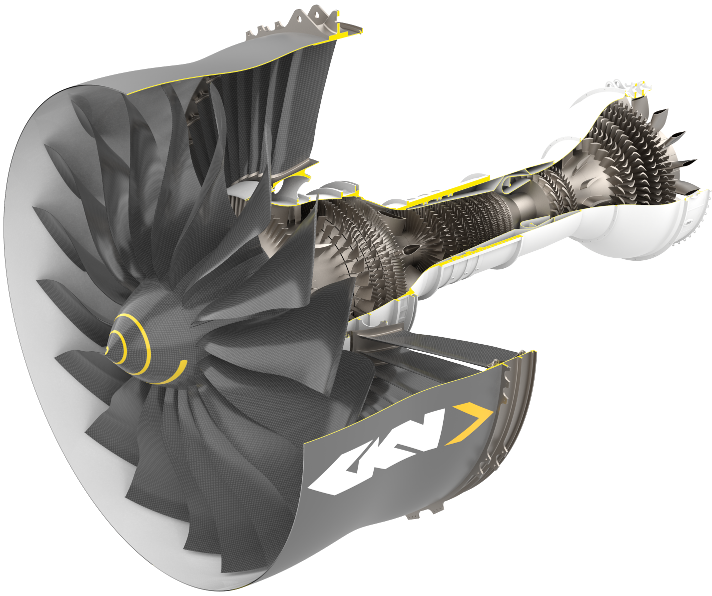
CHALMERS
UNIVERSITY OF TECHNOLOGY



A Method for Using Model-Based Systems Engineering to Drive Multidisciplinary Design Optimization Studies of Jet Engine Components

Exploring how multidisciplinary design optimization studies of a jet engine can be driven by model-based systems engineering

Master's thesis in Applied Mechanics

EDVIN KARLSSON

DEPARTMENT OF MECHANICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Exploring how Model-Based Systems Engineering can be Used to Drive Multidisciplinary Design Optimization

Developing and Implementing an Integration Scheme to Connect a Model-Based Systems Engineering Tool to an Existing Framework for Multidisciplinary Design Analysis and Optimization

EDVIN KARLSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

Exploring how Model-Based Systems Engineering can be Used to Drive Multidisciplinary Design Optimization
Developing and Implementing an Integration Scheme to Connect a Model-Based Systems Engineering Tool to an Existing Framework for Multidisciplinary Design Analysis and Optimization
EDVIN KARLSSON

Supervisor: Petter Andersson, GKN Aerospace AB & Jakob Müller, GKN Aerospace AB
Examiner: Dag Bergsjö, Department of Mechanical Engineering

Master's Thesis 2026
Department of Mechanical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: General visualization and open view showing the common stages of a turbofan engine.

Gothenburg, Sweden 2026

Exploring how Model-Based Systems Engineering can be Used to Drive Multidisciplinary Design Optimization
Developing and Implementing an Integration Scheme to Connect a Model-Based Systems Engineering Tool to an Existing Framework for Multidisciplinary Design Analysis and Optimization
EDVIN KARLSSON
Department of Mechanical Engineering
Chalmers University of Technology

Abstract

The aerospace industry faces immense pressure to maintain efficient innovation due to increasing system complexity, strict environmental regulations and a very competitive market. To manage these conditions, organizations move increasingly towards exploring Model Based Systems Engineering (MBSE) as the potential new way of working to represent the decisions made during requirement breakdown for a novel system. Multidisciplinary Design Optimization (MDO) is a mature method commonly implemented when performing a Design Space Exploration (DSE) and currently MBSE and MDO are implemented in complete digital isolation from one another. This thesis explores the potential benefits and challenges with of adopting MBSE into the designated way of working at GKN Aerospace and develops an integration scheme with a corresponding method to evaluate the possibilities of utilizing MBSE to drive MDO studies. The developed method was evaluated based on three success criteria: **Traceability, automation** and **results reuse ability**. Implementation and evaluation of the developed method revealed that MBSE enables the possibility of having a digital thread from the Product Breakdown (PB) to the MDO environment, however, the current available tools and methods do not satisfy the needs to achieve an efficient integration and there is currently no established way of working with MBSE at GKN.

Keywords: Aerospace Industry, Model-Based Systems Engineering (MBSE), Multidisciplinary Design Optimization (MDO), Traceability, Product Breakdown (PB)

Acknowledgements

This thesis was made possible with the help of significant persons that provided support all the way throughout the project. My utmost gratitude is given to the following people:
My parents and my sister who's continuous support and encouragement provided me with a morale and spirit to push through intense times.
My industrial supervisors at GKN Aerospace, **Jakob Müller** and **Petter Andersson** who through their expertise and feedback provided guidance and a rewarding environment essential for the completion of the thesis.
My academic examiner **Dag Bergsjö** for his valueable feedback along the way and evaluation of the thesis.
My thesis partner **Sanchit Choudhary** who provided excellent collaboration and a complementary view through countless valuable discussions throughout the entire project.
The collegues I have met at GKN Aeropsace for providing a stimulating environment and took their time to participate in several studies and discussion throughout the thesis.

Edvin karlsson, Gothenburg, June 2026

Division of Work

The work presented in this report was carried out in collaboration with a thesis partner. The research objectives, approach and all major decisions were decided jointly throughout the project. While the individual tasks were divided based on each student's area of focus, the work remained closely integrated. The author mainly focused on the Model-Based Systems Engineering area of research while the thesis partner mainly focused on tool integration in the design optimization environment. Constant collaboration and discussions kept the work between the two areas tightly interconnected throughout the project to maintain the common objective of creating a workflow connecting the MBSE and MDO environments.

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

3D	3 Dimensional
AI	Artificial Intelligence
AMOP	Adaptive Metamodel of Optimal Prognosis
API	Application Programming Interface
ARCADIA	ARChitecture Analysis and Design Integrated Approach
BC	Boundary Conditions
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CFD	Computational Fluid Dynamics
DRM	Design Research Methodology
DS-I	Descriptive Study 1
DS-II	Descriptive Study 2
DSE	Design Space Exploration
EPBS	End Product Breakdown Structure
FEA	Finite Element Analysis
FEM	Finite Element Method
GAS	GKN Aerospace Sweden
GKN	Guest, Keen and Nettlefolds
INCOSE	International Council Of Systems Engineering
LA	Logical Architecture
LAB	Logical Architecture Blank
MBD	Multi Body Dynamics
MBSE	Model-Based Systems Engineering
MDAO	Multidisciplinary Design and Analysis Optimization
MDO	Multidisciplinary Design Optimization
MATLAB	MATrix LABoratory
OA	Operational Analysis
OAB	Operational Architecture Blank
OEM	Original Equipment Management
P4C	Python4Capella
PA	Physical Architecture
PAB	Physical Architecture Blank

PB	Product Breakdown
PLM	Product Lifecycle Management
PVG	Property Value Groups
RC	Research Clarification
RO	Research Objective
RQ	Research Question
SA	System Analysis
SAB	System Architecture Blank
SE	Systems Engineering
SMW	System Modeling Workbench
SQL	Structured Query Language
SSOT	Singe Source Of Truth
SysML	Systems Modeling Language
TC	Teamcenter
UI	User Interface
UML	Unified Modeling Language
XML	eXtensible Markup Language

Nomenclature

MDO elements

$\mathbf{x}$	Vector of design variables
$f(\mathbf{x})$	Objective function
c_1, c_2	Design and response constraints

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

The competitive nature of the aerospace industry has for a long time been the source of constant innovation in order to enhance performance, efficiency and safety of aircrafts. On top of that, humans have since the mid-20th century been widely recognized as being a dominant force in shaping the earth's ecosystems, atmosphere and geology. This realization resulted in an increased awareness on sustainability both from the public and regulatory agencies, and with the aerospace industry being responsible for 3.5% of greenhouse gas emissions in 2024, the current demand for novel aerospace technologies is unprecedented [1, 2, 3].

Factors such as long lifecycles, high system integration and having multiple stakeholders result in a high complexity in the systems that are currently employed in the aerospace industry [4]. An existing demand for shorter lead times is coupled with the need for new and innovative solutions which risks increasing the complexity even further. This creates the need for a more profound understanding of the developed systems and their underlying interactions.

Model-based systems engineering (MBSE) is an alternative to using scattered documents as a system representation and is getting more recognized in industry. MBSE is the formalized application of modelling to support system development and representation. MBSE aims to manage increasing system complexity and create a single source of truth (SSOT) for a system. While the actual benefits of MBSE are still under debate, claimed benefits include better communication, reduced rework-risk, reduced cycle time and improved ability to manage complexity [5, 6].

In order to explore new aircraft concepts or compare traditional designs of an aircraft or its subsystem, multidisciplinary design optimization (MDO) is commonly used, sometimes even as early as in the conceptual design phase [7]. The main benefit of implementing the MDO methodology is that it is designed to be able to handle systems where the overall performance is not only based on the performance of the individual constituents, but also the interactions between them which is common in aerospace-related applications. To receive valid results, the most important thing when implementing MDO is to make sure that the analysis tools and optimization software is set up in unison with the problem formulation [8], failure to do so can yield both irrelevant and incorrect data. While MDO provides robust optimization methods, ensuring a clear link to the problem formulation can be hard as the method lacks an integrated way to handle the system rationale.

1.2 Problem Statement

The demand for innovative solutions is ever increasing. In order for industries to keep up with the market, engineering teams need to increase efficiency and accuracy during system development to reduce cost and time spent. For this to be possible, the engineers need a better way to understand the system and more efficiently draw the connection from the system requirements to the final solution.

MDO methodology is used to explore the design space by optimizing and comparing different design alternatives. While it provides strong numerical methods to create relations between inputs and outputs, it lacks integrated ways to verify the validity of the numbers and ensure their consistency to the problem formulation.

MBSE tools have been developed to represent the rationale behind the translation from stake holder needs and system requirements into the available design space. These tools, however, do not possess the ability to perform optimization studies independently.

Hence there currently exists no singular place or standardized way where the rationale behind a system can be directly connected to the MDO study or in a later stage represented in affiliation to the MDO results. Not having available the rationale behind design parameters, objectives and constraints results in a worse system understanding when performing MDO studies. Additionally, not being able to represent the system architecture and connection to the MDO studies also makes retrieving system information less efficient and intuitive.

1.3 Research questions

Based on the identified gaps in the current methods, the thesis is dedicated to exploring the following questions:

- **RQ1:** In what way can MBSE support the DSE of jet engine sub systems by creating a connection from the rationale to the system parameters in the MDO study?
- **RQ2:** What data structures and interfaces are required to connect an MBSE model to parametric CAD and simulation tools in an MDO workflow?
- **RQ3:** How can MDO data be stored and accessed in affiliation to its corresponding system model?

1.4 Research objectives

To understand the impact of MBSE on the modern system engineering approach, the following research objectives has been defined for the thesis:

- **RO1:** Derive a method to map digital system models directly to input parameters for MDO.
- **RO2:** Derive a method to store and make MDO data accessible in direct connection to the related digital system model.

1.5 Success Criteria

In order to evaluate the method for integrating MBSE into the MDO workflow, three success criteria were created. These criteria reflect the key challenges in the current workflow that were identified through a literature and interview study:

- **Traceability:** A promised advantage with MBSE is the ability to maintain relationships between requirements and final architecture and if this trace could be extended digitally into the DSE, it would enable a more consistent connection from the system rationale to the optimization process to support verification that the final design decisions are connected to the system requirements.
- **Automation:** Manual transfer of information between engineering tools is both time consuming and introduces risk for errors. The proposed method should therefor automate the transfer of information between MBSE and MDO tools wherever possible.
- **Results representation and reuse ability:** One of the main benefits with MBSE and having all information in an SSOT is reuse ability of the information. The proposed method should therefor allow for the possibility to access the system rationale and the MDO results in connection to each other to promote a quicker adoption of old knowledge in future projects.

1.6 Scope and delimitations

This study was conducted at the GKN Aerospace Sweden (GAS) site in Trollhättan during the second semester of the school year 2025-2026. The study focuses on exploring the interest areas of MBSE-driven MDO that aligns with the roadmap and experiences of the employees at GKN Aerospace. Any methods tested are demonstrated on the relevant tools used at GAS.

This study is limited to subcomponents of jet engines which means that it does not address large-scale systems comprised of interacting subsystems and the studied systems are kept in a single discipline. The interview study conducted to capture the current state is kept at seven participants which could make quantitative conclusions insufficiently supported. The method development and testing are kept to a single software for each domain (SMW, NX, ANSYS, OptiSLang) which could limit the identified solutions and also results in method demonstrations being kept to these specific softwares.

1.7 Report Outline

The remainder of this thesis will be organized in this following way. Chapter 2 presents the theoretical background relevant to the thesis, introducing Model-Based System Engineering (MBSE) and Multidisciplinary Design Optimization (MDO), and presenting their commonly used tools and methods as well as their relevancy in industry. Chapter 3 presents the research methodology that was employed during the development of the proposed method. Chapter 4 Presents results from the literature and interview study followed by showing the proposed methods and the resulting workflow on a simplified and an industrial use case. Chapter 5 provides a discussion of our employed research methodology regarding its effectiveness and accuracy and a discussion of the developed method based on the research questions and how well it fulfilled its success criteria. Chapter 6 summarizes the main conclusions from the thesis. Chapter 7 proposes potential interest areas for future work, both expanding on the proposed method and also discussing what to keep in mind when moving forward with MBSE in general.

2

Theory

This section will explore the foundational concepts that lay the groundwork for an MBSE-driven MDO. This section will establish a concept of systems engineering (SE) in general and give a view of how MBSE and MDO have evolved as methods to support different areas in this methodology. The global trends and current best practices will be presented which laid the basis for the approach and focus of this work. At last, a description of the utilized softwares and their purpose will be presented to provide understanding of how they are integrated in the explored methods.

2.1 What is a System

The increase in product complexity creates the need to understand how components interact and affect each other, this is captured through the concept of a system. While the definition of a system varies to some extent, the original concept of a system is often regarded as "a whole consisting of interacting parts" and the International Council of Systems Engineering (INCOSE) defines a system in the following way:

"A system is an arrangement of parts or elements that together exhibit behavior or meaning that the individual constituents do not"

A simple coffee mug all the way to a turbofan engine can be examples of systems, where the latter also is composed of several subsystems such as the compressor, combustor and turbine, which even they can be decomposed into additional layers of sub systems and components. If a system is designed or adapted to interact with an anticipated environment, then it is called an engineered system [5]. An inherent trait of a system is that their parts are interdependent on each other, and interactions are woven together which makes the complexity of systems increase rapidly with the number of constituents and interactions [9].

2.2 Systems Engineering

In order to handle complex systems, systems thinking must be applied. Systems thinking can be interpreted as always seeing the big picture and it provides the foundational way of thinking when conceptualizing systems and has two main focus points [10]:

- The success of the entire system is the most important thing.
- Often the interactions between parts are more critical than the parts themselves.

Systems engineering (SE) is a holistic approach to enable the successful realization, use and retirement of engineering systems to achieve a defined goal [10]. It provides a methodology behind translating a problem formulation into a system to solve it. This approach is enabled by systems thinking and aims to consider the systems entire life cycle for optimal implementation. A common framework that is used to visualize a SE approach to developing a system is the V-model which can be seen in figure2.1. In the V-model, the system under development is depicted in a view where it is split into multiple levels. The top level represents the context in which the system operates. The bottom level corresponds to the components that can be obtained and the levels in between depicts the system at different levels of abstraction. When following the V-model, the workflow generally moves from left to right. The left leg represents the top-down development of the system, and the right leg represents a bottom-up integration. The different vertical levels correspond between the two legs and provide an incremental way to validate and verify the system integration to the original system context all the way from component testing to user acceptance [11].

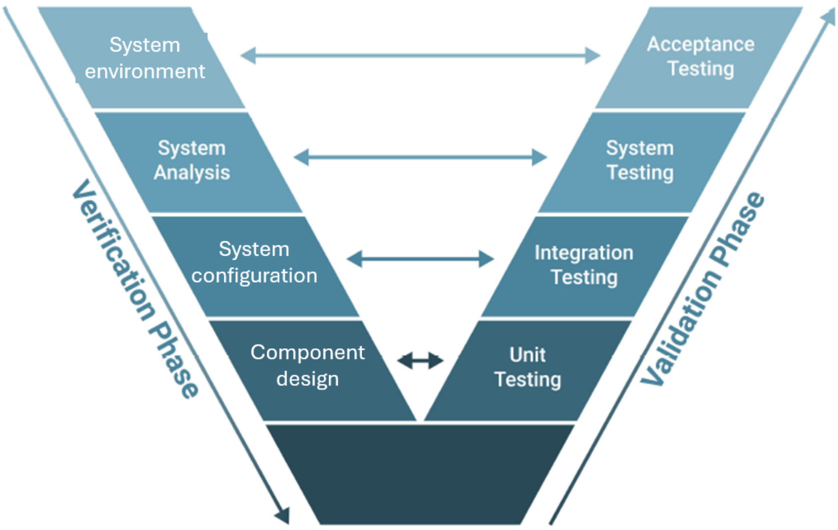


Figure 2.1: V-model

2.3 MBSE

Traditionally, systems engineering has widely been implemented through a document based approach. A document based approach implies that system information such as capabilities, functions and interfaces are captured and communicated through informal diagrams, documents and textual descriptions. However, for several years, the adoption of MBSE as an alternate approach has been growing in industry, MBSE can be interpreted at high level as SE with a computer and a database. In an MBSE based approach, a central descriptive system model is created that represents the information about the system under development in the shape of formal diagrams. This central model is often referred to as the single source of truth (SSOT) and through its several purpose-specific diagrams, it represents the rationale behind the design decisions that are being made during development of the system. The successful implementation of MBSE and a system model requires a sound understanding of the three core pillars of MBSE:

- **Method:** Describes *what* should be modeled as well as *when* and *how* is should be done.
- **Language:** Defines what syntax, semantics and model elements should be used.
- **Tool:** The environment used to create, manage and store the system models.

The increased interest in MBSE stems from its potential to manage complexity and facilitate adaptation in design or system requirements through a formal and dynamic modeling approach. Due to its relatively novel state and lack of know-how, MBSE still remains a large research field regarding what actual value it can provide to SE [6, 5, 12].

2.3.1 SysML to Represent System Models

SysML is a general-purpose modeling language that was developed by evolving a previous modeling language, UML. In SysML, the system under development is represented through the utilization of nine purpose-specific diagrams. There is no particular method that is associated with this language and it can be used from several different tools. Due to the language being general purpose and covering a lot of different applications, SysML has come to be a root from where several other languages take inspiration [23].

2.3.2 ARCADIA to Manage a System Model

One common method that is used to consolidate MBSE is named ARCADIA. ARCADIA describes a structured methodology to manage requirements and constraints in order to define and validate the architecture of complex systems [13]. It is formed to accommodate collaborative work between stakeholders throughout most of the SE phases by providing common system information among relevant stakeholders and allowing for constant iterations in the design phase. The main principles of the ARCADIA methodology include:

- All engineering stakeholders share a common language, methodology and live description of the system and its needs.

- Each system discipline such as cost, performance and safety is officially represented in "viewpoints" from where the system architecture can be verified against the requirements.

- The verification terms and rules for the system architecture are established as early as possible.

The methodology that ARCADIA describes is in strong alignment with the V-model explained in section 2.2 and splits the system design procedures into five different levels in a top-down approach as described below:

- **Operational analysis (OA):** The goal in the OA layer is to identify and understand the needs and objectives of the stakeholders that are going to interact with the system. The systems operational environment is characterized by describing external actors with their interactions and defining the missions the system is supposed to fulfill. This layer completely disregards defining anything about the system itself.

- **System analysis (SA):** The SA introduces the system as a black box into the created environment and identifies what capabilities and functions the system must facilitate to ensure the operational needs are met. It focuses on describing what the system has to do and on identifying the system boundary i.e. how external entities interact with the system.

