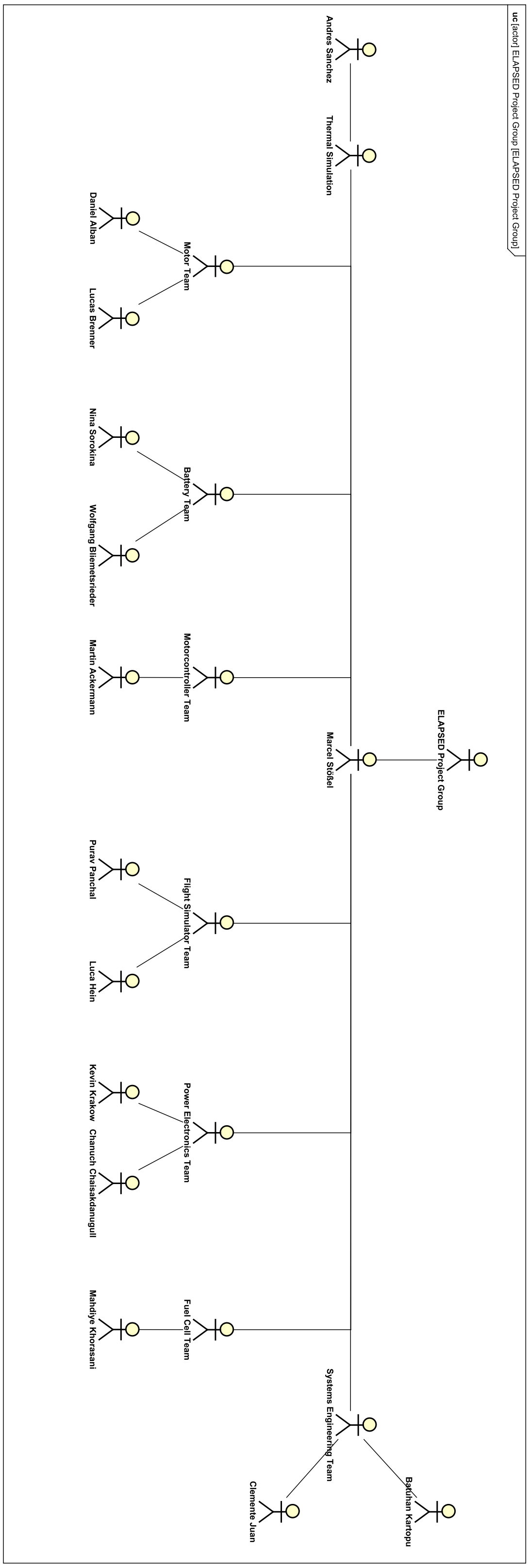

Figure A.2.: Stakeholder Model of project ELAPSED - Layer 2 - Bigger Version

B. Stakeholder Interview - Questions

1. What would you define as the overall goal of the ELAPSED Sub-project 1?
2. Who are the involved stakeholders from your subsystem in ELAPSED Sub-project 1?
3. Which specific components should maintain a connection to the other elements of the virtual laboratory network?
4. Do you have a digital twin of your subsystem?
5. If not, is someone working on the creation of the digital twin for your subsystem?
6. Is your testing environment compatible to work with digital twins of the components?
7. What do you need for the creation of the digital twin and for the connection of the digital components to your lab?
8. What do you want to achieve for your subsystem with the successful completion of ELAPSED Sub-project 1? (Outcome)
9. Which actions and measurable steps are you intending to follow to achieve your goal?
10. What are the technical requirements for achieving these goals?
11. How is your time-plan for the Sub-Project 1?

C. Requirements

C.1. Connectivity and Communication Requirements

- Different avionic test-benches shall be able to communicate with each other using a standard protocol.
- The test-benches shall communicate with a minimum latency of 3 ms.
- The virtual laboratory network shall receive and send signals in real-time.

C.2. Security Requirements

- The virtual laboratory network shall ensure protection against unauthorized access.
- The virtual laboratory network shall have a role-based control system and restrict access to defined parts of the network to authorized users.
- The virtual laboratory shall have at least 3 levels of user access.

C.3. Synchronization and Data Management Requirements

- The virtual laboratory network shall maintain time synchronization between all test-benches with an accuracy of 0.003 s and a maximum of 9 missed ticks.
- The virtual laboratory network shall allow hybrid testing of real components with digital twins.
- The virtual laboratory network shall be able to store and retrieve data for 48 months.