- **Logical architecture (LA):** The LA layer serves the purpose of describing how the system will work to achieve its expected outcome. The system is treated as a white box where system functions are split into sub functions allocated to different logical components that are introduced and represent architectural pattern of the system. This layer is still independent of what technology may implement the functions.

- **Physical architecture (PA):** The PA layer serves a similar purpose to the LA layer, apart from that it introduces the actual physical components that will be used to form the architecture, hence, it shows *how* the system will be built.

- **End product breakdown structure (EPBS):** The EPBS defines the conditions that each component must fulfill to comply with the constraints and design design choices established in the previous architectural layers.

The first two layers are designed to capture the needs of the system under development and the lower three layers describe the solution [13].

2.3.3 Capella as a Tool for the ARCADIA Methodology

Experience shows that it is difficult to get started with MBSE when you originate from either a tool or a language since this means that you are missing the methodology and results in aimlessness. Due to this reason, Thales in 2007 started developing a method which came to be the ARCADIA method described above, a method which implicitly also defined a language. As an outcome of this, the tool developed to accommodate ARCADIA automatically implements both a method and a language, this tool is known as Capella. Capella is an open source software that implements ARCADIA as both the method and the language for the system model. The language that resulted from the ARCADIA methodology is highly related to SysML but with some additional and different-purpose diagrams. These include diagrams for *architecture, dataflow, sequences, functional breakdown, modes and states* and more. The environment that Capella inhibits has a similar ergonomic to PowerPoint and Excel to be intuitive even for novel users [13].

Siemens has integrated Capella into its PLM tool called Teamcenter and calls this software Systems Modeling Workbench or SMW. The thought behind this is that SMW will be used to create and manage the system model and through the PLM structure, the model gets connected to requirements, life cycle data and other artefacts.

2.3.4 Python4Capella

Python4Capella (P4C) is an open source plugin that works for both Capella and SMW. P4C provides SMW with an API which allows the user to create scripts for SMW directly in the tool itself which can support data import and export and several other use cases [24].

2.3.5 MBSE Relevancy in Industry

The aerospace industry is under a lot of pressure to develop novel technology and constantly optimize their systems to keep up with competitors while they simultaneously have to adapt designs to changing environmental goals [7]. Systems that are being sold to customers are called products and in order to stay competitive, an efficient and accurate product development process needs to be established. A successful product development process is characterized by several factors that include product quality, product cost, development time and development cost and there are several challenges that arise when trying to optimize these factors. These challenges include assessing trade-offs in the design, managing dynamic customer needs or regulations, choosing the optimal components and managing everything while being under a short time frame [14]. The immense complexity of the systems being developed in the aerospace industry compounds these challenges and further increases the industrial need to have a structured and robust methodology in place for system development. This makes the exploration of SMW or other MBSE tools a big interest area in terms of research and trying out new workflows.

2.4 MDO

When developing novel systems or products, a design space exploration (DSE) is commonly performed. The purpose of this is to perform a trade study (i.e. analyze trade offs relevant in the system design) in a visual manner in order to build an understanding of how design variations impact the system performance [20]. MDO methods are then often implemented to generate the performance data vital to make informed decisions. MDO approaches rely on numerical optimization techniques to solve complex system design problems involving multiple or overlapping disciplines. MDO methods reformulate multidisciplinary problems into either a single optimization problem or a set of optimization problems that can be solved with established numerical optimization algorithms [19]. Application of MDO methods commonly appear within the aerospace industry due to the high coupling of disciplines [7]. In industrial settings, MDO is usually incorporated by integrating discipline specific analysis tools, such as structural, aerodynamic and thermal models to evaluate system behavior which gives the data for analysis.

2.4.1 Basic Principles of MDO Problem Formulation

There exists several MDO methods but they are all fundamentally based on design variables, objectives and constraints when setting up a problem. These elements can be described as follows:

- **Design variable:** Design variables include any adjustable parameters or specifications that are always under control of the optimizer. These can for example include a geometric length or the type of material.
- **Objectives:** An objective is a specific metric or function that the optimizer is supposed to either minimize or maximize. Examples of objectives include mass, fuel consumption and deformation.
- **Constraints:** A constraint is a relation that needs to be satisfies in order for the system to be regarded as feasible. Constraints can apply both on the design variables and on the performance output of the system.

While there are several different methods to perform the optimization, all of them have their own strengths and weaknesses but all of they follow the same principle:

$$\text{minimize:} \quad f(\mathbf{x}) \tag{2.1}$$

$$\text{w.r.t.:} \quad \mathbf{x} \tag{2.2}$$

$$\text{s.t.:} \quad c_1(\mathbf{x}) \leq 0 \quad \text{and} \quad c_2(\mathbf{x}) = 0 \tag{2.3}$$

where $\mathbf{x}$ represents a vector of design variables, $f(\mathbf{x})$ represents our objective functions i.e. the quantity we want to optimize and $c_1(\mathbf{x})$ and $c_2(\mathbf{x})$ represent inequality and equality constraints i.e. conditions that must be satisfied for the solution to be feasible. [8, 19]

2.5 Additional Tools

2.5.1 NX CAD

NX CAD is a CAE software provided by Siemens that is mainly used for 3D geometric modeling. NX provides the ability for these models to be created using parametric modeling i.e. geometric dimensions and positions are driven by underlying parameters and constraints, enabling efficient modification and design iteration. The 3D model serves as the digital twin for the real world physical product and is widely used in the aerospace industry. The properties the model offers support design processes by enabling illustrations, improved feasibility understanding and digital data sharing [15]. These models also serve as a foundation for several digital analysis methods used to evaluate performance. Such methods include finite element method (FEM), computational fluid dynamics (CFD) and multi body dynamics (MBD).

2.5.2 Ansys

Ansys, Inc offers a wide variety of simulation and analysis tools for evaluating and optimizing engineering designs in a way where they are all integrated into the same environment.

Through Ansys Mechanical, an integrated platform for finite element analysis (FEA) is provided. Mechanical offers a wide range of tools for analyzing structural behavior, including stress distribution, deformation, fatigue and thermal influence. Additionally, Mechanical provides the possibility for scripting in order to enable automation of workflows [17]. Through optiSLang, Ansys provides a base to explore design variation by connecting different tools and applying optimization methods on simulation data from connected tools. These results help finding optimal designs by incorporating methods such as trade studies and sensitivity analyses [18]. Ansys Workbench provides a platform that acts as a central location where you can organize and access several different simulation and analysis tools. OptiSLang provides an optimized way to transfer data between different tools and creates a fluent way to integrate several tools with different purposes into the same workflow [16].

2.6 Design Research Methodology

Design Research methodology (DRM) is a set of guidelines and methods formulated to make design research more efficient. DRM is constructed of a methodology framework with four main stages [25], which in this thesis will be applied to develop a method for integrating MBSE with MDO:

- **Research clarification (RC):** Clarify the current understanding of the research area as well as the aim and plan of the research.
- **Descriptive Study I (DS-I):**Identify the currently known phenomena and understand factors that influence the success of the proposed method.
- **Prescriptive Study (PS):** Develop and formalize the method intended to improve the design process.
- **Descriptive Study II (DS-II):** Evaluate the applicability and usefulness of the proposed method.

3

Methodology

This chapter explains the methodology used to formulate and address the research questions and objectives presented in chapters 1.3 and 1.4. The use of DRM provides a structured and systematic approach for analyzing the current problem, developing solutions and evaluating the results. Accordingly, this thesis applies DRM to investigate and develop a method for leveraging MBSE in driving MDO studies. The research is hence structured around four main steps; *Research Classification, Descriptive Study I, Prescriptive Study* and *Descriptive Study II.* Though DRM presents a very structured approach, these steps are interconnected and several of the early stages were implemented iteratively with the new knowledge and insights obtained.

3.1 Research Clarification and Descriptive Study I

The RC was conducted in high connection with the DS-I. A broad subject with an open ended task was provided, hence the scope and formulation of research questions and research objectives was redefined iteratively based on the level of understanding of the subject.

A literature review was conducted with the aim to establish a foundational understanding about MBSE and MDO, and their respective inclusion in the system development process. The literature review was also aimed at creating an understanding about the underlying principles and physics of a general gas turbine engine. This layed the groundwork for creating a basic understanding of how the MBSE and MDO methods later could be applied in a use case. The chosen sources included basic videos explaining foundational concepts, books and research papers or articles covering different applications of the methods of interest. The sources used were saved in a register in excel.

The insights from the literature study about the two methods were captured in documents, analogue and digital, and analyzed together to realize the different roles the methods play and understand the gap between them. These insights were used to initiate more problem specific research questions to base future research on. The literature study continued, now with the aim of understanding the state of the art of the research gap and identifying any relevant methods implemented in industry or studied in academia that may be relevant to the formed research questions. This stage of the literature review mainly targeted research articles regarding application of any of the methods, stand alone or in unison on an actual project, both aerospace related and not. The results and sources were archived in the same way as previously.

At the same time, an interview study on six employees was conducted at GKN Aerospace with the purpose of understanding the following things:

- The state of the art at GKN Aerospace regarding the utilization of MBSE to drive MDO studies.
- How MDO studies are currently performed at GKN Aerospace, and their connection to the rest of the SE process.
- What gaps exist currently at GKN Aerospace and what improvements do employees expect with a successful implementation of MBSE into the MDO process?
- What are the impressions and ambitions of MBSE at GKN on employee and company level?
- What are the current challenges with MBSE and what lessons have been learnt so far about best practice?

The exact questions that were asked varied depending on the background of each interviewee and the format was also changed along with the increase in interview experience.

A META-study was performed based on the answers from the interview study and the results were formulated into qualitative and quantitative data. These results together with the information from the literature study was utilized to complete the mapping of the problem and understand the current situation including what has been done and what is desired. The results from these studies were used as a reference to finalize research questions, research objectives and success criteria for the thesis.

3.2 Prescriptive Study

The prescriptive study was split into four steps. These included learning softwares, creating a flow chart to map the method to be tested, implementing the method on a simplified use case and lastly implementing the method on an industrial use case. As the understanding of the problem and related softwares increase during implementation, the view of the best integration method also evolved over time leading to continuous enhancement of the developed method.

3.2.1 Conceptual Development of the Integration Method

Software training was carried out on the selected tools with the aim of understanding their interfaces, data structures, capabilities and limitations. By utilizing the knowledge about the softwares' roles in SE from the descriptive study I together with the understanding about the softwares, a mapping was conceptualized around how the different softwares inputs and outputs were related in an SE context. Studies were made regarding how the data structures within each software was built up and how the data could be transferred between tools by the utilization of python. The current perception of the softwares' different roles, data structures and capabilities was used to create a flow chart representing how the information should be communicated between the different softwares in a visual way, there by creating a visual representation of the method. The real-time perception and understanding of the softwares evolved throughout method testing and implementation which led to the flowchart being continuously updated according the newly gained insights.

3.2.2 Simplified Use Case

A simple design optimization problem was formulated in a way where it included all types of design optimization elements explained in chapter 2.4.1. While still including all necessary elements for MDO, the problem was created with the ambition to be kept as simple as possible and in spirit to make the method application to the problem as straight forward as possible. The design problem was used to test and implement the method in a practical sense.

A system model was set up in the SMW software in a way where it represented the system subject to the optimization problem and as well as all the data regarding the elements that were to be exported to the MDO softwares. Parallel to this, a CAD geometry was created in NX that included the same design variables as the system model.

Additionally, an Ansys Mechanical environment, connected to the NX CAD was being setup for simulation and an optiSLang project was created to prepare the optimization environment. This environment was setup in a way that it was entirely dependent on the specific elements created in the system model, meaning that it should be able to read this information and that the inclusion of these elements should be necessary for the ability of the analysis to run.

Once all tools were instantiated to manage the same conceptual data, a python script for reading, transforming and writing data between SMW and the MDO tools was created. The python script was guided by an initial flow chart describing data flow and by evaluating different strengths and weaknesses, the new acquired knowledge was used to update the flow chart for a more efficient method. By iterating this process, a final flow chart representing the data flows in the method was developed and the python script correspondingly finalized to manage the specific data structures and tool connections.

3.2.3 Industrial Use Case

A new problem was formulated to represent an industrial design task. Now, the already defined method was implemented in a similar way as in the simplified case. For this use case, a more overarching system than in the simplified use case was modeled in SMW with the goal of representing how the system can be broken down from a black box to an actual configuration of components and with that, how functions and requirements are broken down as well as how MDO elements are represented for analysis of specific components. This model aimed to keep the simplifications minimal to visualize how the system models and proposed method scaled to a more comprehensive system breakdown.

3.3 Descriptive study II

Once the SMW models had been created and the method had been tested for both use cases, a second interview study was conducted in order to generate an independent evaluation of the results. Two interviews were held, involving one and two participants external to the project so far. The interviews started by a presentation of the thesis objectives and success criteria, followed by presenting the method and later on a detailed walkthrough of the SMW model that was created for the industrial use case. Once the proposed method and used models were shown, open discussions were held with the participants with the aim of evaluating weaknesses, strength and relevancy of the method and later on also discuss the realism of the system model that was created in the industrial use case.

4

Results

With implementation of DRM, this thesis identified research gaps regarding the connection of MBSE to MDO tools and explored different possibilities in order to connect the two areas. Thereafter, an evaluation of the proposed method for MBSE driven MDO was performed to assess its impact. This chapter presents the developed method as well as the results from the studies that drove the problem formulation and the methods' subsequent evaluation study.

4.1 Descriptive Study I

The initial purpose of the DS-I aimed to give the RC with questions and objectives presented in section 1.3 and 1.4. The later part of the DS-I was focused on understanding applications and state of the art of our MBSE and MDO tools and connection, which will be the focus for the results presented here. These results served as the foundational motivation and reference for the method development.

4.1.1 Literature Review

MBSE methods are highly effective at representing the design baseline of a system by capturing the entire design rationale, ranging from stakeholders to the system architecture. They can intuitively show where system functions come from and represent how they are allocated and structured in the system, hence showing both the purpose of the functions and the reasoning behind the systems structural form. This way, MBSE establishes and constrains the design space. However, their capabilities for exploring alternate or multiple design concepts simultaneously are limited and SMW for instance, does not have any built in way at all for variety management [21].

MDO tools on the other hand is often utilized in the analytical a part of a DSE, which is a process designed to compare different design solutions against each other [20]. However, the MDO tools, after defining governing equations, simulation models and the included MDO elements, work by examining numerical relations. They therefor lack the ability to capture the rationale of *why is the model set up this way?* and *why do we test these specific system configurations?*.

Hence when comparing MBSE and MDO, it is evident that:

- Implementation of both MBSE and MDO is relevant when designing a system or product
- The strengths and weaknesses of MBSE and MDO are highly complementary.

There are no tools available right now that can handle both sides but some efforts have been made in order to integrate the available MBSE and MDO tools. Most of them propose workflows that follow data driven coupling, meaning that that primary design variables and other MDO elements are defined in the MBSE model and then transferred to the MDO environment. Louis Timperley et al, [21], proposes a workflow where system parameters and properties are defined in MBSE and supplied directly to the MDAO process for analysis. Note that this acts as a data-backbone for the system and environment but does not directly create the design space itself. This sets up simulation tools by extracting rationalzed parameters but does not entirely structure the optimization problem directly from MBSE export. As for the optimization problem, the workflow contextualizes constraints in MBSE but these have to be manually formulated for the optimization process. As for objectives they are kept out of the MBSE model.

Additional efforts have been made to further improve the connection. One paper, [22], proposes a way to define primary parameters (objectives, constraints and design variables) in the MBSE tool and then import them into the optimization tool. Here, design variables created with an initial value that can be directly used. Constraints are being contextually defined in MBSE together with the actual numerical implementation for MDO and is automatically transferred. Same for objective functions. MATLAB is used to integrate the multidisciplinary analysis model and when it is run, it generates the variable space for each design variable and outputs primary parameters to the optimization component. Once the optimization analysis has been performed, the resulting configuration of design variables, is imported back into the system model to update it with a specific configuration.

4.1.2 Interview Study

The results from the interview study at GKN Aerospace were split into a qualitative and a quantitative part, the results were kept independent of the interviewees position at the company.

4.2 Qualitative results

The qualitative findings from the interviews are organized in five themes:

4.2.1 Current implementation of MDO studies and the general SE process

GKN has been a manufacturer mainly on component and sub-system level for a long time. Customers generally provide requirements, these can be either on the specific component level or on a more universal level. In the latter case, the requirements have split between the different discipline teams and broken down to fit on a component level. Currently a document based approach is mainly being utilized to describe the system information such as requirements and interfaces, and CAD models are used as the main form of communication of the system together with word of mouth. The system information including the CAD models is stored in Teamcenter (TC) where it can be accessed by anyone who needs it. TC does not however communicate efficiently with and CAE tools apart from NX CAD, simulations and analyses for the DSE have to be set up manually and any updates to the information also have to be manually inserted.

4.2.2 Impressions and ambitions with MBSE

The ambitions with MBSE at a company level is to have a new way of implementing systems engineering by 2029. This is partly since GKN is moving toward becoming an OEM on a larger scale and being in a state where they are involved alot earlier in the product development process. MBSE should be used all the way from the conceptual development stage to the end of life of a system. MBSE should represent the system information in one single, structured and always available place, an SSOT. This should aid the system understanding and provide a robust way to define the design space and, capture and justify design decisions. MBSE tools should also connect to other CAE tools. From this should follow:

- Traceability exists from design decisions back to requirements.
- Speed and flexibility is improved throughout the entire SE process
- Automated setup of other tools.
- Less risk for miscommunication.
- Storage of knowledge for similar problems in the future is available.
- Easier communication with customers regarding design decisions is possible.

4.2.3 MBSE state of the art at GKN

MBSE has started to be implemented at GKN but the method is not mature enough or up to speed yet to provide any value in the early SE stages. This is since although SMW is connected to TC, the connection still makes the system feel disconnected from the requirements, resulting in the impression of a zig-zag workflow that does not accommodate iterative processes very well. On top of this, the SMW - TC connection is not consistent enough at this point and results in system models breaking, hence loss of data is a problem. SMW is being implemented though in order to create system models of already somewhat developed products as a way to capture and store information about a concept for future use. By doing this, the system model will serve as a consistent way of communicating system information.

4.2.4 MBSE current gaps and desired improvements

The gaps related to MBSE at GKN are both internal and external of the SMW software. First of all, SMW doesn't communicate digitally with any other CAE tools, hence there exists no digital thread between the different product models. The information from the system model is instead communicated by word of mouth. As previously mentioned, the zig-zag workflow of SMW - TC makes the system feel disconnected from its requirements. As for SMW itself, it does not support both top-down and bottom-up approaches. SMW doesn't have a way to handle variants either which will be even more crucial when GKN is involved earlier in the design phase of upcoming products.

4.2.5 Challenges with implementing MBSE and lessons learned

MBSE is very promising in what it claims to provide but it requires a large commitment to implement it. While MBSE can help manage complexity in systems, this requires that the user can understand the language which can take a lot of effort to learn. The implementation of MBSE is very front loaded for the users which can seem pointless, especially when you are working with very similar and mature products that feel like they are known by heart. In these cases, people will probably respond with "Why do we need MBSE?".

To avoid unnecessarily heavy front loading, it is important that MBSE is implemented from the ground up and not forced onto either the company or the employees. It has to be implemented in a way where the user almost selfishly wants to use MBSE for their task and the training available should be consistent to ensure a consistent usage of the method in order to minimize inaccuracies in models and make it as straight forward as possible to understand a model even if you didnt build it yourself.

All in all, implementing MBSE into the standard way of working is a large investment, you need resources, such as training programs, method development and money for licensing fees but maybe most importantly, the will to utilize it.

4.3 Quantitative results

In addition to the qualitative review of the interview results, a table representing the frequency for how often some key interest areas for the thesis were reflected.

Theme	No. of respondents (out of six)	%
Currently manual data transfer from SMW to CAE tools	6	100%
Main tools for MBSE and PLM are SMW and TC	6	100%
MBSE is a subject that should be explored further	6	100%
Main benefit (or one of) fo MBSE is traceability	6	100%
MBSE should support DSE	5	83%
MBSE should connect to CAE tools	3	50%
Available TC-SMW connection is suboptimal	4	67%
MBSE should act as the main SSOT during product development	4	67%
Cultural resistance makes it harder to implement	4	67%

Table 4.1: Frequency of key subjects during the interviews.

The quantitative analysis reveals that MBSE is currently at a very immature stage at GKN. There is a substantial interest for it and it has started to be used. The only other software tool in the product development realm that it is integrated with though is TC but this connection is lackluster. There is an interest to move MBSE into the main way of doing product development at GKN and integrate it with other tools but it is held back by resistance to change and the fact that it is a big investment of a still very novel method.

4.4 Prescriptive Study

The prescriptive study consists of two different use cases as mentioned previously. In the simplified use case, the method was developed from scratch and implemented to aid solving a design problem, while in the industrial case, the prepared method was implemented on a more extensive, industry-realistic problem.

4.4.1 Method Framework

Several iterations and testing with software integrations to understand what is possible, yielded the general framework for the final method that can be seen in figure 4.1.

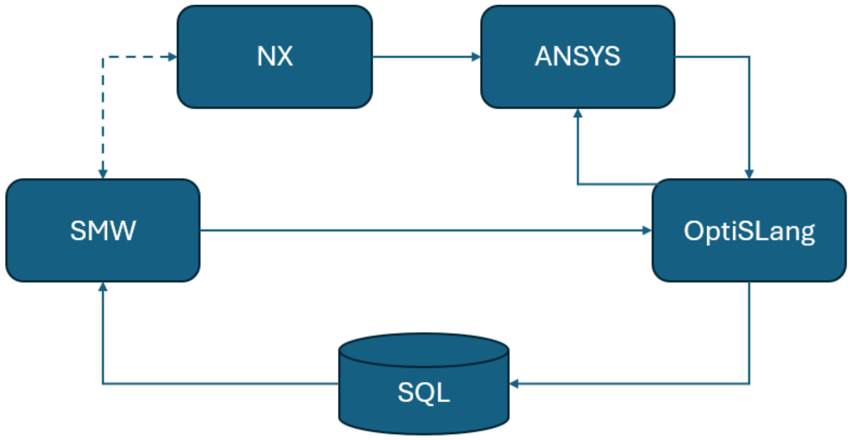


Figure 4.1: Flow of system perception and system data.

We could not identify a way where SMW can be utilized to drive the geometric modeling of a CAD model. Hence, the developed method adheres to the following workflow: An NX CAD model and a system model in SMW is set up in parallel, where communication is accomplished through word of mouth or sharing of visuals. The SMW model captures the requirements through different levels of abstraction of the systems configuration and the 3D model is set up based on this information. By doing this in parallel, the CAD model provides insights into any geometrical constraints that may be overlooked in the system model and these are then fed into SMW manually. Now, the SMW model is set up in a way where all elements in MDO processes are defined, this means that the objective functions, any design constraints, any solution constraints and any design variables are defined as objects within the SMW model. In other words, the entire rationale and logic behind the concept design is captured by working in parallel between NX and SMW. The CAD model will be built in a parameterized way, where any geometric design variables that appear in the system model must be included. Any design variables for the optimization process are tagged, as well as any necessary surface tagging for simulations are set up.

When this is done, an Ansys workbench environment is set up with a Mechanical project. The CAD model is imported to the Ansys Mechanical project where any desired simulation is set up and performed on the geometry. After the simulations, the inputs and desired outputs for MDO processes are marked.

An optiSLang solver takes the desired optimization elements from the simulations. After this, all the MDO elements defined in the SMW model are imported into a sensitivity wizard in optiSLang where they are cross referenced against the inputs from Ansys to make sure the Ansys data is what we actually modeled in the first place. From the same import, optiSLang also recieves the constraints defined in SMW which means that all elements necessary for the optimization are available.

The optiSLang sensitivity wizard then utilizes the solver wizard to run simulations with different input sets and generate the corresponding output. This data is what will be used in the selected optimization model, generating the results for a selected number of samples.

The data from this optimization analysis is saved in csv files and can be used for further analysis or be stored in an SQL database.

4.4.2 Problem Formulation Simplified Use Case

The design problem addressed in the simplified use case is stated in the following way:

What is the minimum number of bolts required in a bolted flange such that the maximum structural deformation under the applied pressure loading remains below a specified threshold, while simultaneously minimizing the total mass of the flange?

A version of the geometry used for the flange can be seen in figure 4.7. The flange will be subject to the following loading:

- **Internal pressure:** 3 bar will be applied on the inside of the part resembling a connected pipe.
- **External pressure:** 1 bar will be applied on the same part.
- **Bolts:** The circular flange has bolt holes periodically spaced that will be treated as a fixed constraint.

Finally, the optimization problem can be formally stated as:

$$\text{minimize:} \quad [\delta_{max}(\mathbf{x}), m(\mathbf{x}), n_{bolts}] \tag{4.1}$$

$$\text{w.r.t.:} \quad \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} \tag{4.2}$$

$$\text{s.t.:} \quad \delta_{max}(\mathbf{x}) \leq 3\mu\text{m} \tag{4.3}$$

Where $\delta_{max}$is the maximum deformation of the flange in its environment, m is the mass of the pipe section n_{bolts} is the number of bolts the flange is secured with and $\mathbf{x}$ represents the design variables that the optimizer can play with.

4.4.3 Method Implementation on Simplified Use Case

Below follows how the proposed method was implemented to manage the design problem in the simplified use case in terms of the storage and transfer of system information.

The created SMW model represents the entire rationale of the flange by implementing OA, SA, LA and PA layers and shows the reason behind the functions the flange needs to accomplish and the environment the flange is exposed to. All necessary

MDO element types were created and connected to either the flange or its allocated functions, depending on what fits the best. The different chosen elements give a visual side and a back end data object that can be parsed. Figure 4.2 shows a part of a SAB, part of the SA-layer, for the flange where it visualizes the load paths appearing for one of the flanges capabilities and also shows how the visual representation of objectives and solution constraints look. The full diagram and all other related diagrams for the flange can be found in appendix A.

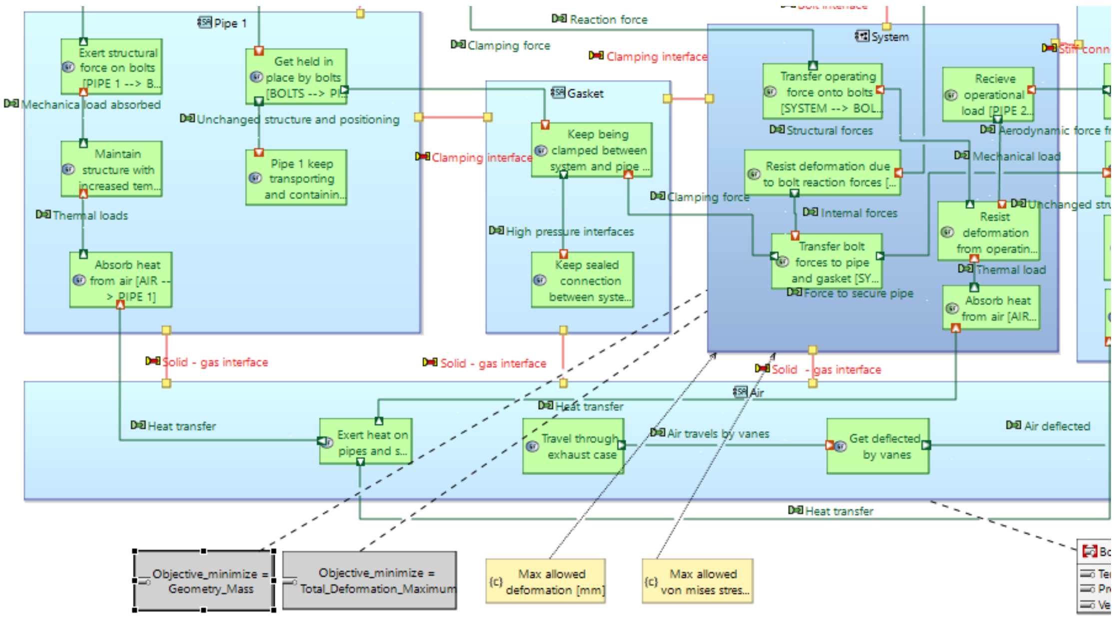
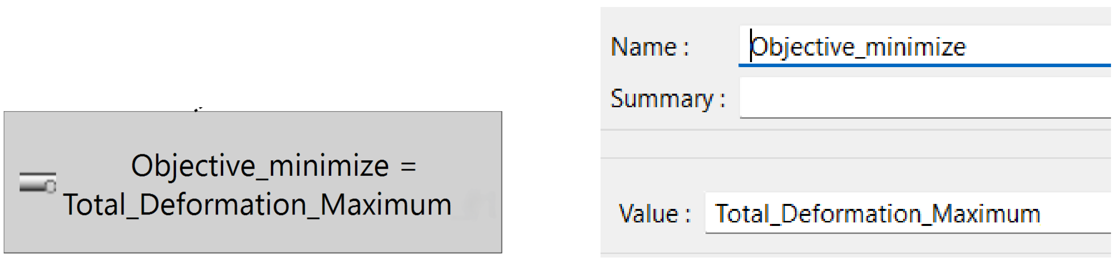


Figure 4.2: SAB diagram showcasing the load paths for one of the flanges (systems) capabilities and showing the visual representation of two MDO elements.

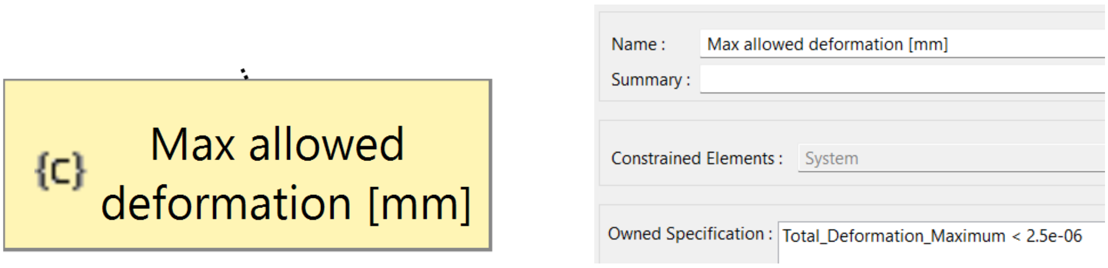
The objectives are represented through the *property* object in SMW which can be seen as the two gray boxes to the left in the figure. Their information is represented in the following way.



(a) (b)

Figure 4.3: Visualization and representation of the deformation objective.

Similarly, constraints are defined through the *constraint* object which are the yellow boxes in figure 4.2. Their information is represented in the following way:



(a) (b)

Figure 4.4: Visualization and representation of the deformation constraint.

The introduction of the design variables and the design constraints was handled in the PA-layer for this use case, where the different parts of the flange (system) has been specified. Figure 4.5 shows part of the physical architecture of the flange, its inherited functions and the representation of some design variables and design constraints. A full version of this diagram can be found in appendix A.

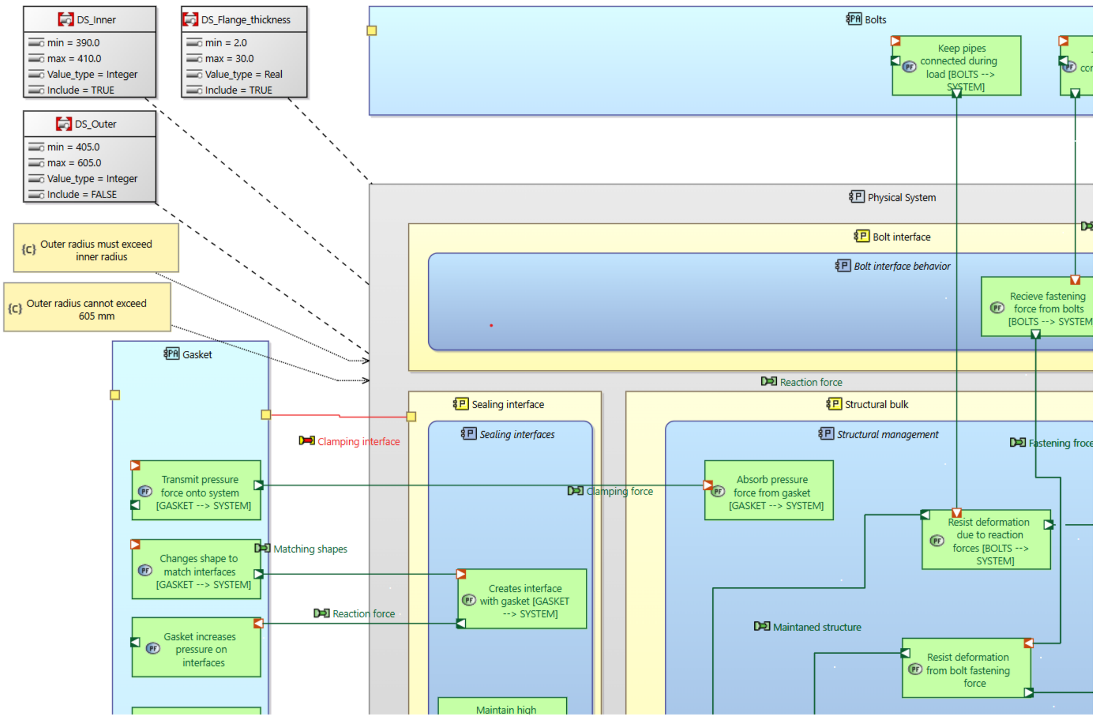


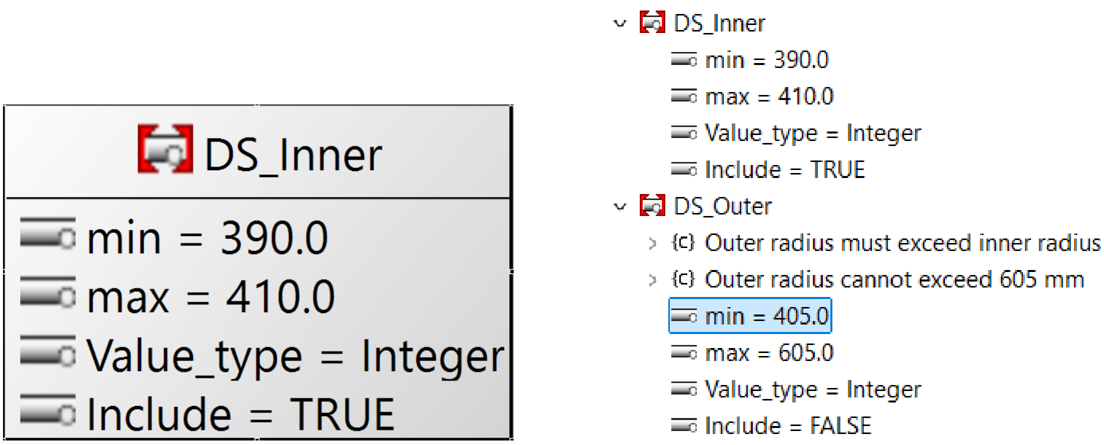
Figure 4.5: PAB diagram showcasing the physical architecture of the flange as well as two types of MDO elements.

The design variables contain more information to read than the objectives or constraints, hence they are represented by Property Value Groups (PVG), in figure 4.5 they can be seen as the gray boxes with a red icon at the top left of the diagram. Each design variable has the following properties inside of them:

- **min:** Lower bound for the design variable to be explored during DSE.

- **max:** Upper bound for the design variable to be explored during DSE.
- **Value_Type:** What type optiSLang will treat the value as during optimization.
- **Include:** Flag for user to choose if this parameter should be treated as a design variable in the DSE.

Figure 4.6 shows the visualization and representation of the design variables in SMW. Each property of the design variable is allocated through the model tree. In 4.6 b, can also be seen some design constraints that apply on a specific design variable. The back end representation of design constraints follow the same principles as the solution constraints.



(a) (b)

Figure 4.6: Visualization and representation of design variables.

The design variables here are tagged with "DS" at the front to later align with their representation from the Ansys simulation that is based on the CAD model.

As mentioned in 4.4.1, the CAD model is created in parallel with the SMW model, they are not digitally connected though. When implementing this workflow, the CAD model needs to be parameterized and any geometric properties that were represented as design variables in SMW need to be named the same way in NX, including the "DS_" tag at the front. Surface tags "NS_" are created for surfaces that are affected by boundary conditions (BC) in the Ansys simulations, to make these changes be automatic. An instance with all the design variables inside their range are used to visualize and export the model to Ansys. Figure 4.7 shows the geometric shape that was studied in the simplified case.

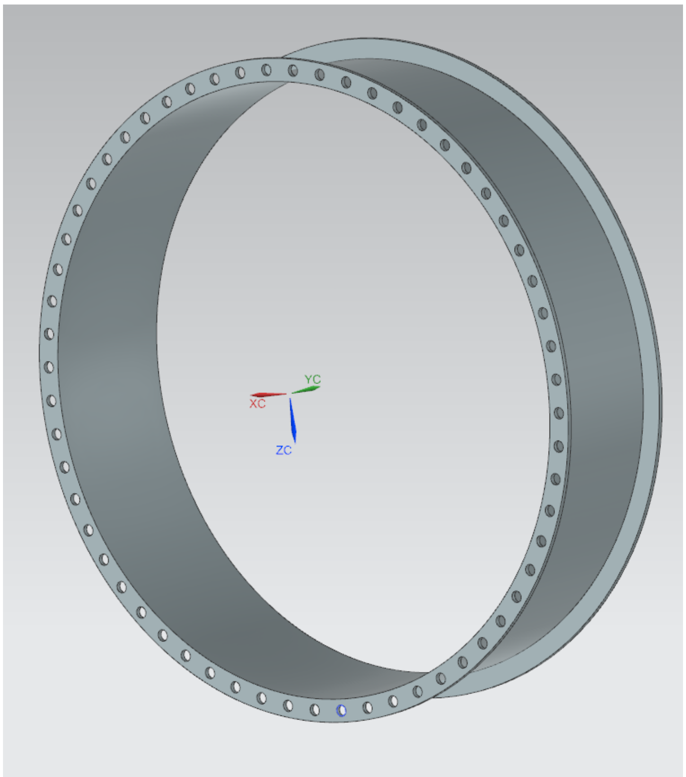


Figure 4.7: CAD model of the flange studies in the simplified use case.

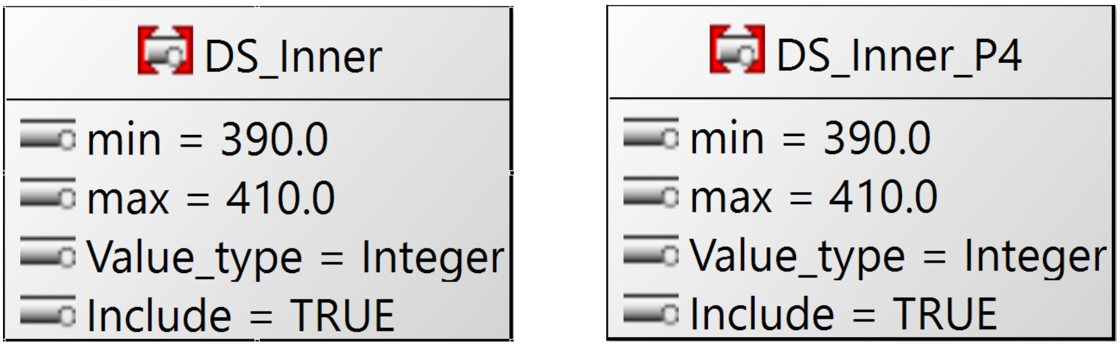
An Ansys project with static structural is created in Ansys workbench where the created CAD model is imported and then simulation boundary conditions i.e. loads and boundary conditions are applied manually. The desired outputs from the simulations have to be selected in the solver. Afterwards, parameters for the optimization project are defined in the workbench based on the inputs and outputs in the simulation. In our case this includes number of bolt holes, inner diameter for the flange (design variables), total maximum deformation and mass (objectives), this is shown in figure 4.8.