C.4. Remote Access and Control Requirements

- The virtual laboratory network shall use a Simulink interface and be accessible over the internet.
- The virtual laboratory network shall handle at least 10 users simultaneously.

C.5. Simulation Requirements

- The virtual laboratory network shall be able to simulate real-world scenarios including the behaviour of the engine, battery, flight dynamics and the pilots reactions.
- The virtual laboratory network shall simulate software in the loop, hardware in the loop and pilot in the loop elements simultaneously.

C.6. Usability Requirements

- The user interface of the virtual laboratory network shall be user-friendly.
- The virtual laboratory network shall have detailed documentation, handbook and training materials for the users.
- The virtual laboratory network shall be able to diagnose and troubleshoot the issues remotely.
- The virtual laboratory network shall provide virtualization and simulation support for multiple OS and software.

C.7. Scalability Requirements

- The virtual laboratory network shall have a modular architecture.
- The virtual laboratory network shall be able to scale with additional components, laboratories and systems.

D. System Architecture Models

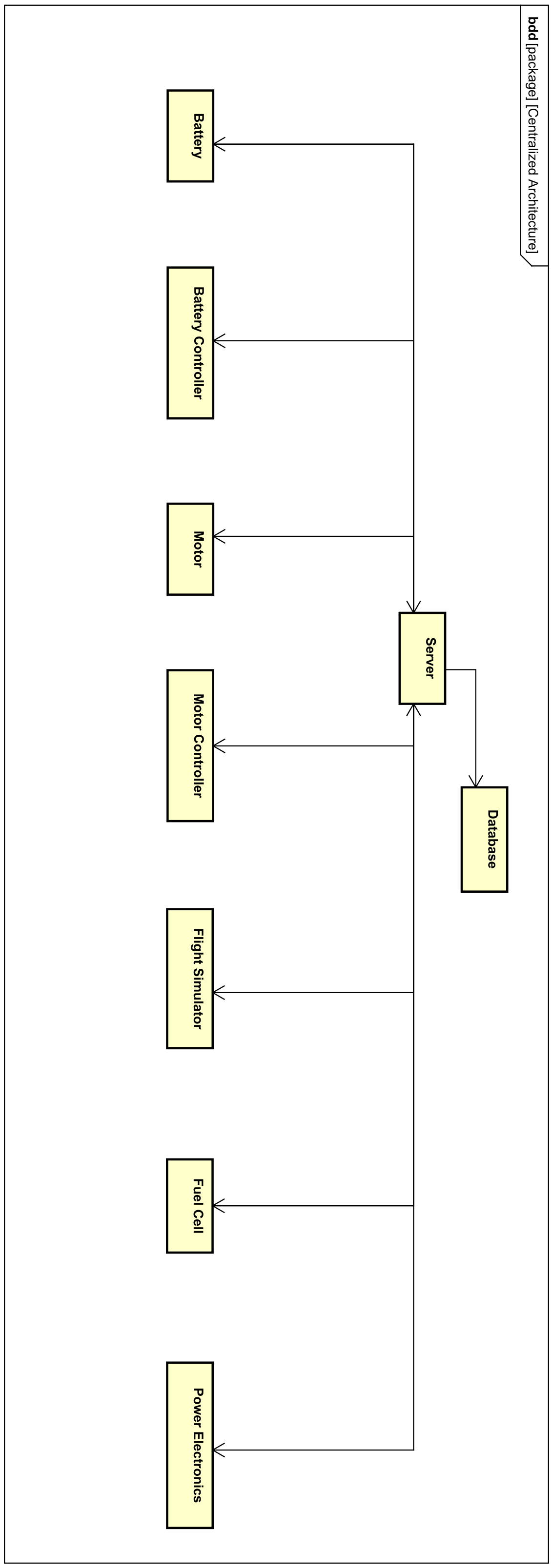

Figure D.1.: Block Definition Diagram of the Centralized Architecture - Bigger Version

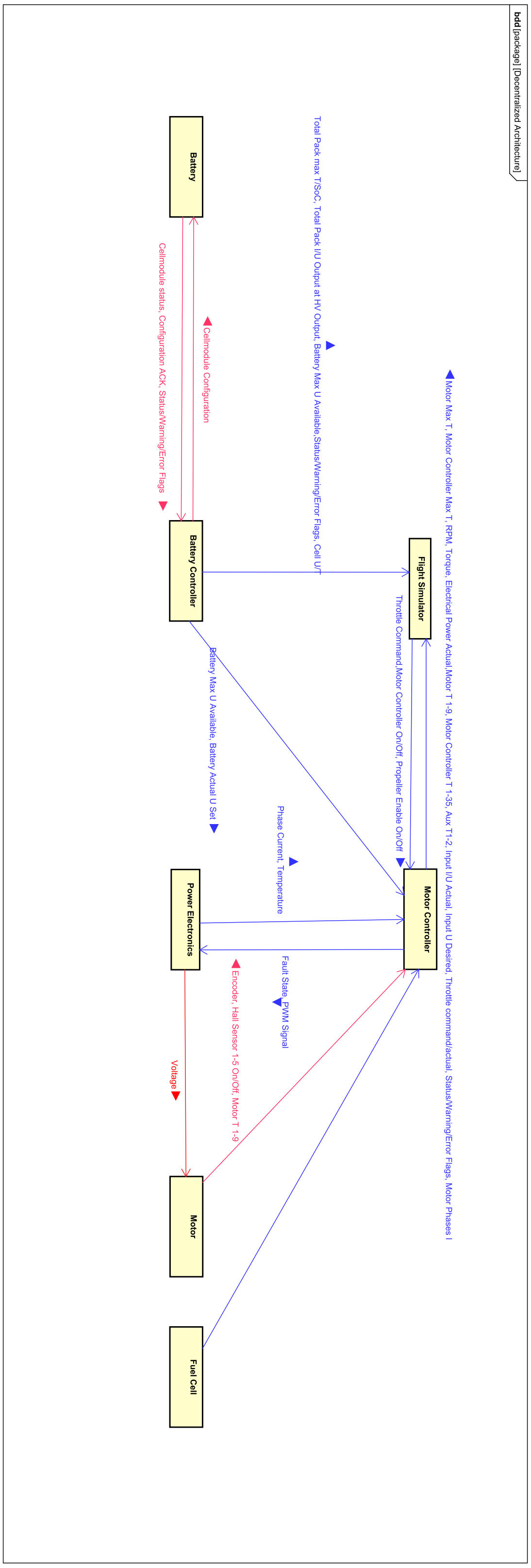

Figure D.2.: Block Definition Diagram of the Decentralized Architecture - Bigger Version