Outline of All Parameters

	A	B	C	D
1	ID	Parameter Name	Value	Unit
2	Input Parameters			
3	Static Structural (B1)			
4	P1	DS_N_holes	80	
5	P4	DS_Inner	405	
*	New input parameter	New name	New expression	
7	Output Parameters			
8	Static Structural (B1)			
9	P2	Total Deformation Maximum	2,4842E-06	m
10	P3	NX_parameters_TestCase_Generec_TRS_Model_for_testing - Copy Mass	29,573	kg
11	P5	Equivalent Stress Maximum	7,4364E+07	Pa
*	New output parameter		New expression	
13	Charts			

Figure 4.8: Ansys UI for choosing parameters for the optimizer.

As can be seen in this figure, the input and output parameters get assigned a tag "Px", this will be attached to the name itself when later transferred into optiSLang, hence the user must read these parameter tags and manually edit the names of the related elements in the SMW model to match. This applies to both design variables, objectives and constraints that include any of these design variables or objectives in its logical statement. This change on a design variable will therefor look like this:



(a) Old name (b) New name

Figure 4.9: Design variable in SMW adapted for name matching.

Once the Ansys simulation environment has been setup and connected to the elements for the optimization and these elements names have been corrected in the SMW model, the user has to run a python script that parses the SMW models file which is an XML structure and stores all elements for MDO in csv files.

When the data is stored in csv files, it's time to create an optiSLang project. In the optiSLang environment, a solver wizard (MainCase_08_04 in figure) connects to the Ansys workbench that was just created and a sensitivity wizard (amop in figure) is connected to that solver wizard as visualized in figure 4.10.

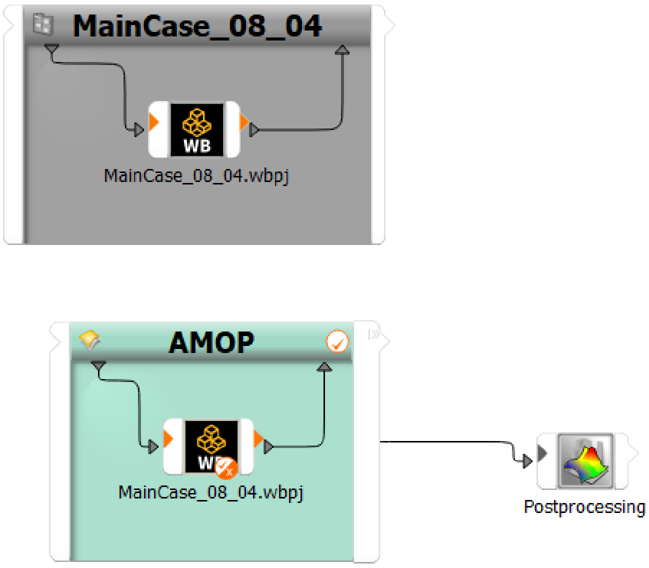


Figure 4.10: OptiSLang process chain.

In the sensitivity wizard, under the *parameter* tab, the user has to upload a csv file with the design variables and their properties which was read from SMW. A result of doing this can be seen in figure 4.11. The inputs from this csv file sets up the design space for the DSE.

	Name	Parameter type	Reference value	Constant	Value type	Resolution	Range	Range plot
1	DS_Inner_P4	Optimization	400	☐	INTEGER	Discrete by value	[390-410];	
2	DS_N_holes_P3	Optimization	45	☐	INTEGER	Discrete by value	[10-80];	

Figure 4.11: Design variables entered to optiSLang from SMW.

Under the criteria tab in the same wizard, a csv file storing the constraints and objectives from SMW need to be imported. These can be seen represented in the *Criteria* section in figure 4.12. The two top boxes, *Parameters* and *Responses* come directly from the Ansys environment, from the inputs and outputs that were defined earlier.

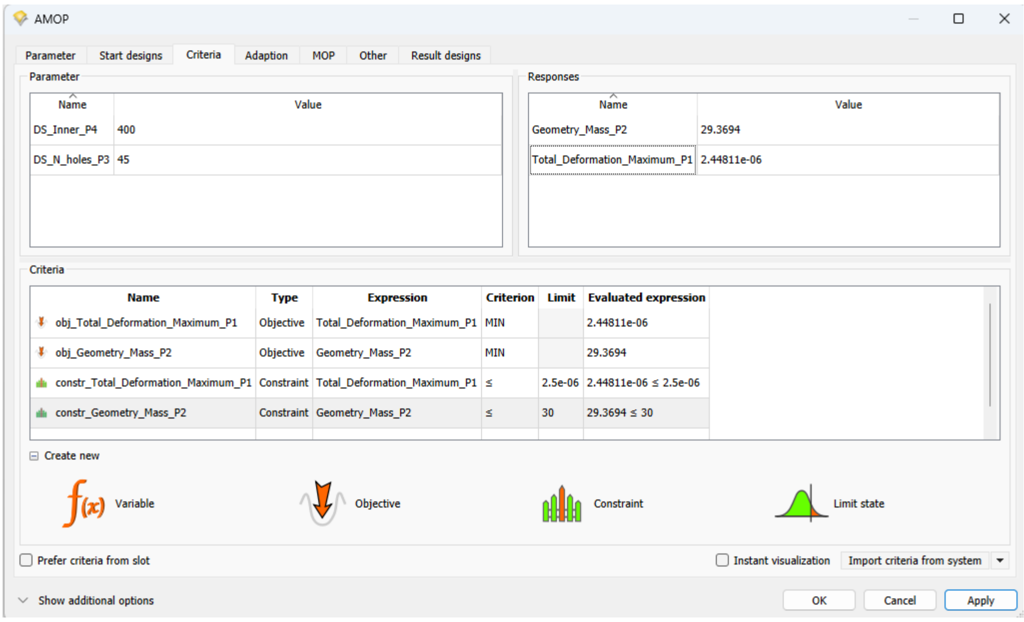


Figure 4.12: Elements from NX and Ansys meets constraints and objectives SMW.

Now, the objectives that were imported from SMW has to match 1:1 with the objectives we got from the Ansys environment and same goes for the design variables, otherwise the optimization process wont be able to run. This acts as a fail safe in case of human error. The constraints were never created anywhere in the NX or Ansys environment, but were just imported directly from SMW.

With this, the MDO environment is now completely set up and after choosing a model for optimization and the desired amount of design points to explore, the sensitivity wizard will each iteration generate a new instance of design variable values, feed them into the solver wizard, update the geometry and run the simulation to then get the related outputs. These relations that are generated between design points and objective values are analyzed to get an understanding about the system behavior.

After the optimization study is done, the data from the DSE can be exported from optiSLang as a csv file and stored in an SQL database for future reference.

In terms of reaching the MDO data in the future in connection to the SMW model, there are some ways this can be done, but which one is the best depends heavily on *what* you want to do with the data. Whatever the user wants to do, the plugin Python4Capella (P4C) can be used with great success. P4C allows for creating executable directly inside the SMW environment. By just utilizing python scripts that are connected to the location where the data is stored, it can be reached and viewed in different ways by just running that python script directly from SMW model where all desired scripts can be accessed. Examples include having scripts to open the related optiSLang project from directly inside of the system model, immediately opening a MDO-data analysis tool with the data loaded for visualization or just accessing a git or database.

4.4.4 Industrial Use Case formulation

The industrial use case aims to study how a turbine blisk can be represented in SMW by including both the breakdown of its functions and requirements and also how SMW from this can represent the MDO elements and rationale for a specific jet engine component. The entire MDO procedure will not be carried out since it follows the same principles as for the simplified case.

4.4.5 Industrial Use Case System Model

This section will present the industrial use case SMW model and how it breaks down the system to represent the MDO elements on a component level. This showcases how the SMW model changes and scales to a more overarching system definition.

Since the turbine blisk is a component of a jet engine, the SMW model is setup to represent the jet engine as the system to be broken down. The jet engine needs to enable several different capabilities and depending on their nature, the operational environment can be represented through several different OAB diagrams. One example can be seen in figure 4.13, the rest can be found in appendix B.

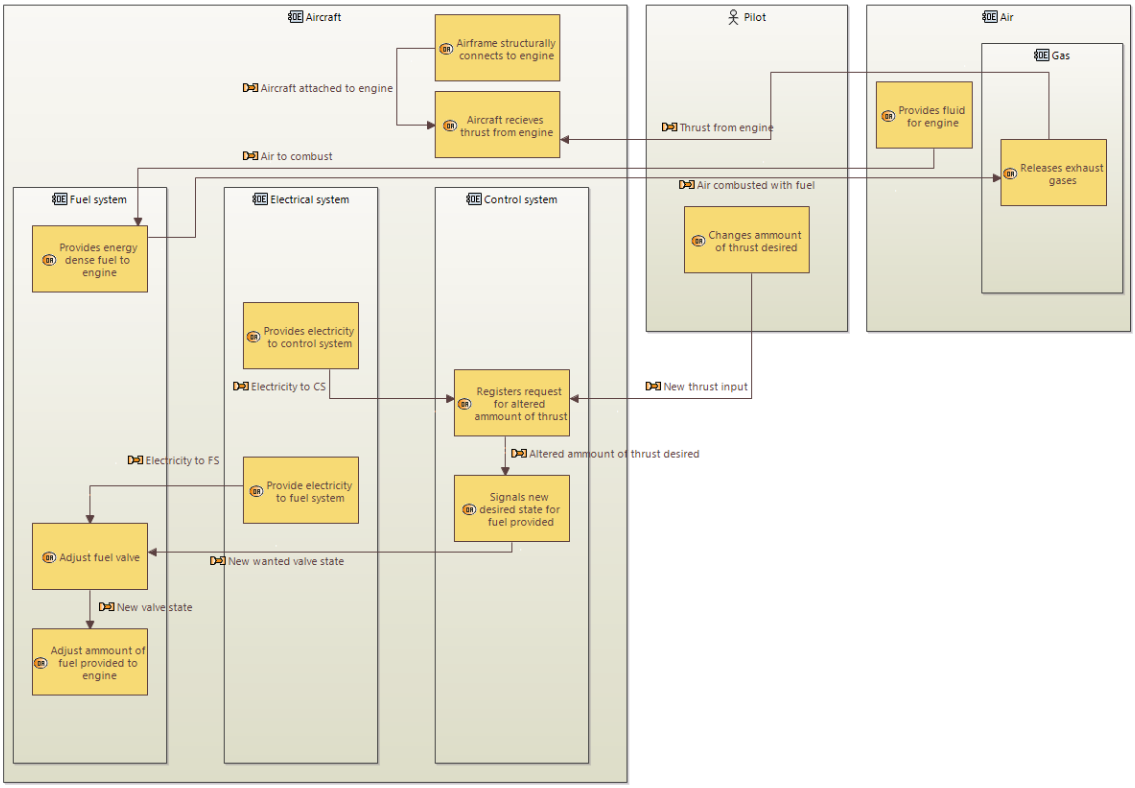


Figure 4.13: OAB diagram for creating thrust to aircraft.

By combining all operational environments, the system, in this case the engine, can be introduced by transitioning to the SA layer. Figure 4.14 visualizes all the functions that the system inherits. An alternative view is then created by grouping functions into general parent functions which reveals a less cluttered image shown in figure 4.15 that also represents the adjacent actors and an example of how system requirements can be represented.

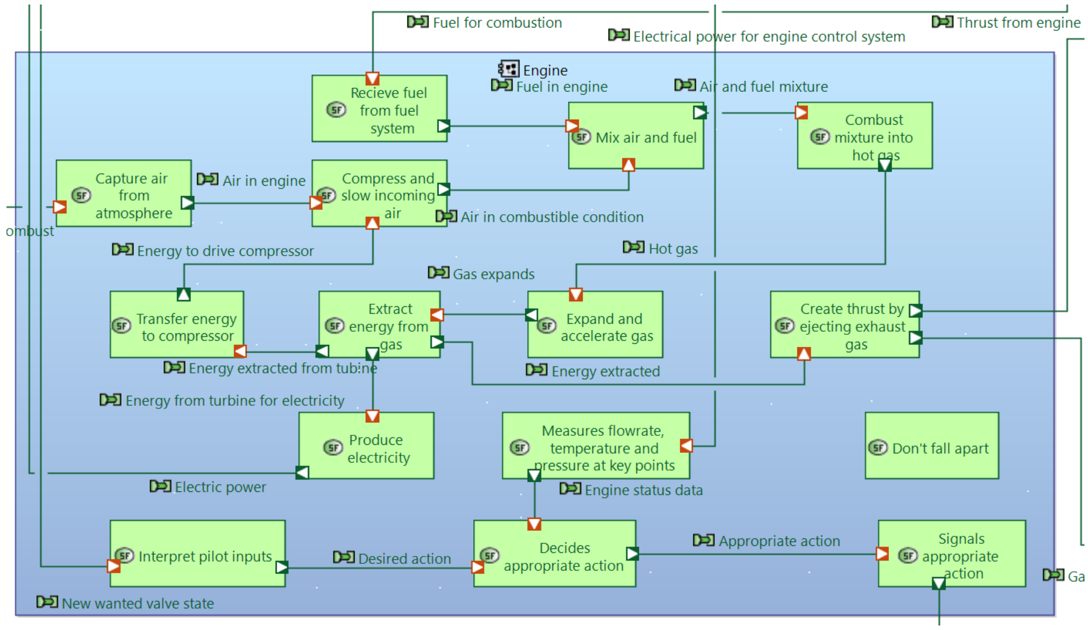


Figure 4.14: SAB diagram for system.

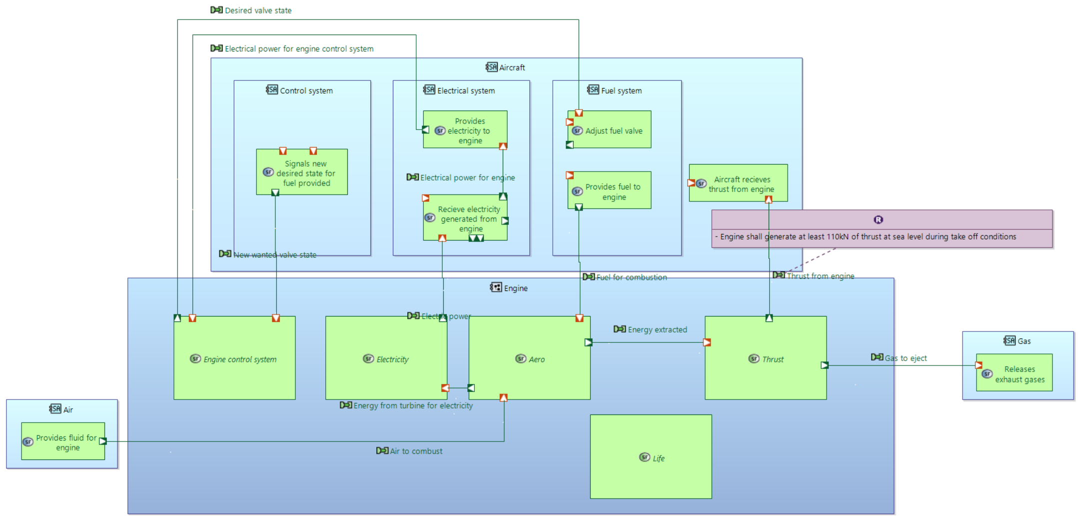


Figure 4.15: SAB diagram grouping system functions.

As mentioned earlier, the industrial case shall provide the rationale behind the functions and interfaces for the turbine blisk, hence the objective functions are not created at the same level in the industrial use case as in the simplified use case.

By transitioning to the LA layer, the different functions can be allocated to different sub systems within the engine and be broken down further to be more specific This level can also visualize how a higher level requirement can be specified to a sub system requirement. Figure 4.16 shows a snippet of the LA layer, zooming in on the turbine sub system and by this visualizing the specifications of the *Life* and *Aero* functions from the SA layer that the turbine inherits as well as an example of a requirement broken down to apply to the turbine sub system.

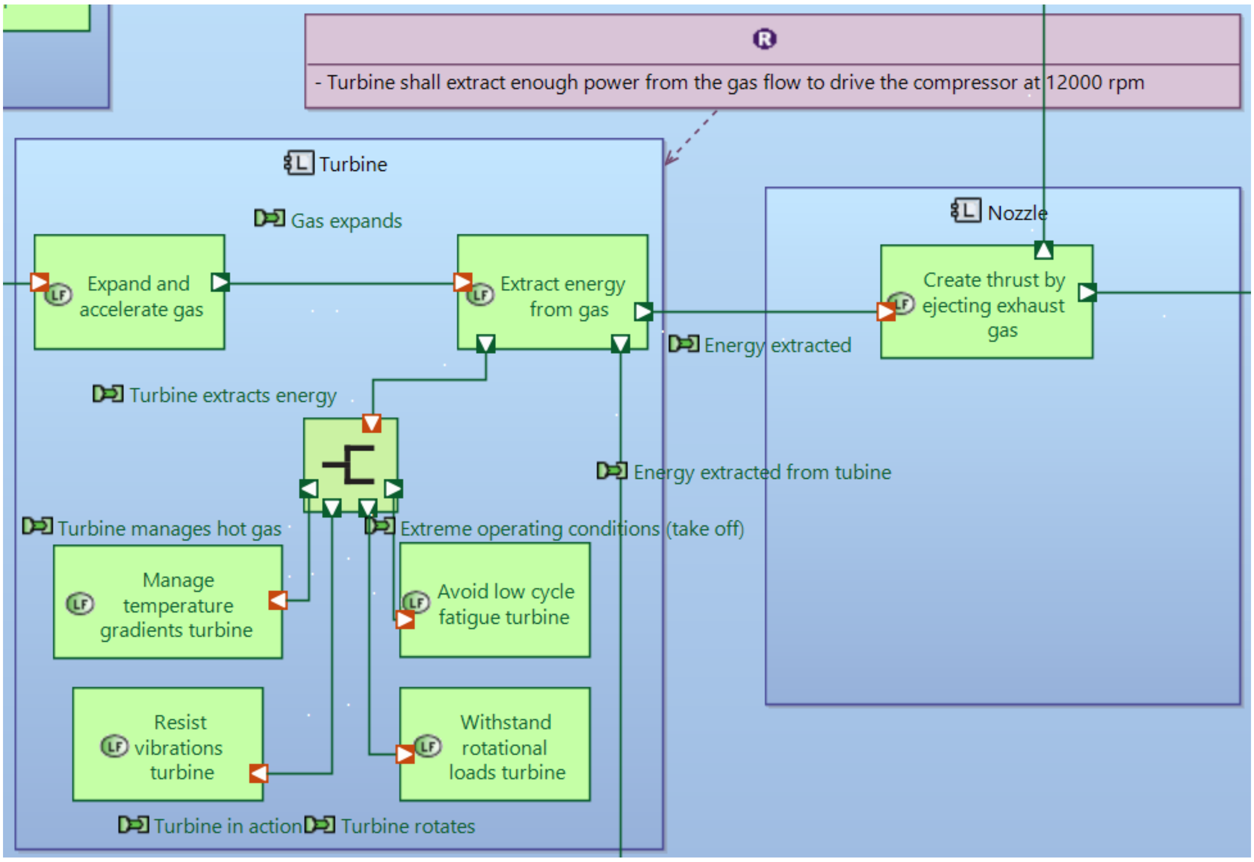


Figure 4.16: LAB diagram to define sub systems.

Moving on, the physical layer is created and showcases the included constituents, how they interact and what functions they fulfill. Since the focus of the industrial use case was the turbine blisk, only the turbine has been modeled in detail. Figure 4.17 shows a zoomed in picture of the PA surrounding the turbine as well as an additional example of how the requirements are broken down further, to component level.