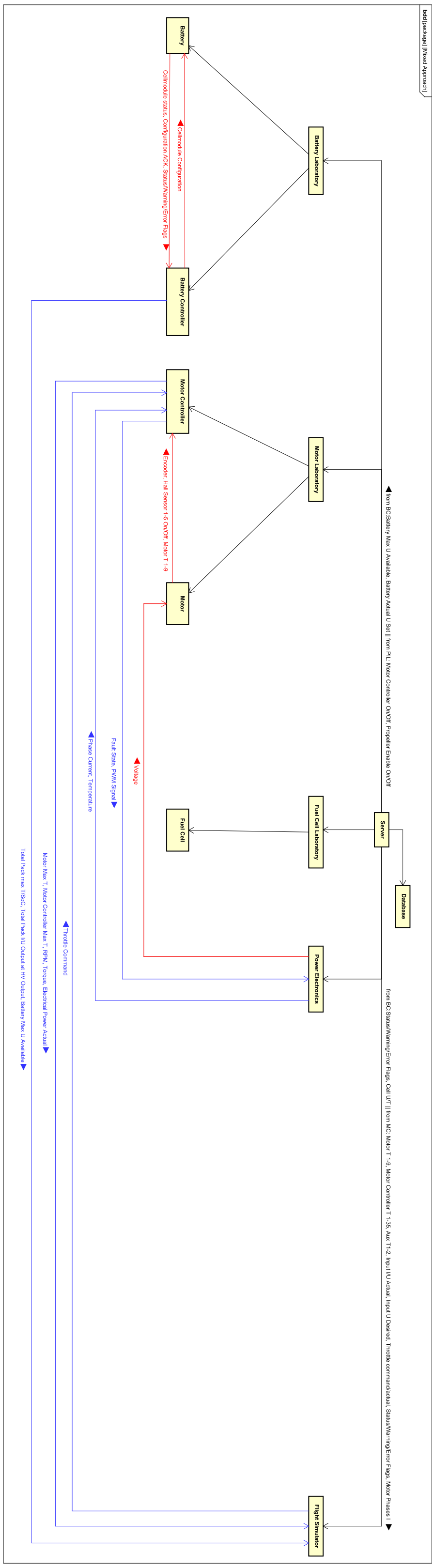

Figure D.3.: Block Definition Diagram of the Mixed Approach - Bigger Version

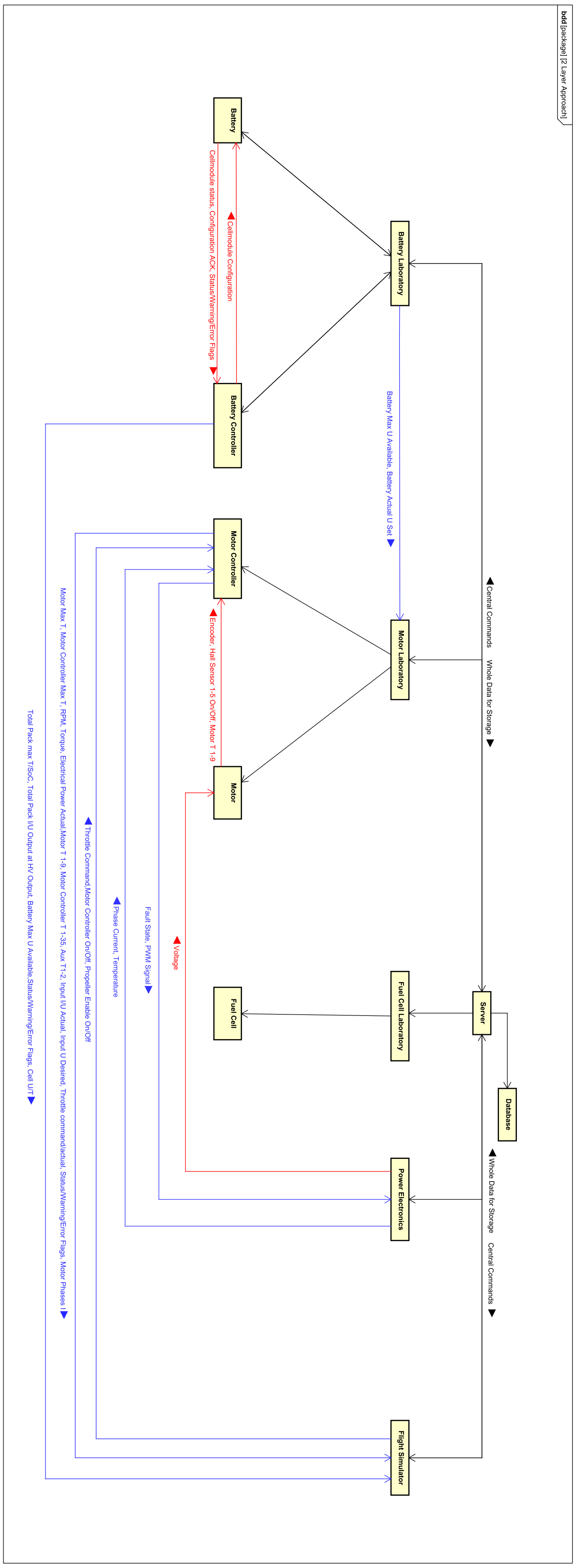

Figure D.4.: Block Definition Diagram of the 2 Layer Approach - Bigger Version

I confirm that this master thesis is my own work and I have documented all sources and material used.

Munich, 31.05.2023

Batuhan Kartopu

Batuhan.K

Bibliography

Batarseh, O. & McGinnis, L. F. (2012), System modeling in sysml and system analysis in arena, *in* 'Proceedings of the 2012 Winter Simulation Conference (WSC)', pp. 1–12.

Boardman, J. & Sauser, B. (2006), System of systems - the meaning of of, *in* '2006 IEEE/SMC International Conference on System of Systems Engineering', pp. 6 pp.–.

Buede, D. M. & Miller, W. D. (2016), 'The engineering design of systems: models and methods'.

Eisner, H., McMillan, R., Marciniak, J. & Pragluski, W. (1993), Rcasse: Rapid computer-aided system of systems (s2) engineering, *in* 'INCOSE International Symposium', Vol. 3, Wiley Online Library, pp. 267–273.

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

Georg, M. (2006), 'A survey of tcp optimizations for wireless channels', *mgeorg@ arl. wustl. edu* .

Hall, A. D. (1962), 'A methodology for systems engineering.-princeton'.

Hause, M. et al. (2006), The sysml modelling language, *in* 'Fifteenth European Systems Engineering Conference', Vol. 9, pp. 1–12.