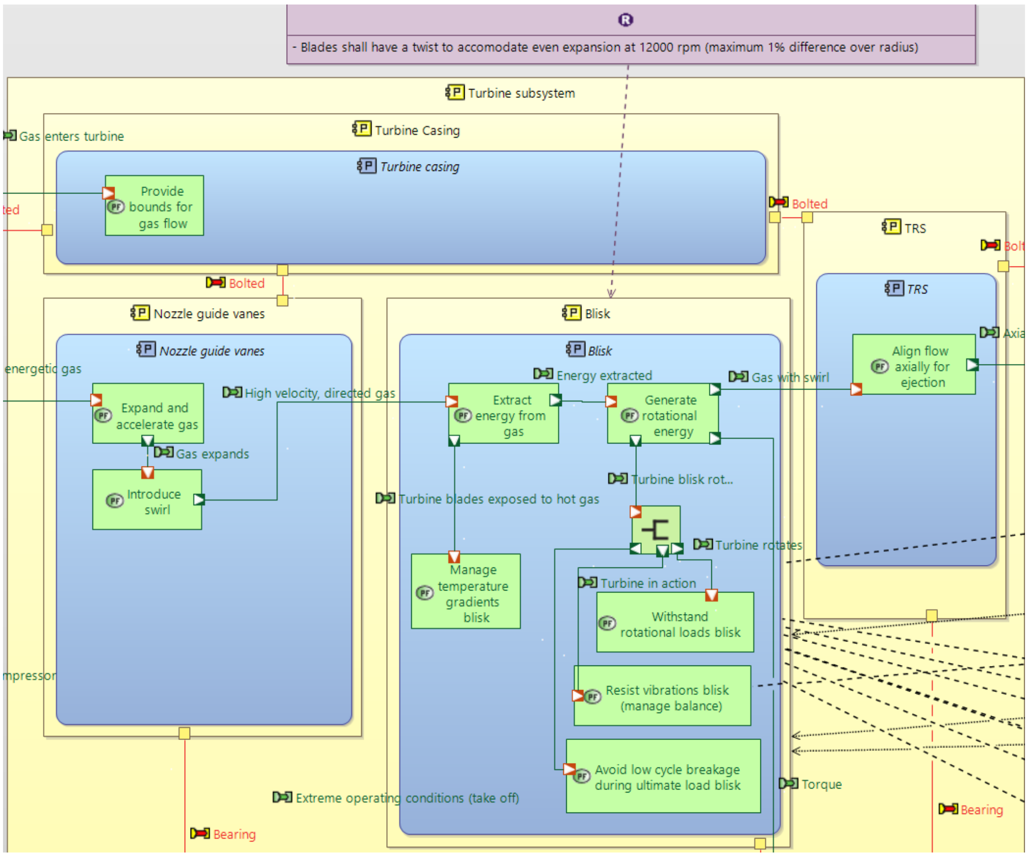


Figure 4.17: Turbine constituents in PA.

As can be seen, this level introduces the turbine blisk into the system model as a physical component. As a result of this, the MDO elements, including design varaibles, objectives and constraints, are also created here and allocated to the turbine blisk as can be seen in figure 4.18

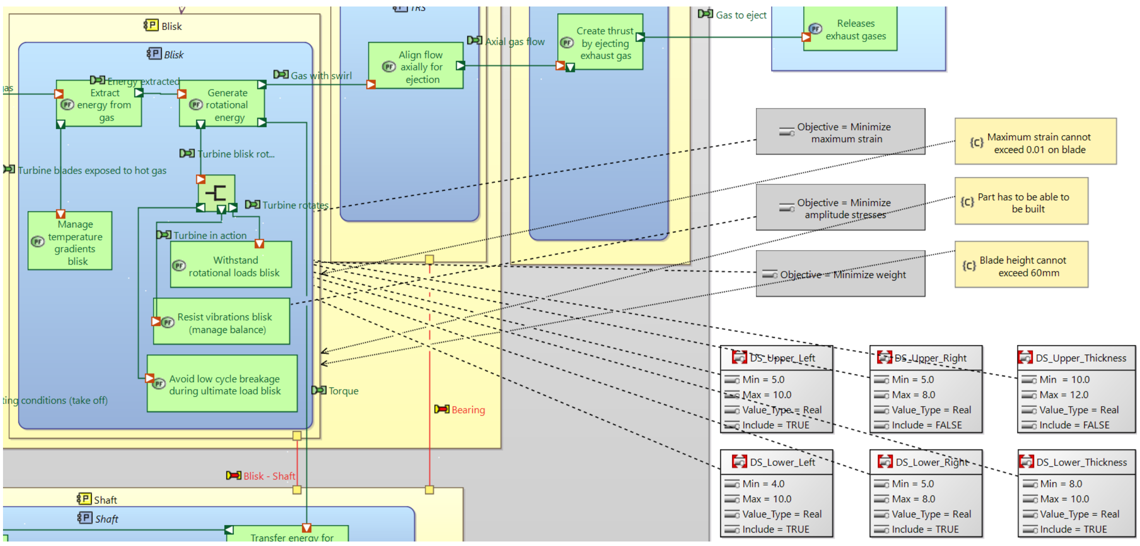


Figure 4.18: MDO elements created in relation to their component to analyze.

This shows how the operational environment for the overarching system, the engine, breaks down to be represented on a component level and also create the setup for any MDO studies.

Full frame diagrams of the industrial use case can be found in Appendix B, also including the functional break down diagrams for understanding how the grouped functions in the SA level summarize the entire system functionality and how it distributes to lower stages.

4.4.6 Descriptive Study II

This section will present the results from the interviews performed in the DS-II, aimed at evaluating the proposed method and the relevance of the industrial system model from a qualitative perspective.

4.4.6.1 Proposed Method

The method manages to achieve a mapping between the system parameters in the system model to the input parameters for MDO. There are several concerns though to be taken into consideration when exploring this approach. Firstly, there is a lot of work that still has to be done by the engineer when following this method. One problem that could appear as well is that the system model goes out of sync when several people can edit it, this since testing is sometimes done step wise and this could lead to either sub systems or the model itself being out of sync of what is going on. In the MDO integration it would also be interesting to have an intermediate step between Ansys and optiSLang to get results for a sensitivity analysis and it would be nice to have a way to take a decision on an entire group of inputs mid way to if there is a trend which makes it evident that a specific design variable can't be over a specific threshold for example.

There is a problem on the way that data is stored to be accessible for future work and that is data security risks. Different people within the company have access to different data so there needs to be some way built in to make sure only ones with access can obtain the data. There is also a risk since GKN collaborates with several different companies and requirements for example or other technical data can not be shared or presented between them. On top of this, there could be some struggles to implement a method like this since people can view things differently and then you get different views of the system if people aren't taught the same language.

4.4.6.2 Industrial System Model

One first impression with MBSE overall is that it can be hard to learn or understand it, hence giving people different views of the same system, however the feedback on the industrial SMW model indicated that it makes functions and interactions very clear and easy to find and follow. There are however a lot of things that could be improved on the model. While showcasing the basics of how a larger scale system can be broken down, many functions could be more specified and additional ones could be added. The model doesn't for example take into account manufacturing or several additional interactions such as a system for cooling air or blade tip clearance. On top of this, often times when starting a project, we don't know what the best design will be so we would need a way to evaluate different concepts. Additionally, one interesting thing that also would be nice is to see how design variables are connected to different failure modes such as burst or blade out in the case of a turbine blisk.

5

Discussion

This section will discuss and evaluate the presented method by summarizing its strengths and weaknesses, connecting it to the formulated research questions and assessing it against the success criteria. On top of this, the implemented research methodology is evaluated to understand its effectiveness and identify how this may have affected the results.

5.1 Results

The literature review and the interview study yielded very interesting results that laid the foundation for the entire method that was developed. While the literature study and qualitative interview analysis stand on a solid foundation, I think the quantitative part of the interview study should be kept away from being used as grounds for decisions or data gathering due to the following reasons. First of all, the data is based on only six different participants so the results are underpowered for generalization. Second of all, since an interview study like this and the subject area as well was new, the questions and interview structure for each participant was a little different, so the reason for these themes being mentioned or not mentioned can be as a result for how the interviewer led the interview which makes any conclusions based on this quantitative data unreliable. However, the findings from the literature study and qualitative summarization of the interviews provided a sound picture of MBSE in general and at GKN in specific.

While MBSE was the main new area of interest in this thesis, the software training and literature study gave the impression that while SMW enables a great representation of the system rationale, in its current state, it completely lacks any equipments necessary to perform numerical comparisons and optimizations on the needed scale in isolation from other tools. As a result of this, the proposed method combines multiple different tools and showcases an integrated framework between them in an ambition to make the system parameters in the system rationale be the driving factor in the external optimization tools. This results in the proposed method having to include steps that may feel disconnected to the user at first since you have to go back and forth between different software tools and it makes the integration into one streamlined process a bit harder than if everything could be solved within one tool since the data objects in these tools are built differently from each other.

For a consumer though, the utilization of a variety of software tools may be desired since this makes sure that the company is not completely reliant on one other organization. This keeps the market price for tools down and it puts the company in a stronger position when negotiating since it is easier to adopt other options if you would only have to switch a small part of the tools.

The proposed method however, acts as an indication for the possibility of the creation of a centralized tool to manage both MBSE and MDO in the future. This since all the data that is necessary for the optimization procedures can be represented in an intuitive way in connection to the system model. In the proposed method, the simulation model is set up manually by creating the same system parameters in NX and Ansys as already were created in SMW. The system parameters,digitally though, are only in the end imported into optiSLang as a way of making sure that the optimization problem is set up in compliance to the system rationale. In a perfect world where one tool could manage this, assuming that simulation and optimization methods are built in, the main thing that would have to be changed is the current representation of system parameters in our SMW tool. The system parameters would just need to be visualized in the exact same way as they are currently (just intuitive for the user to read) but it needs to have a data structure in the back end that aligns with how these elements are understood in our CAD or simulation tools. So instead of us right now, looking at the system parameter in SMW and then creating the same exact one in Ansys, this could just be a drag and drop if the back end data structure fits. This could work for both the optimization and simulation part. As for the CAD part, it should be possible to assign system parameters from the system model onto the geometric design if the general shape of the design is already created just that the parameters we used now comes from the system model directly. In terms of automating the entire creation of a CAD model from the system model, that is a lot more complicated and harder to ensure quality. This could maybe be possible though to some extent either by having saved a large quantity of similar older designs or with the implementation of AI to read the functions and interfaces in the system model.

5.2 Research Questions

5.2.1 In what way can MBSE support the DSE of jet engine sub systems by creating a connection from the rationale to the system parameters in the MDO study?

MBSE represents system information and connects system system architecture to its underlying stakeholder needs. By setting up the design space inside the MBSE model before performing the DSE, the use of MBSE ensures that the elements included in and optimization procedures comply with the initial system requirements.

By saving the system information in a dynamic model that is more flexible, collected and interpretable than separate documents, the information can be reused in future design studies as well to get a quick initial understanding of how design variables affect outcome.

5.2.2 What data structures and interfaces are required to connect an MBSE model to parametric CAD and simulation tools into an MDO workflow?

Since the DSE was performed through Ansys and optiSLang, the relevant system data needed to be formatted to fit into ways that optiSLang can read. OptiSLang has a built in way for importing csv files, hence python was used as an interface between SMW and optiSLang by transforming system data into a csv structure. Since SMW is read by parsing an XML file, which has standard methods in python, python can be used as an interface to transform the data to fit whatever other tool one may want to integrate it with. Python has an upper hand on other languages here due to the access to P4C which can be run directly from the model and introduces additional methods to read the SMW objects.

5.2.3 How can MDO data be stored and accessed in affiliation to its corresponding system model?

When analyzing data from an MDO study, it is often desirable to have it interactive in order to view it from different perspectives, compare different data sets or even add new data. SMW does not provide a build in environment that supports these kinds of visualization methods. However with the use of P4C, it is possible to store the MDO data basically anywhere that makes it reachable with a python script. This can be in a database, on a network drive or in a GIT. Python scripts can also connect to open applications or connect to other tools to display data in specific ways. This means that by aiming your python scripts in a specific SMW model to the specific location where the MDO results are stored, these results can be opened and viewed directly from the system model.

5.3 Success Criteria

5.3.1 Traceability

The proposed method does not digitally utilize the SMW objects when setting up the design space in NX and Ansys, instead manual copies are created for design variables and objectives. When setting up the sensitivity analysis in optiSLang however, elements from Ansys are checked against the elements that were defined in the system model to make sure that the DSE is set up completely in compliance to the system model. The constraints used in the sensitivity analysis and the behaviors of the design variables is directly imported from SMW though. Hence this creates a complete one-to-one mapping between all MDO elements and the system parameters and makes sure that the DSE is created in complete accordance to the underlying system rationale.

5.3.2 Automation

Automation of DSE comes with the main benefit of saving time. This method automates some of the procedures during DSE but leaves some to still have to be done manually. The entrance of the constraints and behavior of the design variables is completely automated as mentioned in the previous section. The creation of design variables and objectives in the DSE is still completely manual and on top of that, in order to ensure traceability the user has the extra step of editing the element names in SMW to fit their respective parameters. In terms of scalability, this can introduce additional confusion if there is a high amount of elements to edit spread over several diagrams which can also results in a lot of extra work.

5.3.3 Results storage and reuse ability

The future use of data has been touched upon earlier in section 5.2.3. The reuse of old data when starting a design problem provides engineers with a quick way to understand rough behaviors of the system they are designing, given that the system the available data belongs to is related to the new project. The point with utilizing MBSE to enhance understanding of the MDO data in the future is that the user can view the systems purpose, configuration and relevant interfaces as a reference for what the MDO data represents and thereby be given more context for the old data. The use of P4C enables this in a seamless way, assuming that the scripts has been created and points towards the correct data from the original study.

As mentioned in the DS-II though, there needs to be a robust way for managing data security. This can probably be handled in different ways depending on how the data is stored. If the data is stored on project discs, then it is only reachable to them who have access to it. If the data is stored in an SQL or a git, then there could probably be a way to include an access key in the python scripts that makes the data only accessible to specific people.

5.4 Method

The implementation of DRM and the structure for the methodology fit well to the task at hand and the initial level of knowledge we had for the area. By iterating the RC and DS-I stages, we could build an understanding of the theory and current situation, collect the thoughts in a research statement and then take aim of what areas we were missing information in to then research that more which led to refinement of our research statement until finally RQs and ROs reasonable for our time frame could be defined. This approach was a bit time consuming but since we started with an extremely open problem description over a relatively new area universally, this approach provided us with a sound understanding over a majority of the included tools and methods, both academically and industrially which was necessary to be able to generate a vision for the project.

Understanding of the method was split into two main parts, firstly understanding the purpose of all available tools and methods and the industrial state of the art and secondly, understanding how these tools functioned, i.e. how to use them, *what is possible to do within them?* and *what is the standard way to do things?*. Learning the tools was also necessary for conceptualizing the general data flow in the method and while some software training was inherently filtered into the implementation of the simplified use case for free, this still took some effort that could have been put immediately into method development in case the tool usage was second nature from the beginning.

This learning process made the project very front heavy in a sense which doesn't inherently need to be problematic since it gives a solid foundation. However, in this case we did not structure and document this process efficiently enough to justify the time investment of about seven to eight weeks and this resulted in us being pressed for time in the latter parts of the thesis.

5.5 Is MBSE of Value?

This thesis implemented a method where MBSE was used to drive MDO studies. This method provided good traceability and indicated that MBSE can be used as a good SSOT. However, there is a lot that could be better in terms of automation and workflow to achieve a smooth process. While it has been under consideration during development, the method has not been tested in terms of scalability or managing multiple design concepts simultaneously. In short, the method has proved that the design space for a product can be represented in available MBSE tools, SMW in this case, but there are still a lot of issues that need to be fixed.

Now, does MBSE show any potential of providing value if it was to be integrated into the workflow in these types of design problems? I believe that the proposed method shows that MBSE has great potential when it comes to providing traceability and system understanding. However, SMW takes a long time to set up since the current workflow requires a lot of repetitive steps where it almost feels like we are doing the same thing over and over. Hence, the implementation of MBSE in cases where the products are not that complex is completely unnecessary and just adds extra workload for a diminishing return.

When managing larger more complex systems though, I think that the traceability showcased in the proposed method and the increased possibilities for automation compared to a document based approach should not be overlooked and since GKN want to take part earlier in the product development process resulting in more free design and larger more complex systems, the implementation of MBSE into the standard way of working should not be disregarded.

Granted, SMW has several weaknesses, including the modeling process being repetitive, the tool not having automatic integration to other tools and the lack of an intended way to evaluate or compare several concepts at once. There may be other tools that manage this better but the sole inclusion of P4C in SMW gives SMW a good argument against other tools since it provides SMW with both internal and external operations and I think that P4C is a really important area to look into before excluding either SMW or MBSE as a whole from the standard workflow.

Now, implementing MBSE at a company is a huge investment in terms of any licensing fees, restructuring of digital workflows and training of people to be able to use the software. I recently took a training module for SMW and what I find to be the biggest problem or obstacle with implementing MBSE is the lack of know-how. A new user is faced with countless options of how to start modeling which makes the user aimless. Forums are available online but even there, people give contradictory opinions of how things should be done so there seems to not be any designated clear way to use SMW at this point. This makes it very hard for a new user since it is hard to know exactly what you want to get out of it and people seem to interpret the language a bit differently if these two things are not defined clearly enough.

Therefor, if GKN was to adopt MBSE in any form, these things need to be clearly defined. To do this, it has to be determined *What does GKN want to achieve with MBSE?*. With this question answered, an investment in exploring *what is the best tool and language for this purpose?* is necessary. Once this is known, a clear definition and way of working needs to be set; *This is how we do MBSE at GKN* to create a common language. With this in place it is possible to put together a training course that actually makes employees learn the same standards and gives them a red thread to follow when creating models. For an efficient implementation of MBSE, it should be rewarding and as easy as possible for people to pick it up which means that it needs value and clear directives.

6

Conclusions

This thesis was dedicated to investigating the potential for Model Based Systems Engineering (MBSE) to drive Multidisciplinary Design Optimization (MDO) studies of a jet engine components at GKN. An initial descriptive study revealed that there is a widely known gap between the two methodologies in general and the same is true at GKN where MBSE models are setup in System Modeling Workbench (SMW) in digital isolation from NX, Ansys and optiSLang which are used during Design Space Exploration and accommodates the MDO process. The thesis developed an integration scheme and corresponding method to create a digital thread between, and integrate SMW with the currently employed framework of MDO tools at GKN. The method was tested by implementing it on a simple design optimization problem in the form of a bolted flange and another industrial scale component in the form of a turbine blade. Testing of the method revealed that an integration of MBSE driven MDO is technically feasible with the current tools available at GKN but there is still a lot of additional work to be completed for an integration to become mature enough to integrate seamlessly into the current workflow. The implementation of MBSE showed desired results in the area regarding increased traceability and in terms of minimizing risk of human error to create inconsistencies in the analysis however the proposed method does not achieve a high level of workflow automation. These are all interest areas as to why GKN is exploring MBSE but the adoption of MBSE into the standard way of working is a huge investment and the available possibilities and their respective potential needs to be evaluated further to confirm the best way of moving forward.

7

Future Work

The way forward when it comes to using MBSE to drive MDO studies has several aspects to take into consideration. This chapter will suggest and discuss interest areas for future work, both from the basis of the method connecting MBSE to MDO, but also regarding the steps that need to be taken to evaluate the adoption of MBSE as a whole and make it as successful as possible.

7.1 Future work for proposed method

There are several directions that could be taken in terms of developing the proposed method further. One of the main things realized with the developed approach is that all the needed elements in MDO studies are able to be represented in the SMW environment which ensures traceability. A high level of automation was not achieved though since the MDO environment still has to be set up manually as an initial step. It would therefor be interesting to look into the possibility of importing the MDO elements from SMW earlier in the MDO workflow i.e. in the NX or Ansys stage to keep less things manual.

There is a tool called PyAnsys that would be interesting to explore in terms of enabling easier imports and exports or a more interactive environment in Ansys. If imports into the Ansys environment from SMW can be managed, then it would also be interesting to setup boundary conditions in SMW as well to also import the operating environment directly from the system model.

Another very important aspect where MBSE in general is lacking right now it the management of multi architectural models i.e. models that represent several different concepts at once in an efficient way. This is a very important aspect if MBSE is to be implemented in the conceptual design stage since the best system configuration may not be known at that point. SMW does not have any intended built in way to do this at all. I think though that functional chains could be an object type to look into and maybe further develop on their underlying concept. Functional chains highlight specific pathways in the system model and in the current state, by being selective with what functional chains you include in a diagram, one could utilize that to visualize how for example loads paths behave depending on what concept you highlight in a model. It has to be taken into consideration though that representing multiple architectures or concepts at once puts higher technical requirements on the integration method to external tools.

As for future work, it could also be very interesting to look deeper into P4C and what it is able to do. At this point it has been shown that it helps with imports, exports and connection to external tools but it would be very interesting to see if P4C can be utilized to change the user experience within SMW. One example is that SMW feels very repetitive to use sometimes and it feels like you are recreating almost the same diagrams when transitioning between layers just that you have a higher resolution with each step. This can feel like a lot of unnecessary work. Hence, in some cases, it could be of benefit to transition diagram structures (where these new diagrams can be edited of course) between layers. This would still save the top layers as the history of where we came from and they would represent their dedicated granularity but the user would experience a more streamlined process. This was just one example but all in all it would be interesting to see if P4C could be used to enhance the user experience inside if SMW.

7.2 Future work for general adoption of MBSE

MBSE is still a novel method that promises a lot but has a lot of issues as well. The adoption of MBSE in industry is not a guarantee in its current state but the matter should be given some attention. If the decision is made to adopt any time in the future it should be implemented from the ground up. A call needs to be made to put effort into investigating: *What are the best tools and methods for our desired capabilities?* and *How does GKN want to do MBSE?* Define a structured way of working and put together a training program with concrete implementations of MBSE in order for people to see the value it can bring and learn the method in a common way. The adoption needs to be made as natural as possible for the employees.

Bibliography

[1] Jagtap, S. S., Strehlow, A., Reitz, M., Kestler, S., and Cinar, G. (2025) Model-Based Systems Engineering Approach for a Systematic Design of Aircraft Engine Inlet. In: *AIAA SciTech Forum*, Orlando, FL, USA, January. DOI: 10.2514/6.2025-1410.

[2] Hallstedt, S. I., Isaksson, O., Nylander, J. W., Andersson, P., and Knuts, S. (2023) Sustainable product development in aeroengine manufacturing. *Design Science*, 9, e22. DOI: 10.1017/dsj.2023.22.

[3] Léonard, P. L. Y., Hallstedt, S. I., Nylander, J. W., and Isaksson, O. (2024) Sustainability transition for aerospace manufacturing. In: *Advances in Transdisciplinary Engineering*, Vol. 52, pp. 577–588. DOI: 10.3233/ATDE240200.

[4] Hou, R., Song, X., Deng, Q. Y., Zhang, X., and Deng, J. (2025) Synergistic innovation pathways in aviation ecosystems. *Sustainability*, 17(17), 7650. DOI: 10.3390/su17177650.

[5] Henderson, K., and Salado, A. (2021) Value of Model-Based Systems Engineering (MBSE). *Systems Engineering*, 24(1), pp. 51–66. DOI: 10.1002/sys.21566.

[6] INCOSE (2023) *Systems Engineering Handbook*, 5th ed. John Wiley & Sons, Hoboken, NJ, USA.

[7] Jeyaraj, A., Tabesh, N., and Hanke, S. L. (2021) Connecting MBSE and MDAO for aircraft systems. In: *AIAA AVIATION Forum*. DOI: 10.2514/6.2021-3077.

[8] Martins, J. R. R. A., and Lambe, A. B. (2013) Multidisciplinary Design Optimization: A Survey. *AIAA Journal*, 51(9), pp. 2049–2075. DOI: 10.2514/1.J051895.

[9] Haberfellner, R., de Weck, O. L., Fricke, E., and Vössner, S. (2019) *Systems Engineering Fundamentals and Applications*. Springer Nature.

[10] INCOSE (2026) About Systems Engineering. Available at: `https://www.incose.org/about-systems-engineering/` [Accessed 6 May 2026].

[11] Weilkiens, T., Lamm, J. G., Roth, S., and Walker, M. (2016) *Model-Based Systems Architecture*. John Wiley & Sons.

[12] Schumacher, T., Müller, C.-K., and Inkermann, D. (2025) SysML process chains in MBSE. *Digital Engineering*, 4, 100037. DOI: 10.1016/j.dte.2025.100037.

[13] Roques, P. (2018) *Systems Architecture Modeling with the Arcadia Method*. ISTE Press.

[14] Ulrich, K. T., and Eppinger, S. D. (2014) *Produktutveckling Konstruktion och Design*. Studentlitteratur AB.

[15] Siemens (2026) NX CAD software. Available at: `https://www.siemens.com/en-us/products/designcenter/nx-cad-software/` [Accessed 28 May 2026].

[16] Ansys Inc. (n.d.) Ansys Workbench. Available at: `https://www.ansys.com/products/ansys-workbench` [Accessed 28 May 2026].

[17] Ansys Inc. (n.d.) Ansys Mechanical. Available at: `https://www.ansys.com/products/structures/ansys-mechanical` [Accessed 28 May 2026].

[18] Ansys Inc. (n.d.) Ansys optiSLang. Available at: `https://www.ansys.com/products/connect/ansys-optislang` [Accessed 28 May 2026].

[19] Tedford, N., and Martins, J. R. R. A. (2010) Benchmarking MDO algorithms. *Optimization and Engineering*, 11(1), pp. 159–183.

[20] Al Atik, L., Srinivasan, N. R., Kallou, E., Bagdatli, B., and Mavris, D. N. Design Space Exploration Methodology. In: *AIAA Proceedings*.

[21] Timperley, L., Berthoud, L., Snider, C., and Tryfonas, T. (2024) Mapping MBSE and design space exploration.

[22] Qi, K., Luo, Z., Zhao, H., and Guo, Y. (2024) Linking MBSE and MDAO. *Journal of Physics: Conference Series*, 2755, 012003. DOI: 10.1088/1742-6596/2755/1/012003.

[23] Capella MBSE Community (n.d.) Arcadia, Capella, SysML and MBSE Tools. Available at: `https://mbse-capella.org/arcadia_capella_sysml_tool.html` [Accessed 1 June 2026].

[24] labs4capella (n.d.) Python4Capella. Available at: `https://github.com/labs4capella/python4capella/tree/master` [Accessed 4 June 2026].

[25] Calderon, M. L. (2010) Design Research Methodology in engineering education. In: *E&PDE 2010 Conference*, Trondheim, Norway.

A

System Model Simplified Use Case

This appendix includes all the diagrams created for the simplified use case, showcasing the rationale behind the design of a flange used to connect two pipes.

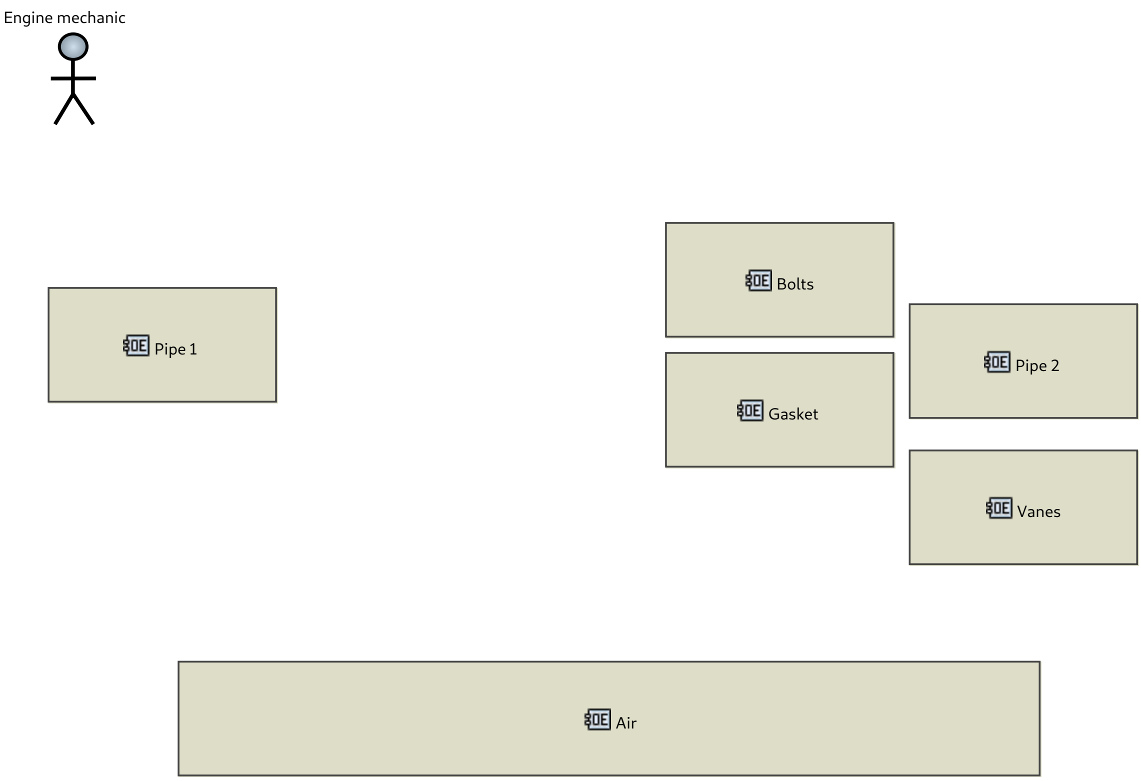


Figure A.1: Operational entities

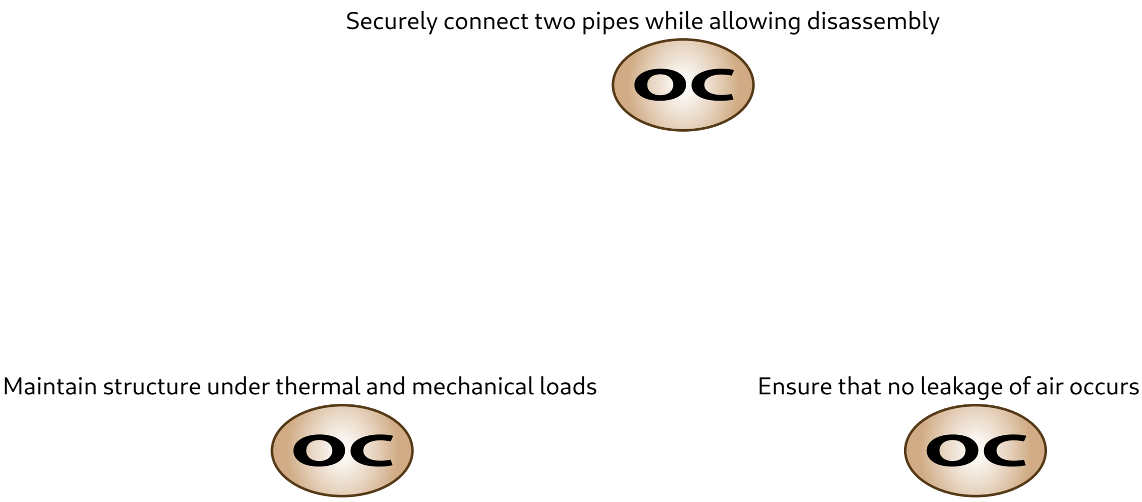


Figure A.2: Operational capabilities

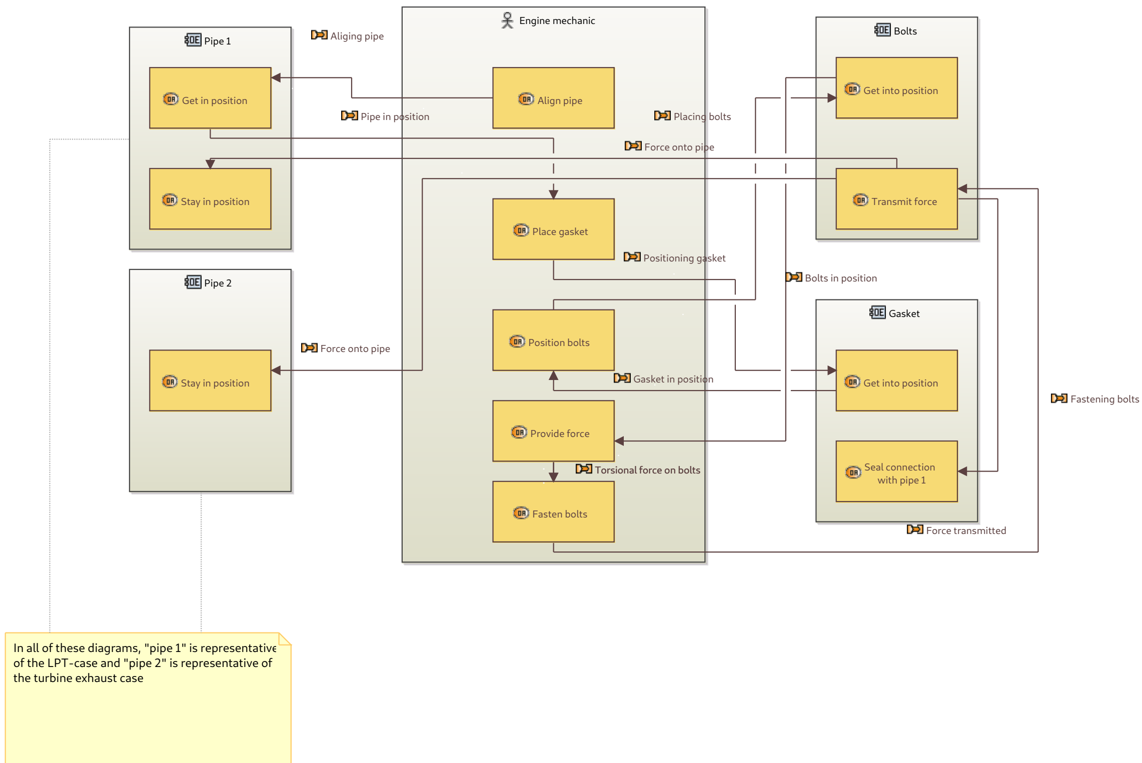


Figure A.3: OAB for pipes being assembled.

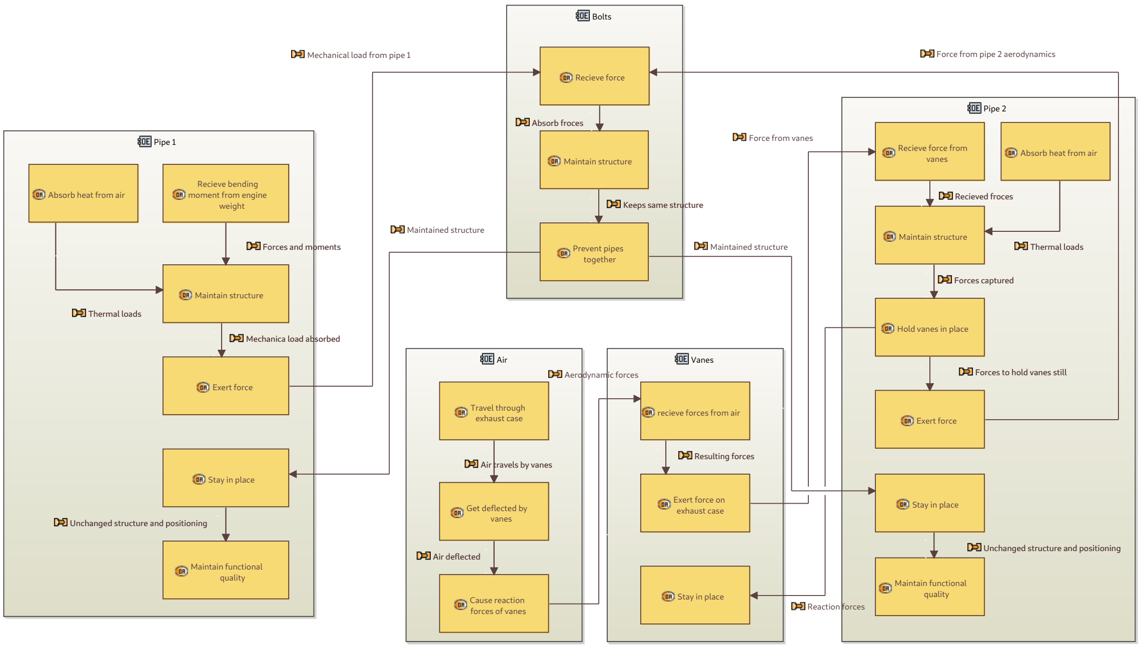


Figure A.4: OAB maintain structure under load.