Hirshorn, S. R. (2017), 'Nasa systems engineering handbook revision 2'.
URL: *https://www.nasa.gov/connect/ebooks/nasa-systems-engineering-handbook*

Hossain, N. U. I., Jaradat, R. M., Hamilton, M. A., Keating, C. B. & Goerger, S. R. (2020), 'A historical perspective on development of systems engineering discipline: a review and analysis', *Journal of Systems Science and Systems Engineering* **29**, 1–35.

Kumar, S., Dalal, S. & Dixit, V. (2014), 'The osi model: Overview on the seven layers of computer networks', *International Journal of Computer Science and Information Technology Research* **2**(3), 461–466.

Kumar, S. & Rai, S. (2012), 'Survey on transport layer protocols: Tcp & udp', *International Journal of Computer Applications* **46**(7), 20–25.

Li, L., Aslam, S., Wileman, A. & Perinpanayagam, S. (2022), 'Digital twin in aerospace industry: A gentle introduction', *IEEE Access* **10**, 9543–9562.

Liu, M., Fang, S., Dong, H. & Xu, C. (2021), 'Review of digital twin about concepts, technologies, and industrial applications', *Journal of Manufacturing Systems* **58**, 346–361. Digital Twin towards Smart Manufacturing and Industry 4.0.
URL: *https://www.sciencedirect.com/science/article/pii/S0278612520301072*

Maier, M. W. (1998), 'Architecting principles for systems-of-systems', *Systems Engineering: The Journal of the International Council on Systems Engineering* **1**(4), 267–284.

Martinen, D. H., Lagalaye, M., Pfefferkorn, J. & Casteres, J. (2017), Modular and open test bench architecture for distributed testing, Technical report, SAE Technical Paper.

Nations, U. (2015), 'Paris agreement'.
URL: *https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en*

Pouzolz, R., Schmitz, O. & Klingels, H. (2021), 'Evaluation of the climate impact reduction potential of the water-enhanced turbofan (wet) concept. aerospace 2021, 8, 59', *Stress8* p. 31.

Rahmani, M., Pettiti, A., Biersack, E., Steinbach, E. & Hillebrand, J. (2008), A comparative study of network transport protocols for in-vehicle media streaming, *in* '2008 IEEE International Conference on Multimedia and Expo', IEEE, pp. 441–444.

Roman, R., Zhou, J. & Lopez, J. (2013), 'On the features and challenges of security and privacy in distributed internet of things', *Computer Networks* **57**(10), 2266–2279. Towards a Science of Cyber Security Security and Identity Architecture for the Future Internet.
URL: *https://www.sciencedirect.com/science/article/pii/S1389128613000054*

Schlager, K. J. (1956), 'Systems engineering-key to modern development', *IRE Transactions on Engineering Management* **EM-3**(3), 64–66.

Shenhar, A. J. (1995), 2.5. 1 a new systems engineering taxonomy, *in* 'INCOSE International Symposium', Vol. 5, Wiley Online Library, pp. 723–732.

SysML FAQ: What are the SysML diagram types? (n.d.), `https://sysml.org/sysml-faq/what-are-sysml-diagram-types.html`. Accessed: 2023-04-15.

Tao, F., Xiao, B., Qi, Q., Cheng, J. & Ji, P. (2022), 'Digital twin modeling', *Journal of Manufacturing Systems* **64**, 372–389.
URL: *https://www.sciencedirect.com/science/article/pii/S0278612522001108*

Tchana de Tchana, Y., Ducellier, G. & Sébastien, R. (2019), 'Designing a unique digital twin for linear infrastructures lifecycle management', *Procedia CIRP* **84**, 545–549.

Thomson, R. (2020), 'Electric propulsion is finally on the map', `https://www.rolandberger.com/en/Insights/Publications/Electric-propulsion-is-finally-on-the-map.html`. Accessed: 2023-04-09.

Trainor, T. & Parnell, G. (2007), '2.2.2 using stakeholder analysis to define the problem in systems engineering', *INCOSE International Symposium* **17**, 296–309.

Wright, L. & Davidson, S. (2020), 'How to tell the difference between a model and a digital twin', *Advanced Modeling and Simulation in Engineering Sciences* **7**(1), 1–13.

Wuebbles, D. J. & Jain, A. K. (2001), 'Concerns about climate change and the role of fossil fuel use', *Fuel Processing Technology* **71**(1), 99–119. Fuel science in the year 2000: Where do we stand and where do we go from here?
URL: *https://www.sciencedirect.com/science/article/pii/S0378382001001394*