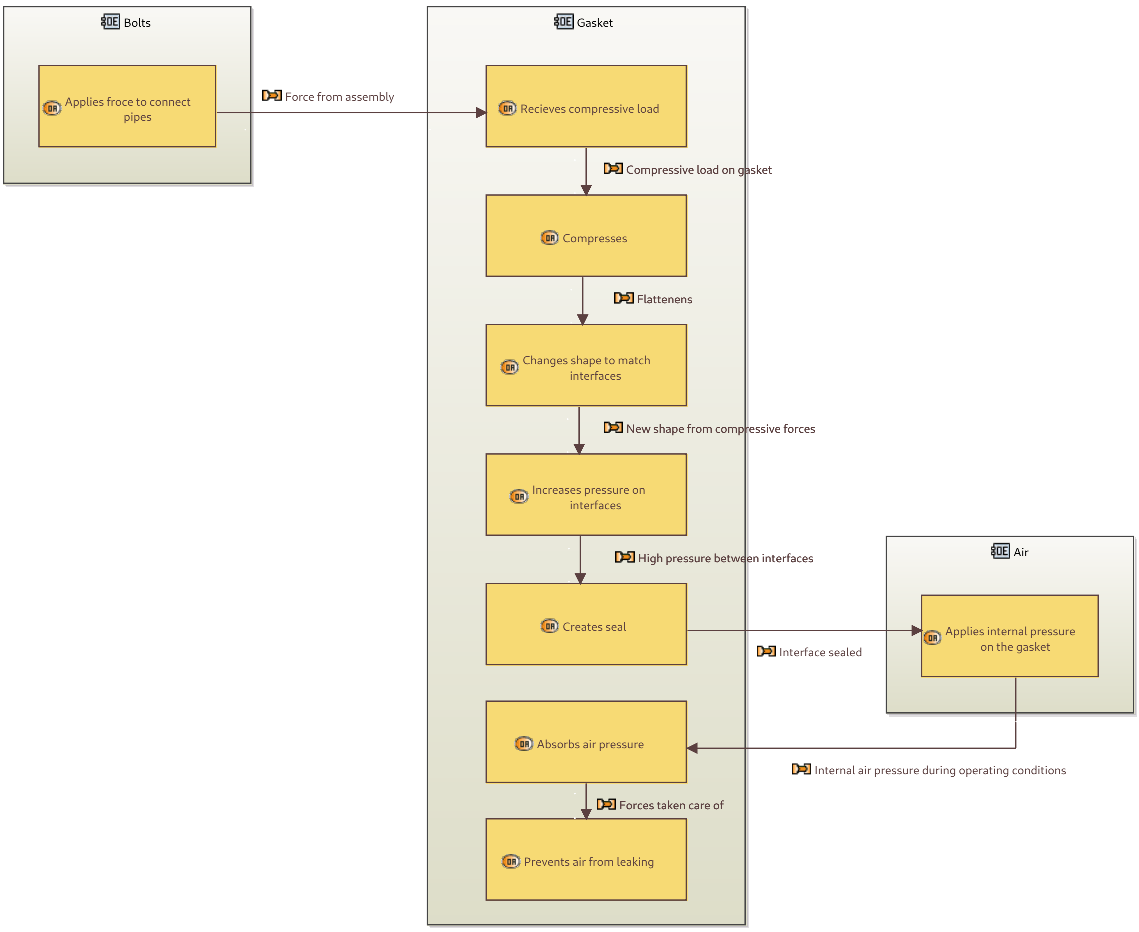


Figure A.5: OAB seal leakage

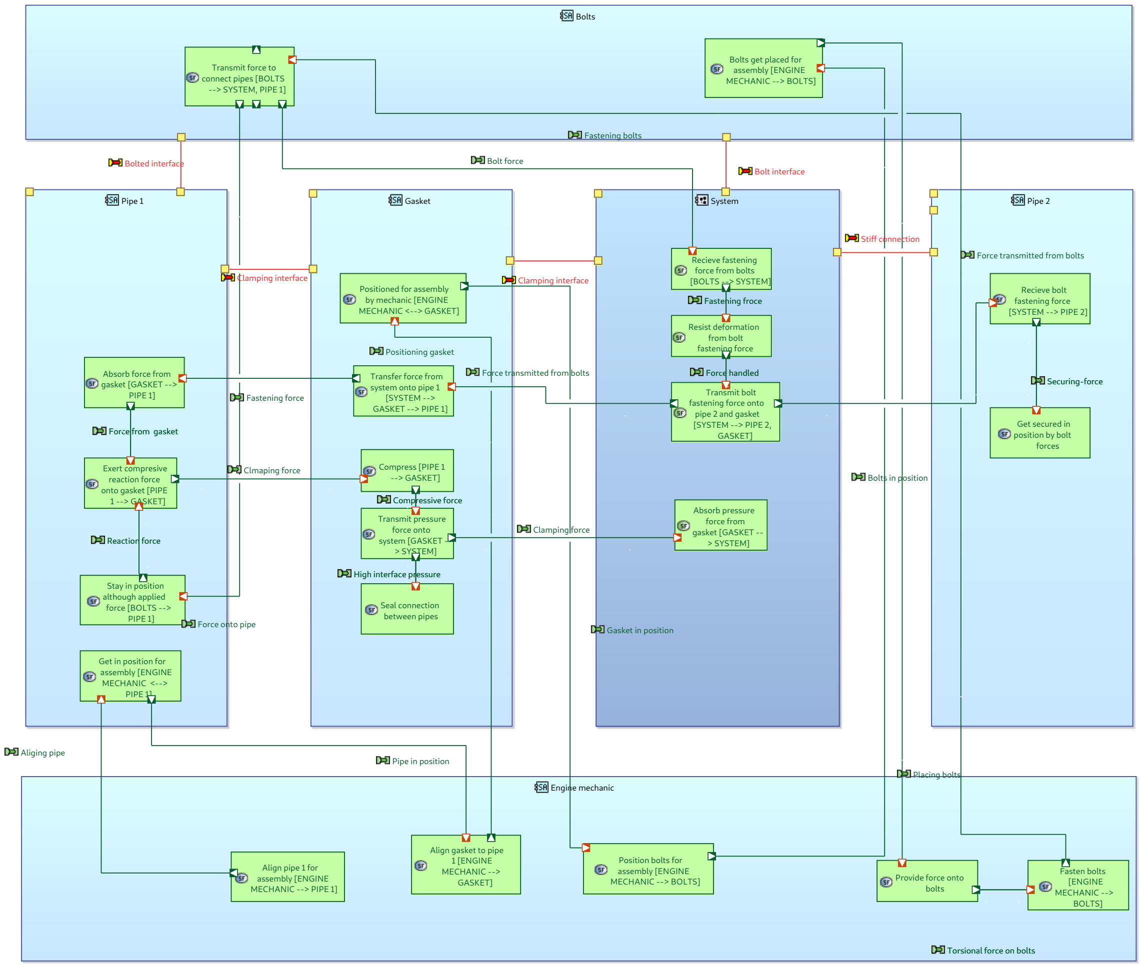


Figure A.6: SAB for pipes being assembled.

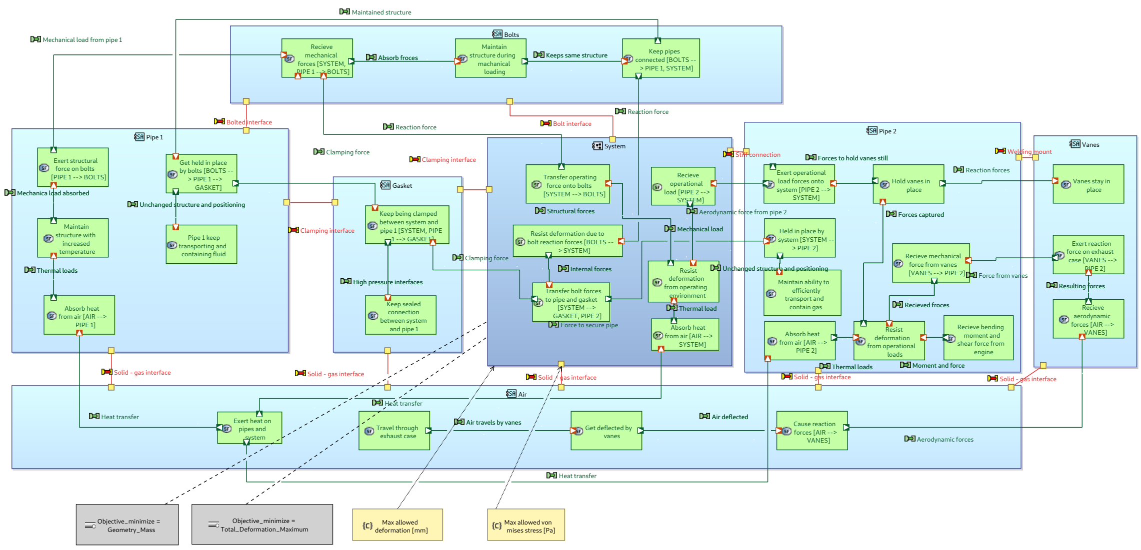


Figure A.7: SAB for pipes maintaining structure under load.

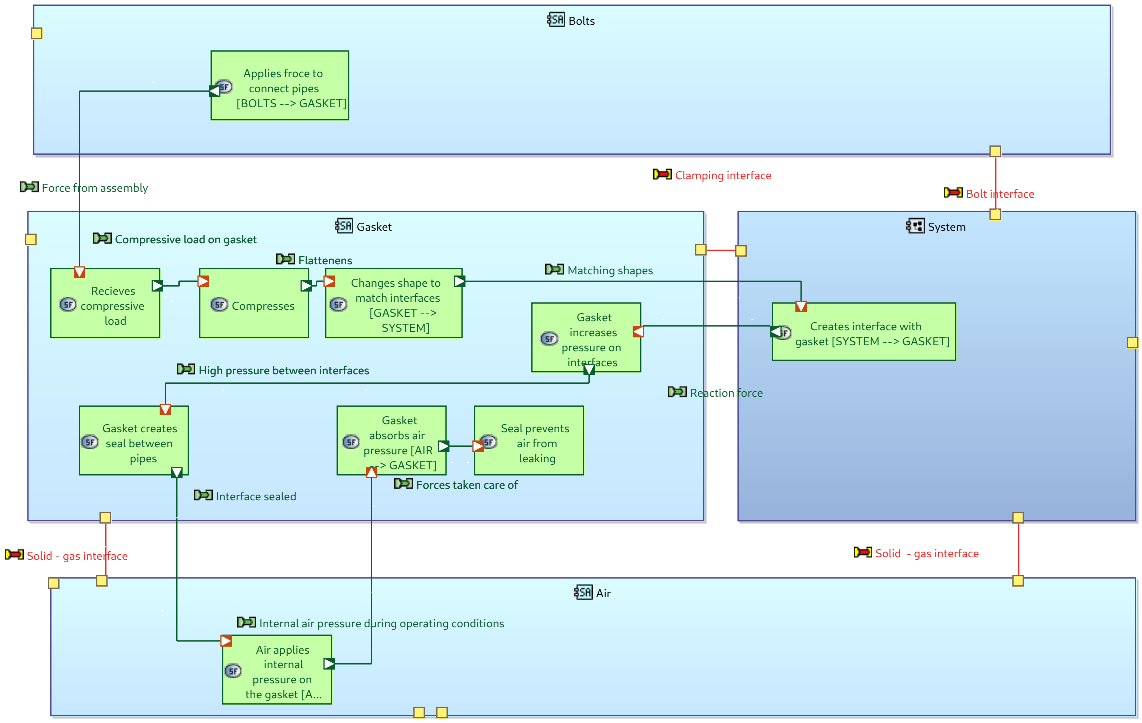


Figure A.8: SAB for seal preventing air leakage.

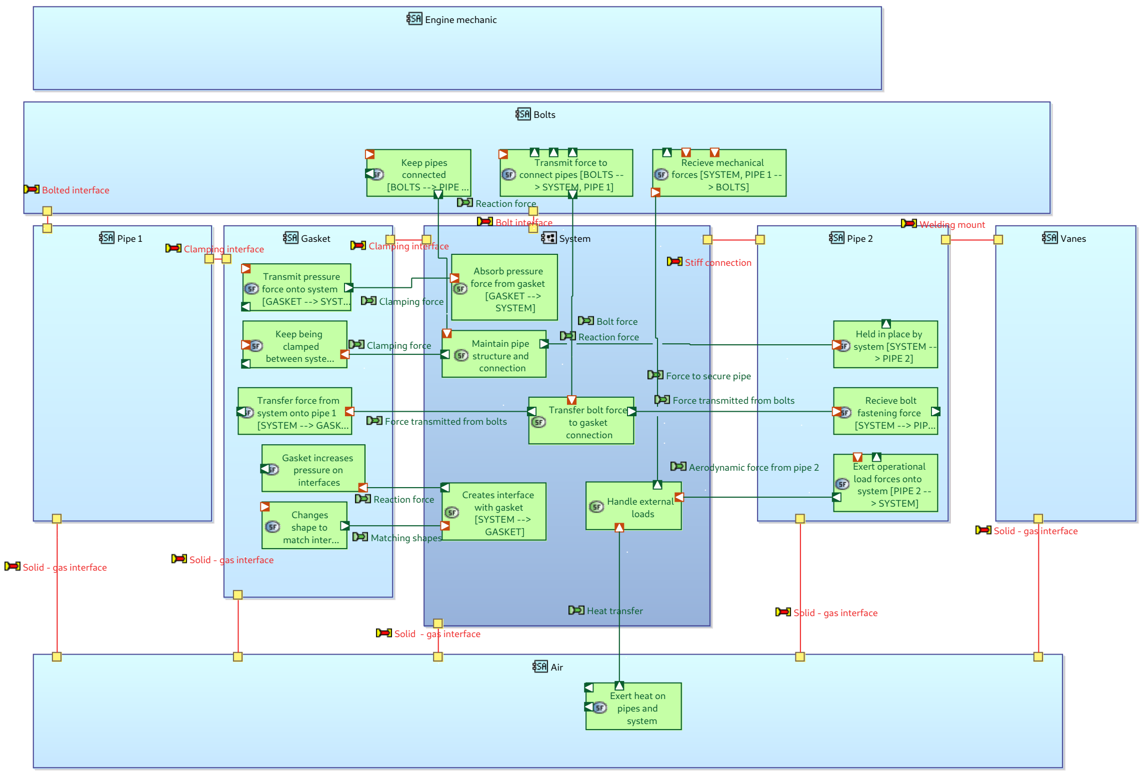


Figure A.9: SAB showing only adjacent functions and grouping related system functions.

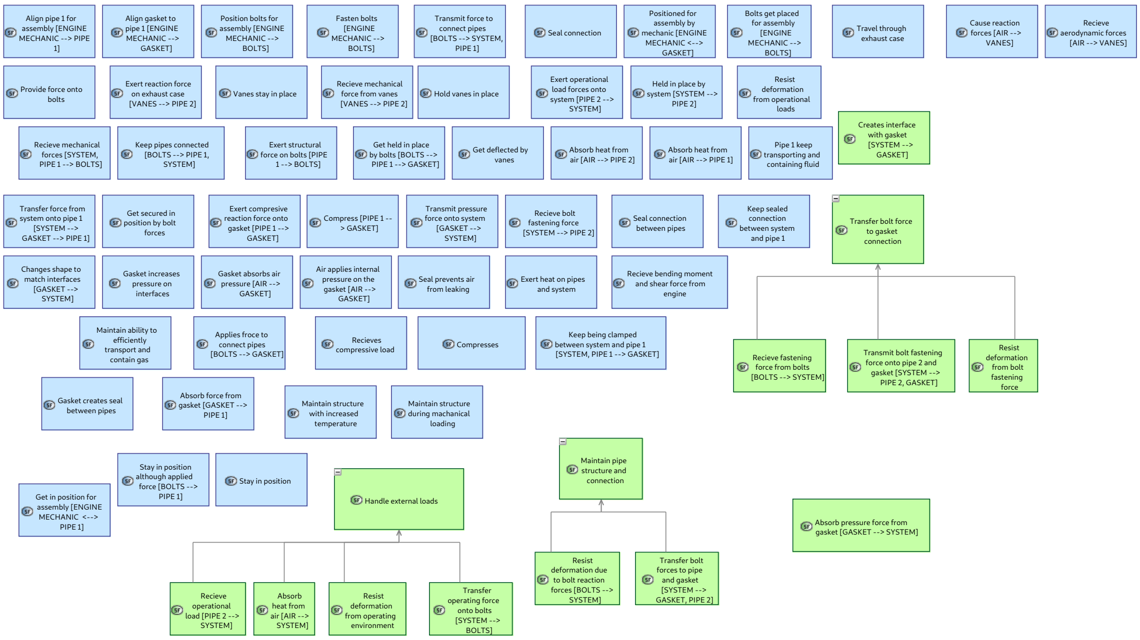


Figure A.10: All included functions and their breakdowns or grouping.

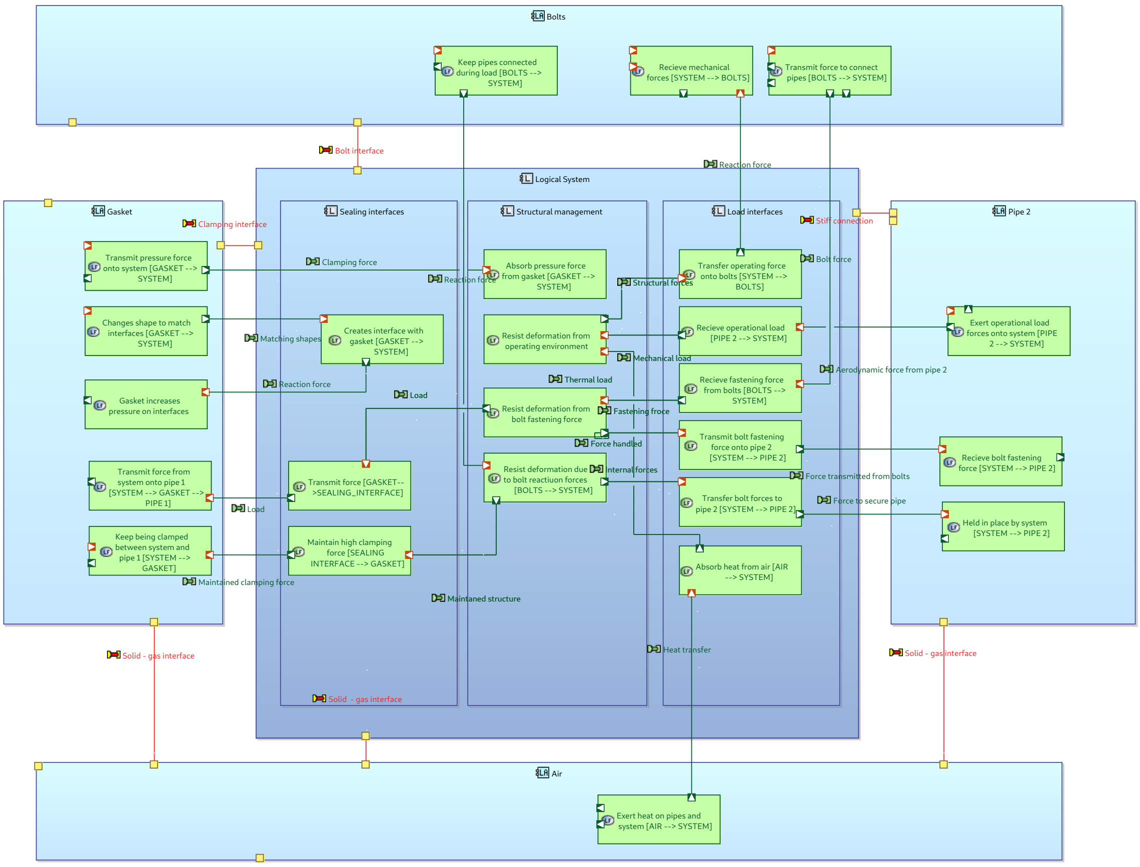


Figure A.11: LAB of flange showcasing its different parts and their function allocation

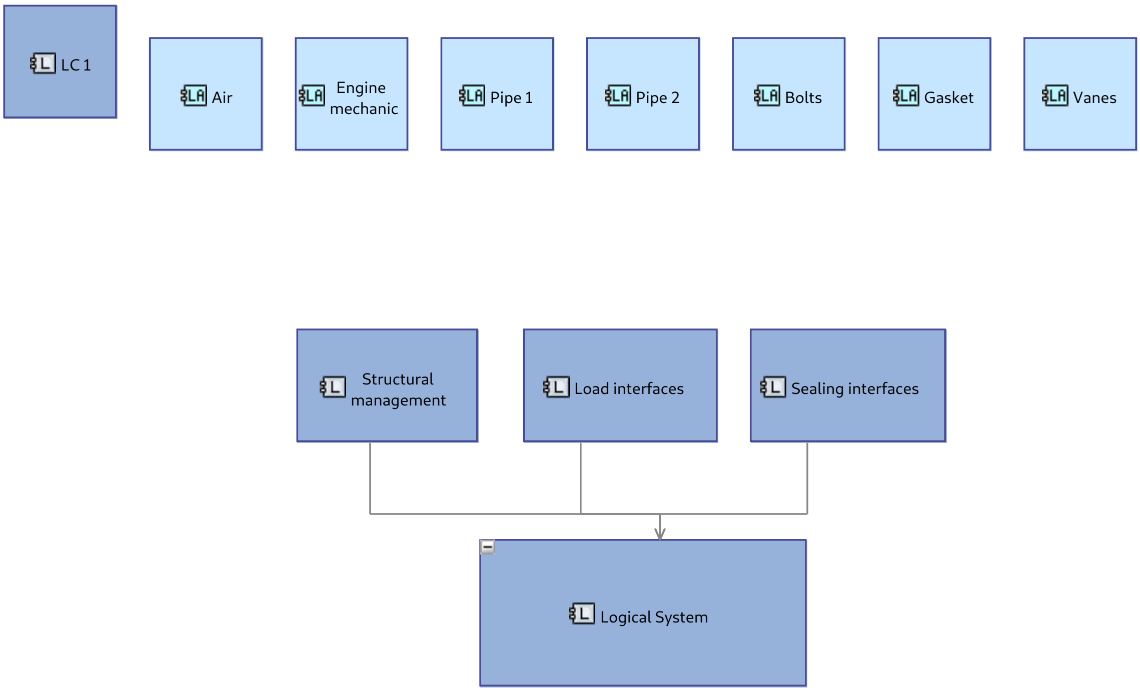


Figure A.12: Breakdown of logical system

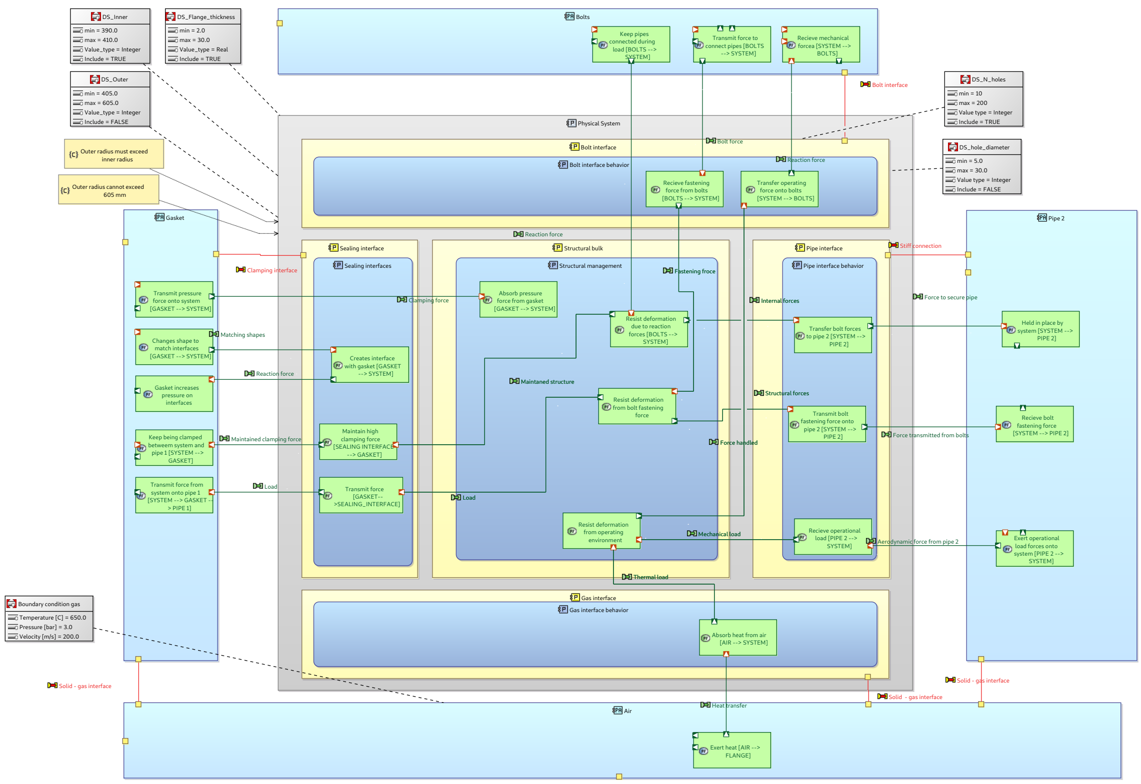


Figure A.13: PAB of the flange, showing its physical parts and their purpose, also including the design variables and constraints.

B

System Model Industrial Use Case

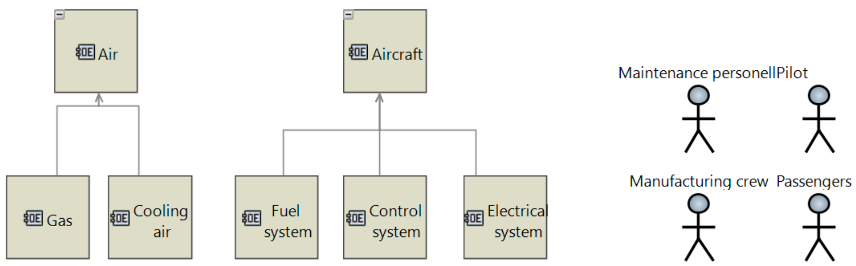


Figure B.1: Operational entities.

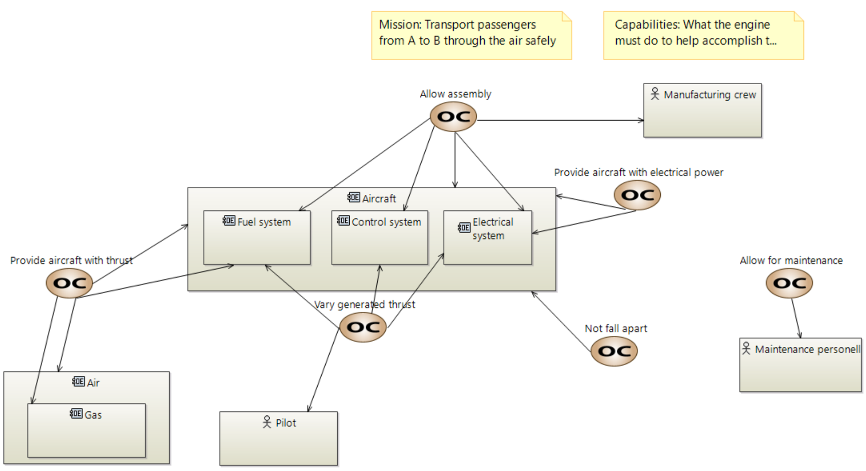


Figure B.2: Missions and capabilities.

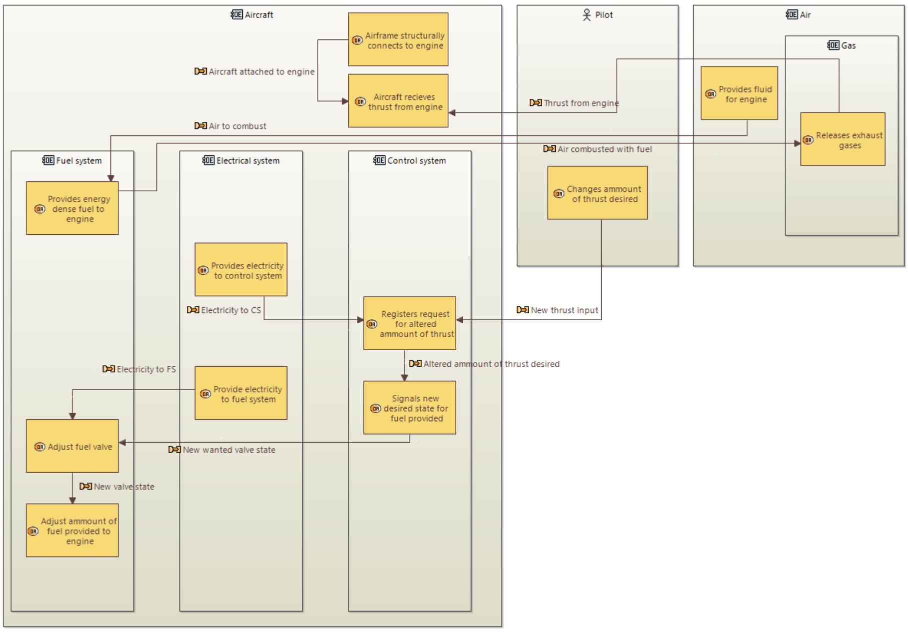


Figure B.3: OAB generating and varying thrust.

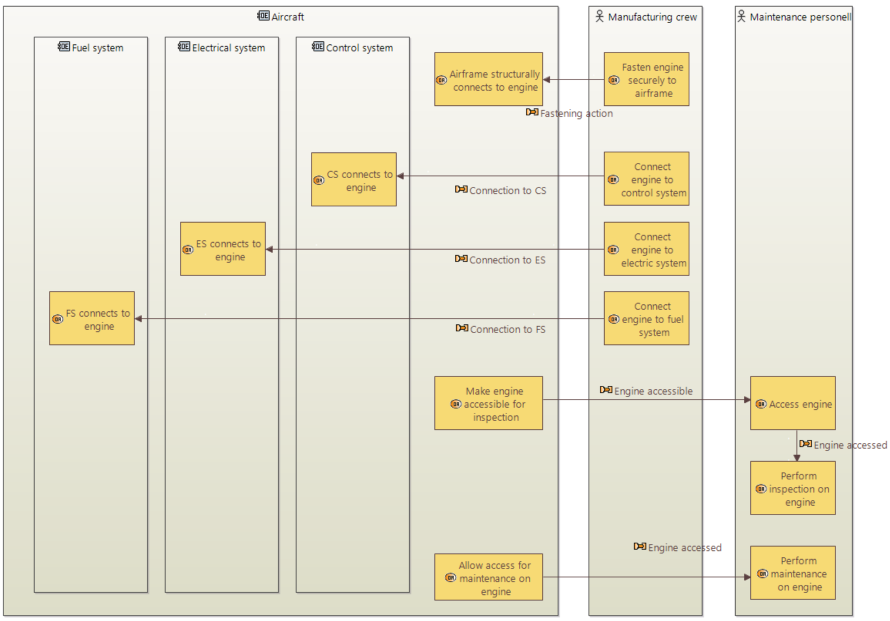


Figure B.4: OAB assembling aircraft.

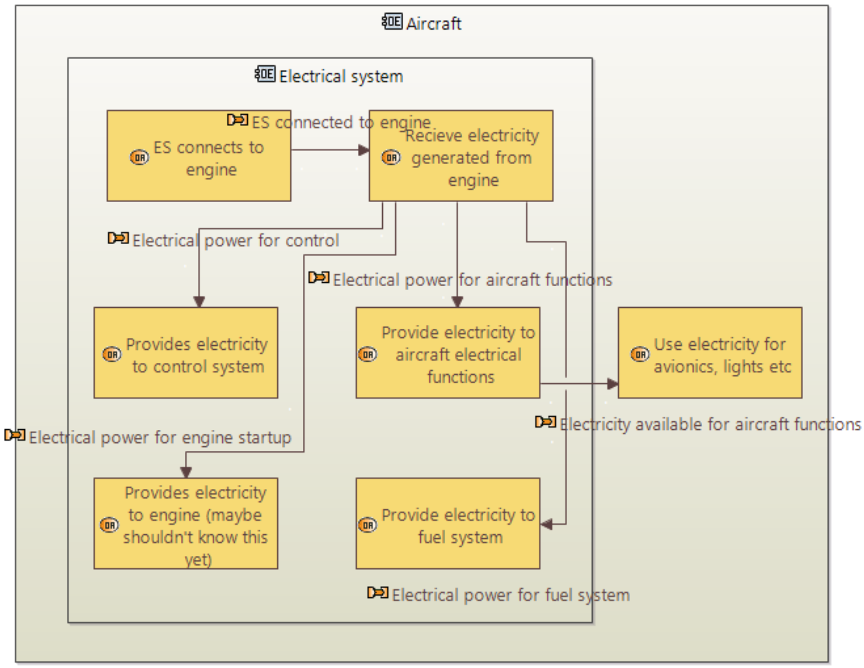


Figure B.5: OAB generate electricity.

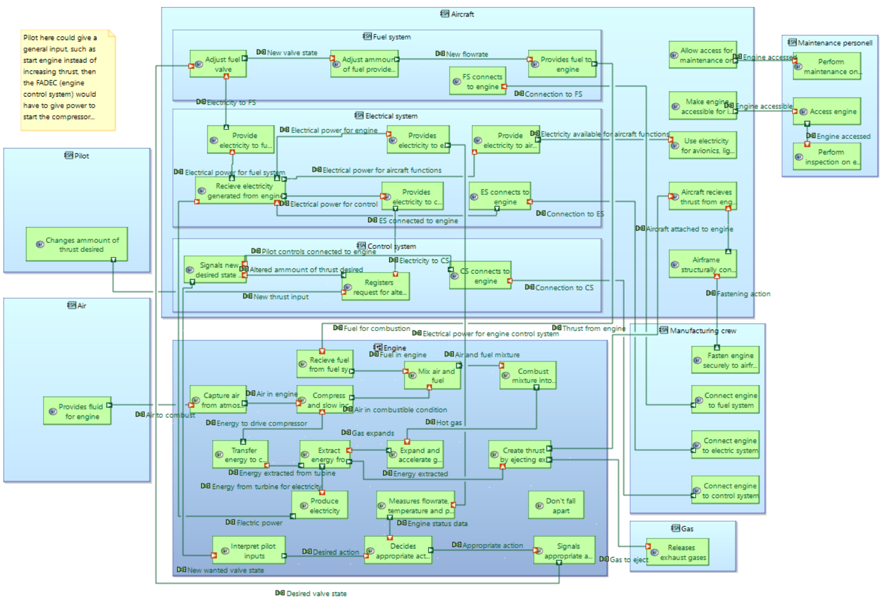


Figure B.6: SAB engine system.

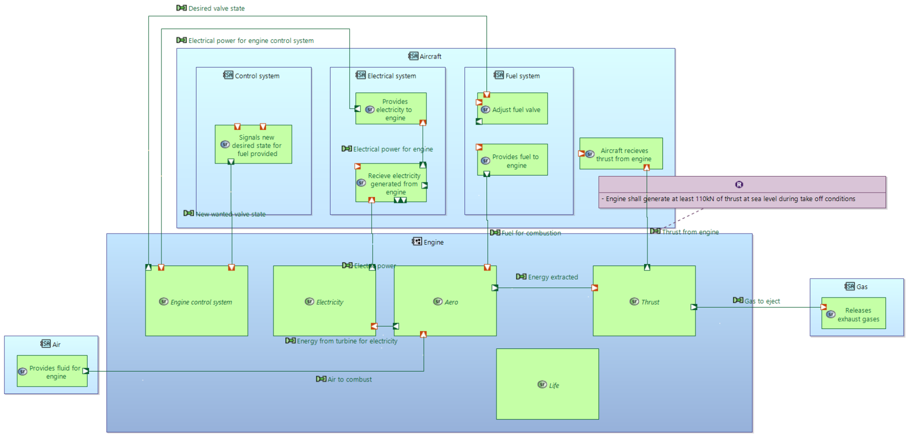


Figure B.7: SAB for engine system with grouped functions and only system-adjacent actors.

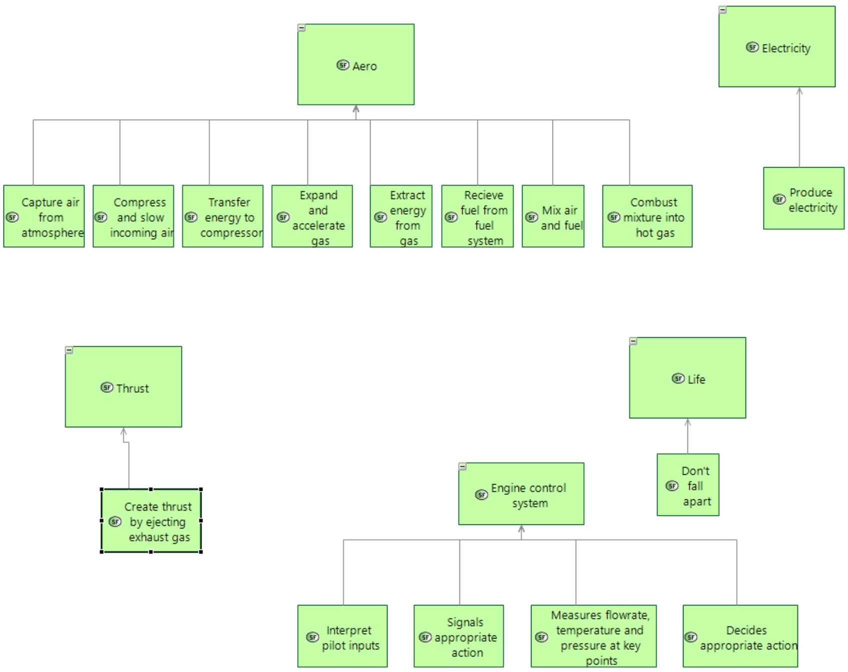


Figure B.8: SFBD, shows how grouped functions are broken down.

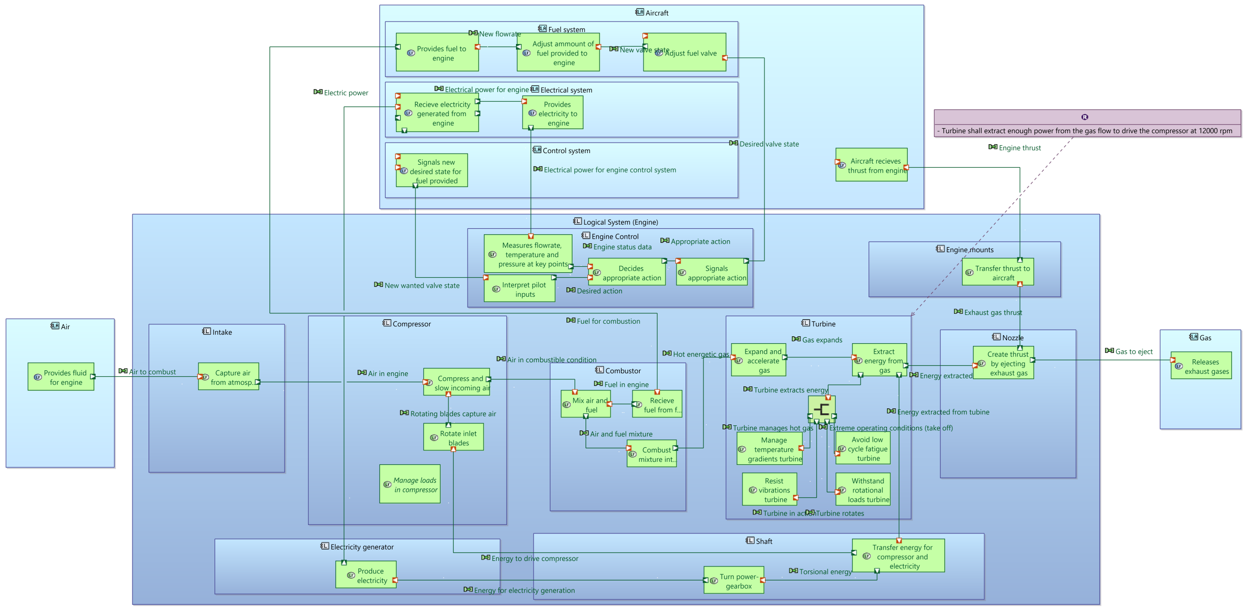


Figure B.9: LAB for full engine.

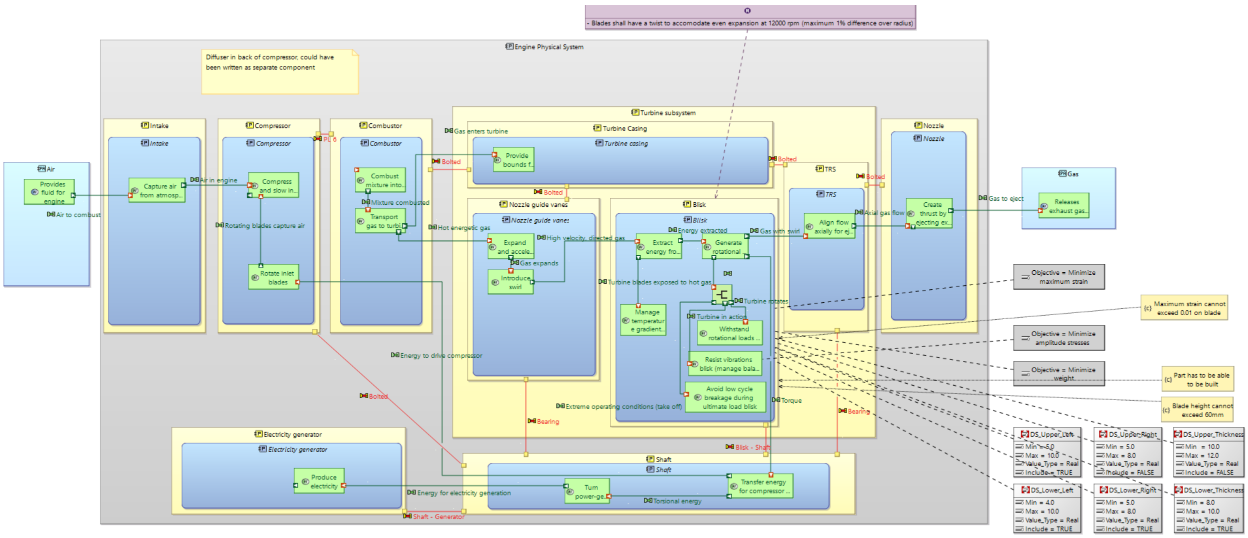


Figure B.10: PAB for full engine structure.

DEPARTMENT OF MECHANICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY