Dynamic Multi-disciplinary Analysis and Optimization workflows to enable Design for Assembly

Msc. Thesis final report

K.H. Vlessert

Technische Universiteit Delft

Delft University of Technology

Challenge the future

Dynamic Multi-disciplinary Analysis and Optimization workflows to enable Design for Assembly

MSc. Thesis final report

by

K.H. Vlessert

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on March 20th, 2025

Student number:	5425263	
Project duration:	February 20, 2024 – March 20, 2025	
Thesis committee:	Dr. ir. G. La Rocca,	TU Delft, Supervisor
	Dr. ir. O. K. Bergsma,	TU Delft, Examiner
	Dr. C. Varriale,	TU Delft, Chair
	Ir. A.M.R.M. Bruggeman,	FPP/GKN Fokker, Supervisor

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Cover picture retrieved from Mitsubishi Heavy Industries,
https://www.mhi.com/company/domain/pdf/mhica_2022.pdf (accessed on 10/12/2023)

TU Delft Delft University of Technology

PREFACE

The paper before you marks not only the completion of my thesis but also the end of my entire academic journey. This journey began in 2017 with my BSc. at the University of Applied Sciences Inholland, followed by a bridging program, and ultimately led me to the Flight Performance and Propulsion track at the TU Delft. After completing my bachelor's degree, I felt that my learning journey was far from over, which motivated me to persue further studies at the Technical University of Delft

During my thesis, I developed valuable new skills and am grateful for the opportunity to contribute to new methodologies. In my view, Multi-disciplinary Design Analysis and Optimization (MDAO) will play a large role in shaping the future, enabling faster technical innovations through digital collaboration across multiple disciplines. I enjoyed the adventure of the past years, gaining insights into fields such as system engineering, knowledge-based engineering and the academic world.

I would like to express my gratitude to my supervisors, Dr. ir. G. La Rocca and ir. A.M.R.M. Bruggeman, for their continuous guidance, valuable insights, and support throughout this thesis. Their expertise and experience in the field were indispensable. I also extend my thanks to ir. D. Bansal, who supervised the first three months of my thesis.

K.H. Vlessert
Delft, May 2025

Executive Summary

As technical designs grow more complex, involving more disciplines and more advanced tools, there is a growing need for methodologies that accelerate the design process while improving product performance. One such methodology is Multidisciplinary Design Analysis and Optimization (MDAO), which enables engineers to analyze and optimize complex systems by integrating various tools and disciplines. MDAO combines systems engineering principles with numerical optimization methods to automate design analysis and optimizations.

However, a drawback of MDAO workflows is the insufficient integration of manufacturing and assembly considerations. As a result, optimized designs are often impractical to manufacture or assemble, requiring costly post-processing adjustments. This research proposes a methodology of the inclusion of assembly considerations into MDAO workflows, building further on research by the TU Delft that includes manufacturing considerations for single part optimizations.

The primary objective of this thesis is: *Investigate the integration of manufacturing and assembly into early product design stages by using Multi-disciplinary Design Analysis and Optimization (MDAO), substantiated by a case study focused on a section of a wingbox.*

A new methodology was developed to incorporate assembly into the optimization process. Firstly, it is defined *how* assemblies can be assessed. This assessment is done by distinguishing all assembly requirements into two groups: Geometry Independent Assembly (GIA) requirements and Geometry Dependent Assembly (GDA) requirements, these are grouped based on when they can be evaluated in the workflow.

- Geometry Independent Assembly (GIA) requirements: These do not depend on geometry and are based on properties such as material type. In this research, they are implemented as a material compatibility tool. This tool checks if the materials of the two connected parts in the assembly will galvanically corrode, if so, the tool gives back a negative grade, meaning infeasibility.

- Geometry Dependent Assembly (GDA) requirements: These depend on the geometric representation of the part such as the flange area or weight. GDA constraints are evaluated using Joint Assessment Methods (JAM), which simulate specific joining methods. Within the research, four JAMs are made: Riveting, Bonding and two bolting JAMs, each assessing a joint based on its own requirements, giving a joint feasibility grade.

The requirements of the JAM are integrated into the RMM in a similar way as the requirements of the manufacturing constraints, ensuring consistency. To validate the methodology, a case study was performed on a wing rib-skin panel combination optimized for cost. Three workflows were compared to evaluate the effect of including manufacturing and assembly constraints. Results show that manufacturing constraints alone have limited impact on the design feasibility. However, assembly constraints have a major effect: design optimized without these constraints are often physically not assemblable, leading to more post-processing.

In conclusion, incorporating assembly constraints into MDAO workflows has a large impact on design feasibility. Design that ignore these factors may be cheaper but can often not be assembled in practice. This research demonstrates the potential of integrating assembly considerations into the workflow.

The document in front of you exist of three parts, all delivered in partial fulfillment of the Thesis (AE4020) of Aerospace Engineering. Part I contains the literature research, containing the initial research including scoping and background information. Part II contains the academic article, combining the findings of this research. Part III contains further technical appendices providing better understanding of some thesis parts. The supporting documentation elaborates two aspects further: (1) The cost calculations of the joint via CATMAC and (2) the defined geometric model.

Contents

List of Figures

Nomenclature

Abbreviations

$AGILE$	Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts
AI	Artificial Intelligence
AIF	Advanced Integration Framework
$CATMAC$	Cost Analysis Tool for Manufacturing of AIrcraft Components
$CMDOWS$	Common MDO Workflow Schema
$CPACS$	Parametric Aircraft Configuration Schema
DEE	Design and Engineering Engine
$DEFAINE$	Design Exploration Framework based on AI for froNt-loaded Engineering
DfA	Design for Assembly
DfM	Design for Manufacturing
$DfMA$	Design for Manufacturing and Assembly
DOE	Design of Experiments
$DV(B)$	Design Variable (Bound)
EL	Engineering Library
ELW	Engineering Language Workbench
ES	Enabling System
FPG	Fundamental Problem Graph
$FrAECs$	Future Enhanced Aircraft Configurations
GDA	Geometry Dependent Assembly
GIA	Geometry Independent Assembly
GUI	Graphical User Interface
$HDOT$	Hinge-System Design and Optimization Tool
$IDEaliSM$	Integrated & Distributed Engineering Services Framework for MDO
JAM	Joint Assessment Method
KA	Knowledge Architecture
$KADMOS$	Knowledge- and graph-based Agile Design for Multidisciplinary Optimization System
KBE	Knowledge Based Engineering
KPI	Key Performance Indicators
lb	lower bound

$MBSE$ Model-Based Systems Engineering

MDA Multi-disciplinary Design Analysis

$MDAO$ Multi-disciplinary Design, Analysis and Optimization

MDO Multi-disciplinary Design and Optimization

$PIDO$ Process Integration and Design Optimization

QoI Quantities of Interest

RCE Remote Component Environment

RCG Repository Connectivity Graph

$SADM$ System Architectural Design Space Model

$SADS$ System Architectural Design Space

SME Subject Matter Experts

SoI System of Interest

SWF Subworkflow

ub upper bound

$VISTOMS$ Visualization Tool for MDO Systems

$XDSM$ eXtended Design Structure Matrix

Symbol

$\tau_{allowable}$ Allowable shear flow $[kN/m]$

τ_{max} Maximum shear flow $[kN/m]$

A_f Fastener area $[N]$

A_f Fastener area $[mm^2]$

D_f Fastener diameter $[mm]$

e_f edge distance $[-]$

F_{max} maximum force $[mm^2]$

L_{flange} Length flange $[mm]$

L_{joint} Length of joint $[mm]$

S_f fastener spacing $[-]$

t_{flange} flange thickness $[mm]$

t_{rib} rib thickness $[mm]$

W_{flange} Width flange $[mm]$

Appendices

A

Literature research

Enable Design for Manufacturing and Assembly in Dynamic MDAO workflows

Msc. Thesis Literature study

K.H. Vlessert

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Enable Design for Manufacturing and Assembly in Dynamic MDAO workflows

MSc. Thesis Literature study

by

K.H. Vlessert

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
As part of the main thesis.

Supervisors: Dr. ir. G. La Rocca,
ir. A.M.R.M. Bruggeman,

Note: Cover picture retrieved from Mitsubishi Heavy Industries,
https://www.mhi.com/company/domain/pdf/mhica_2022.pdf (accessed on 10/12/2023)

Executive Summary

Before you lie the literature research for the MSc. Thesis of K.H. Vlessert, titled *"Enable Design for Manufacturing and Assembly to Dynamic MDAO workflows"*. The literature study (AE4020) precedes the thesis, providing the necessary background information and outline of the main thesis. The goal of the thesis, and thus the literature study, is to enable Design for Manufacturing and Assembly (DfMA) to the Multidisciplinary Design Analysis and Optimization (MDAO) workflow. This extends upon the current Design and Engineering (DEE), created by ir. A.M.R.M. Bruggeman, who is also one of the thesis supervisors.

As technical designs become increasingly complex, involving more people and higher fidelity tools, the need arises for methodologies that could speed up the design process while increasing the performance of newly designed products. MDAO is such a tool, it allows users to analyse complex multidisciplinary problems. MDAO combines system engineering principles and numerical optimization methods to automatically calculate and analyse designs. However, a significant challenge of MDAO is the absence of production considerations; designs developed by MDAO are often not producible and require further analysis and modification, resulting in additional costs and decreased performance.

This thesis assignment proposes a solution to extend the DEE with assembly considerations. The current DEE (as of February 2024) projects incorporate manufacturing considerations that can only optimize a single part with respect to manufacturing considerations. Although this process is validated, it only produces one simplified part. To fully integrate DfMA into the DEE workflow, multiple parts should be created and joined automatically, including both manufacturing and assembly considerations.

To achieve this goal, a research plan has been created, describing a step-by-step procedure for adding assembly considerations into the DEE. Initially, the current framework will be extended to create and optimize multiple parts in one run, starting with two parts to validate the concept. Automated Assembly Assessment Methods (AAAM) will be created to assess the feasibility of the assembly. This tool will be integrated into the workflow such that the workflow accounts for both the manufacturing and assembly considerations. Subsequently, an analysis will be conducted to increase the efficiency of data transfer, thereby reducing the computational time. Once achieved, the number of parts within the proof of concept will be increased to ensure scalability.

Finally, an extension of the DEE is delivered, marking the completion of the last full part of the DEE project. The DEE, under which this thesis falls has been performed in the framework of AGILE 4.0 (Towards Cyber-physical Collaborative Aircraft Development) and DEFAINE (Design Exploration Framework based on AI for froNt-loaded Engineering).

K.H. Vlessert
Delft, February 2024

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

$AAAM$ Automated Assembly Assessment Method

$AGILE$ Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts

AI Artificial Intelligence

AIF Advanced Integration Framework

$CATMAC$ Cost Analysis Tool for Manufacturing of AIrcraft Components

$CMDOWS$ Common MDO Workflow Schema

$CPACS$ Parametric Aircraft Configuration Schema

DEE Design and Engineering Engine

$DEFAINE$ Design Exploration Framework based on AI for froNt-loaded Engineering

DfA Design for Assembly

DfM Design for Manufacturing

$DfMA$ Design for Manufacturing and Assembly

DOE Design of Experiments

EL Engineering Library

ELW Engineering Language Workbench

FEM First Element Method

FPG Fundamental Problem Graph

$FrAECs$ Future Enhanced Aircraft Configurations

GUI Graphical User Interface

$HDOT$ Hinge-System Design and Optimization Tool

$IDEaliSM$ Integrated & Distributed Engineering Services Framework for MDO

KA Knowledge Architecture

$KADMOS$ Knowledge- and graph-based Agile Design for Multidisciplinary Optimization System

KBE Knowledge Based Engineering

$MBSE$ Model-Based Systems Engineering

$MDAO$ Multidisciplinary Design, Analysis and Optimization

MDO Multidisciplinary Design and Optimization

$PIDO$ Process Integration and Design Optimization

RCE Remote Component Environment

RCG Repository Connectivity Graph

$SADM$ System Architectural Design Space Model

SME Subject Matter Experts

$VISTOMS$ Visualization Tool for MDO Systems

$XDSM$ eXtended Design Structure Matrix

1 Introduction

In recent years, the world has undergone rapid transformations, witnessing increasing global challenges spanning geopolitics, finance, and environmental concerns. Within these challenges, the aviation industry is tasked to make growth and sustainability compatible. This can be achieved by creating innovative solutions faster, reducing the time-to-market, and enhancing the sustainability of new and/or existing products. During the technical design process, where these products are created, significant steps have been taken. More accessible technologies, such as Finite Element Methods (FEM), and Artificial intelligence (AI) already introduced a leap forward in efficient design process. However, integrating these technologies, especially with the increasing size of projects, introduces complexity. Therefore, the introduction of new methodologies becomes important to effectively create and optimize innovative products with a minimized time-to-market and increased sustainability [1, 2].

Multidisciplinary Design Analysis and Optimization (MDAO) has the potential to reduce this time-to-market. Developed over three decades ago, MDAO has a long track record of research-proven potential [3]. MDAO enables users to study complex multidisciplinary problems by streamlining the design process. It combines system engineering principles and numerical optimization methods to include various disciplines in an optimization tool, exploiting their synergies. MDAO contains the capability to automatically evaluate different design options against predefined constraints and objectives, such as cost-effectiveness or weight-minimization [4]. Additionally, it can be used to analyse existing designs to investigate their characteristics. Despite these applications, MDAO is not widely adopted by the industry [5].

Definition: Difference between MDO and MDAO

MDO (Multidisciplinary Design Optimization) and MDAO (Multidisciplinary Design Analysis and Optimization) are often used interchangeably while defining the same system. The distinction lies in the emphasis on both analysis and optimization, users of the MDAO framework can both optimize products or focus on the analysis of products, without optimization. MDO solely focuses on optimization without significant emphasis on analysis.

One of the primary challenges within MDAO, particularly in optimizing preliminary or conceptual designs, is the absence of production and assembly considerations. The impact of these optimized early-phase designs is profound on the subsequent more detailed stages of the design process. According to research by Shahpar [6], over 80% of the program's life-cycle cost is set after the preliminary design review, even though a small portion of this cost has been spent. The *"optimized early-design"* often needs alterations to change it into a feasible and producible detailed design, resulting in increased cost and weight. Furthermore, these design alterations can change the performance of the design such that the final design is not the optimized design [2, 7–9].
Production methods are often disregarded as design teams may have limited knowledge about these aspects. Either processes are not well documented, or subject matter experts (SMEs) become involved at larger stages. This system lacks a formal structure [1, 6].

Besides these challenges, recent years have seen projects grow in scale, incorporating more disciplines and involving a greater number of experts. This expansion made older versions of MDAO less transparent and nearly impossible to manage, contributing to its limited adoption within the industry [3, 6, 10].

Various collaborative projects have been initiated to fight these shortcomings, these projects include FrEACs (Future Enhanced Aircraft Configurations) [11], IDEaliSM[1] (Integrated & Distributed Engineering Services Framework for MDO), DEFAINE[2] (Design Exploration Framework based on AI for froNt-loaded Engineering), and AGILE (Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts). [2, 5]

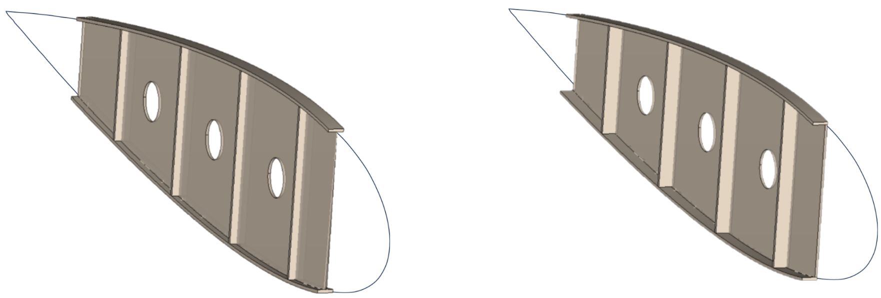
The Design and Engineering Engine (DEE) is a framework developed through the experiences and lessons learned from the aforementioned collaborative projects. The goal of the DEE is to generate a blueprint for the automatic creation of MDAO workflow with the integration of the MBSE (Model-Based Systems Engineering) principle. MBSE focuses on the digitalization of the system engineering process, where traditional document-based products are replaced by interconnected digital models [13]. The DEE enables users to define MDAO workflows, from requirements to the final product, and automatic verification. This thesis project further extends the DEE framework with the Design for Manufacturing and Assembly (DfMA) principles.

Definition: Model-based System Engineering

MBSE is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life phases. [12]

1.1. STATE-OF-THE-ART

The DEE framework, initiated by Bruggeman *et al.* [8], is a PhD research project and has resulted in the creation of multiple sub-tools. One of these sub-tools is the research presented by Nikitin [9]. This research expands the capabilities of the DEE by incorporating manufacturing methods within the dynamic MDAO workflow, therefore enabling Design for Manufacturing (DfM). During this research, a workflow was developed that uses the "manufacturing method" as a design variable, dynamically guiding the MDAO workflow to select the corresponding analysis disciplines corresponding to the chosen manufacturing method. However this system is a huge leap into the inclusion of the DfMA considerations, it is currently limited to optimizing one single part. The proof of concept for this approach was successfully demonstrated through the optimization of a wing rib as illustrated in fig. 1.1.



(a) Optimized aluminium wing rib

(b) Optimized composite wing rib

Figure 1.1: Proof of concept of the DEE extended with production considerations [8]

In addition to the inclusion of DfM, the DEE framework has facilitated the development of various other tools [13], including a Requirement Management Module (RMM) and the Manufacturing Information Model (MIM). The RMM ensures the efficient integration of the requirements into the MDAO workflow. Users can

[1] https://idealism.ifb.uni-stuttgart.de/index.html, accessed on 17-01-2024
[2] https://www.defaine.eu/, accessed on 17-01-2024

convert document-based requirements into machine-readable model-based requirements, enabling direct utilization within the workflow, and can automatically generate a compliance report upon completion of the optimization. The MIM streamlines the data management process by including a database and system information, thereby automatically making data directly accessible. While the RMM is integrated into the workflow, the MIM is currently not.
The outcome of the research in Bruggeman *et al.* [8] showed two optimized designs; an optimized aluminium wing rib as shown in fig. 1.1a which is optimized for cost, and a composite wing rib that was optimized for weight as illustrated in fig. 1.1b. Both designs have been proven producible through the inclusion of production methods.

Expanding upon the findings of this research, the thesis discussed in this literature research introduces the assembly methods into the DEE, therefore fully enabling DfMA. This integration enables the MDAO workflow to scale up, enabling the analysis and optimization of multi-part designs and supporting the growth and development of more complex systems.

1.2. RESEARCH OBJECTIVE AND RESEARCH QUESTIONS

The primary aim of this thesis is to investigate how the manufacturing assembly methods can be further taken into account into MDAO workflows. Therefore, the research objective is written as:

Investigate how to integrate manufacturing and assembly into early product design stages by using Multidisciplinary Design Analysis and Optimization (MDAO) framework, substantiated by a case study focused on (a section of) a wingbox.

To achieve research objective, four research questions have been set up, these eventually lead to the research objective:

1. What automated methods can be developed to assess the feasibility of assembly/joining of parts, and how can these **assembly assessment methods** be incorporated within the MDAO framework?
2. Would the incorporation of the **assembly assessment methods** and the **production assessment methods** result into problems within the MDAO workflows
3. How can **assembly related requirements**, such as material compatibility be incorporated into MDAO workflow?
4. By what means could the challenges posed by a **large combination of parts and joints** and their corresponding manufacturing and assembly methods be effectively addressed within an MDAO workflow?

This report serves as the foundation for the main thesis, marking the first eight weeks dedicated to literature research. Its objective is to assess the current state-of-the-art DEE version as specified by Bruggeman *et al.* [8], upon which this thesis will be built. Furthermore, it will outline the necessary steps to achieve the aforementioned research objective: *"Integrate manufacturing and assembly into early product design stages by using MDAO framework"*.
To begin, chapter 2 introduces generic terms and definitions within the MDAO workflow, tracing its historical evolution from the first versions to the current version. Chapter 3 provides a more in-depth discussion of the state-of-art and modules/frameworks. Following these two chapters, chapter 4 introduces the wingbox required for validation and outlines the assembly process. Subsequently, chapter 5 focuses on possible assembly methods and the assembly-related requirements.
After these chapters, the research plan and the conclusion are given in chapter 6 and chapter 7 respectively.

2

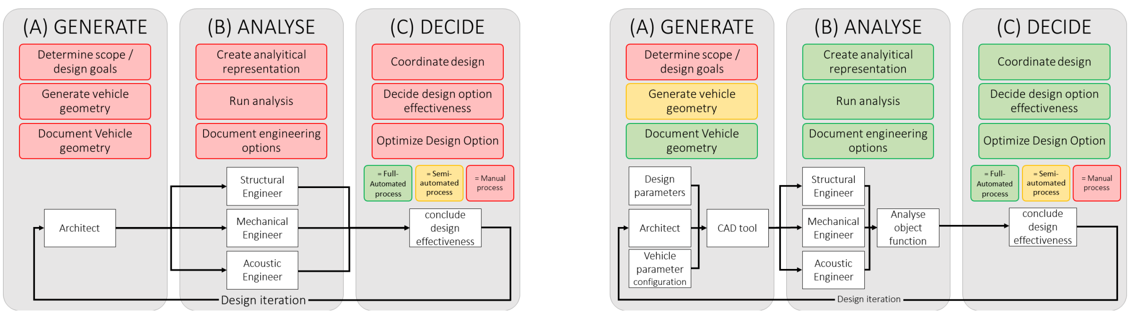
Introduction to MDAO Workflows

Understanding the MDAO principles is one of the key aspects of understanding why and how the production considerations should be involved within MDAO. This chapter introduces the generic MDAO workflows and large collaborative projects that have worked on these workflows.
In section 2.1, the history of MDAO is discussed, providing a brief comparison between conventional non-MDAO projects and MDAO projects. This section also discusses the barriers that have hindered the widespread adoption of MDAO within the industry.
Section 2.2 discusses the evolution of MDAO over the years, introducing three generations of MDAO. This discussion is followed by a summary of four collaborative projects (section 2.3) Subsequently, section 2.4 delves into the sub-tools created and used, particularly those relevant to this thesis. The chapter then addresses the shortcomings of current projects in section 2.5 before concluding with section 2.6.

2.1. History of MDAO and its evolvement through the years

. The design process has undergone a significant evolution over the past centuries. Initially, engineers worked in small teams, but with the increase of complex projects like the Boeing 747, involving over 4500 engineers, the design process became increasingly intricate. Modern product design processes involve multiple disciplines, stakeholders, and complex mathematical models, resulting in a lot of recurring tasks [3].
The period between the 1970s and 1990s saw a shift in complexity within these projects. Teams began incorporating new technologies like computer-aided design (CAD), and Finite Element Method (FEM), allowing for more elaborate projects and larger team sizes. These advancements increased the need for MDAO methodologies to effectively manage the complexities arising from these new projects.

2.1.1. MDAO system compared to the conventional design processes



(a) conventional design process

(b) Simplified MDAO process

Figure 2.1: Comparison of a conventional design method and the MDAO-process (figures based on Haymaker and Lee Flager [4]). The top coloured boxes illustrate the tasks involved per phase, as shown, much more tasks are automated (green) in the MDAO process. The bottom of the picture shows a flowchart with the design process

The design process can be generalized into nine different tasks divided into three phases, as illustrated in the top fig. 2.1a and fig. 2.1b.
In the *"Generate"* phase, the goals and initial geometrical model of the to-be-designed product are generated. Subsequently, in the *"Analyse"* phase, this initial model is shared with various disciplines (e.g., Structural engineers), and the performance of this product is assessed and reported.
Moving to the third phase, the reports generated in the analysis phase are evaluated to determine whether the design aligns with the created design goals in the first phase. If not, the design iterates by making alterations in the geometrical model and restarting the first two phases. This iterative process continues until a design is found that satisfies the requirements in the first phase [4].
The bottom parts of fig. 2.1 shows the workflow of this process and the communication lines.

CONVENTIONAL DESIGN PROCESSES

The conventional design process illustrated in fig. 2.1a is characterized by the work together of various departments, with all individual tasks colored red, meaning that all these tasks are performed manually. This manual approach is time-consuming and can result in miscommunications [4].
Research conducted Haymaker and Lee Flager [4] involved an extensive survey between architects and engineers. The survey aimed to determine (1) the number of possible design iterations within three months and (2) The relative amount of time spent on key tasks. As presented in fig. 2.2, the outcome reveals that (1) this conventional design process allows only a few iterations within three months and (2) management consumes over 50% of the entire process, this phase includes the communication and documenting, which is required for each iteration. This makes the overall design process highly ineffective.

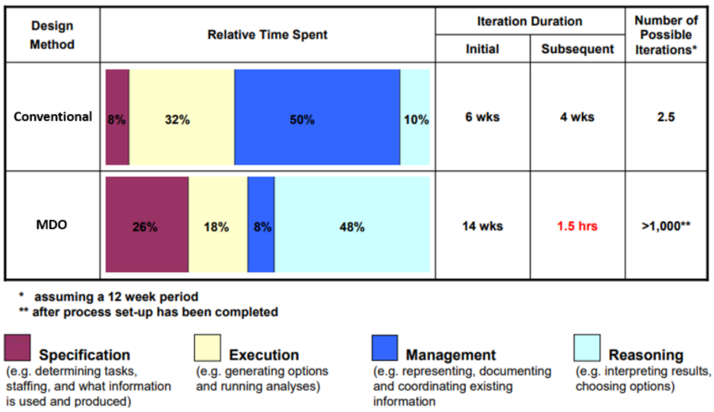


Figure 2.2: Comparison of legacy and MDAO process metrics for the design of an hypersonic aircraft [4]

CURRENT AEROSPACE PRACTICE WITH MDAO

Design processes with MDAO are fundamentally different compared to the conventional design process. Figure 2.1b shows the main components of the structure used in MDAO along with a flowchart of the MDAO process. A significant part of the tasks are automated (highlighted in green). Initially, in the *"generate"*-phase, the product is geometrized in a CAD tool which can be automatically shared with various automated disciplines, making the analysis more efficient by eliminating the need for human engineers further in the design process.
The research done by Haymaker and Lee Flager [4] showed that (1) iteration time reduced from weeks to hours, increasing the number of possible iterations in three months to thousands, and (2) management time reduced to 4%, while the specification time increased. The advantage lies in the fact that this specification only needs to be conducted at the beginning of the design process, as illustrated in fig. 2.2.

COMPARISON

Two design processes have been discussed: the "conventional" method and the "MDAO" method. As shown in Figure 2.2, an enhanced survey conducted by Haymaker and Lee Flager [4] found that in their example, the first phase of the MDAO process takes 14 weeks compared to the 4 weeks required for the conventional system. The longer time required is due to the project initialization and the set-up of the entire process. However, after the first iteration, the time per subsequent iteration is reduced since all tasks within the MDAO workflow are automated. This makes it possible to perform over 1000 iterations in three months compared to just 2.5 iterations for the conventional design process. Furthermore, according to scientists from Boeing Phantom Works [14], MDAO has the ability to improve the objective performance of innovative design by 8-10% and by 40-50% for novel unconventional concepts [15].

2.1.2. BARRIERS FOR THE USE OF MDAO IN PRACTICE

While the previous section showed the potential for efficiency increase, MDAO has not yet been used within the industry [3, 6, 10]. There are several reasons why MDAO and its capabilities are not implemented into the system, these are both technical issues and non-technical issues, both discussed below.

TECHNICAL ISSUES

Several technical challenges arise when considering MDAO:

- **Geometrical representation:** The to-be optimized or analysed object should be fully parametrized, which can make the less transparent for users. For example, representing complex shapes like NACA airfoils often requires alternative methods such as polynomial representations. Parametrization can be complicated and not well-established, adding complexity to the system engineering process.

- **Simulation tools:** Integrating various simulation tools from different disciplines poses in a challenge, as these tools may be proprietary and owned by third parties who may be reluctant to share information, and want the tool to be kept locally on their computer. Additionally, each discipline's analysis code may be written in different programming languages, and solvers may not be designed to be easily coupled to other disciplines [16].

- **post-processing and meshing:** Meshing, a critical step in simulation, is a complex task that requires significant human involvement. However, MDAO aims to automate this process, requiring a fail-safe and transparent system for accurate results.

- **Costs:** MDAO systems can be computationally expensive, as running these tasks can be demanding for normal computers. An expensive IT infrastructure is required to be able to set up and run these projects.

NON-TECHNICAL ISSUES

Non-technical issues are defined as those that are not related to specialized knowledge and/or methods used. Firstly, a set of cultural-determined issues arise: MDAO is a non-traditional innovative tool and is not a product of a company, thus the set-up, implementation, and troubleshooting have to be done internally with own in-house experts, which are hard to acquire. Furthermore, engineers can perceive MDAO as a potential threat, as it has the capability to take over human jobs. Many companies are also focused on cost avoidance and are reluctant to invest in new technologies if they are not fully proven yet. Consequently, when budgetary cuts have to be made, the MDAO is de-prioritized. [6, 10].
Beyond these cultural issues, the "black-box" perception is generally negatively seen. People prefer to verify what steps the program does, this leads to one of the major drawbacks: when engineers can't follow the steps taken into the system, they tend to have less confidence in the outcome. As Shahpar [6] stated: *"MDAO can be perceived as a mix of magic, hope, and hype all wrapped in a tidy yet unverifiable program"*.

2.2. EVOLUTION OF MDAO THROUGHOUT THE YEARS

While the essence of the MDAO systems has remained relatively constant, following the same three phases for set-up: setup, operational, and solution, as illustrated in fig. 2.3

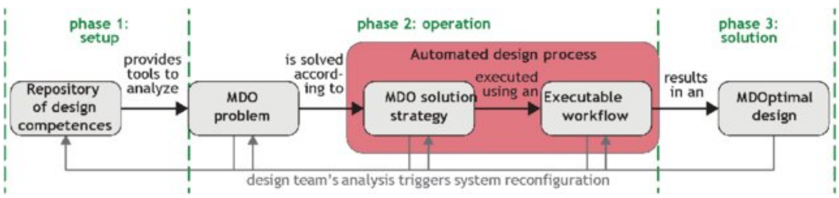
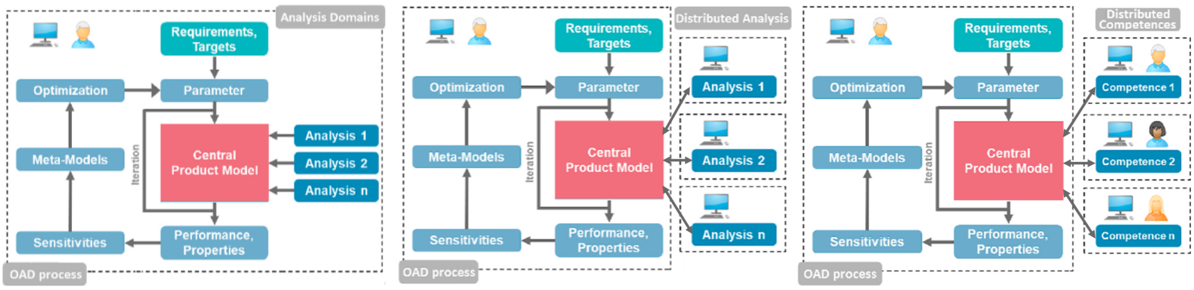


Figure 2.3: Three phases of the MDAO workflow [5]

In the **setup phase** the formulation of the design task and the definition takes place. Design drivers are selected, and the required IT infrastructure is established to facilitate efficient data transfers between all systems. Followed by the **operational phase**, this phase transformed the formalized workflow into a workflow that can be executed. Human judgment is often involved to verify the system and assess the correctness of the results generated. In the **solution phase**, optimization is executed, and convergence is sought after. The primary goal of this phase is to find a robust and optimal solution through an iterative efficient.
While the general structure remains consistent, significant improvements have been made aimed at mitigating the challenges discussed in section 2.1.2. Three generations of MDAO have come by, each introducing their own improvements. These generations are briefly discussed below [2]:



(a) First generation: Monolithic system (b) Second generation: Shared system (c) Third generation: Shared knowledge

Figure 2.4: Overview of the three generations of MDAO [2]

- **Generation 1:** The original monolithic MDAO workflow (as shown in fig. 2.4a) featured tightly integrated disciplines and was typically created by one central team or person known as the "architect"[1]. They were required to possess expertise in each analysis discipline or have the ability to transform a discipline into a workable tool.
- **Generation 2**: This generation introduces dedicated experts responsible for their respective disciplines, with a single team overseeing the design of the MDAO workflow (as shown in fig. 2.4b).
- **Generation 3:** Building on the second generation, Generation 3 of MDAO further distributes design tasks, allowing not only the analysis capabilities but also the entire design task. When optimization is run, the script automatically sends required inputs for an analysis tool to a third party, which then runs their own analysis locally, sending back the required outputs. This is all done automatically while sensitive data is kept internally. This distributed approach enhances human judgment and allows for easy verification of data by third parties[2].

The third generation MDAO systems still had some major challenges. As discussed in section 2.1.1. In this example, the first iteration of a hypersonic vehicle design tool 14 weeks using an MDAO approach, which is over twice as long as the conventional design process. The complexity that comes with the creation of an MDAO project make it less interesting for companies. Various collaborative initiatives have been undertaken to advance the field of MDAO, gaining notable success. The following section discusses a few of these projects.

2.3. COLLABORATIVE PROJECTS

To overcome the challenges of MDAO practices, such as long and complex iteration times, various collaborative initiatives have been initiated. The overarching DEE framework, upon which this thesis is built, falls under the umbrella of AGILE and DEFAINE, both discussed in section 2.3.3 and section 2.3.4, respectively. The DEE framework itself will be elaborated in chapter 1. This section will discuss a few of the collaborative projects created:

2.3.1. IDEALISM - 2017

IDEaliSM (Integrated & Distributed Engineering Services Framework for MDO) is a European project under the ITEA2 program comprises of 14 partners across 5 countries. The main goal of this project is to re-

[1] The architect is the main responsible team for the specification of the design case and is the main lead/responsible throughout the process [17]

duce time-to-market and development costs by integrating people, processes, and technologies within the MDAO workflow. This is done by changing the production development process by enabling integration of distributed- and specialized development teams [18]. The IDEaliSM framework exists of three main building blocks: the Advanced Integration Framework (AIF), the Engineering Language Workbench (ELW), and the Engineering Library (EL). In the first phase ("build cases"), a team of specialists utilizes the AIF and ELW to prepare the general workflow, while engineering services are prepared for the "Configure/execute" phase (see figure Figure 2.5).

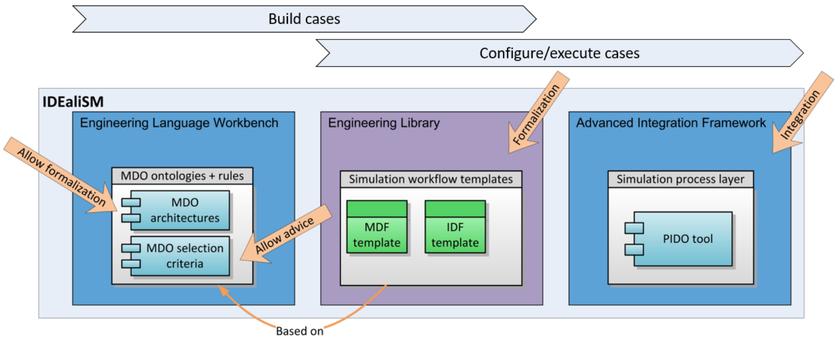


Figure 2.5: Graphical representation of the IDEaliSM framework, picture based on [18]

The blocks shown in fig. 2.5 contains the following information:

- **Engineering Language Workbench:** The ELW consists of different development toolkits, which can include various analysis toolkits such as MATLAB, Python, and ParaPy.
- **Engineering Library:** The EL serves as a bridge between the "build cases" and "configure/execute" phases. It generates different generic templates, engineering services, and data standards include the ELW into the MDAO workflow.
- **Advanced Integration Framework:** The AIF generates, applies, and reuses the engineering knowledge based on the ELW. It aims to support the integration of processes across different teams.

The IDEaliSM project is validated by a test use case, which demonstrated a significant lead time reduction of over 90% in the specific test case for building large workflows [2].

2.3.2. FrEACs - 2017

FrEACs (Future Enhanced Aircraft Configurations) is a project performed under the German Aerospace Center (DLR), it is a collaborative project involving eleven departments. The main aim of FrEACs is *"to quantify uncertainties in the design process for new aircraft and to apply them to the design of two unconventional configurations"* [11], where both a strut-based configuration and a wing blended wing body was researched. These configurations are researched with inputs from specialists in four disciplines; aerodynamics, loads, structures, and aeroelasticity [11].
In the FrEACs project, specialists from these four disciplines create tools that are incorporated through automated integration. However, the distributed workflow requires manual building, which is a time-consuming and error-prone process. According to Erwin Moerland, the project lead for FrEACs, "one of the key lessons learned was that the setup of the collaborative workflows including leaping over the associated hurdles along the way consumed a large part of the available project time"[3].
Despite the challenges in setting up collaborative workflows, FrEACs have proven to be a promising tool. It effectively fulfilled its objective of designing two non-conventional aircraft with respect to the aforementioned disciplines.

2.3.3. AGILE-PARADIGM - 2020

The AGILE (Aircraft 3rd Generation MDO for Innovative Collaboration of Heterogeneous Teams of Experts) project is built further up on the third generation of MDAO, aiming to improve workflow efficiency and sim-

[2]https://idealism.ifb.uni-stuttgart.de/, accessed on 10/02/2024
[3]Retracted from an interview by FrEACs project lead Erwin Moerland, on April 4th, 2019 performed by van Gent [19]

plify the design process. Over a period of 15 months, AGILE successfully addressed and optimized seven MDAO projects. demonstrating time savings of over 40% in the setup phase and 20% in the convergence time compared to the 2nd generation MDAO environment as illustrated on fig. 2.6 [2].

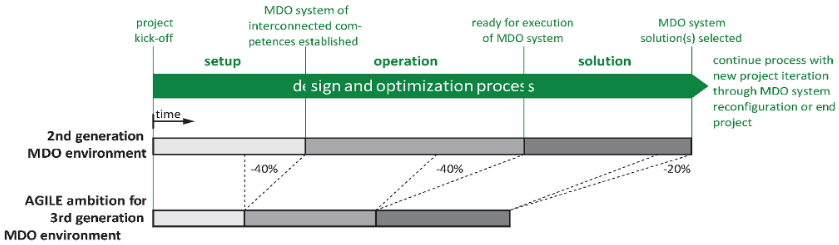


Figure 2.6: Conceptual overview of the AGILE-paradigm, picture based on [2]

KNOWLEDGE ARCHITECTURE OF AGILE

The Knowledge Architecture (KA) of the AGILE paradigm exists of a four-layer structure, as illustrated in fig. 2.7.
Firstly, the top layer defines the **development process**, in this phase, all tasks are required to be defined, monitored, and managed in one overarching process. This layer controls all other applications used within the rest of the KA. The intermediate layer **automated design** provides the formalizes the computational architecture, this architecture defines the exact path that needs to be followed.
The bottom layer **design competences** is the layer where the actual disciplinary tools or MDAO tools are stored, these include CFD or FEM tools. The last layer is the **data schemes**, the information placed in this layer is used by the workflow to effectively come to the required solution.

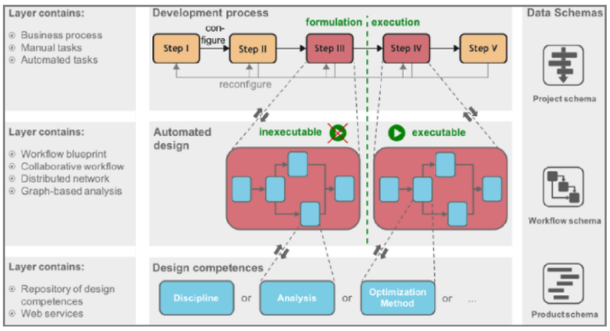


Figure 2.7: The AGILE Knowledge Architecture of a supported design in heterogeneous teams of experts [5]

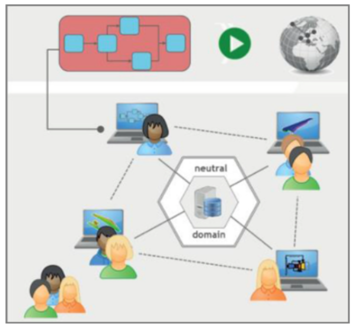


Figure 2.8: Execution phase with distributed competences [5]

Figure 2.8 illustrates the generalized concept of distributed competences, where multiple specialized companies, spread across the world, collaborate from a neutral domain. This domain ensures homogeneity in both the utilized and delivered results, thereby preventing errors, such as discrepancies between inches and millimeters.

STEPS OF AGILE - 2023

An important aspect of the KA lies in its formalized and clear structure, involving multiple agents[4] involved. Figure 2.9 shows the five stages of the AGILE development process layer, these five steps combined cover the entire process, from the initial set-up to the delivery of the solution.
This figure shows the steps, description, agents involved, deliverables, and the MDAO support applications which will be discussed after this paragraph.

[4]An agent is a heterogeneous stakeholder within the MDAO workflow

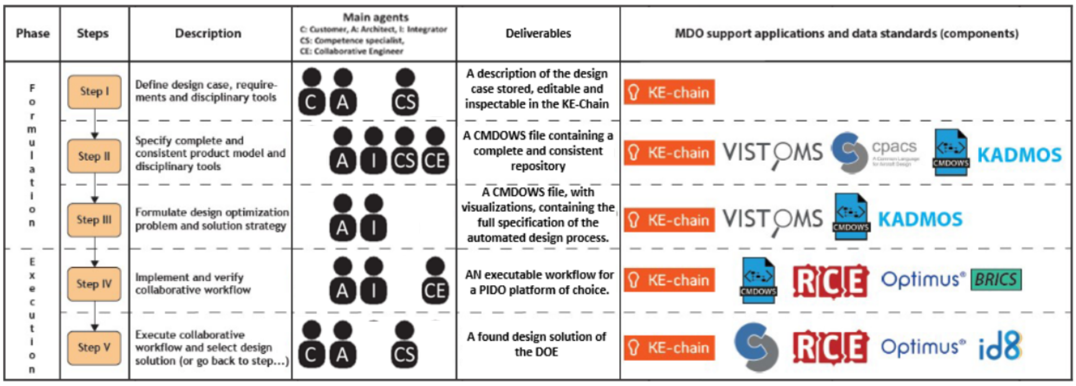


Figure 2.9: Five stages of the AGILE development process layer picture based on [2, 5]

- Formulation phase
  - **Step I: Define design case and requirements** - The customer, architect, and compliance specialist set up the research objective and the top-level requirements.
  - **Step II: Specify complete and consistent product model and design competences** - The entire team analyses the consistency and validation of the data, and links to common product model from the data schema layer.
  - **Step III: Formulate design optimization problem and solution strategy** - The automated design process is formalized based on the requirements.
- Execution phase
  - **Step IV: Implement and verify collaborative workflow.** The formulated architecture is translated into an executable workflow for the PIDO platform.
  - **Step V: Execute collaborative workflow, select design solution(s), and/or go back to an earlier step for reconfigurations** This final phase involves running the designed MDAO system and generating the results. It can either deliver an optimized or analysed product or the necessity of adjustments in the workflow.

The figure shown in fig. 2.9 also highlights the KE-Chain applications used within the AGILE framework. These applications, such as KADMOS, VISTOMS, and cpacsPy are used to provide essential functionalities. PIDO Tools (Process Integration and Design Optimization) (RCE, Optimus, and OpenMDAO) are used for the integration of executable computational frameworks.

2.3.4. DEFAINE - 2024

The DEFAINE project (Design Exploration Framework based on AI for froNt-loaded Engineering) is a collaborative project under the ITEA4 program. The main goal is *"deliver an advanced design exploration framework which reduces recurring design costs and lead time for design updates"*[5].
DEFAINE focuses on efficiently exploring large-scale designs by incorporating machine learning capabilities to analyse the designs further. By integrating front-loaded principles into the systems, project teams can mitigate the inefficiencies of the design process, resulting in a projected reduction of recurring costs by 10% and lead-time by 50%.
DEFAINE offers an effective method of this front-loading of the system such as the "AI-Enriched" development framework as shown in fig. 2.10. This novel front-loaded development process allows engineers to explore over 10 times as many options within the same timeframe.

[5]Retrieved from defaine.eu/projects-details, accessed on the 29/01/2024

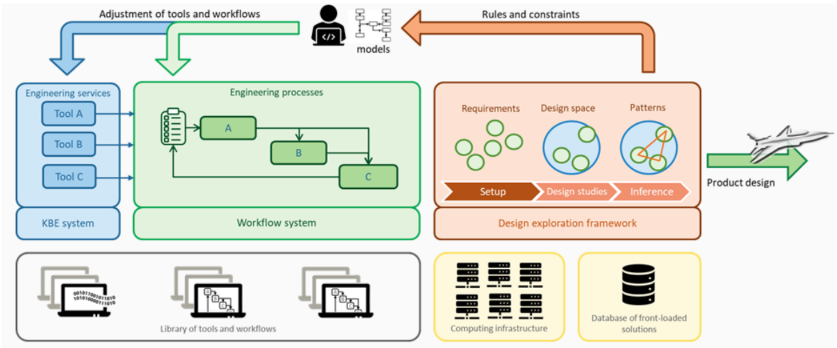


Figure 2.10: DEFAINE framework architecture [20]

The aforementioned front-loaded development process exists of four solutions:

- Integrating knowledge-based engineering (KBE) methodologies that accelerate the automated design solutions that can be adjusted. This is done by combining KBE with MBSE to enhance the traceability and ease of verification and validation.
- A tool capable of automatically (re)formulating MDAO workflows based on the design requirements.
- A virtual computing infrastructure that is both scalable and cost-efficient.
- Machine-learning and AI methodologies for identifying trends within large datasets, thereby increasing the number of analysed designs within the same timeframe.

2.3.5. CONCLUSION

Novel projects within the world of MDAO have made significant improvements in the design process, from enhancing user-friendliness to reducing lead times. While projects like IDEaliSM may still require a considerable amount of manual work, lessons learned from such projects have been integrated into newer frameworks like AGILE and DEFAINE, these two projects, delivered in 2023 and 2025 respectively, have shown an increase in efficiency of the setup time and in user-friendliness.

In the timespan of these projects, numerous sub-tools have been developed by various partners, such as KADMOS, CMDOWS, RCE, and Optimus, which find applications across multiple tools. For instance, KADMOS is utilized in IDEaliSM, AGILE, and DEFAINE [19].
While many collaborative projects have focused on improving a part of the MDAO process, like improving the efficiency or decreasing the setup time, none have fully covered the entire MDAO process from requirement setup to requirements verification, therefore the Design and Engineering Engine is created by Bruggeman and La Rocca [13], this tool, which became part of AGILE and DEFAINE created a blueprint how MDAO workflows could be set-up, with the potential to include Design for Manufacturing and Assembly into the workflow. The DEE framework is elaborated in chapter 3.

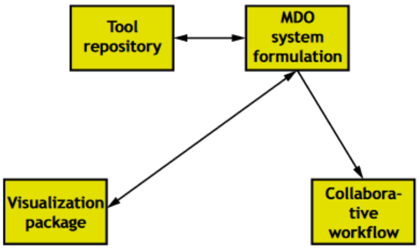
2.4. RELEVANT SUB-TOOLS IMPLEMENTED IN AGILE AND DEFAINE

Numerous tools are implemented into various collaborative projects, few of the tools are highlighted in this section, these tools include the tools necessary to run the workflow of AGILE and DEFAINE for this thesis project. This includes firstly the formal language used by the MDAO to store and exchange MDAO systems; CMDOWS (Common MDO Workflow Schema) [21] in section 2.4.1, followed by KADMOS, (Knowledge- and graph-based Agile Design for Multidisciplinary Optimization System), KADMOS is a manner to formulate MDAO systems [22] in section 2.4.2. Via CMDOWS, VISTOMS (Visualization Tool for MDO Systems) creates an interactive platform for how the structure of the MDAO workflow is visualized in [23] in section 2.4.3. After the workflow is set up and checked, it can be created and executed by a PIDO tool (Process Integration and Design Optimization), these are discussed in section 2.4.4.

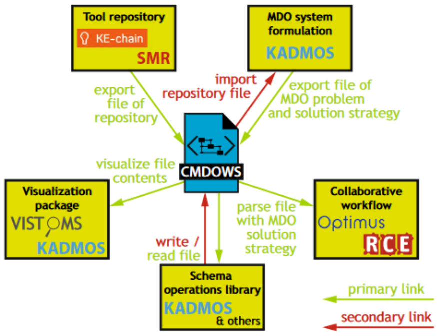
2.4.1. CMDOWS

During the setup of MDAO systems, the formal specification of such computational systems, a number of tools, data, and connections between the tools evolve. The open-source system CMDOWS (Common MDO

Workflow Schema) has been developed by Van Gent *et al.* [21] to streamline the process of combining all these tools. CMDOWS uses an XML-based data representation, meaning that any stored MDAO system is directly shared among the design team members and the applications, and therefore is on the basis of transforming the formulated MDAO workflow into executable workflows. CMDOWS transformed the structure from fig. 2.11a to the centralized structure of fig. 2.11b.



(a) Direct coupling approach



(b) Central workflow schema approach

Figure 2.11: Links between various MDAO framework categories [21]

The CMDOWS definition is created such that it is both machine-interpretable and human-readable whilst being neutral, meaning that it can be used throughout multiple MDAO projects while working in all three stages.

2.4.2. KADMOS

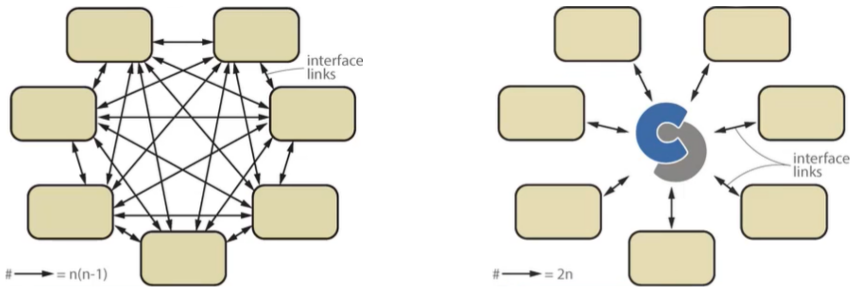
KADMOS (Knowledge- and graph-based Agile Design for Multidisciplinary Optimization System) is a tool designed to formalize a specification of the MDAO system before its actually implemented. KADMOS increases the agility of the MDAO scheme such that it can be easily assembled, adjusted and adapted to the design team needs [22] . KADMOS has three basic functionalities:

- Reduce the set-up time MDAO systems, especially for larger project with more disciplines.
- Enabling the design teams to inspect and debug the model if necessary. This increases the thrust into the system due to the increasing ability of validation.
- Manipulate the model such that optimization strategies can automatically be created and reconfigured.

KADMOS takes over a part of the formulation phase (as shown in fig. 2.9) where the formal specification is generated, this formal specification will later-on be transformed by a PIDO tool into an executable MDAO workflow. KADMOS exists of four main technologies:

1. Central data schema:

The concept of a central data schema is to have a standardized structure available that allows for a seamless flow of data through the workflow. Every tool has its own inputs and outputs, without a central data scheme, the information generated by one tool is shared with all other tools involved in the process, resulting in significant data traffic, as illustrated in fig. 2.12a. While this approach may suffice for small projects, the number of interconnections grows rapidly as the number of tools increases, since the number of links is specified by $\# = N(N-1)$, where N is the number of disciplines. Centralizing the data reduces the number of possible tool interfaces to the minimum, making the tool more efficient as shown in fig. 2.12b.
Various options for data management tools are created, but within KADMOS (for AGILE, IDEaliISM, and DEFAINE), CPACS (Parametric Aircraft Configuration Schema) is used [22]. This tool, developed by the DLR, serves as a collaborative common language between various tools [24, 25], consequently reducing the number of interconnection links from $\# = N(N-1)$ to $\# = 2N$.



(a) Old data scheme integration (b) CPACS Data scheme

Figure 2.12: Data scheme, the left picture shows the non-centralized data scheme, where all output information is shared with each discipline. The right picture shows the CPACS data scheme where the data is only stored at a centralized hub [25]

2. Knowledge based technologies:

The knowledge-based technologies within KADMOS store four different types of information: 1) The schema definition, this XML-based definition houses the central data scheme. 2) Tool into file, this file stores the meta-data of the tool such as the license information, accuracy, etc. 3) Tool input files, these files contain the data elements required to run the analysis, and latest 4) tool output files, the XML file containing those data elements which are written as output for the tool [22].

3. Graph-based approach:

KADMOS extends the graph-based approach created by Pate *et al.* [26], containing both the graphs belonging to the solution strategy and the connectivity between several tools. This graph-based approach enables users the easier creation, debug, and manipulate the model. The entire system overview is stored in the standardized format CMDOWS. However, these aforementioned graphs lack readability when projects scale up in size. Consequently, KADMOS is extended by VISTOMS (Visualization Tool for MDO Systems) to enhance the debugging and manipulation of large MDAO workflows, as will be discussed in section 2.4.3

4. Simulation Workflow packages (SWF):

SWF software packages enable the execution of the workflows, serving as the PIDO tools that are used to execute the workflow. Within KADMOS, both RCE and Optimus can be integrated, playing a key element in executing workflows across distributed networks. This integration allows for collaboration between various companies while ensuring the protection of intellectual property rights and security measures of all partners and disciplines.

Methodologies

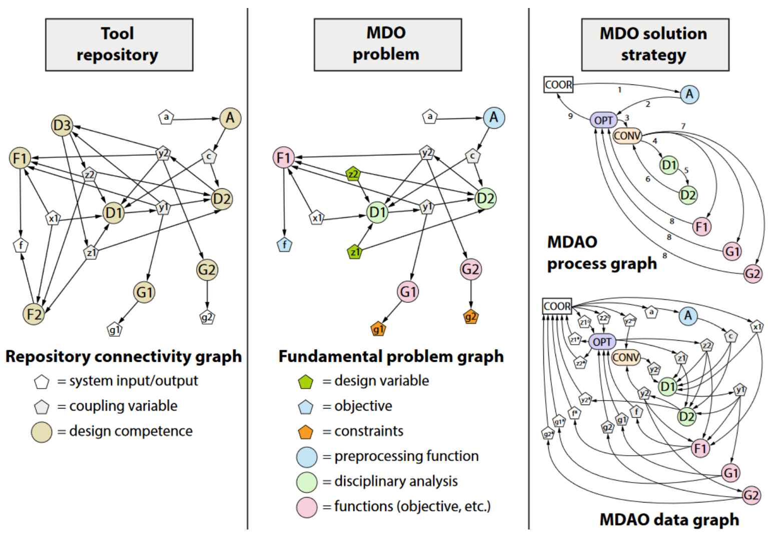


Figure 2.13: Top-level overview of KADMOS and its automatic creation of the workflow formulation. This process is discussed in Pate *et al.* [26] and based on Sellar *et al.* [27]

To formalize the workflow, KADMOS initially generates a Repository Connectivity Graph (RCG) (shown on the left in fig. 2.13). This graph represents the connections between inputs, outputs, and design competences, showcasing it as a web of data. For smaller datasets, the RCG is easily constructed using CPACS. However, as depicted in fig. 2.13, the RCG with only eight disciplines can already become quite complex, resulting in readability challenges for larger tools with more disciplines [23].
This RCG graph is transformed into the Fundamental Problem Graph (FPG), which extends the RCG with attributes required to specify the MDAO problem, such as the objective, constraints, and design variables. This graph is stored in two separate graphs (as shown on the right of fig. 2.13): 1) the MDAO Process Graph, which stores the process execution flow of all system components, and 2) The MDAO Data Graph, which specifies all data exchanges between components.

While KADMOS graphs contain all the information for visualizing the entire process, certain details are implicitly stored, which can be shown explicitly by VISTOMS. For instance, input and output variables are stored in the MDAO Data Graph, but when retrieving this information, one has to loop over all the connections corresponding to that tool. In contrast, VISTOMS can present this information more clearly.

In summary, KADMOS enables users to automatically create formalized collaborative workflows in a matter of minutes. CPACS enables effective data transfer increasing the computational efficiency. Additionally, KADMOS employs VISTOMS to make the workflow interactive, thereby increasing the flexibility and transparency for the user. Further details on VISTOMS are provided the following section.

2.4.3. VISTOMS

As previously mentioned, while KADMOS increases the ease of formulating, altering, and debugging the system. However, for more complicated systems, the graph-based approach lacks readability and transparency. Therefore, RWTH Aachen University developed a tool that complements KADMOS, enabling efficient visualization and inspection of MDAO system specifications. Figure 2.14 shows the top-level overview of how VISTOMS (Visualization Tool for MDO Systems) is integrated into the system. KADMOS is based on the CMDOWS files and the information is shared via JSON files to the VISTOMS visualisation tools.

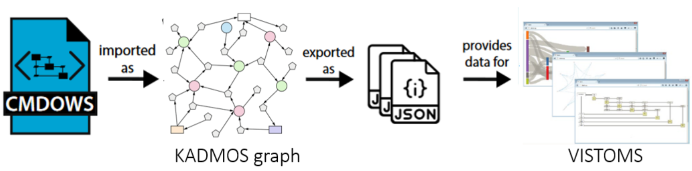
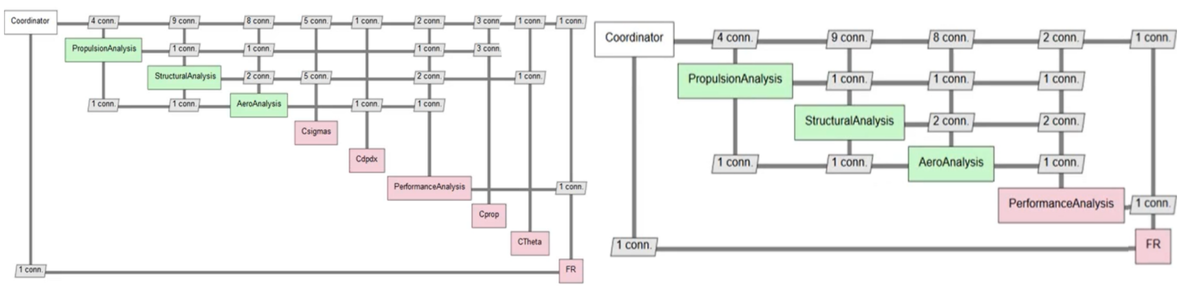


Figure 2.14: Top-level overview of the dynamic visualization approach [23]

VISTOMS has several advantages, including the ability to display system information through three interactive graphs, directly the formulated workflow definition. This allows users to directly identify unused disciplines which can be excluded from the optimization, this is shown by fig. 2.15. Besides the XDSMs, VISTOMS also generate[6] an Edge Bundling Diagram and Sankey diagrams. These tools effectively check and adjust the preferred path required by the MDAO system, enhancing tool transparency.



(a) XDSM with unused competences created by VISTOMS (b) fundamental problem graph

Figure 2.15: XDSMs adapted by VISTOMS to remove the unused competences [28]

[6]https://www.agile-project.eu/files/VISTOMS_SellarProblem/, accessed on 10/02/2024

2.4.4. PIDO TOOLS

After the workflow is specified by KADMOS, the execution phase begins, during which the formulated workflow is transformed into an executable workflow by the so-called PIDO tools (Process Integration and Design Optimization). Various PIDO tools can achieve this, including Optimus, OpenMDAO, and RCE which are discussed below. Each of those PIDO tools shares the common goal of using the formalized system output in CMDOWS created by KADMOS to create an executable MDAO workflow. All PIDO platforms provide a graphic user interface (GUI) for assembling and executing MDAO systems [19].
Note: While there are numerous PIDO tools available, this research focuses on discussing a select number of tools.

RCE - DLR

RCE (Remote Component Environment) is an open-source software developed by the DLR to assist users to create, manage en execute complex workflows[7]. It allows certain tools within the workflow to run on the host's computer, enabling effective integration of sensitive tools and data sharing. A workflow in RCE is defined by multiple components, each with its own inputs and outputs connected to each other. Standard connections are established between multiple components (e.g., simulation tools) within RCA, some standard connections can be found in a library, like the optimizer. Furthermore, the user can define its own tools, enhancing scalability. RCE integrates the defined but non-executable workflow from KADMOS via CMDOWS in a workflow that can be run [29]. RCE supports both optimization and design of experiments [30].

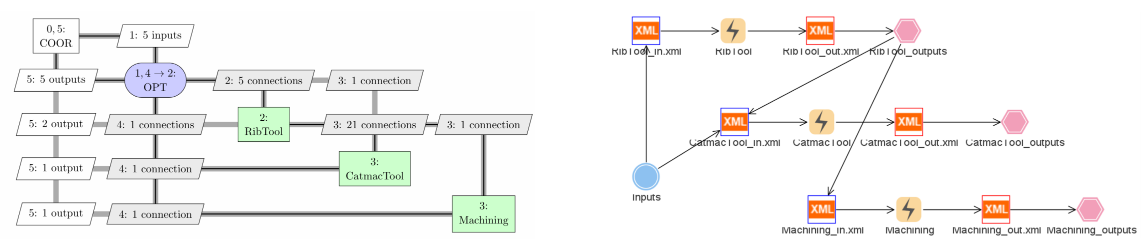
OpenMDAO - NASA

Originally designed by NASA, OpenMDAO is an open-source software specifically designed for MDAO. OpenMDAO includes optimization capabilities and is the only PIDO tool that includes a library for standardized MDAO architectures. However, a downside of these standard MDAO architectures is the requirement for extensive manual programming when integrating new MDAO architectures or when external tools need to be incorporated into the workflow [8].

Optimus - Noesis Systems

Optimus is a commercial package developed by Noesis Solutions and is utilized within AGILE. Through a GUI, users can create workflows, while Python scripts allow for customization according to individual preferences. Optimus automatically integrates various disciplines such as MATLAB and Excel. Its architecture distinguishes between simulation workflow and design methods (such as optimization or DOE), facilitating the reuse of the same workflow for different calculations, and making it easy to adapt to different user cases.[9].
Apart from these benefits, Optimus manages and executes the entire workflow and can integrate various tools located on different servers, enabling seamless data flow. It also supports multi-objective iterations.

Optimus transforms the formulated workflow by KADMOS where VISTOMS create the XDSM as illustrated in fig. 2.16a, into an executable workflow. This workflow can be shown with the Optimus integrated workflow graph (fig. 2.16b).



(a) Simplififed XDSM (b) Optimized implementation

Figure 2.16: Simplified XDSM with the Optimus implementation [9]

[7] https://www.dlr.de/sc/en/desktopdefault.aspx/tabid-5625/9170_read-17513/, accessed on 01/02/2024
[8] NASA - OpenMDAO, https://openmdao.org/, accessed on 01/02/2024
[9] Optimus - Noesis Systems, https://www.noesissolutions.com/our-products/optimus, Accessed on 01-02-2024

Conclusions All the PIDO tools mentioned above provide a GUI, which enhances usability and reduces the black-box nature of the process by increasing the transparency. The tools ensure that the workflow can be executed effectively and that various tools and disciplines are integrated within the workflow. table 2.1 shows the advantages and disadvantages per discussed PIDO tool.

PIDO Tool	Access	Advantages	Disadvantages
RCE	Open source	• Multi-purpose • Can be run on host devices • Supports DOE and optimization	• Significantly slower [31] • No API is provided
OpenMDAO	Open source	• Fast • Standardized MDAO architectures • Python-build	• Manual programming extensive • Less flexible
Optimus	Commercial	• allows multi-objective optimizations • Less uncertainties compared with open-source • Can be run on external devices	• Commercial[10]

Table 2.1: Advantages and disadvantages of the discussed PIDO tools

.

As discussed in table 2.1, both OpenMDAO and RCE have large disadvantages over the commercial tool Optimus. While RCE is more user-friendly compared to OpenMDAO, it is significantly slower [31], while Optimus combines these two advantages in a tool which is both flexible and computational fast.

2.5. LIMITATIONS

The various collaborative projects discussed in this chapter have demonstrated the potential to reduce lead time and enhance user-friendliness in MDAO systems. As a result, some of the limitations discussed in section 2.1.2 are mitigated. However, some overarching relevant limitations remain for the inclusion of Design for Manufacturing and Assembly (DfMA) into MDAO workflows. **Note:** The limitations discussed down below are from collaborative projects without the inclusion of the Design and Engineering Engine (DEE), the objective discussed in this literature research extends this DEE and so it will be discussed separately in chapter 3.

1. **Exclusion of production considerations:** Manufacturing and assembly considerations are neglected within the presented collaborative projects. This means that once the product is optimized, further discussions regarding production considerations are necessary, which can increase costs and lower product performance.

2. **Static workflows:** Generated workflows in one of the discussed projects are static, meaning that workflows remain constant when run, and the same path is followed at any stage of time and iteration. When non-continuous variables are used as design variables (such as material choice), the system will always perform the same analysis, which can result in discrepancies.

3. **Data handling:** Increasing the number of parts skyrockets the number of possibilities. For example, in a study by Raju Kulkarni *et al.* [32], a seven-part product optimization resulted in a total of $205e^6$ different options. As projects scale up further, this number of possible options increases, necessitating effective data handling to prevent feedback couplings and effective coding, keeping the required computational power minimized.

To address production considerations and enable non-continuous variables to alter the workflow, the DEE has been established. This research, which extends the capabilities of both DEFAINE and AGILE, is researched by Bruggeman *et al.* [8] and further elaborated in chapter 3.

2.6. CONCLUSION

MDAO is a widely proven and researched design methodology, promising increased design efficiency and reduced time-to-market for products. It has the potential to improve optimization objectives by 8-10% for innovative aircraft and 40-50% for novel unconventional designs. Despite research spanning over thirty years, MDAO has not been widely adopted within the industry due to various technical and non-technical problems.

Over the years, MDAO has evolved through three generations, with a growing emphasis on a shared data scheme. This data scheme allows disciplines from third parties to be run locally on the computers of the respective companies, enhancing transparency, data security, and ease of data sharing.

One of the largest challenges of MDAO is the complexity involved in setting up the workflow, research showed that this process takes way longer compared to setting up conventional non-MDAO workflows. Collaborative projects such as IDEaliSM, FrEACs, AGILE, DEFAINE, and others, have shown significant improvements in this challenge. Each of these projects follows a similar three-phase workflow: setup, operation, and conclusion of the optimal design. During the setup phase, the design competences are researched, followed by the operation phase where the problem is defined and transformed into an executable workflow, and lastly, during the solution phase, the optimal design is concluded for compliance, and the project is re-iterated or finalized. Various sub-tools have been developed to support these steps, with AGILE and DEFAINE employing similar tools which will be used in the proposed thesis in this literature research.

CMDOWS serves as a data format used by the workflow to centrally store, house, and share details of the MDAO system. Via KADMOS, various tools can communicate via a common-language. KADMOS formalizes the structure of workflow by connecting the outputs and inputs of all disciplines to each other. VISTOMS plays an important role within KADMOS to give the user a clear overview of the process and a way to evaluate user demands. Once the workflow is formalized, it can be transformed into an executable workflow by using PIDO tools such as RCE or Optimus. Both AGILE and DEFAINE have chosen the commercial tool Optimus for executing workflows due to its user-friendly GUI and the ease of how capabilities are added.

However, despite many improvements, the discussed collaborative projects still had some issues, especially with the static nature of MDAO workflows, where the same exact route is followed every iteration. This poses a problem as if not disciplines are relevant for all designs. Additionally, DfMA considerations are excluded from these workflows, resulting in design changes after the optimization is run. To address these challenges, the DEE has been developed, on which this thesis will build upon.

3

DfM; an overview of the State-of-the-art MDAO workflow

The frameworks in presented in section 2.3 often do not adequately consider the production during product analysis and optimization. Manufacturing and assembly, while challenging to include due to their reliance on expert experience and often limited documentation, have significant impact the costs and performance of products, as design adjustments are often necessary to ensure feasibility. Bruggeman *et al.* [8] developed a new methodology that enables the inclusion of both manufacturing and assembly into the design process, forming the foundation of this thesis. This chapter will provide an analysis of this framework, including its relevant sub-tools.

This chapter begins by discussing the reasons why production considerations have not yet been included (section 3.1), followed by an overview of the proposed DEE blueprint (section 3.3). Subsequently, section 3.5 introduces the *"Manufacturing Information Model"* (MIM) and section 3.4 *"the requirements management module"* (RMM) in section 3.4. Then, the DfM inclusion is discussed (section 3.7), and finally, section 3.6 introduces CATMAC, a tool designed by GKN Fokker for cost estimations.

3.1. Problems with the inclusion of product considerations

Although the inclusion of production considerations may seem straightforward, it is far from simple. Numerous rules must be integrated to include the DfMA problems, such as joining methods, and material comparability which can directly affect the design. For instance, various assembly methods can affect geometrical requirements like required thickness. The significance of the absence is evident in conventional MDAO processes. Exclusion of DfMA consideration can lead to 1.) A higher risk for design changes at later stages of the development process, which may lead to higher costs and time delays, especially in novel aircraft design [33]. And 2.) to mitigate the risk of 1.), design choices are made by specialists who often choose conservative options to ensure feasibility, potentially hindering the efficiency [34].

Conventional design processes incorporate production considerations via two methods: First and foremost, production considerations are made manually, where information is retrieved from 1.) documents like guidelines, and best practices, 2.) discussions with manufacturing and test engineers, and 3.) through their own experiences. However, this method has drawbacks, including a high risk of errors and reliance on engineers' experiences, which are subjected to change and time-sensitive [33]. Consequently, designs generated through this method may not always be the most optimal design. The second method involves automated approaches, which are often done by either very specialized, commercial tools or still require human involvement for certain processes, rendering them not fully automated. The inclusion of the DfMA problem into conventional MDAO workflows was one of the primary objectives of the research on the DEE [13], which is explained in the following section.

3.2. Design and Engineering overview

The thesis extends the Design and Engineering (DEE) framework presented in Bruggeman and La Rocca [13]. While most projects discussed in chapter 2 focused on specific aspects of the MDAO process, such as the setup phase (e.g., AGILE and DEFAINE) or numerical uncertainties (e.g., FrAECs [11]), the DEE stands out as a generic blueprint for setting up the entire MDAO process. The DEE defines the workflow from requirement set-up to requirement validation. The DEE incorporates findings from previous collaborative projects and has been performed in the framework of AGILE 4.0 (successor of AGILE) and DEFAINE.

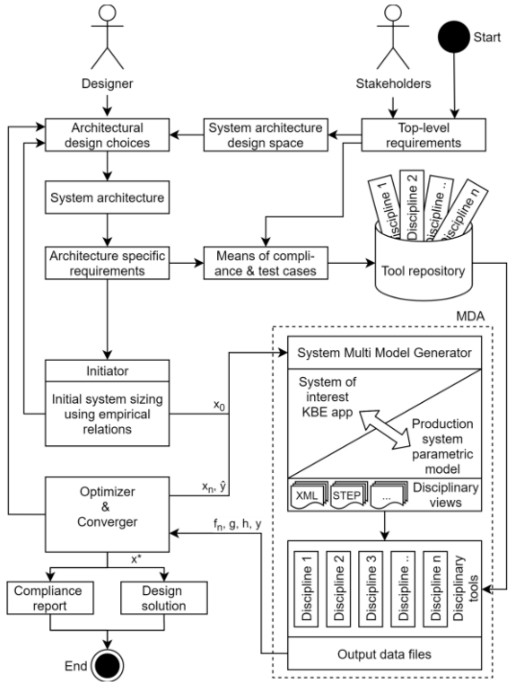


Figure 3.1: Overview of the DEE for simultaneous product and production system design and optimization [13].

The DEE is the only framework available that combines requirement management, automatic requirement verification, MDAO workflow formulation, and execution of the workflow. Figure 3.1 provides an overview of the DEE simultaneous aircraft and production system design and optimization.
Initially, stakeholders (clients) define top-level requirements, which are then formulated into machine readable requirements. These requirements are incorporated into the MDAO workflow to ensure compliance-by-design. When architectural design choices are generated, different specific requirements become active, and therefore different disciplinary tools will appear in the MDAO workflow. The workflow generates a KBE model, utilized in different disciplines to calculate required outputs. All data is provided to the converger, which checks for design convergence. If necessary, new design choices are considered, changing the requirements and the workflow until convergence is reached.
The DEE consists of multiple sub-tools, as illustrated in fig. 3.2. The inner ring shows all (relevant for this thesis) functionalities incorporated within the DEE, while the outer ring indicates the tool that performs these functionalities. Notably, the inclusion of assembly considerations is defined as the objective of this thesis.

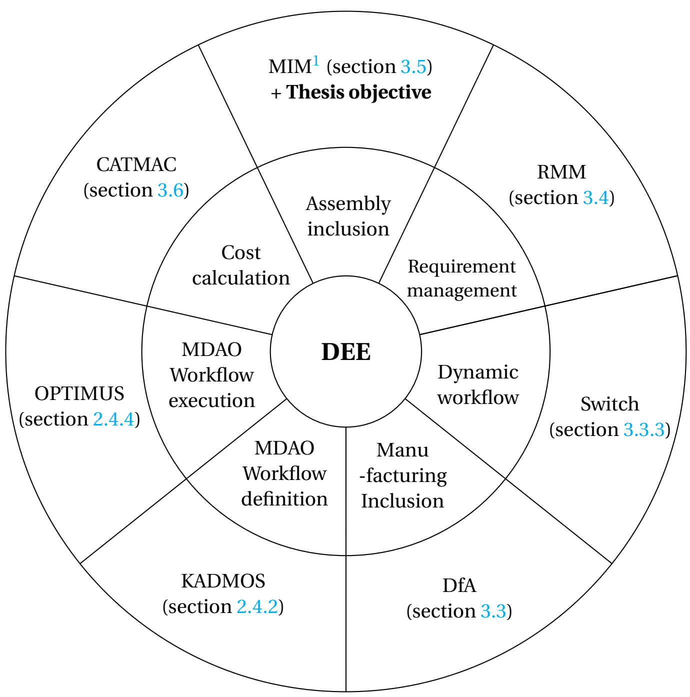


Figure 3.2: The Design and Engineering Engine with it subtasks that are discussed in this chapter, the middle ring contains the sub-tasks and the outer ring gives the corresponding tool and the reference where the tool is discussed, as highlighted in the top of the picture, this thesis researches the assembly inclusion into the DEE

3.3. METHODOLOGY OF DFMA INCLUSION WITHIN THE DEE

The integration of the DfMA considerations within the DEE is outlined in Bruggeman *et al.* [8], this outline is shown in Figure 3.3. Firstly, the System Architectural Design Space Model is set up (section 3.3.1), followed by the requirement inventarisation (section 3.3.2), and lastly the dynamic MDAO workflow formulation (section 3.3.3).

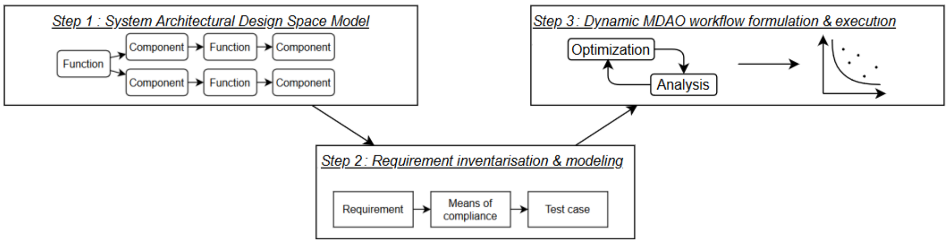
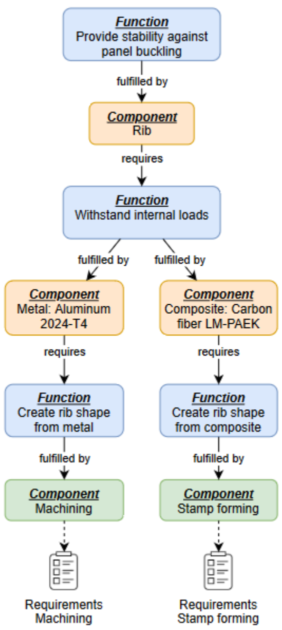


Figure 3.3: Schematic overview of the proposed methodology to add manufacturing models [8].

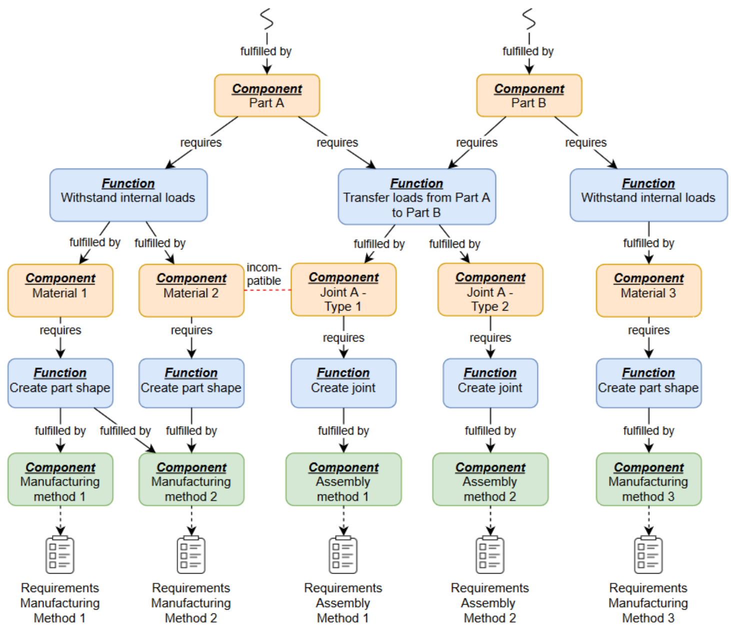
3.3.1. STEP 1: SYSTEM ARCHITECTURAL DESIGN SPACE MODEL

The first step involves creating the System Architectural Design Space Model (SADM) of the DfMA problem. The SADM incorporates the entire design space into the MDAO workflow, including all possible design options, materials, and production considerations (enabling DfMA). The SADM depicts the hierarchical structure of the designed object as shown in Figure 3.4a, using a wing rib as an example [8]. At the top level, main functions that need fulfillment, such as *"provide stability against panel buckling"* for a wing rib, are defined, this can be achieved by multiple components. Each component has its own derived function that must be fulfilled, continuing until all functions are satisfied.

> **Definition:** System Architectural Design Space Model
>
> This model is a conceptual model that defines the entire structure of the design options, eventually, a path is chosen that results in the most optimal design. The SADM shows the trade-off of all options possible.



(a) SADM-example of a wing-rib



(b) SADM-example for a generic two part primitive design,with incompatibilities links included to show infeasible options

Figure 3.4: System Architectural Design Space Models (SADM) [8]

Figure 3.4a focuses solely on the SADM of a single part primitive. However, an example of the inclusion of multiple-part primitives, as will be done in this thesis, is illustrated in fig. 3.4b. This figure displays two different "generic" parts and their assembly. The SADM incorporates incompatibility links that specify incompatibilities between options, such as a manufacturing method with a certain material or material incompatibility. This approach enables the workflow to make efficient trade-offs. For instance, when loads need to be transferred from part A to part B via a joint, the MDAO workflow can choose between Type 1 Joint and Type 2 Joint, allowing it to automatically select the corresponding set of requirements and observe the effect, choosing the most optimal design path.

3.3.2. STEP 2: REQUIREMENT INVENTARISATION & MODELING

While Step 1 defined the manufacturing system within the architectural design space, this step ensures that the workflow can automatically access relevant and the corresponding requirements depending on the chosen design. This step guarantees the same source-of-truth for requirements, facilitating direct accessibility by the system. The implementation of this step is done via the Requirement Management Module (RMM) which is elaborated in section 3.4. As the manufacturing and assembly methods impose different requirements on the product, each method (both manufacturing and assembly) should have its own requirement database (as illustrated in fig. 3.4b).

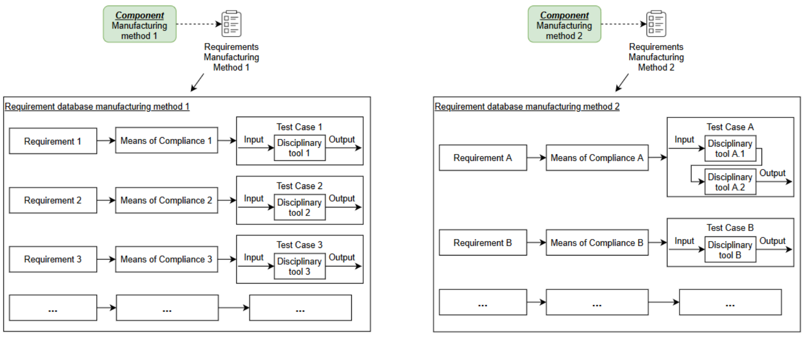


Figure 3.5: Generic modelling of the requirement databases for each of the manufacturing method. Picture retrieved from [8]

3.3.3. STEP 3: DYNAMIC MDAO WORKFLOW FORMULATION & EXECUTION

After the RMM is incorporated, the full workflow can be formulated and executed into an MDAO framework. Depending on the chosen "path" by the MDAO workflow, different requirements may become active, thereby altering the design variables, disciplinary tools, and constraints. Changing these tools and constants can be done manually or automatically via dynamic MDAO workflows.

1. **Manual changing the runs (static MDAO workflow):** Initially, the optimization is run with production method 1, followed by a manual adjustment of the MDAO workflow with a change of production method. All results need manual comparison and human judgment to choose the most optimal design. This method is suitable for small, uncomplicated projects, such as part primitive design with a limited number of production methods.

2. **Creating a dynamic workflow:** This workflow has the ability to autonomously execute and choose the correct workflow depending on the design options. In the aforementioned example, it will automatically run all options for production methods and choose the best one, seamlessly "switching" between methods. This method is proven to be more efficient since fewer iterations are required and no manually switching needs to be done [8].

Dynamic MDAO workflows have advantages over static workflows since a larger design space can be covered without manual changing the runs. For example, when optimizing a part assembly consisting of five part primitives, each with five possible production methods and three assembly methods, a total of 75 different options can be generated. This makes static workflows less practical due to the sheer size of data used, 75 different options would result in 75 individual runs. The DEE is the first instance where this type of dynamic flexibility is included.
To enable dynamic MDAO workflows, three new components are introduced into the workflow [8]:

- **Sub-workflows:** These are workflows within in a workflow. In the top-level workflow, a block represents a lower-level workflow such as an analysis tool, a list of tools, or even an entire optimization workflow. An example is shown in fig. 3.6.

- **Branches:** Represented by mathematical functions, analyses, or sub-workflows. Branches are executed when a boolean condition at the beginning of the branch evaluates true. If the condition is met, the branch is executed, and if this condition is not met, the branch is skipped. This evaluation is performed by the "switch" Branches are located after the switch.

- **Switches:** Switches can activate different branches of a workflow depending on the variables of the design at that time. For instance, if the design variable "composites" is chosen, the switch will automatically select the manufacturing process associated with "composites". The outputs of the switch, termed *"Decision variables"*, can be True or False, determining which branch is executed during each iteration.

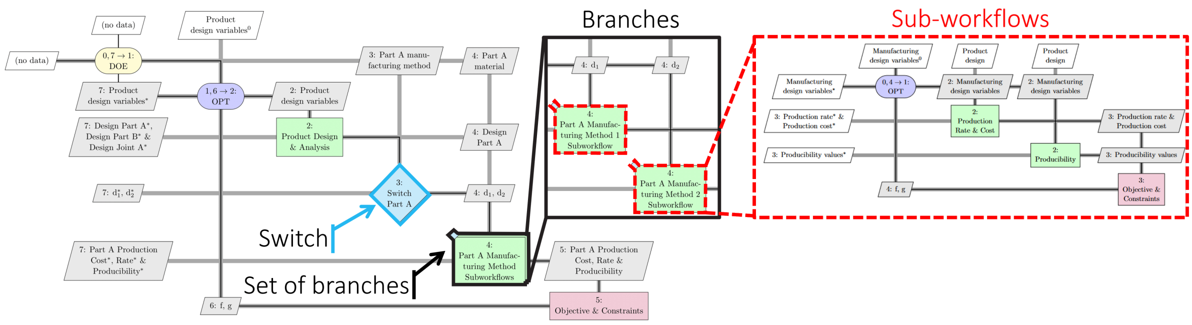


Figure 3.6: XDSM of a DfM problem. The blue diamonds are switches that could lead to either one of the disciplines located below it, these disciplines are depicted as a sub-workflow, highlighted in the red box.

In fig. 3.6, an overview of a generic DfM problem is shown with the three aforementioned parts included: The Switch, Branch, and Sub-workflows. The "blue diamonds" represent the switch where one of the branches underneath is chosen. The black box marked by *"Set of branches"* shows a stacked block with all the branches underneath it. In this example, there are two branches after the switch. Each of these branches can be an analysis, tool, or even an entire optimization as the sub-workflows on the right shows.

3.4. REQUIREMENTS MANAGEMENT MODULE (RMM)

Research by Smith [35] concluded that 47% of all unsuccessful projects are seen as not successful due to poor requirement management. Given the complexity of MDAO projects, it is important to systematically consider the requirements during the design process. Bruggeman *et al.* [36] proposed a framework that transitions the requirements from document-based to model-based using MBSE (Model-Based System Engineering), thereby making them machine-readable and interpretable. This integration of requirements into the design process enables their consideration during optimization, from setup to automatic compliance validation.
This model-based approach has several benefits. Firstly, it provides clear oversight and maintenance by establishing a single source-of-truth from which all necessary information is derived, thus enhancing consistency within the design process. Furthermore, it improves the reusability of the approach across multiple projects, enhancing overall efficiency. Lastly, it enhances transparency for the user regarding the final design and its capabilities.
This section explains how the RMM works. Firstly, it will provide a high-level overview of how the RMM is integrated within the MDAO process (illustrated in section 3.4.1), followed by the explanation of the automatic requirement verification in (section 3.4.2).

3.4.1. THE RMM AS PART OF THE MDAO PROCESS

The goal of the RMM is to integrate the requirements within the design process thus improving requirement traceability, hence, improving the transparency of the process to all agents. Furthermore, the RMM enables the process to automatically generate a compliance report of the product requirements.

The RMM consists of two main parts, as illustrated in fig. 3.7: The Automatic Requirement Verification (upper block) and the Requirements to Drive the Design Process (lower block).

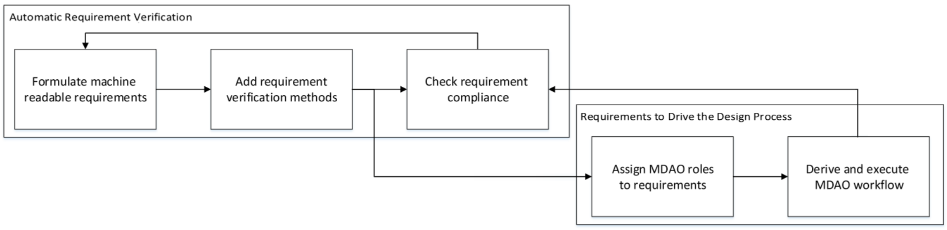


Figure 3.7: Overview of the RMM within the design process [36]

The first part of the RMM involves automatic requirement verification. Here, machine-readable requirements are set up along with their verification methods, defining what objective needs to be met for successful compliance. This information is then passed to the second part, where the MDAO roles are assigned to the requirements. This assignment enables the requirements to be used and taken into account by the MDAO workflow, allowing the information to be passed back to the first block, where an automatic compliance report is generated. These two blocks are more in-depth discussed in section 3.4.2 and section 3.4.3 respectively.

3.4.2. AUTOMATIC REQUIREMENT VERIFICATION

This subsection discusses the first part of the RMM, which involves the systematic set-up of automatic requirement verification within the system. In this phase, stakeholders' requirements are identified and derived into system requirements. Generally, there are two types of requirements: 1.) **Functional requirements:** Requirement that specifies a function that a system will perform. However, they are not quantifiable. 2.) **Non-functional requirements**, these are specified by ISO [37] as a *"measurable criterion that identifies a quality attribute of a function or how well a function requirement shall be accomplished.* Since only non-functional requirements are quantifiable, only those are taken into account in the RMM [33].
The systematic design of requirements involves three steps, as illustrated in the smaller blocks in fig. 3.7:

Step 1: Machine Readable Requirements

During the setup of the MDAO process, one of the first tasks for the project team is to collaborate with the clients to create a document outlining the overall requirements. This document-based list is then transformed into model-based requirements, following the systematic setup illustrated in fig. 3.8. This figure shows the various types of requirements implemented within the RMM. Performance requirements specify the desired performance that the part must achieve, while design constraints define limits on the feasible design space. Environmental requirements specify the condition under which the part must be performed. Sustainability includes all the '-ilities' such as maintainability, producibility, and reliability.

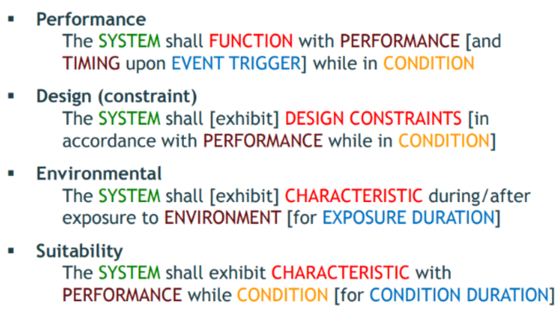


Figure 3.8: Requirements systematically set up [38]

By categorizing requirements into these four categories, the workflow can effectively read and interpret them, facilitating the creation of a design-for-compliance approach.
Step 2: Requirement Verification Methods
Each requirement created in the first step needs to have a corresponding verification method to demonstrate compliance with the design requirement [36]. This verification method consists of two parts:

- **Means of compliance:** An agreement between the client and the architect on how compliance will be achieved, with each requirement potentially having multiple means of compliances. These means may involve agreements regarding software tools, versions, assumptions, etc.

  - *Example:* The requirement will be verified with a stress analysis using FEM
    - *"The mesh size shall be 5 mm"*

- **Test case:** This represents the technical implementation of the means of compliance and includes all the models, tools, and physical tests, along with their required inputs and outputs. This can include FEM, or CFD analysis with the input of a CAD model. Every means of compliance can have multiple test cases.

An illustration of both the means of compliance is presented in fig. 3.9.

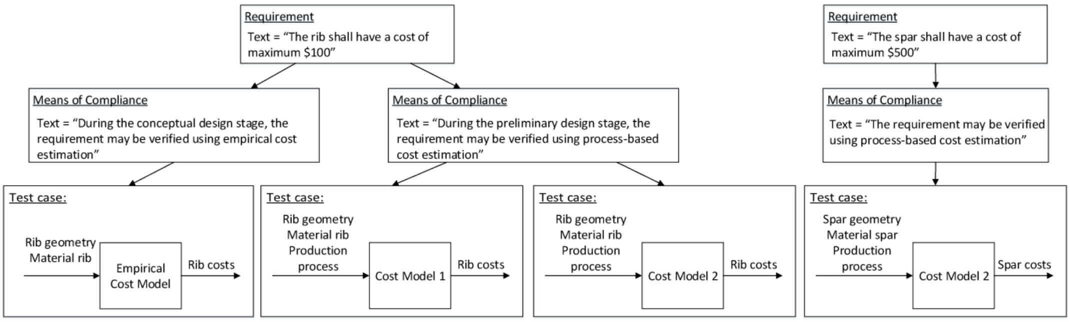


Figure 3.9: Example indicating the link between requirements, means of compliance and test cases. As shown, each requirement can have multiple means of compliances and every means of compliance can have multiple test cases [36]

Step 3: Compliance report

Once the requirements have been established and the script executed, an automated compliance report can be generated. This report combines both the first and second step. An example of such a requirement report is shown in Figure 3.10.

As shown, the report presents the outcomes of the final design, indicated whether designs have been met or not. This information can be used to re-evaluate the requirements. For instance, if the total cost exceeds the initial budget, stakeholders can reassess this financial requirement and increase the budget.

	Requirements	Textual Requirements	Compliance	Value	Unit	Difference
R-1001	total cost	The movable shall have a total cost of less than $5000	False	5088.42	$	-1.77%
R-1004	total bracket cost	The movable shall have a total bracket cost of less than $2000	True	1562.48	$	21.88%
R-1005	single bracket cost	The movable shall have a single bracket cost of less than $400	True	312.5	$	21.88%
R-1006	total skin cost	The movable shall have a total skin cost of less than $1500	True	1396.89	$	6.87%
R-1003	total mass	The movable shall have a total mass of less than 50 kg.	True	19.38	kg	61.24%
R-2005	root chord	The movable shall have a root chord of less than 3000 mm.	False	3537.21	mm	-17.91%
R-2006	tip chord	The movable shall have a tip chord of less than 3000 mm.	True	1507.25	mm	49.76%
R-2007	span	The movable shall have a span of less than 3000 mm.	False	3757.34	mm	-25.24%

Figure 3.10: Example of an requirement compliance report [36]

3.4.3. REQUIREMENTS TO DRIVE THE DESIGN PROCESS

The second part of the process, as introduced in fig. 3.7, involves assigning MDAO roles to requirements before executing the workflow. Each requirement restricts the feasible design space, but there are various ways in which the restriction imposed by a requirement can be incorporated into the MDAO workflow. A requirement does not have to be implemented as a fixed constraint; instead, there are five different requirement roles [36], which are discussed below:

- **Design variable:** A requirement that can specify the allowed options for a design variable.

- **Design variable bound:** A requirement that specifies the upper and/or lower bound of a design variable.

- **Input parameter:** A requirement that can specify a fixed value for an input parameter.

- **Constraint:** A constraint limits the feasible design space.
- **Objective:**The objective indicates the quantity that should be minimized or maximized during the optimization method.
- **Quantity of interest:** This does not put a limitation on the design space but these values will be provided to the user after the optimization is done.

The roles assigned to requirements within the MDAO workflow are important as they allow for automatic quantification of how the workflow utilizes each requirement and assesses whether compliance is guaranteed. This is illustrated in table 3.1.

Roles	Compliance guaranteed	What the system does
Design variable, Design variable bound, Input parameter	Yes	All designs explores by the optimizer are compliant with the requirement
Constraint	Yes	Optimizer will evaluate design options for which the requirement can be violated
Objective	No	No guarantee that the final requirement is met, but the objective is minimized or maximized
Quantity of interest	No	No guarantee the final requirement is met and not taken into account during the optimization

Table 3.1: Requirement roles and whether the compliance is guaranteed within the MDAO workflow, information retrieved from Bruggeman *et al.* [36]

After all MDAO roles are assigned to requirements, the workflow can be executed in accordance with fig. 3.2.

3.5. Manufacturing Information Model (MIM)

incorporation of the manufacturing into the MDAO workflow is more complex than one might expect, as discussed in chapter 1. Manufacturing processes are mostly known by subject matter experts and are only considered when the design is completed. This can lead to design changes later in the process and therefore to extra costs and a decreased efficiency, contradicting the primary goal of the optimization. It is crucial to address these considerations early in the design process since a design cannot be successful if it can't be manufactured. To streamline data management and include manufacturing information within the MDAO workflow, a Manufacturing Information Model (MIM) has been developed by Bansal [33]. The MIM captures and organizes all production-related information in the system [8]. This captured data can be used to calculate the mass, costs, and production rate of the product.
The MIM is a Python package that can implemented within KBE via a Python plug-in. The MIM package consists of three sub-packages; The Manufacturing Model, The Database, and the Assembly Model. The Unified Modelling Language (UML) of the MIM is shown in fig. 3.11.
Note: the MIM is not included within the work of Bruggeman *et al.* [8]; but (partial) integration falls under the scope of the thesis discussed in this literature research.

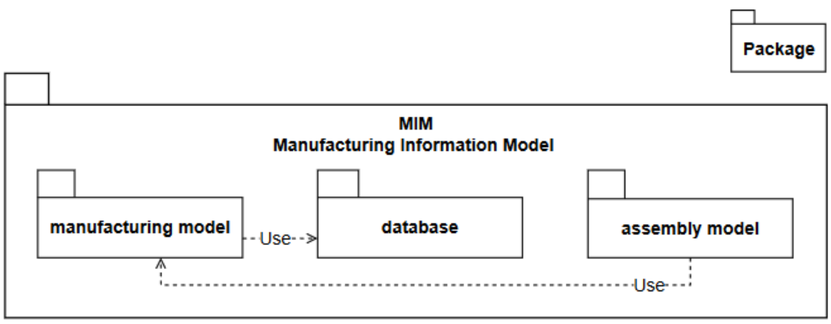


Figure 3.11: UML diagram of the developed MIM [33]

3.5.1. PACKAGE 1, MANUFACTURING MODEL

Goal: The manufacturing model serves as an accessible repository for storing manufacturing data for each primitive.
The manufacturing contains all the manufacturing information to a primitive. Here, a *primitive* is defined as *"a parametric building block to define the product"* [39], which could represent a physical object or a joint. The manufacturing model enhances the primitive by adding information categorized into five information categories; Design Specifications, Method, Material, Equipment Set, and Site. Additionally, the package has a graphical user interface (GUI)

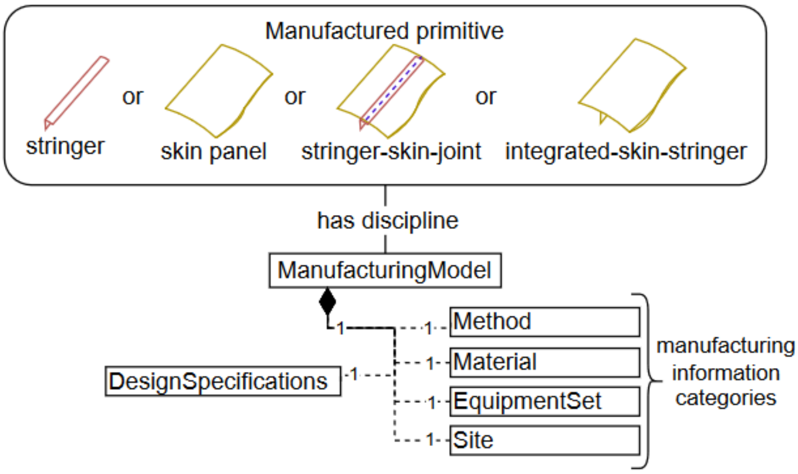


Figure 3.12: Relation between the manufacturing model and manufactured primitives of a product with the manufacturing model information [33].

When a user needs to define a new product that needs to be developed in-house, they can select a specific "Site" (the location where the product is built), and the manufacturing model can automatically filter out available materials and methods. This ensures that only feasible options are further researched [33]. Overall, integrating the MIM into the DEE framework can enhance the efficiency and effectiveness of the design process by enabling more informed decision-making since data can be accessed earlier and more conveniently.

3.5.2. PACKAGE 2, DATABASE

Goal: The second package contains all data regarding the five subcategories of the manufacturing model in an easily accessible way.
The database supports the first subpackage. This database is structured using JSON[2] format files, a lightweight data-interchange format that is both human-readable and machine-readable. JSON files provide efficient means of storing and accessing information, ensuring easy access throughout the process.
Within the database, separate files are dedicated to each subcategory of the manufacturing model, including manufacturing, equipment, manufacturing site, and material choice, holding values for compliances. These files contain data and specifications relevant to each category. For example, *"Manufacturing site "A" can manufacture material "I"* or *"Material "I" can't be connected to Material "II"*

3.5.3. PACKAGE 3, ASSEMBLY MODEL

Goal: Capturing information regarding the required steps for the assembly of a product
The assembly model aims to capture information regarding the assembly of a set of primitives and the sequence in which these steps need to occur. It provides a structured representation of the assembly process, including the sequence of steps and the time required for each operation. An example of two parts joined together is shown in fig. 3.13.

[2]https://www.json.org/json-en.html (accessed 10 January 2024)

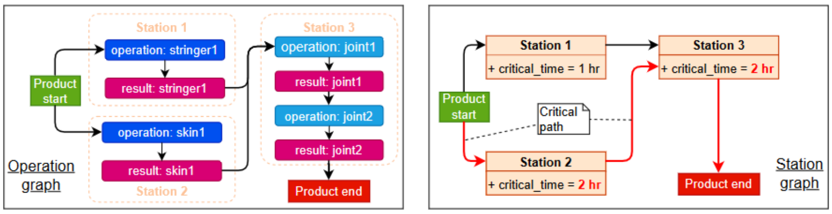


Figure 3.13: Simplified version of the operation graph and the stations graph [8]

In the Operation graph on the left side of fig. 3.13, nodes represent individual operations involved in the assembly process. The station graph, depicted on the right side of fig. 3.13, displays the time spent at each station and identifies the critical path. The critical path represents the longest path through the assembly process and determines the overall production time [33].

3.6. CATMAC

CATMAC (Cost Analysis Tool for Manufacturing of AIrcraft Components) is a tool developed by GKN Fokker and integrated into the Knowledge-Based Engineering (KBE) workflow through Parapy [40]. While the MDAO analysis during the conceptual design stage may not require highly accurate cost calculations, understanding the relative impact of manufacturing choices is crucial. CATMAC serves this purpose that providing comparative effects of choices made by the MDAO workflow. CATMAC uses various mathematical models to make cost estimations of products, taking into account factors such as material and manufacturing methods. The tool accesses a large database containing variables related to material costs, manufacturing options, complexity of objects, production time, and hiring cost of equipment, all these considerations are eventually used to make cost estimations.

It is important to note that the outcomes of CATMAC are not accurate and can have offsets of over 50%. As a result, the tool can only be used to give qualitative feedback on the manufacturing costs and to define which paths of the architecture lead to the most optimal design. The trends shown by CATMAC are highly valuable since research by Van Der Laan and Van Den Berg [40] validated that these trends can be used to estimate the most efficient design.

3.7. Design for Manufacturing

Nikitin [9] included manufacturing considerations into the DEE enabling DfM. During this research, two manufacturing methods where included in the workflow, allowing for an evaluation of whether products where feasible to produce. A proof of concept was made which demonstrated that 35% of the created designs for composite materials were deemed not manufacturable, while only 7.5% of steel parts faced similar challenges.

This section will focus on the process of enabling DfM within DEE workflow, beginning with an overview of its inclusion, followed by descriptions of two tools: the Machining producibility tool [9], and the drape-tool [41].

3.7.1. DfM inclusion within DEE

In the MDAO experiment where the DfM problem is incorporated, the manufacturing method is integrated as a design variable, as illustrated in the XDSM in fig. 3.14. This includes the workflow to explore various manufacturing methods and corresponding materials and select the optimal option. The workflow automatically defines the geometry of the part developed and evaluates its producibility. In this DfM problem, two different production techniques were assessed, and chosen by the optimiser. This process of testing these techniques is elaborated in section 3.7.2. Following this, the MDAO workflow incorporated CATMAC, the cost estimation tool. This workflow is iteration through until a feasible and optimized design is obtained The optimizer follows this by including CATMAC; the cost estimation tool and iterates this process until a feasible and optimized design is found [9].

One of the significant advantages of this setup is its scalability. While the current research focused only on two production method and their corresponding materials, the framework can be scaled up to accommodate additional materials and manufacturing methods as required by the user [9]. Each of the manufacturing tools should have its own manufacturing assessment which can be included in the dynamic MDAO workflow.

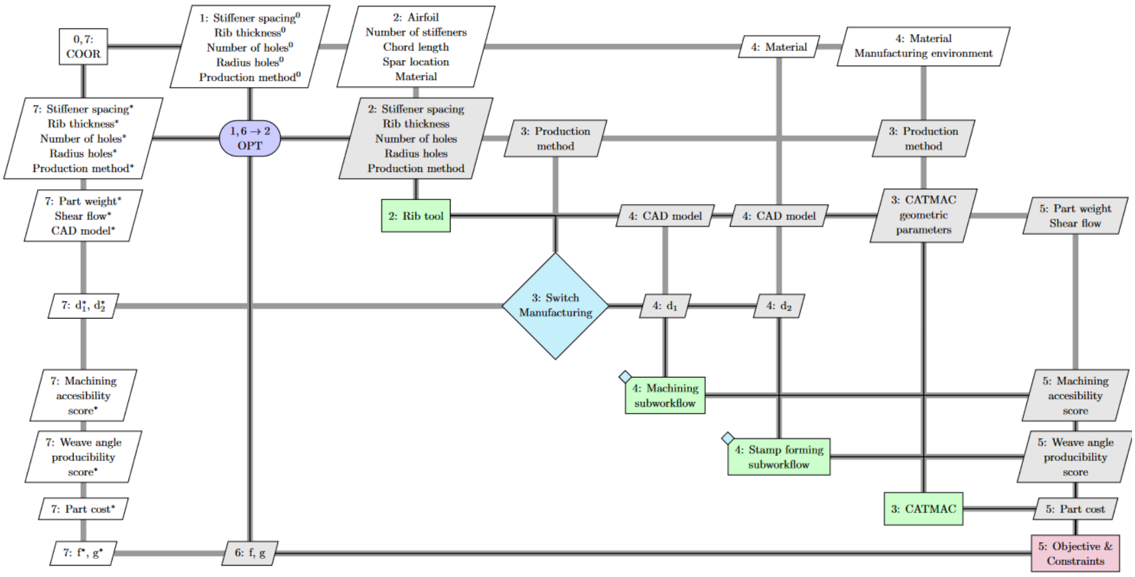


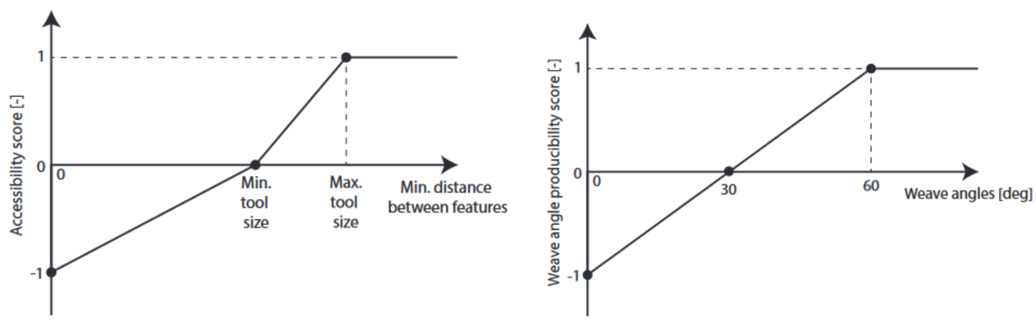
Figure 3.14: XDSM of an proof of concept of DfM Dynamic-workflow definition where two manufacturing techniques are included; both machining (for Aluminium parts) and Stamp forming (for composite parts) [8].

3.7.2. MANUFACTURING ASSESSMENT TOOLS

To evaluate manufacturability, two methods have been integrated into the workflow. For machining, the Machining Producibility Tool (MPT) was created and used, while for composites, the drape tool, created by Dr. Ir. Otto, Bergsma at the Delft University of Technology [41] was employed. These tools quantify manufacturability by assigning a continuous variable between -1 and 1, where 1 means easy producibility, 0 denotes just producible, and negative numbers mean not producible. This continuous variable enables the optimizer to identify trends and make design changes more effectively, leading to fewer required iterations.

- **Machining producibility tool** measures how well the tool can reach the material that needs to be removed, A score value of 0 means that the smallest distance between two features equals the smallest available tool. A higher grade can be achieved when there are possibilities to use bigger tools, this is beneficial because it can remove more material in less time. Figure 3.15a shows the relation between the accessibility score and the tool size.

- **Drape tool**: The drape tool looks at the fibres angle within the rib after deformation (weave angles). The Drape tool takes the CAD model of the part as well as the fabric type and outputs a text-style file. This file contains all the weave angles of the fibres and can be analysed in accordance with fig. 3.15b to give a scale of manufacturability.



(a) Plot of the accessibility score used for the machining process

(b) Plot of the weave angle producibility score used for stamp forming

Figure 3.15: Visualization of the producibility scores [8]

3.7.3. Important considerations

Despite the inclusion of DfM into The Dynamic MDAO workflow, there are some shortcomings. Firstly, the complexity of the designed part in the proof of concept was relatively simplified, which made the results not very indicative of real-life examples. This simplification limits the applicability of the results to more complex designs encountered in practice. Secondly, the workflow only focussed on the manufacturing of a single part-primitive design without any secondary parts or assembly. Real-world problems involve multiple interconnected components, this increases the complexity of the design process and should therefore be included in the workflow.

Moreover, the creation of just one part in the proof of concept may not be sufficient to draw a conclusive comparison between dynamic and static workflows. While the research by Nikitin [9] stated that dynamic workflows reduced time by 16% compared to static workflow, for the proof of concept, this needs to be further researched.

Furthermore, although the MIM would be beneficial for DfM problems, it is not included in the DEE/ However, for DfMA problems, the MIM will be of extra importance due to its features regarding assembly. Therefore, it should be partly incorporated into the workflow to enhance its capabilities to address manufacturing considerations.

Lastly, the addition of assembly methods significantly impacts the number of possible design options and thus computational time. A method needs to be incorporated so the workflow can still be executed efficiently despite the increased complexity, this can be partially done by VISTOMS, where the workflow is visualized and can be adapted.

3.8. Conclusions

The Design and Engineering Engine (DEE), upon which this thesis will be built, serves as a blueprint for creating MDAO projects. The state-of-the-art version of this framework includes Design For Manufacturing (DfM), enabling the automatic design of one single part primitive, where two manufacturing assessment tools are included. Various tools within the DEE streamline the entire workflow, from requirement creation to requirements verification.

The Requirement Management Module (RMM) introduces a systematic method for setting up machine-interpretable requirements, which can then be used to design products that comply with these requirements. The Manufacturing Information Model (MIM) is a promising tool that can include manufacturing considerations and enhance data management in the MDAO workflow. However, despite its potential, the MIM has not been integrated into the workflow yet.

Although a proof of concept has been developed to incorporate DfM into the DEE, this version is relatively simplified and only optimizes for a single-part primitive. To fully enable Design for Manufacturing and Assembly (DfMA) via the DEE, the workflow should create multiple part primitives and joints. This requires new assembly assessment tools and the integration of the MIM to effectively handle data. Additionally, a challenge lies in the effective management of the sheer amount of data that is being produced.

4

ASSEMBLY METHODS FOR WINGBOX STRUCTURAL DESIGNS

The DEE included the optimization of a single wing rib, which serves as the starting point of this thesis. The subsequent phase of this research focusses to scale up this research, adding multiple parts and include assembly considerations. This section will focus on the wingbox design which will be used as a proof of concept, together with assembly considerations.
In section 4.1, the structural layout of the wingbox will be quantified, where the distinction will be lied between part primitives and joint primitives. Following the definition of the wingbox, various assembly methods currently used within the industry will be discussed. Section 4.2.1 discusses assembly techniques for combining two metal parts, while section 4.2.2 will focus on assembly methods for combining composite parts with either metal or composite parts.
The chapter will conclude with a discussion on excluded parts from the research and the selection of three assembly methods for further analysis in section 4.4 and section 4.5, respectively.

4.1. WINGBOX DESIGN

The primary purpose of the wingbox is to support the wing, generate lift, and maintain structural stability [43]. While real-world wingboxes consist of various components such as tanks, engine pylons, and different types of stringers and ribs. Within this thesis, the wingbox is simplified to three primary components; ribs, spars and, skin panels.
The wingbox is designed to form a robust torsion box capable of withstanding all internal wing loads generated by lift and the weight of the fuselage. Typically, this torsion box normally consists of two spars and two stiffened skin panels [43]. To enhance clarity throughout this discussion, the terms part and joint are used to refer to "part-primitives" and "joint-primitives" as discussed below [33].

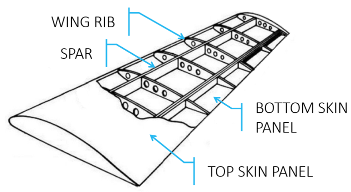


Figure 4.1: Generalized design of a wingbox (picture based on [42])

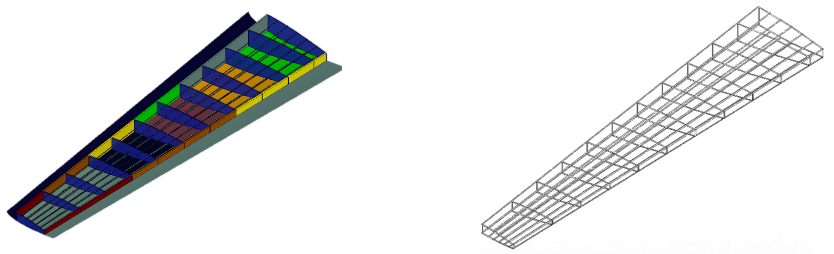
4.1.1. WINGBOX PART PRIMITIVES

Part primitives are sub-assemblies resulting in individual parts. These parts are shown in fig. 4.2a and exist of the following components:

- **Ribs:** Ribs serve to maintain the shape of the wing and are positioned across the entire width of the aircraft. They introduce and transfer internal loads to the skin panels and spars. A wingrib typically consists of flanges, a web, and stringers to enhance its strength [43].
- **Spars:** Spars are designed to transfer bending loads in the wing to the fuselage. Usually configured as I-beams, spars have a web that bears shear loads to prevent deformation. [43]
- **Skin panel:** Skin panels are responsible for holding the entire wing together and generating lift. They should have a smooth outer surface and internal stringers to ensure stiffness. As part of the torsion box, skin panels must withstand shear forces along the skin.

4.1.2. WING BOX JOINTS PRIMITIVES

Joint primitives have the function of adding two part primitives together. As illustrated in fig. 4.2b, there are several types of joints that can be found in this research, these include the: stringer-skin joints, rib-skin joints, rib-spar joints, and spar-skin joints.



(a) Wingbox with part primitives including skin panels, stringers.

(b) Wingbox joint primitives located in the wingbox

Figure 4.2: Baseline concept of a wingbox [33]

4.2. ASSEMBLY METHODS

Design for Assembly (DfA) represents a relatively recent development in the design process, with research effort dating back to the 1960s. In recent years, there has been an increased focus on DfA, especially in optimizing assembly processes to reduce costs and assembly time [44]. During the DfA process, more ellaborate background is given in the research by [33] were assembly considerations are used to calculate the required time for the assembly process.
Within this research, emphasis will be placed on two families of materials: metals and composites. Each part may consists of either one, leading to various assembly configurations. As such, joints between two parts can involve two metal parts, a metal part connected to a composite part, or two composite parts. this section discusses the various options for assembling such parts.

4.2.1. ASSEMBLY METHODS FOR METALS TO METALS

Metal-to-metal assembly has a long history, with well-established production methods dating back centuries. While an ideal structure would consist of a single unit made from homogeneous material, practical limitations require the assembly of parts, especially for large wingboxes [45]. C. Y. Niu [45] imposes several specific guidelines how metal parts should be joined, including considerations for repair and maintenance. Several joining techniques have been developed for the aviation industry, which are discussed below:

MECHANICAL FASTENING

Mechanical fastening is widely used and well established methods within the aviation industry. Fastening can either be done *permanent* or *non-permanent*.. Rivets (illustrated on the right of fig. 4.3) are used for permanent fastening, as removing them will lead to damages on the material and the rivets while bolts (illustrated on the left), at the other hand, offer the flexibility of being removable for maintenance purposes.
Comparing rivets and bolts, conventional rivets have limited clamp load and are primarily suited for shear forces. One advantage of rivets is their ability to be installed from one side and resist rotating or loosing due to vibrations, unlike bolts [46]. While mechanical fastening offers widespread adoption and easy detection of damages (e.g., missing fasteners), it also adds significant weight to the structure and is labour intensive.

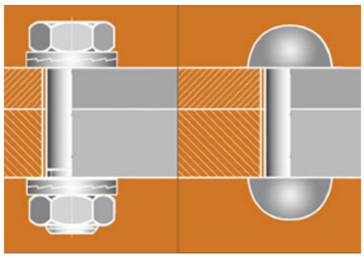


Figure 4.3: Example of bolted (left) and rivets (right)

In the aviation industry, rivets are the preferred choice for steel-to-steel combinations due to their reliability and suitability for permanent fastening applications.

WELDED JOINTS

Welding involves joining two metal parts by heating their surfaces to the point of melting, or by adding molten metal between them to fuse them together. While welding can create strong bonds, the strength of the weld largely depends on the skill of the welder. Mostly, welding is used in joints involving shear or compression

forces, and not in tension [45].
Compared to other joining methods, welding adds relatively little material, making it a lightweight option. However, one drawback of welds is their chance for instantaneous failure. As a result, additional technologies such as ultrasonic or X-ray sampling inspection are often used to ensure a weld's quality. When performed correctly, welding offers advantages such as reduced stress concentrations and a good force load transfer.

BONDED JOINTS

Bonding is often used as a secondary assembly method. It is primarily used to enhance fatigue life by reducing internal vibrations. Bonding has a limited application in metal parts due to concerns regarding the reliability.
Unlike welding, the quality of boned joints are less dependent on the skills of the workers, and the application process is relatively easy and cost-effective. To achieve strong bonds, careful preparations of the surfaces needs to be ensured.
Two major drawbacks of bonding is the sensitivity to temperature fluctuations and exposure to environmental factors such as salt [45].

GUSSET JOINTS

When the shape of materials are incompatible for direct bonding, gusset plates can be introduce. These plates serve to connect two parts and enhance the load transfer between them. However, it is important to note that gusset joints are not considered as an assembly method, it is combined with one of the above mentioned joints methods [47].
Gusset plates must be custom-designed for its specific location, which increases the weight and costs of the product.

CONCLUSIONS

In conclusion, the four discussed assembly method can be summarized in table 4.1

Method	Anticipated benefits	Limitations
Mechanical fastening	• Mature Technology • Baseline for cost data • Could supplement weld/bond	• Increased weight • Labour intensive • Sensitive for fatigue
Welded joints	• Mature Technology • Easy to atomize	• High risk • Quality depend on welder • Harder to inspect • Requires secondary seal
Bonded joints	• Less fatigue • Cheaper	• Higher risk • Not used in industry
Gusset joints	• High load transfer	• High added weight • High cost due to specialization • Difficult repair or to maintain

Table 4.1: Comparison of currently used joining methods of two metal parts

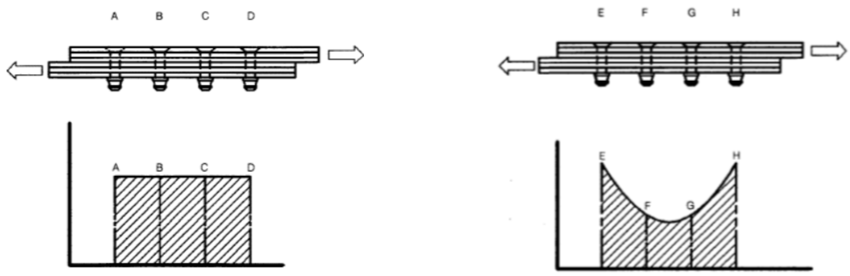
.

4.2.2. ASSEMBLY METHODS FOR COMPOSITES

Composite joining plays a critical role in aircraft structures, as failures in these joints are one of the most common sources of failure in aircraft structures [48]. Composites are popular for their potential to significantly reduce weight compared to their metal counterparts. To fully realize this weight-saving potential, it is important to minimize the number of joints while ensuring that joint strength is sufficient. This challenge is particularly profound with composite materials due to their brittle nature compared to metals.
This subsection explores various assembly methods for both composite-composite and composite-steel joints.

MECHANICAL FASTENING

Mechanical fastening in composite materials involves unique considerations compared to its application in metal parts, as discussed in section 4.2.1. Composite materials exist of fibres and a matrix that is not homogeneous, resulting in a different response to fasteners compared to metals, as illustrated in fig. 4.4. The load distribution of fasteners in a steel plate is evenly distributed, while the same force results in load peaks for composite parts, as shown in fig. 4.4b, which can damage the part if placed wrongly.



(a) Load distribution metal-metal

(b) Load distribution composites-composites

Figure 4.4: Comparison between metal and composite joints [48]

Compared to other assembly methods, fastening offers provides structural reliability without the need for extensive surface preparation, aside from predrilling holes. For composites, bolts are preferred due to their better clamp strength, reducing the chance of pull-through. However, fastening can lead to severe strength degradation in the laminate, resulting in a weight penalty due to the requirement for local reinforcement. Despite this drawback, fastening provides engineers the ease of joint inspection and it is easy to assemble and disassembly for fabrication replacement, unlike other techniques.

ADHESIVE BONDING

Bonding is the standard practice for joining two composite parts. The strength of the adhesive bond is independent of the length of the overlap as long as it exceeds a certain threshold, making bonding a relatively lightweight assembly method. One of its key advantages is the homogeneity of force transfer, resulting in a uniformly distributed load compared to bolds. Adhesives can belong to the same family as the matrix, ensuring compatibility.

However, some joints may develop peel stresses due to eccentricities in the load path or external peel loads, as illustrated in fig. 4.5. Tapering the ends of the plates can help mitigate peeling, and a proposed solution by Silva and Adams [49] involves ensuring the joint is balanced, meaning that the adherents carrying the load in the two directions equally, this can be solved by a double-lap joint.

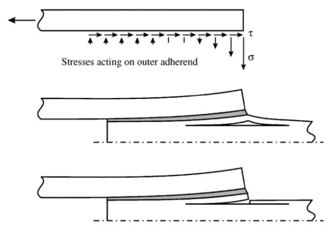


Figure 4.5: Peel stresses resulting in failure [49]

The surface does need treatment before the bonding can be applied. Adhesive bonding is often combined with mechanical fasteners since fasteners reduce the chances of peel failure and adhesive bonding distributes the load evenly, levering the strengths of both methods.

THERMOPLASTIC WELDING

In thermoplastic welding, heat energy is used to melt the resin and achieve the bonding. There are three primary types of thermoplastic welding: resistance, ultrasonic, and induction.

Resistance welding involves passing an electric current through the element, generating heat that melts the resin. It has the potential to weld large areas without thickness limitations but requires high power. Ultrasonic welding is used for parts with special surface preparations, featuring added bumps. An ultrasonic horn melts these bumps while pressure is added, resulting in molten material that solidifies into a strong modular bond. While quick and producing very strong properties, ultrasonic welding is limited to small spots and requires access from both sizes.

Induction welding uses a specialized non-contact tool that generates an electric field, inducing heat in the targeted area to melt and join the materials. It requires only single-sided access and can bond complex shapes, but it is thickness-limited and carries a high risk regarding strength [45]. Overall, thermoplastic bonding carries a higher risk but is a lightweight method for assembling parts.

COCURRING

Cocurring is one of the most attractive features, as it enables the production of complex shapes. Unlike adhesive bonding, cocurring involves fabricating individual parts that are then cured together into single units, eliminating the need for fasteners. This can be defined as *curing of a laminate with stiffeners and attachments all in one process operation.* [48]. Cocurring offers the advantage of reducing weight and increasing structural efficiency. However, it is relatively costly and limited in terms of shapes and sizes that can be produced. Despite its benefits, cocurring is a complex process with a higher risk of errors such as bad layup or misalignment. Additionally, inspection becomes challenging, especially for hidden parts.

CONCLUSION

Table 4.1 shows the four discussed assembly methods relevant with joints on composite parts.

<table>
<tr><th colspan="2">Method</th><th>Anticipated benefits</th><th>Limitations</th></tr>
<tr><td colspan="2">Mechanical fastening</td><td>• Mature Technology
• Baseline for cost data
• Could supplement weld/bond</td><td>• Low risk
• Increased weight
• Labour intensive
• Requires secondary seal</td></tr>
<tr><td colspan="2">Adhesive bonding</td><td>• Low added weight</td><td>• Moderate risk
• Cure cycle required
• Tooling</td></tr>
<tr><td rowspan="3">Thermoplastic welding</td><td>Resistance</td><td>• Can be automated process
• Continuous weld
• Reduced fastener count/weight</td><td>• Moderate risk
• Requires 2 side access</td></tr>
<tr><td>Ultrasonic</td><td>• Can be automated process
• Possible continuous weld
• Reduced fastener count weight</td><td>• Moderate risk
• Requires 2 side access</td></tr>
<tr><td>Induction</td><td>• Requires 1 side access
• Can be automated process
• Continuous weld
• Low added weight</td><td>• Moderate - high risk
• Requires magnetic mat</td></tr>
<tr><td colspan="2">Cocurring</td><td>• Total homogeneous weld joint
• Probable elimination of seal</td><td>• High risk
• Part size/shape limited</td></tr>
</table>

Table 4.2: Comparison of currently used joining methods [48]

.

4.2.3. ASSEMBLY CHOICES

Various assembly methods have been discussed in this section. To address the three combinations of joints in the DfMA problem: metal to metal, composite to metal, and metal to composite, an assembly method is required for each scenario.

For metal to metal parts, riveting is chosen due to its widespread acceptance in the industry, good resistance to fatigue, and ease of assembly. By choosing an industry-standard method, it is expected that the results will be most accurate to real-life scenarios.

Due to foreseeable time constraints in this thesis, the decision has been made not to include double-assembly methods for two primitives (e.g., bolting-adhesive). Therefore, adhesive bonding is selected as the assembly method for two composite parts, and bolting is chosen for a composite part and a metal part. Both of these methods are well-documented and widely adopted within the industry.

Conclusion: Assembly methods

Material A:	Material B:	Assembly method
Metal	Metal	Riveting
Metal	Composites	Bolting
Composites	Composites	Adhesive bonding

Table 4.3: Assembly methods used within this thesis

.

Adhesive bonding is preferred for two composite parts because it is commonly used in practice since the resin is often used as the adhesive. However, metal-composite bonding with adhesives is rare due to material

incompatibilities. Bolting is more common for metal-composites, as metal is easy to fasten mechanically and seals can be used to prevent corrosion.
Table 4.3 show the conclusion of this section. These three methods will be used in the main thesis in the DfMA problem and will be included in the workflow.

4.3. Material incompatibility

Various materials can have material incompatibilities, this can result into various forms of corrosion, resulting in a weak structure after a time. Most corrosion occurs between various types of graphite/epoxy based composites such as carbon fibre and metals. The research of Nikitin [9] focussed on carbon-fibre with various sorts of metals. Down below, a list with compatibilities materials is given [48].

- **Acceptable:**, Titanium, Ti Alloys, Various high tensile metal alloys (MP-35N, INCO600)
- **Marginally Acceptable:** Stainless steel CRES alloys (PH13-8MO, A286)
- **Not compatible:** Stainless steel, silver, Aluminium, Aluminium alloys, most platings (Nickel, Zinc)

4.4. Important considerations

Unfortunately, not all considerations can be taken into account. Due to time limitations, there are some limitations to the project:

- **Thermal expansion:** The thesis will not account for the effects of thermal expansion on materials within the wingbox assembly, despite its relevance [50].
- **Fastener/adhesive selection:** Detailed specifications for fasteners or adhesives will not be considered.
- **Structural considerations:** Structural disciplines such as FEM analysis will not be incorporated into the MDAO workflow.
- **Assembly preparations:** Pre-assembly preparations such as surface roughening for adhesive bonding or pre-drilling for mechanical fastening will not be addressed during the analysis.

4.5. Conclusion

The wingbox is a multi-part assembly of the aircraft, designed to support the wing, generate lift, and maintain stability. Wingboxes consist of multiple parts, but for this thesis, the wingbox is simplified to ribs, spars, and skin panels. The research focuses on composite and steel materials, resulting in three different assembly options: one for joining two steel parts, one for joining two composite parts, and one for connecting a composite and a metal part.
For the combination of two metal parts, riveting is chosen to incorporate within the DfMA problem. Riveting is widely adopted in the industry due to its ease of use, reliability, and resistance to fatigue. For joining two composite parts, adhesive bonding is used. This method is also common in the industry and often uses adhesives of the same material family as the resin, ensuring compatibility and an even load distribution. Bolting is chosen for joining metal to composite parts due to its ability to resist peeling, a common load case for composite structures.
These three assembly methods will be incorporated into the dynamic workflow of the DEE to analyse their performance and suitability for the wingbox assembly.

5

GUIDELINES OF ASSEMBLING MULTI-PART OBJECTIVES

The last chapter (chapter 4) discussed various assembly methods suitable for use within the aviation industry. Three of those methods were chosen to be included in the DfMA process within the DEE: riveting, bolting, and adhesive bonding. In this chapter, these three methods will be further discussed. Furthermore, design guidelines and rules are given which can be used for the assessment of assembly.
Firstly, the riveting of two metal parts is discussed in section 5.1, followed by the guidelines of guidelines for bolting composite materials to metal materials in section 5.2. Then, the adhesive bonding of two composite materials will be discussed in section 5.3.

5.1. RIVETING FOR JOINING TWO METAL PARTS

Riveting involves deforming each fastener in a controlled manner to hold two parts together, as shown in fig. 5.1. Pneumatic tools are often used to pull the mandrel which ensures the ensure locking of two parts. The process begins with pre-drilling holes within the structure, and if necessary, countersinking can be added to ensure the rivet lies flush with the surface of the part. This is especially important for skin panels since any roughness can disturb the airflow.
Some novel techniques have been developed to eliminate the need for pre-drilling [51], which can improve manufacturing efficiency. This would lower the required time tremendously but is not further discussed within this project, since the drilling methods are left out of the scope as stated in section 4.4

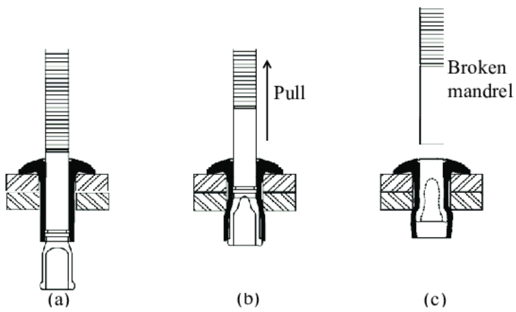


Figure 5.1: Process (a) the blind is inserted in the correct location the correct location, (b) the mandrel is pulled by a rivet gun which expands the end of the rivet and (c) the mandrel breaks and finishes the notch [51]

5.1.1. DESIGN GUIDELINES FOR THE RIVETED PARTS

When two metal parts are combined via riveting, both guidelines and design rules need to be established. Design guidelines are not mandatory but are helpful suggestions for optimizing assembly and ensures ease of manufacturing. Design rules are fixed requirements that need to be met.
The primary function of the joint is to transfer forces from part "A" to part "B" and vice versa. The failure modes should be considered when determining the number of rivets via an analysis during the thesis self.

DESIGN GUIDELINES

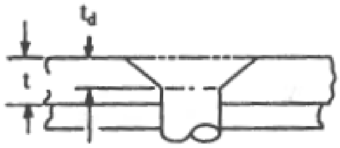
1. A fitting factor of 1.15 will used for calculations regarding the number of required rivets.

- The fitting factor is used to define the extra required strength of the joint, the joint should be 15% stronger than the critical loadcase, and the critical loadcase will be defined within the MDAO workflow.

2. Joints should be supported

   - To improve the joint efficiency of two overlapping plates, the joint should be located on or near support structures such as stringers.
   - For unsupported joints, eccentricities are produced in a joint which leads to a higher tensile stress which is defined by $f = \frac{4P}{t}$ where f equals the tensile stress, P equals the load of the object.

3. Knife-Edges should be avoided since this severely limits the load the rivet can transfer.



(a) Preferable (No Knife-Edge)

(b) Poor (Knife-Edge)

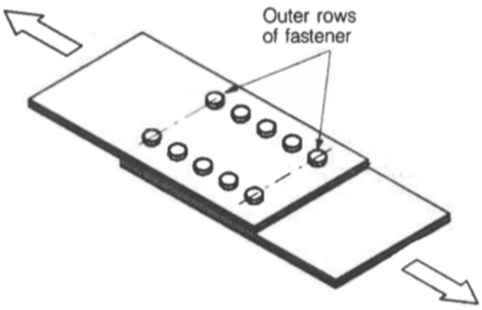
Figure 5.2: Countersunk Fastener Knife Edge [52]

DESIGN RULES

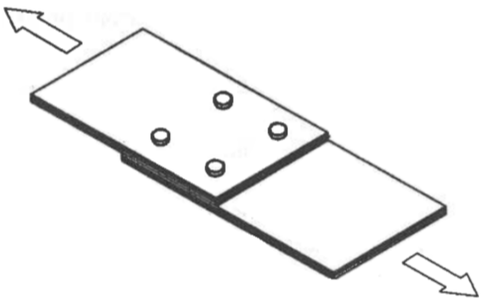
1. an minimum edge distance of e/D[1] =2 is required in the direction of the load.
2. A minimum edge distance of eD=1.5 may be used if not in the direction of the load.
3. The rivet diameter is equal to the thickest plate, such that the hole should be 0.03 inch larger than the thickest plate.

5.2. BOLTING OF A COMPOSITE PART TO A METAL PART

Bolting, defined as removable fasteners, is widely used within the aviation industry. Although it is relatively similar for both composite and metal parts, their behaviour differs significantly due to two reasons: The relative brittleness of the material and the occurrence of laminate failure, (peel off).
Mechanical fastening offers several advantages, the most notable difference between bolting and riveting is that bolting is non permanent, enabling engineers to check and replace parts if necessary. Bolting does require the same preparations as riveting, where holes need to be pre-drilled, but in contrast to riveting, access from both sides is required. Furthermore, normal steel plates often have multiple rows of bolts/rivets as shown in fig. 5.3a while composites often have two rows of fasteners, which leads to lower stress concentrations [48].



(a) Multi-row joint (avoid for composites)



(b) Two-row joint (preferred for composites)

Figure 5.3: Composites materials prefer a low amount of joints due to low ductility of the material [48]

[1] D is the diameter of the fastener and e is the distance from the center of the fastener to the edge of the part

5.2.1. DESIGN GUIDELINES OF THE BOLTED METAL AND COMPOSITES PARTS

DESIGN GUIDELINES

1. Net tension failure occurs when too many fibres are damaged by the bolts, in order to prevent this:

   - Two rows of holes should be used as shown in fig. 5.3b
   - The sheet thickness should be larger than the feather-edge as illustrated in b of fig. 5.2
   - Shallow head and large diameter shear fasteners will be used where t/d[2] < 1
   - Tension fasteners are more efficient for thicker laminates with t/d>1.

DESIGN RULES

1. If laminates are dominated by 0^o fibres, it is highly likely for the joint to fail by shear out. This can be prevented by

   - Edge distances compared to bolt diameters should be at least 3 (e/D>3)
   - Use a minimum of 40% of $\pm 45^o$ plies
   - Use a minimum of 10% of 90^o plies

2. fastener spacing should be approximately four times the fastener diameter

5.3. ADHESIVE BONDING FOR TWO COMPOSITES PARTS

Adhesive bonding is widely used within the aerospace industry. The strength is independent of the overlap as long as it is above a certain limit. Too little overlap will result in creep failure, while longer overlaps do not significantly increase strength or durability, only adding to weight and costs.
Ductile adhesives improve the load distribution by reducing the pressure peaks. Adhesive-bonded joints are mostly used to transfer shear loads, but some joints develop peel loads, to counteract these peel loads, materials can be tapered, as shown in fig. 5.4. This tapering helps by lowering the forces on the ends of the composite parts.

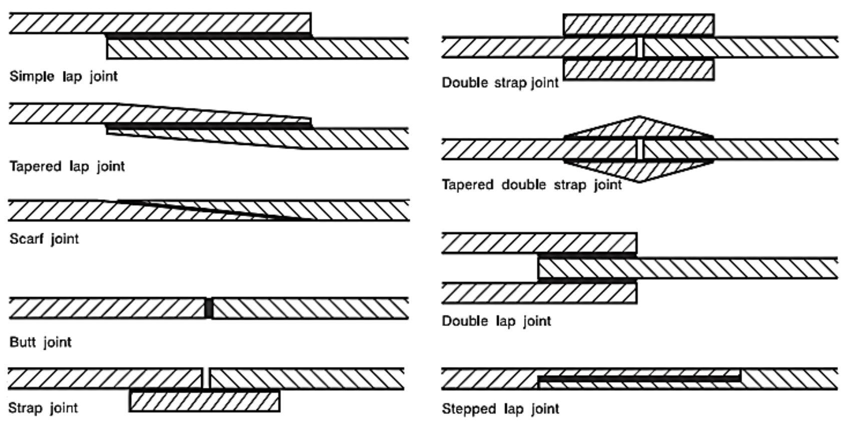
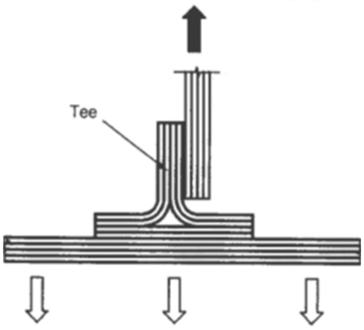


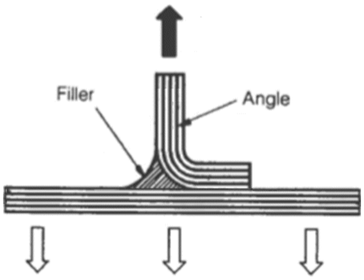
Figure 5.4: Common examples of tapered joints [53]

Adhesive bonding is most effective in scenarios involving shear stress. For other load cases, such as perpendicular load transfer, additional elements must be added. This can either be a "Tee", as shown in fig. 5.5a, or an "Angle", as shown in fig. 5.5b. It is important to note that a filler is required for the "Angle", to prevent peel-off.

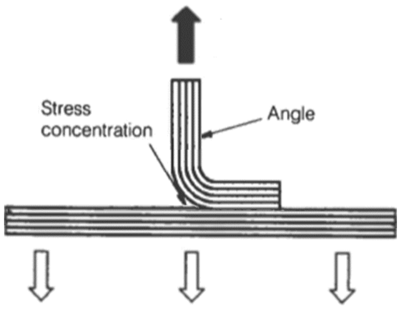
[2] thickness of the plate is less than the diameter of the hole



(a) Recommended



(b) Use with caution



(c) Do not use

Figure 5.5: Angle Stiffener Bonded to a Skin [48]

5.3.1. DESIGN GUIDELINES FOR THE PART PRIMITIVES

DESIGN GUIDELINES

There are a number of guidelines for adhesive bonding [48].

1. Joints with short overlap lengths are more efficient in avoiding adhesive shear failure.
2. Peel stress needs to be minimized, this can be done by the following:
    - Peel stress is an inverse function of overlap
    - Taper ends to a thickness of about 0.03 inch and a slope of 1/10
3. Tooling used in bonding operations must be carefully designed to minimized thermal distortion and residual stresses within the joint.
4. Thermosetting resins are commonly used (e.g. Epoxy). Pressure must be applied for these resins, requiring access from both sides.
5. When thermosetting resins can't be applied paste adhesives can be used but these have lower strength.
6. Bonded joints are less suitable for hot-wet or cold-dry conditions and need more maintenance.
7. The most ductile adhesive should be chosen since these can withstand the highest shear.
8. For the connection of two perpendicular parts, an extra T-stiffener is added with respect to fig. 5.5.
    - This problem is more significant when the skin buckles or rotates
    - The ends of the T-flanges must be tapered

DESIGN RULES

set rules:

1. outer plies can't have 90^o orientations, this can only be $\pm 45^o$ to prevent peel stress
2. For composite-composite joints, the bonds should have the following guidelines
    - 80t overlap in single-lap
    - 30t overlap in double shear
    - 1/50 slope for scarf joint

5.4. CONCLUDING REMARKS

Three assembly methods will be implemented into the DEE, these includes riveting, bolting and adhesive bonding. This chapter introduced various guidelines and rules regarding these methods. These rules can during the thesis itself be transformed into the loose requirement databases. Each assembly method should have a requirement database as stated in section 3.3.2.

6

RESEARCH PLAN

The research plan defines the methodologies and the plan for achieving the results. The main goal of this thesis is to include DfMA considerations into the DEE framework, this can be done by performing the research objective:

Recap: Research objective

Investigate how to integrate manufacturing and assembly into early product design stages by using Multidisciplinary Design Analysis and Optimization (MDAO) framework, substantiated by a case study focused on (a section of) a wingbox.

To investigate how these methods can be incorporated, four research questions were identified, each contributing to the overall understanding of the research objective.

6.1. THESIS NARROWING DOWN

The main objective states that eventually, a proof of concept must be delivered that demonstrates that manufacturability methods can be included within the DEE for the wingbox. The wingbox consists of various parts, as discussed in chapter 4, this includes the skin panels, stringers, and wing ribs, each of these can be joined against one and another. These parts can be made from different materials and assembled in various ways, resulting in a large number of possibilities.
Due to time limitations, initially, the thesis will focus on two parts with one joint to assess its feasibility, Once this is achieved, the methodology can be scaled up to a product consisting of three-part primitives: two skin panels and one wing rib.

6.2. METHODOLOGIES

The research questions in general together form the main objective, thus these needs to be answered accordingly. This section presents the steps required and inputs to answer the research question.

6.2.1. RESEARCH QUESTION 1: AUTOMATED ASSEMBLY ASSESSMENT METHODS

Recap: Research questions

1. What automated methods can be developed to assess the feasibility of assembly/joining of parts, and how can these **assembly assessment methods** be incorporated within the MDAO framework?

INPUTS

Data	File format	Relevance	Storage location
"Composite Airframe Structures" by M. Nui	Literature	Quantification of assembly related requirements for composite structures.	Local
"Steel airframe structures" by M. Nui	Literature	Quantification of assembly related requirements for steel structures	Local
Assembly methods and requirements	.xls	This is the an excel file of the RMM tool, containing relevant requirements	Git
Requirement Management Module	.Py	Incorporate the requirements into the workflow	Git
Manufacturing Information Model	.Py and .JSON	Manufacturing information model	Git
Automated Production Assessment Tool	.Py	Two tools that can automatically assess the production, can be used as inspiration.	Git

Table 6.1: Inputs required to solve Research Question 1

METHODOLOGIES

1. The state-of-the-art part manufacturing needs to be *analysed* and used to create two-part primitives,

2. The two production assessment methods need to be *analysed* where their state-of-the-art software is learned.

3. The list of requirements is set up via a *descriptive research*

4. A new method should be created that grades the assembly as feasible or not feasible.

OUTPUTS

Data	File format	Relevance	Storage location
Automated Assembly Assessment Methods (AAAM)	TBD	An individual tool/sets of tool that can assess the assembly feasibly based on a set of requirements.	Git

Table 6.2: Required outputs to solve the first research question.

6.2.2. RESEARCH QUESTION 2: AAAM INCLUSION INTO WORKFLOW

Recap: Research questions

2. Would the incorporation of the **assembly assessment methods** and the **production assessment methods** result into problems within the MDAO workflows

INPUTS

Data	File format	Relevance	Storage location
Automated Assembly Assessment Methods (AAAM)	TBD	The output of Research Question 1	Git
Requirement Management Module	.xls	Streamline the management of the requirements throughout the workflow	Git
Manufacturing Information Module	.py and .JSON	Streamlining the manufacturing model throughout the workflow	Git
MDAO workflow	Various	Relevant modules of DEE workflow used by Bruggeman and La Rocca [13]	Git
CATMAC	py	Cost estimation tool created by GKN Fokker	Git

Table 6.3: Inputs required to solve Research Question 2

METHODOLOGIES

1. The AAAM (output of RQ01) must be added to the workflow. Therefore, the workflow needs to be *analysed*.
2. The MIM, RMM, and CADMAC need to be fully understood and integrated into the system
3. Any challenges of including the AAAM and previous production assessment tool should be listed and solved.

OUTPUTS

Data	File format	Relevance	Storage location
Workflow with AAAM incorperated	Various	A KBE supported applications that can optimize the manufacturing (production and assembly) of two part primitives	Git

Table 6.4: Required outputs to solve the second research question.

6.2.3. RESEARCH QUESTION 3: ASSEMBLY RELATED REQUIREMENTS

Recap: Research questions

3. How can **assembly related requirements**, such as material compatibility be incorporated into MDAO workflow?

INPUTS

Data	File format	Relevance	Storage location
MDAO worflow	Various	The entire MDAO workflow as an output of RQ03	Git
Manufacturing Information Model	.py and .JSON	The .JSON files contain non-geometrical requirements.	Git
Assembly requirements	Literature	Various books and literature is used when required to enhance the MIM	Git

Table 6.5: Inputs required to solve Research Question 3

METHODOLOGIES

1. Further required non-geometrical assembly requirements such as material incompatibility are listed in .JSON files (database) or in the MIM input.

2. The MIM needs further research on how to be included in the workflow, this is currently not done yet

OUTPUTS

Data	File format	Relevance	Storage location
MDAO Workflow	Various	Workflow including assembly methods and full inclusion of MIM and requirements.	Git

Table 6.6: Required outputs to solve the third research question.

6.2.4. RESEARCH QUESTION 4: DATA MANAGEMENT

Recap: Research questions

4. By what means could the challenges posed by a **large combination of parts and joints** and their corresponding manufacturing and assembly methods be effectively addressed within a MDAO workflow?

INPUTS

Data	File format	Relevance	Storage location
MDAO Workflow	Various	The entire MDAO workflows which can be used for validation	Git
Static validation options	.xls	Various options delivered by SME's used for validation of the static workflow	Local

Table 6.7: Inputs required to solve Research Question 4

METHODOLOGIES

1. Run the optimization scheme with a dynamic workflow and keep track of the running time, iterations, and characteristics of the optimized model
2. Run the static MDAO-*analysation* of three case studies, these case studies are determined by the SMEs.
3. Do an *comparative analysis* of the results of tasks 1 and 2 where the running time, iterations, and overall performance are compared
4. Conduct whether the dynamic workflow has advantages over static workflows.

OUTPUTS

Data	File format	Relevance	Storage location
Conclusions regarding Dynamic vs Static	Text	Part of report	Local

Table 6.8: Required outputs to solve the fourth research question.

6.3. DELIVERABLES

There are two main deliverables from this master thesis, besides the overall MDAO infrastructure must be shared via Git, a report must be shared, there are two possibilities for a report.

- **Report:** The report should contain the setup, considerations, and information regarding the thesis project. If chosen for a report, a draft must be handed in during the green light review and the final report must be handed in on the final hand-in date as described in table 6.12.

- **Scientific paper:** The scientific paper publishes the findings of the method. If chosen, the final version must be handed in during the green light. accompanying documentation such as other considerations and the process is attached as appendixes and handed in during the final hand-in date.

The chosen option will be done during the thesis itself. All the deliverables are shown in table 6.9

Type of data	File format	How will the data be collected	Purpose of sharing	Storage location	Who will have access
Report / paper	.PDF	Written	Knowledge transfer	Git	Stakeholders
MDAO Workflow	.ZIP	All technical document created and altered within this thesis	Transferring data for easy access	Git	Stakeholders

Table 6.9: Deliverables of MSc Thesis

6.4. WORKPACKAGES

The work and methodologies are divided in the workpackages as shown in table 6.10. The workpackages are introduced in table 6.10, this table includes the calculated hours, the research question it solves and a summary of the containing.

WP name	Hours	Solves	Goal
WP1: Management and Organization	50		Manage and structure the thesis correctly, ensure compliance with the planning.
WP2: Familiarize with state-of-art	160		Understand and learn all the state-of-art projects on which this thesis will build upon.
WP3: Skin panel formulation	100		Formulate the “primitive” skin panel such that an assembly can be made with the existing wing rib design.
WP4: Assembly assessment method formulation	190	RQ1	Create a tool that can automatically assess if two parts can be assembled together by using a list of requirements.
WP5: Method incorporation and workflow definition	250	RQ2, RQ3	Incorporate the method created by WP4 and the overall assembly related requirements within the workflow.
WP6: Validation and verification	100	RQ4	Validate the entire workflow created in WP5 and research the efficiency increase compared to conventional analysis.
WP7: Complexity increase to 3 primitives	150	RQ1+, RQ2+, RQ3+	Repeat WP2 to WP6 for three primitives instead of two. If time does not allow, this WP can be marked redundant.
WP8: Report and defence preparations	TBD		After the green light review, all technical aspects needs to be finished, this workpackages starts after this deadline and plans time for the report finalization and defence preparation and includes all work spend between the green light review and the Defence.
Total	**1000**		

Table 6.10: Workpackages

These workpackages are elaborated in appendix A, these forms show the containing of the workpackages including clearly defined inputs and outputs. when any unforeseen delays are encountered, workpackage 7 can be skipped in its entirety since all the research questions are answered.

6.5. IMPORTANT CONSIDERATION REGARDING SUPERVISORS

During the thesis, some changes occur regarding supervision, these changes are depicted in table 6.11.

Name	Supervisor from date	Supervisor end date
Dr.ir. G. La Rocca	Beginning	End
ir. A.M.R.M. Bruggeman	Beginning	26th of May, 2024
ir. D. bansal	Beginning	25th of December, 2023

Table 6.11: Supervisors

6.6. SCHEDULE

Proper planning is important to monitor progress and minimize the chance of delays. Table 6.12 outlines the external and internal deadlines for this thesis. *Internal deadlines* acts as guidelines for internal tracking purposes while *external deadlines* are shared with third parties and are thus less flexible. Note that these dates are subject to changes, the deadlines are set up according to the guidelines of the Technical University of Delft.

Date	Type	Deadline	Deadline
13-05-2024	AAAM finished	Internal	The automated assembly assessment methods needs to be finished
07-06-2024	Midterm	External	The midterm review is a go/no-go moment where progress is presented and where there will be reviewed if the planning is on track or if adjustments needs to be made.
09-08-2024	AAAM incorporated	Internal	The automated assembly assessment methods need to be incorporated into the system
16-08-2024	Green light review	External	The report needs to be handed in (iaw section 6.3), this serves as a go-no go moment
13-09-2024	Hand in	External	
TBD	MSc Defence	External	twenty working days after the hand in, an defence presentation occurs were the findings will be published

Table 6.12: Deadlines, where the kick-off meeting takes place at the 19th of February, 2024

fig. 6.1 shows the Gantt chart planning based the project plan, the planning takes holidays into account. WP7, as marked red, can be skipped entirely if any delays occur, this increases flexibility of the planning.

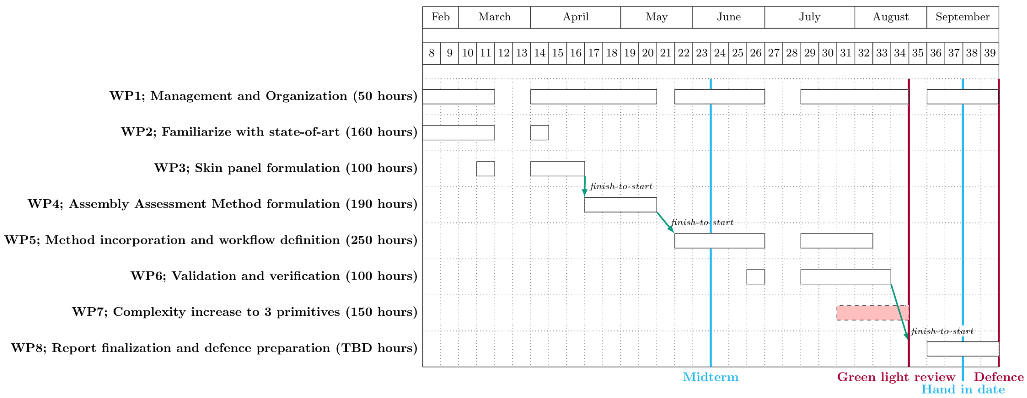


Figure 6.1: Gantt Chart of Thesis

6.7. CONCLUSION

This chapter discussed the research plan, including the work that needs to be done in order to fulfill the thesis. In the first instance, a setup with two-part primitives and one joint will be investigated, the automated assembly assessment method will find a layout that is structurally possible. After this method is researched and integrated within the overall MDAO workflow, it can be validated and run to check the effect of the dynamic optimization scheme. This process will be repeated for three primitives.

7 Conclusion

This report serves as the literature study for the MSc. Thesis titled "Enable Design for Manufacturing and Assembly in Dynamic MDAO workflows" by K.H. Vlessert. This study is conducted as parts of the requirements for the Master's degree.

The world finds itself in an era where conventional design processes are becoming more collaborative, involving multiple partners and disciplines in the design process of one single part. New innovative designs have to be delivered faster with better performance, while the complexity of modern designs increases. Multidisciplinary Design Analysis and Optimization (MDAO) is a method developed over thirty years ago that holds the potential to meet these demands. By creating a tool that can automatically incorporate various disciplines (such as strength analysis, and cost estimation), products can be designed faster and more accurately.

Although developed over 30 years ago, it is not widely adopted within the industry due to a number of technical and non-technical reasons. One significant limitation has been the absence of production considerations within the MDAO process itself. Currently, these considerations are typically addressed at later stages by involving production experts, who then modify the design to make it feasible for production. However, such alterations often result in increased costs and compromised product performance.
To solve this issue, this thesis aims to incorporate the Design for Manufacturing and Assembly (DfMA) methods into the MDAO workflow. By integrating DfMA into MDAO, the optimization process can be enhanced, resulting in more optimized outcomes where the need for design alterations is reduced. This is done by concluding the following research objective:

Investigate how to integrate manufacturing and assembly into early product design stages by using Multidisciplinary Design Analysis and Optimization (MDAO) framework, substantiated by a case study focused on (a section of) a wingbox.

The study extends the existing Design and Engineering Engine (DEE) by ir. A.M.R.M. Bruggeman, which lays out a blueprint for the setup of the entire MDAO system, from the requirement to the final validation. The latest version (as of January 2024) incorporates Design for Manufacturing (DfM) considerations into the workflow, allowing for the MDAO system for automatic optimization of a wing rib. The following components are building blocks of the DEE and therefore key aspects of the thesis:

- **Manufacturing Inclusion:** Research of ir. M. Nikitin focused on integrating manufacturing considerations into the DEE. This enabled the workflow to automatically optimize a wing rib, incorporating two manufacturing methods: machining for metal parts and stamp forming for composite parts.
- **Dynamic workflows:** Dynamic workflows can alter the performed workflow based on input variables. For manufacturing inclusion, different manufacturing methods required distinct analysis tools (e.g., one for stamp forming and one for machining). The dynamic workflow automatically selects the corresponding tool based on the manufacturing process type.
- **Requirements Management Module (RMM):** The RMM systematically incorporates the requirements into the workflows. The document-based requirements that are transformed into model-based require-

ments which are machine-readable. The system automatically takes them into account and can generate a compliance report for the final design.

- **Knowledge- and graph-based Agile Design for Multidisciplinary Optimization System (KADMOS):** KADMOS, an open-source package, formulates the workflow by analysing disciplines, inputs, and outputs. It creates a visual representation of how the workflow can be executed. These visual graphs can be altered to the customer's demand.
- **Optimus:** Optimus is a PIDO tool (Process Integration and Design Optimization) tool. This tool takes the formulated workflow generated by KADMOS and transforms it into an executable workflow.
- **Cost Analysis Tool for Manufacturing of AIrcraft Components (CATMAC):** CATMAC, developed by GKN Fokker, estimates the costs of the designed part by using analytical methods. It considers factors such as material costs, machining costs, and hourly wages. This tool allows for cost minimization as an objective in the design process.
- **Manufacturing Information Model (MIM):** The MIM streamlines manufacturing data throughout the process, connecting information to parts regarding the method, material, equipment, and site. Although developed, the MIM has not yet been integrated into the DEE.

During this thesis, the state-of-the-art workflow will be extended by adding multiple parts and assembly methods. Each of these parts can be either metal or composite, resulting in joints between metal-metal, composite-composite, or composite-metal parts. Three corresponding assembly methods have been selected to be incorporated within the DEE:

1. **Riveting:** This method will be used for joining two metal parts. Riveting is chosen since it is an industry-standard and has ease of assembly. Riveting needs one-sided access while assembled.
2. **Adhesive bonding:** Two composite parts will be joined using adhesive bonding. This method is widely adopted within the industry and ensures uniform load distribution, enhancing structural integrity. Bonding needs one-sided access.
3. **Bolting:** Metal parts will be bolted to composite parts. Bolting is a well-established technique that helps against the delayering of composite plates. Bolting needs double sided access.

To incorporate assembly considerations within the DEE framework, a research plan is created. Firstly, time is planned to familiarize with the current state-of-the-art framework. Following this, a new part (skin panel) will be optimized in a similar manner to the wing rib studied by ir. M. Nikitin. Once both the wing rib and the skin panel are defined, a tool needs to be developed to assess if the two products can be assembled. This Automatic Assembly Assessment Method (AAAM) will be integrated into the workflow to determine the feasibility of assembly. Furthermore, as this method may generate a large amount of data, research will be done on efficient use of data.
Firstly, research will be done on an assembly of two parts, followed by an increasing number of parts. The ultimate goal is to demonstrate that DfMA can be effectively incorporated into the DEE.

Bibliography

[1] I. Dostaler, *Avoiding rework in product design: evidence from the aerospace industry,* International journal Quality and Reliability Management **27**, 5 (2010).

[2] P. D. Ciampa and B. Nagel, *Agile paradigm: The next generation collaborative mdo for the development of aeronautical systems,* Progress in Aerospace Sciences **119** (2020), 10.1016/j.paerosci.2020.100643.

[3] J. Sobieszczanski-Sobieski, A. Morris, M. v. Tooren, G. L. Rocca, and W. Yao, in *Multidisciplinary design optimization: Supported by Knowledge Based Engineering* (Wiley, 2015) pp. 1–378.

[4] J. Haymaker and F. Lee Flager, *A comparison of multidisciplinary design, analysis and optimization processes in the building construction and aerospace industries,* International Conference on Information Technology in Construction **84** (2007).

[5] I. van Gent, B. Aigner, B. Beijer, J. Jepsen, and G. La Rocca, *Knowledge architecture supporting the next generation of mdo in the agile paradigm,* Progress in Aerospace Sciences **119**, 100642 (2020).

[6] S. Shahpar, *Challenges to overcome for routine usage of automatic optimisation in the propulsion industry,* Aeronautical Journal **115**, 615 (2011).

[7] T. Van Der Laan and L. Hootsmans, *Consequences of multi or single disciplinary optimization results on aircraft component cost and weight, 2018 Aviation Technology, Integration, and Operations Conference,* (2018), 10.2514/6.2018-3026.

[8] A.-L. Bruggeman, M. Nikitin, G. La Rocca, and O. K. Bergsma, *Model-based approach for the simultaneous design of airframe components and their production process using dynamic mdao workflows,* (2023).

[9] M. Nikitin, *Dynamic MDAO workflows to Enable Design for Manufacturing of Aircraft Structural Components*, Msc thesis, Delft University of Technology (2024).

[10] R. Belie, *Non-technical barriers to multidisciplinary optimization in the aerospace industry,* AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization **9** (2002), 10.2514/6.2002-5439.

[11] E. Moerland, T. Pfeiffer, D. Böhnke, J. Jepsen, S. Freund, C. M. Liersch, G. Pinho Chiozzotto, C. Klein, J. Scherer, Y. J. Hasan, and J. Flink, *On the design of a strut-braced wing configuration in a collaborative design environment,* AIAA Aviation Technology, Integration, and Operations Conference **17** (2017), 10.2514/6.2017-4397.

[12] T. Op., *System Engineering Vision 2020* (International Council on Systems Engineering (INCOSE), 2020) p. 15, document No: INCOSE-TP-2004-004-02.

[13] A. M. Bruggeman and G. La Rocca, *From requirements to product: an mbse approach for the digitalization of the aircraft design process,* INCOSE International Symposium **33**, 1688 (2023).

[14] J. Vandenbrande, T. Grandine, and T. Hogan, *The search for the perfect body: Shape control for multidisciplinary design optimization,* .

[15] G. La Rocca, *Knowledge Based Engineering Techniques to Support Aircraft Design and Optimization*, Phd thesis, Delft University of Technology (2011).

[16] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, *OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization,* Structural and Multidisciplinary Optimization **59**, 1075 (2019).

[17] P. D. Ciampa and B. Nagel, *Towards the 3rd generation mdo collaborative environment,* Congress of the international council of the Aeronautical Sciences **30** (2016).

[18] M. Hoogreef, *Advise, Formalize and Integrate MDO Architectures*, Phd thesis, Delft University of Technology (2017).

[19] I. van Gent, *Agile MDAO Systems: A Graph-based Methodology to Enhance Collaborative Multidisciplinary Design*, Phd thesis, Delft University of Technology (2019).

[20] M. Baan, B. van Manen, P. Andersson, R. Augustinus, R. Braun, C. Jouannet, G. La Rocca, B. Beijer, M. Panarotto, and P. Edholm, *Defaine - design exploration framework based on ai for front-loaded engineering,* European Conference for Aero-Space Sciences **10th** (2023), 10.13009/EUCASS2023-698.

[21] I. Van Gent, G. La Rocca, and M. F. M. Hoogreef, *CMDOWS: a proposed new standard to store and exchange MDO systems,* CEAS Aeronautical Journal **9**, 607 (2018).

[22] I. Van Gent, G. La Rocca, and L. L. Veldhuis, *Composing MDAO symphonies: graph-based generation and manipulation of large multidisciplinary systems, 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (2017), 10.2514/6.2017-3663.

[23] B. Aigner, I. van Gent, G. La Rocca, E. Stumpf, and L. Veldhuis, *Graph-based algorithms and data-driven documents for formulation and visualization of large mdo systems,* CEAS Aeronautical Journal **9**, 695 (2018).

[24] B. Nagel, D. Böhnke, V. Gollnick, P. Schmollgruber, A. Rizzi, G. LaRocca, and J. Alonso, *Communication in aircraft design: Can we establish a common language?* International Congress of the Aeronautical Sciences, Brisbane, Australia **28** (2012).

[25] M. Alder, E. Moerland, J. Jepsen, and B. Nagel, *RECENT ADVANCES IN ESTABLISHING A COMMON LANGUAGE FOR AIRCRAFT DESIGN WITH CPACS,* Aerospace Europe Conference 2020 (2020).

[26] D. Pate, J. Gray, and B. German, *A graph theoretic approach to problem formulation for multidisciplinary design analysis and optimization,* Structural and Multidisciplinary Optimization **49**, 743 (2013).

[27] R. Sellar, S. Batill, and J. Renaud, *Response surface based, concurrent subspace optimization for multidisciplinary system design,* **714** (1996), 10.2514/6.1996-714.

[28] B. Aigner, *Tutorial of agile; formulating mdo problems and solution strategies,* .

[29] *RCE User Guide*, 8.3.0.201907051305 (DLR, 2019).

[30] M.F.M. Hoogreef, *Advise, Formalize and Integrate MDO Architectures: A Methodology and Implementation*, Phd thesis, Delft University of Technology (2017).

[31] A.-L. Bruggeman, *Automated Execution Process Formulation using Sequencing and Decomposition Algorithms for Collaborative MDAO*, Msc thesis (2019).

[32] A. Raju Kulkarni, M. Hoogreef, and G. La Rocca, *Combining semantic web technologies and KBE to solve industrial MDO problems, 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (2017), 10.2514/6.2017-3823.

[33] D. Bansal, *Automating Inclusion of Production Considerations in the Conceptual Design of Aircraft Structures*, Msc thesis, Delft, universtiy of technology (2022).

[34] A. Raju Kulkarni, G. La Rocca, T. Berg, and R. Dijk, *A knowledge based engineering tool to support front-loading and multidisciplinary design optimization of the fin-rudder interface,* Aerospace EUrope CAES Conference **6** (2017).

[35] A. Smith, *PMI's Pulse of the Profession: Requirements Management — A Core Competency for Project and Program Success*, Tech. Rep. (Project Management Institute, 2014).

[36] A.-L. Bruggeman, B. Van Manen, T. Van Der Laan, T. Van Den Berg, and G. La Rocca, *An mbse-based requirement verification framework to support the mdao process, AIAA AVIATION 2022 Forum*, (2022), 10.2514/6.2022-3722.

[37] *ISO/IEC/IEEE 24765:2017 Systems and software engineering — Vocabulary* (International Organization for Standardization, Geneva, Switserland, 2017) accessed: 10/02/2024.

[38] L. Boggero, P. D. Ciampa, and N. Björn, *An mbse architectural framework for the agile definition of system stakeholders, needs and requirements,* AIAA AVIATION 2021 FORUM 10.2514/6.2021-3076, https://arc.aiaa.org/doi/pdf/10.2514/6.2021-3076 .

[39] T. Berg and T. Laan, *A multidisciplinary modeling system for structural design applied to aircraft moveables, AIAA AVIATION 2021 FORUM,* Aerospace Europe Conference 2023 - 10th EUCASS **10** (2021), 10.2514/6.2021-3079, https://arc.aiaa.org/doi/pdf/10.2514/6.2021-3079 .

[40] T. Van Der Laan and T. Van Den Berg, *An open source part cost estimation tool for mdo purposes, AIAA AVIATION 2021 FORUM,* (2021), 10.2514/6.2021-3058.

[41] O. Bergsma, *A Three dimensional simulation of fabric draping,* Phd thesis, Delft, universtiy of technology (1995).

[42] S. P. Venkatesan, B. N, J. Jayaprabakar, and P. N. Kadiresh, *Modelling and analysis of aircraft wing with and without winglet,* International Journal of Ambient Energy **42**, 1 (2018).

[43] J. Teuwen, *Lecture notes 3.4.6 introduction to aerospace structures and materials. main takeaways on wing structures.* (2022).

[44] A. Desai, *Ease of product assembly through a time-based design methodology,* Assembly Automation **39**, 881 (2019).

[45] M. C. Y. Niu, *Airframe Stress Analysis and Sizing,* 3rd ed. (Hong Kong Conmilit Press LT., Hong Kong, 2016) pp. 273–393.

[46] D. A. Fanella, R. Amon, B. Knobloch, and A. Mazumder, *Bolts and rivets,* in *Steel Design for Engineers and Architects* (Springer US, Boston, MA, 1992) pp. 186–230.

[47] G. Sun, S. Xiao, J. Wu, S. Yu, M. Wei, J. Qin, and M. Zang, *Study on the static stability of aluminum alloy single-layer spherical reticulated shell,* Journal of Building Engineering **84** (2024), https://doi.org/10.1016/j.jobe.2024.108595.

[48] M. C. Y. Niu, *Composite Airframe Structures,* 3rd ed. (Hong Kong Conmilit Press LT., Hong Kong, 1996) pp. 285–325.

[49] L. F. M. d. Silva and R. D. Adams, *Techniques to reduce the peel stresses in adhesive joints with composites,* International Journal of Adhesion and Adhesives **27**, 227 (2007).

[50] J. M. Guerrero, A. Sasikumar, J. Llobet, and J. Costa, *Experimental and virtual testing of a composite-aluminium aircraft wingbox under thermal loading,* Aerospace Science and Technology **138**, 108329 (2023).

[51] J. Min, J. Li, B. Carlson, Y. Li, and J. Lin, *Mechanical property of al alloy joints by friction stir blind riveting,* Procedia Engineering **81**, 2036 (2014).

[52] C. Rans and P. V. Straznicky, *Avoiding knife-edge countersinks in glare through dimpling,* Fatigue & Fracture of Engineering Materials & Structures **28**, 633 (2005).

[53] Thiago Destri Cabral, *A First Approach to Structural Health Monitoring of Adhesive Bonded Joints in Pipelines Using Integrated Fiber Optic Sensors,* (2016), 10.13140/RG.2.2.10258.25283, publisher: Unpublished.

Appendices

A

WORKPACKAGES

WP 1: Management and Organization	
Number of hours:	50
Begin date:	29th of January 2024
Final date:	21st of July 2024
Purpose	Manage the project as clear as possible, preparing the weekly meetings, midterm meeting and rest of the generic meetings

Inputs:
- Project plan
- Brightspace documents, rubrics, and thesis guidelines

Research methodology:

Subtasks:

- Meetings	30 hours
- Rest	20 hours

Outputs:
- Agenda and minutes

Milestones

WP Description made at 6th of January 2024

WP 2: Familiarize with state-of-art	
Number of hours:	160
Begin date:	29^{th} of January 2024
Final date:	21^{st} of April 2024
Purpose	Since this thesis builds further upon the existing research of A.L Bruggeman, it is important to fully understand the already exiting work. Within this workpackage, time is spent to familiarize with the existing work to a sufficient level that the thesis student can alter and change the workflow to it owns liking.

Inputs:

- Design Engineering Engine (Documents of PHD A.L. Bruggeman)
- Knowledge of A.L. Bruggeman and M. Nikitin (PHD student and MSc student)
- Parapy

Research methodology:

1. Gaining access to all documents and relevant pieces.
2. Running the original scripts, without extra parts.
3. Alter and run parts individually to see the individual effects.
4. Design several test cases within the workflow

Subtasks:

- Research overall workflow
- Research requirement management module (RMM)
- Research data input (such as available materials)
- Options to change Dynamic workflows to static workflows and vice versa

Outputs:

- The entire system overview

Milestones:

WP Description made at 6^{th} of January 2024

WP 3: Skin panel formulation	
Number of hours:	100
Begin date:	26th of February 2024
Final date:	24th of March 2024
Purpose	During the research of MSc student M. Nikitin, the production of a wing rib is designed via the DEE. In this workpackage, a skin panel needs to be formulated and automatically created such that an assembly can be made with two products.

Inputs:

- Rib optimization method by KBE method
- Structural data calculations (book NUI)

Research methodology:

1. Use the knowledge from WP2 to the geometrics of the wing rib
2. Analyze the rules of a wing planform and how it is established
3. Create a skin panel within the existing MDAO workflow
4. Define how the skin panel internal forces are defined and analyzed
5. Create both the wing rib and the skin panel in the tool simultaneously

Subtasks:

- Research required wing planform
- Research method of wing ribs
- Research and analyze required thickness of skin panel
- Research on how both the wing rib and skin panel can be created simultaneously

Outputs:

- Automatic creation of both the skin panel and the wing rib simultaneously within the MDAO workflow

Milestones

WP Description made at 6th of January 2024

WP 4: Assembly assessment method formulation	
Number of hours:	190
Begin date:	18th of March 2024
Final date:	21st of April 2024
Purpose	A method should be created that assesses if an object can be assembled. Multiple primitives created by WP3 should be assembled. This Assembly Assessment Method (AAM) will grade the assembly by a scale of -1 to 1, where 1 means that its assembled and -1 meaning that it can not be assembled. **Note** that this workpackage will solely assess this and wont be integrated into the system)

Inputs:

- Production assessment method of M. Nikitin
- Output WP3: relevant "primitives" that are used in the assembly
- Assembly related requirements (book by NIU)
- Manufacturing Information Model, first two modules by MSc Student D. Bansal

Research methodology:

1. Analyze production assessment tool
2. List the requirements for assembly (geometrical requirements)
3. Create an Automated Assembly Assessment Tool (AAAT)
4. Quantify the possibilities of incorporating the MIM within the AAAT

Subtasks:

- Ensuring that the system locates each primitive correctly
- Translating the assembly guidelines into requirements
- Scaling the terms assembly to -1 and 1
- Connecting a skin panel to a wing rib

Outputs:

- An automated assessment method

Milestones:

- Answer RQ1: *"What method can be employed to automatically assess the feasibility of part assembly in MDAO workflows?"*

WP Description made at 6th of January 2024

WP5: Assembly method incorporation and workflow definition	
Number of hours:	250
Begin date:	22nd of April 2024
Final date:	23rd of June 2024
Purpose	To incorporate the assembly method made in WP4 into the entire overview researched in WP2 and to streamline the workflow such that both the two options

Inputs:
- Output WP2: Entire system overview
- Output WP4: automated assessment method
- Subtools such as RMM and MIM
- CADMAC (Cost analysis tool by Fokker)

Research methodology:
1. Research how the production techniques are incorporated in the workflow
2. Incorporate the AAAT into the workflow
3. Continuously test and run test cases to see the compliance
4. Incorporate the MIM within the MDAO worfkow

Subtasks:
- Including the method within the MDAO workflow
- Incorperate the RMM to its fulleste extend
- Incorperate the switch
- Incorporate CADMAC (cost extimation tool)

Outputs:
- Entire updated system overview
- Optimizable workflow

Milestones
- Answer to RQ2: “*What challenges arise when incorporating the assembly and manufacturing assessment methods within the MDAO workflows?*”
- Answer to RQ3: “*How can assembly related requirements, such as material compatibility be incorporated into MDAO workflows?*”

WP Description made at 6th of January 2024

WP 6: Validation and Verification	
Number of hours:	100
Begin date:	20th of May 2024
Final date:	6th of July 2021
Purpose	Both validate and verification of the working product by checking with various test cases of realistic and correct results. Also check the time advantages of Dynamic workflows compared to static workflows.

Inputs:

- WP 5: entire system
- Various test cases

Research methodology:

1. Set up test cases for validation and verification
2. Run test cases and check for errors
3. Set up possible designs with SMEs for static analysis and perform this analysis
4. Set up dynamic MDAO workflow and run this
5. Compare the dynamic and static results for concluding efficiency.

Subtasks:

- Testing of system robustness
- Testing system dynamic vs static ability
- Validation via means of test cases

Outputs:

- Validated workflow

Milestones :

- Answer to RQ4: *By what means could the challenges posed by a large combination of parts and joints and their corresponding manufacturing and assembly methods be effectively addressed within a MDAO workflow?*

WP Description made at 6th of January 2024

WP7: Complexity increase to 3 primitives	
Number of hours:	150
Begin date:	24th of June 2024
Final date:	21st of July 2024
Purpose	The first iteration was solely focused on the assembly of two primitives, after this is fully validated and time allows the research will be broaden by increasing the number of primitives. This will redo all the tasks as described in earlier work packages with more primitives.

Inputs:

- Output WP6: Validated MDAO system
- Knowledge of Work packages 2 to 6.

Research methodology:

- Perform the research methodologies done from workpackage 2 to 6

Subtasks:

- As defined in WP 2 to 6, extended to multiple primitives.

Outputs:

- Validated workflow for 3 primitives

Milestones

- Answer RQ1+: *"What method can be employed to automatically assess the feasibility of part assembly in MDAO workflows?"* For 3 primitives.
- Answer RQ2+: *"What challenges arise when incorporating the assembly and manufacturing assessment methods within the MDAO workflows?"* For 3 primitives.
- Answer RQ3+: "*How can assembly related requirements, such as material compatibility be incorporated into MDAO workflows?"* For 3 primitives.

WP Description made at 6th of January 2024

B

Final publication

Dynamic Multi-disciplinary Design Analysis and Optimization workflows to enable Design for Assembly

Vlessert, Kees[1]
Delft, Universtiy of Technology, Kluyverweg 1, 2629 HS, The Netherlands

Multi-disciplinary Design Analysis and Optimization (MDAO) combines system engineering principles with numerical optimization methods to evaluate and optimize design configurations. However, manufacturing and assembly considerations are often neglected, resulting in theoretically optimal designs that require costly redesigns to ensure producibility. This research integrates assembly requirements for multi-part designs into MDAO workflows, continuing on previous research by the TU Delft which addressed manufacturing considerations for single-part products. The methodology incorporates Joint Assessment Methods (JAM) and Geometry Independent Assembly (GIA) tools within the Design and Engineering Engine (DEE). The DEE is a framework for building MDAO workflows based on design requirements. JAMs assess joint feasibility for specific assembly methods, while GIAs evaluate non-joining constraints such as material incompatibility. A wing rib-skin panel use case demonstrates this approach, with the optimizer selecting an optimal, producible solution. Although effective, the method introduces significant computational overhead, with iteration time ranging from 85 to 205 seconds, depending on the assembly method. Future research may reduce this through early break-off of unfeasible solutions and can include new assembly requirements, to optimize more complex products.

Nomenclature

DEE	= Design and Engineering Engine	KBE	= Knowledge Based Engineering
DfA	= Design for Assembly	KPI	= Key Performance Indicators
DfM	= Design for Manufacturing	MDA	= Multi-disciplinary Design Analysis
DfMA	= Design for Manufacturing and Assembly	MDAO	= Multi-disciplinary Design Analysis and Optimization
DV(B)	= Design Variable (Bound)	QoI	= Quantity of Interest
ES	= Enabling System	SADS	= System Architectural Design Space
GDA	= Geometry Dependent Assembly	SoI	= System of Interest
GIA	= Geometry Independent Assembly	SWF	= Subworkflow
JAM	= Joint Assessment Method	XDSM	= eXtended Design Structure Matrix

I. Introduction

The aviation industry faces the dual challenge of sustaining growth while reducing environmental impact. Addressing this requires faster development of innovative products, shorter time-to-market, and optimized performance. During product design, engineers rely on input from multiple disciplines that work together, such as strength or aerodynamic simulations. With growing projects which have become more complex, these

[1]MSc Student, Flight Performance and Propulsion

disciplines can no longer function as isolated tools; instead, they must operate as an integrated system. To enable this, design processes must be automated to create these new complex products [1–3].

One approach that supports this integration is Multi-disciplinary Design Analysis and Optimization (MDAO). Developed over 30 years ago, MDAO combines system engineering principles with numerical optimization methods [4]. It enables the evaluation of various design configurations against predefined constraints and objectives, such as cost and weight [5]. Despite its potential, MDAO adoption in industry remains limited, partly due to the high complexity of the initial setup [6].

One of MDAOs main advantages is the ability incorporate various disciplines into a single workflow. However, there has been limited research on incorporating Design for Manufacturing and Assembly (DfMA) considerations. Although some projects include manufacturability or assemblability aspects, they do not necessarily drive design decisions. As a result, theoretically optimal designs often require later changes to make them producible, affecting both performance and cost. [2, 7–9].

This paper presents a method for integrating manufacturing (part-primitive creation) and assembly (joining multiple parts-primitive together) considerations into MDAO workflows. Where a part primitive is defined as *"a parametric building block to define the product"* [10], (e.g., a skin panel). The approach integrates the DfMA requirements into the Design and Engineering Engine (DEE) and is thereby included into the workflow. The DEE is a blueprint to translate requirements into an MDAO formalization (a workflow that can be executed) [11]. This approach enables DfMA considerations to be evaluated. This reduces late-stage modifications, lowering both cost and time-to-market [12]

Previous research has focused on optimizing single part-primitives, including manufacturability (ease of manufacturing) considerations [8, 9]. This work extends that by incorporating assemblability, or ease of assembly, on optimizations with multiple part-primitives. It includes assessing joint feasibility and addressing assembly requirements that are not directly tied to geometry, such as material compatibility. This is done by addressing the research objective and research questions outlined in Section I.A.

A. Research objective and research questions

The primary objective of this thesis is:

Investigate the integration of manufacturing and assembly into early product design stages by using Multi-disciplinary Design Analysis and Optimization (MDAO), substantiated by a case study focused on a section of a wingbox.

To achieve this. The following research questions are addressed:

- **(RQ-1)** What assembly assessment methods can be developed to assess the feasibility of assembly/joining of parts?
- **(RQ-2)** Would the incorporation of the **assembly assessment methods** and the **manufacturing assessment methods** result into challenges within the design and engineering engine?
- **(RQ-3)** What are the advantages and disadvantages of a dynamic MDAO workflow compared to a static one when manufacturing and assembly considerations are included?

Within these research questions, the first question, RQ-1, focuses on the question *how* the assembly can be evaluated. Where the second research question integrates this assessment methodology into the DEE. The third research question evaluates dynamic and static workflows, this is an ongoing research spanning over various projects [9, 13].

B. Overview of MDAO workflow with Design for Manufacturing inclusion

This research builds on the Design and Engineering Engine (DEE). The goal of the DEE is to generate a blueprint for the automatic creation of MDAO workflows. It enables users to define workflows from requirements to the final product [11]. The DEE exists of multiple components: the Requirement Management Module (RMM), KADMOS, and CMDOWS. The DEE transforms requirements into a workflow. KADMOS uses the RMM data to generate Extended Design Structure Matrix (XDSM), which visualizes MDAO workflows. CMDOWS stores workflow information in a standardized format. Prior work by Nikitin [9] integrated requirements into the DEE framework.

A main component used throughout this research is the RMM. Within the RMM, users define machine-readable requirements in an Excel tool, which are processed by the RMM into an MDAO formalization. The provides a standardized way to integrate and use requirements for the optimization process. The MDAO workflow is run in Optimus[1] [13]. Optimus is a third party software which is used to run, alter and post-process MDAO processes. The overview of the steps within the RMM are illustrated in Fig. 1.

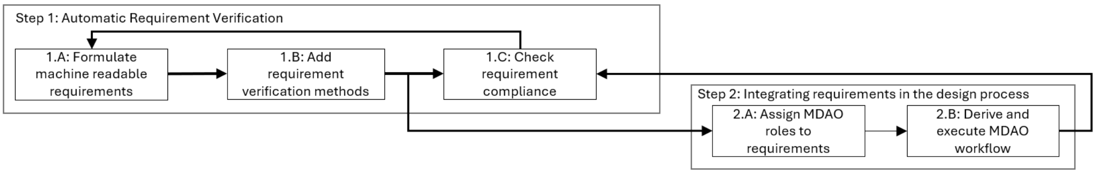


Figure 1. Overview of the process steps in the Requirement Management Module (based on [13]).

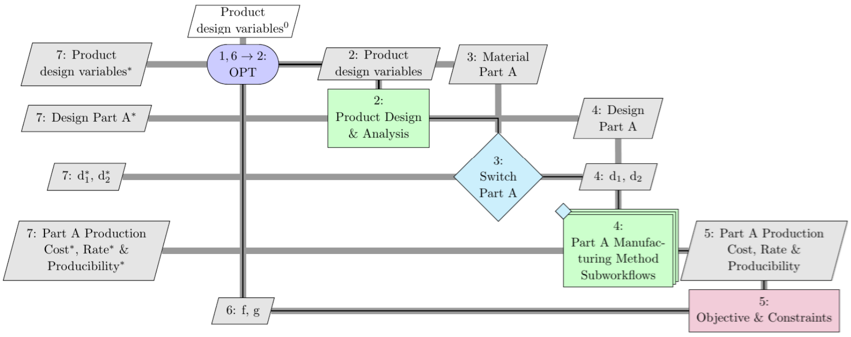
As shown in Fig. 1, The RMM consists of two steps [13]:

- **Step 1: Automatic requirement verification**
  - (1.A) *Formulate machine readable requirements*: Transforms document-based requirements into machine-readable requirements, this includes requirements regarding manufacturing.
  - (1.B) *Add requirements verification methods*: Each requirement is assigned a verification method, consisting of a means of compliance and a test case. The means of compliance defines how compliance will be demonstrated, while the test case is defined as the "*the technical implementation of the means of compliance*" [13] and includes the simulation tools used to show compliance .
  - (1.C) *Check requirements compliance*: Compliance is verified by evaluating the test case results (1.B) against predefined criteria (1.A).
- **Step 2: Requirements to drive the design process**
  - (2.A) *Assign MDAO roles to requirements*: Requirements are categorized as constraints, objectives, design variables, parameter input or quantity of interest (QoI), determining how each requirement is implemented into the workflow. The definition of these MDAO roles is given in Bruggeman et al. [13].
  - (2.B) *Derive MDAO workflow*: Once roles are assigned, the MDAO workflow is formalized. The formalization based on Bruggeman et al. [8] is discussed below.

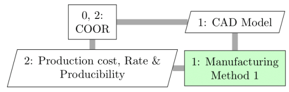
Fig. 2a shows a dynamic MDAO workflow that includes DfM considerations. “Dynamic” means that certain requirements may become active depending on design choices e.g. material selection. In Fig. 2a, based on the outcome of a certain condition, either Fig. 2b and Fig. 2c is activated. In contrast, static workflows remain unchanged regardless of design choices. Within static workflows, each tool is always performed while in dynamic workflows, some tools may be unused depending on the conditions on which the tool is activated.

Three key concepts enable dynamic workflows: subworkflows (SWF), switches and branches. SWFs are workflows in a workflow. The top-level workflow represents the System of Interest (SoI), which is the product

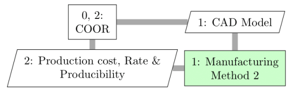
[1] https://www.noesissolutions.com/our-products/optimus, accessed on 02-02-2025



(a) XDSM illustrating a dynamic MDAO workflow used to optimize a single-part primitive w.r.t. manufacturing cost, rate and manufacturability



(b) XDSM manufacturing method 1



(c) XDSM manufacturing method 2

Figure 2. Dynamic MDAO formalization with SWF which are activated depending on material selection [9].

analyzed or optimized. Within his top-level workflow, a executable block points towards a lower-level SWF. This SWF can be a tool, a chain of tools or an entire optimization [8]. Within this research, the SWF is an Enabling System (ES), defined as a "*system that supports a system of interest during its life cycle stages but does not necessarily contribute directly to its function during operation* [14], such as the manufacturing method.

A branch is a conditional block that can represent an analysis, SWF, or mathematical function. If the switch activates the branch, its executed, otherwise, it is ignored. A swich (show as a blue diamond in Fig. 2a) activates a branch based on a Boolean condition, e.g., "*If <parameter> is <value> , then <branch>*", meaning that if a certain parameter equals a certain value, then a certain branch is activated and executed.

In the workflow of Fig. 2a, the optimizer selects a design vector, which is used to create a parameterized geometrical representation in ParaPy[2]. Based on the material selection, the switch then corresponds the correct manufacturing SWF, which calculates manufacturability, production cost, and rate.

This baseline setup optimizes a single-part primitive. This research expands the workflow by (1) increasing the number of part primitives and (2) introducing joints for multi-part assembly. The methodology for these additions is described in Section II, an user case is used as a proof of concept in cref:Testcase, and this case is evaluated in Section IV.

II. Methodology

This section outlines how assembly requirements are imposed during an optimization. Figure 3 shows how these requirements are integrated into the DEE and what steps are taken to ensure alignment with the RMM, so that the RMM can formulate the workflows correctly.

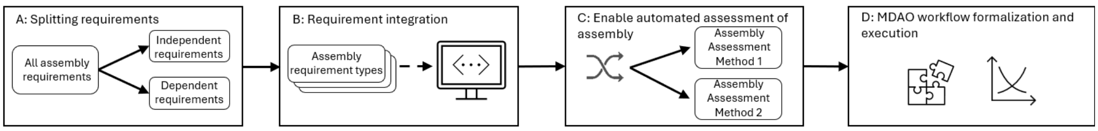


Figure 3. Schematic overview the proposed methodology. Step A - Assembly requirements are categorized based on when they are imposed in the MDAO process. Step B - The requirement sets are incorporated into the RMM. Step C - The switch is updated such that the correct requirements can be activated at the correct times. Step D, a dynamic workflow is formulated via the RMM

[2]Parapy.nl, accessed on 02-02-2025

Step A distinguishes all assembly requirements in to various types based on when they are imposed, this distinguishment is elaborated in Section II.A. Step B shows how these requirement types are integrated into the RMM (Section II.B). Step C focuses on the switch mechanism (Section II.C), the switch is updated so that the correct requirements can be imposed at the correct time. Step D integrates all components required for assembly inclusion into the baseline workflow (Section II.D).

A. Types of assembly requirements

Assembly processes introduce new requirements on the product. Not all requirements need to be evaluated at the same stage. By categorizing them according to *when* they apply, computational efficiency could potentially be improved by removing infeasible designs earlier. This research categorizes all assembly-related requirements into two types:

- **Geometry Dependent Assembly (GDA) Requirements:** These depends on the SoI geometry. These includes the requirements induced by specific assembly methods. For example, adhesive bonding may require that the glue application area exceeds 100 cm^2 or for welding, the weld length must be above 50 cm.
- **Geometry Independent Assembly (GIA) Requirements:** These requirements are independent of geometry, or information extracted from the geometric model. They can be evaluated using the design vector, information from a database, or a disciplinary tool. Since GIA requirements are independent of the created design, they can be assessed before the design is created. For example, a requirement like "withstand 150 ^{O}C" can be evaluated early if the material is included in the design vector.

GIA requirements can be checked early and are tied to design variables, allowing for early rejection of infeasible solutions. GDA requirements require geometry and are therefore applied later. Although other categorizations could be explored, this division aligns with the stages of the current approach, allowing for early elimination of infeasible designs and more efficient integration of assembly constraints.

B. Requirement integration into the workflow

The following subsections describe how GIA and GDA requirements are integrated in the MDAO workflow.

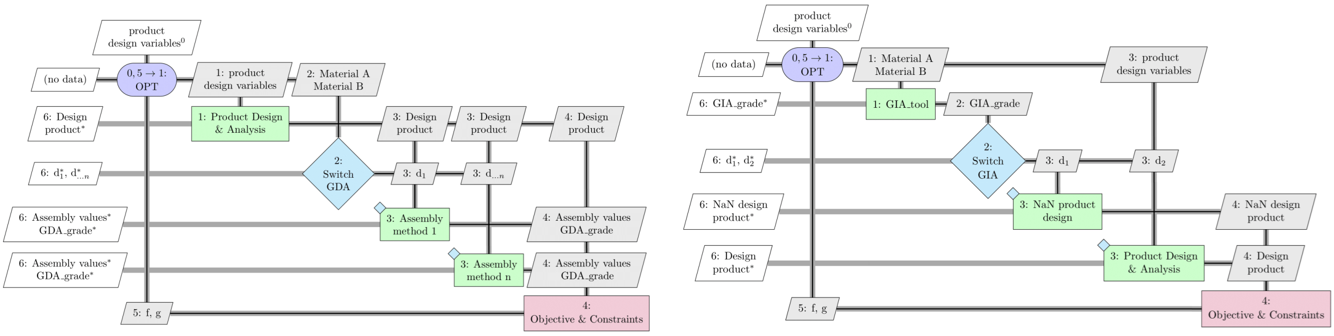
1. Geometry Dependent Assembly (GDA) requirements

GDA-requirements are used in this research to assess joint feasibility based on geometry, such as the available adhesive area or weld length. For each assembly method, a Joint Assessment Method (JAM) is created. The JAMs can be calculations, SWFs, or optimizations. This method is similar to the inclusion of manufacturing requirements [9], where the feasibility is graded (GDA_grade) on a scale from -1 to 1. A grade ≥ 0 indicates a feasible joint; < 0 indicates infeasibility. An example of how this is implemented is shown in Fig. 4a: The optimizer selects the design vector that is used for the geometrical definition. A switch then activates the correct branch corresponding to an assembly method (e.g., welding or bonding), each with a dedicated JAM SWF that calculates the grades and other quantities of interest (QoI).

2. Geometry Independent Assembly (GIA) Requirements

GIA requirements evaluate the compatibility of the parts regardless of the geometry or the assembly method. As shown in Fig. 4b, they can be checked before the geometric representation is made. After the optimizer sets the design vector, a GIA tool assesses compliance and outputs a Boolean value. In principle, this early check could be used to conditionally activate the Product Design & Analysis phase, where the geometry is generated and analyzed. It could also skip this phase, based on the Boolean output as shown in Fig. 4b, resulting in a more efficient workflow, since a portion of unfeasible designs are not evaluated.

However, in the current implementation this is not feasible due to extra implementation complications. Therefore, GIA requirements are incorporated as normal constraints within the optimization, in the same manner as manufacturing considerations. This results in the geometry to be always created and evaluated, independently on the outcome of the GIA tool. This will elaborated more in Section III.



(a) XDSM of GDA requirements

(b) XDSM of GIA requirements

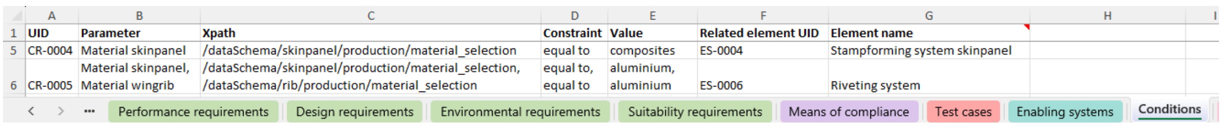
Figure 4. XDSM for GIA checked before and GDA checked after the geometrical definitions.

C. Integration of multi-conditional switches

As discussed in Section I.B, the conditions specified in the Excel tool are transformed by the RMM into Boolean expressions, which Optimus uses to let the Switch activate the appropriate branch [15]. Currently, this transformation is only implemented for single-conditional switches. For example, manufacturing condition CR-0004 (Fig. 5) is translated into: "*If Material is Composites, then Stampforming*", meaning composites should be manufactured through stamp forming in this research.

For more complex cases, such as those needed to evaluate assembly where the chosen JAM is based on multiple parameters, the RMM must convert multi-conditional logic into Optimus syntax. To support this, the RMM has been updated to detect and handle such cases automatically. Optimus supports logical operators like *AND*, enabling expressions such as: "*If (<parameter*$_1$*> is <value*$_1$*>) AND (<parameter*$_2$*> is <value*$_2$*>), then <subworkflow>*" to be generated.

With this update, users can define these complex rules directly in the Excel tool. For instance, the multi-conditional entry CR-0005 in Fig. 5 is automatically converted into "*If (Material skinpanel is aluminium) AND (Material wingrib is aluminium) , then Riveting system*". In this way, Optimus knows that if both two materials are aluminium, the joint is riveting according to CR-0005.



	A	B	C	D	E	F	G	H
1	UID	Parameter	Xpath	Constraint	Value	Related element UID	Element name	
5	CR-0004	Material skinpanel	/dataSchema/skinpanel/production/material_selection	equal to	composites	ES-0004	Stampforming system skinpanel	
6	CR-0005	Material skinpanel, Material wingrib	/dataSchema/skinpanel/production/material_selection, /dataSchema/rib/production/material_selection	equal to, equal to	aluminium, aluminium	ES-0006	Riveting system	

Figure 5. the Excel tool conditions sheet. Where CR-0004 is single-conditional and CR-0005 a multi-conditional

D. Dynamic MDAO workflow formalization

With the integration of the GIA and GDA mechanisms and the updated switch, all necessary elements are in place for incorporating assembly considerations into the MDAO workflow. Based on Fig. 2a, the complete workflow shown in Fig. 6 integrates both a second part primitive and the joint assessment.

In this example, a Design of Experiments (DoE) and optimization are performed, where the DoE varies both the manufacturing methods of Part A, Part B, and the assembly method. This generates a set of optimal designs, one for each possible design option proposed by the DoE.

As shown in Fig. 6, The GIA_tool is applied first within the optimizer to check compatibility constraints. The geometry is then generated, and key parameters such as weight and dimensions are extracted. The workflow continues with two switches, one for each part. Each switch activates the relevant manufacturing SWF based on the input. Each SWF returns the manufacturing cost, rate, and grade, which are included as constraints. These evaluations are based on the geometry produced during the Product Design & Analysis phase.

Lastly, the assembly feasibility is checked by assessing GDA requirements, these are evaluated via JAMs. The corresponding switch activates the correct JAM branch. These tools assess joint feasibility using the geometry of both parts and calculating the cost, rate, and assemblability.

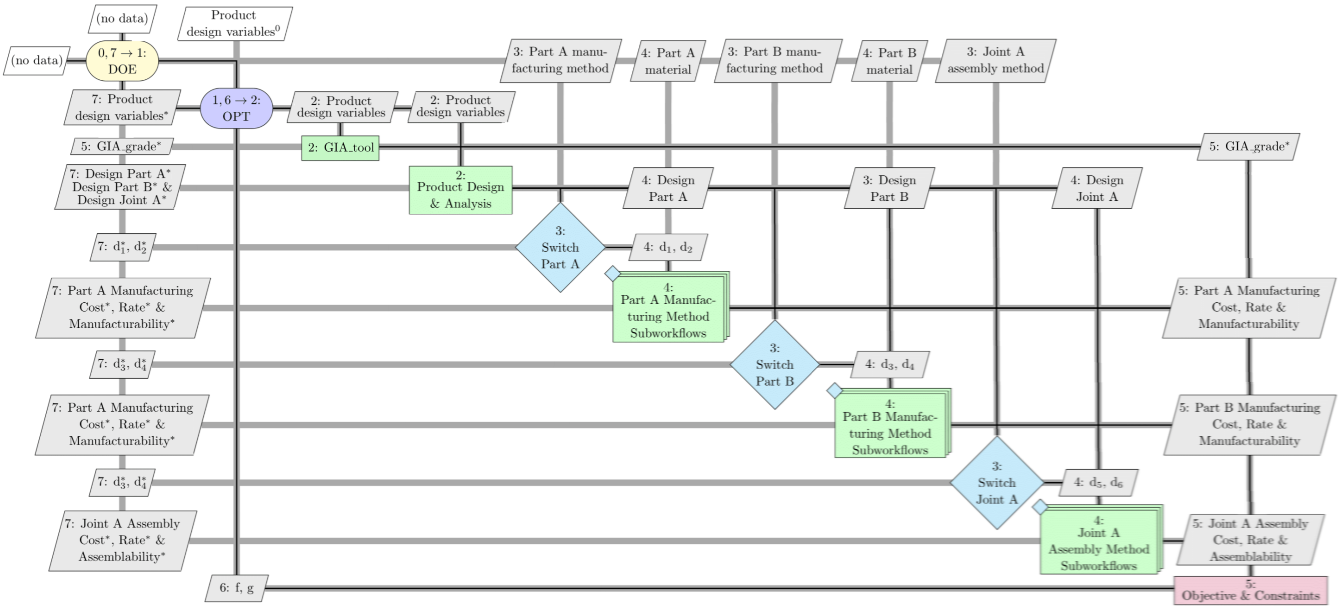


Figure 6. XDSM formalization of dynamic workflow for the optimization of a two-part primitive design. Both parts are tested for manufacturing and assembly considerations. After each switch, a stacked green block with a small blue diamond is illustrated, this indicates multiple possible branches [8]. A full page version of this XDSM is shown in Appendix A.

III. Use case

To validate the methodology proposed in the previous section, a use case is developed to optimize the cost of a wingbox. This wingbox consists of a rib, a skin panel, and a joint between them. An overview of the implementation is shown in Fig. 7.

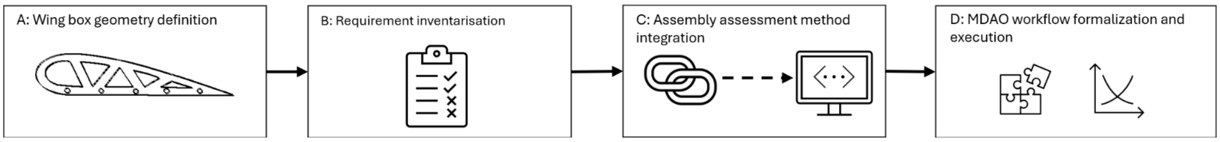


Figure 7. Overview of use case implementation. Step A - Define the geometrical representation of the wingbox. Step B - Inventarisation of all requirements related to the SoI and ES. Step C - Create and integrate JAM to assess joints. Step D, derive the MDAO workflow with DfMA considerations included and execute it.

The process begins by defining the parameterized geometry of the SoI (wingbox), as discussed in Section III.A. This model forms the basis for assessing manufacturability, assemblability, and cost. Requirements for both the SoI and ES are covered in Section III.B. Joint Assessment Methods, representing GDA requirements, are integrated into the workflow as outlined in Section III.C. Finally, the RMM formalizes these data in MDAO workflows, detailed in Section III.D. These workflows are used for the evaluation of the research objective and the research questions, as elaborated in Section IV.

A. Geometrical definition

The geometry used in the optimization process consists of a wing rib and a skin panel, shown in Fig. 8. The geometry of the rib is defined in Nikitin [9]. The skin panel is a flat plate reinforced with stringers, where the geometry is independent on material selection. The parameterized model, developed in ParaPy, includes seven design variables which are introduced in Section III.B, and fully defined in Appendix B.

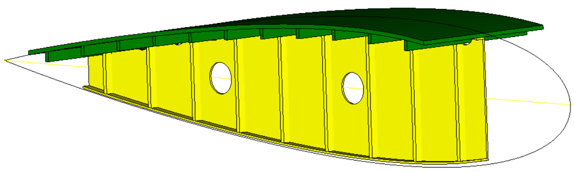


Figure 8. Wing rib (yellow) and Skin panel (green)

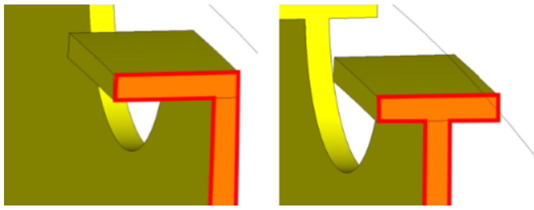


Figure 9. L-Flange (left) and T-Flange (right)

Material selection is directly related to the manufacturing method, which significantly influences the resulting rib geometry [9]. The use case includes two material options: Aluminium 2024-T4 (referred to as aluminium) and carbon fiber LM-PAEK (referred to as composite). In this case, aluminium part primitives are manufactured through machining, and composite parts through stamp forming. Machining provides access to both sides of the part, creating a T-flange as shown on the right in Fig. 9. In stamp forming, a sheet is pressed against a mold. This process results in an L-shaped flange, with the flange bent only to one side, as shown on the left in Fig. 9 [9].

B. Requirement inventarisation

During the requirements inventarisation process, relevant requirements are identified, and the *System Architectural Design Space* (SADS) model is created, as shown in Fig. 10.

This model represents the design space and serves as a basis for the qualification and quantification of the requirement databases. The process begins with the boundary function "*Carry flight loads*", which is performed by the wingbox. The wingbox has several functions; where this research focused on two sub functions: "*Provide stability against panel buckling*" and *"Withstand shear forces"*, fulfilled by the rib and the skin panel respectively. Each of them can be aluminium or composite and must be created (e.g. *Create part shape*). Depending on the material, the manufacturing method and relevant requirements are selected:

machining for aluminium and stamp forming for composites [8]. Joints fulfill the function "*transfer load from part A to part B*". Within this research, it is assumed that the joining method depends on the material selection of both parts. Specifically, aluminium-aluminium connections are riveted, composite-composite joints are connected via adhesive bonding, and bolting is used for aluminium-composite and composite-aluminium connection. Each combination has unique requirements.

The SADS results in nine requirement databases: one for the wingbox (SoI), and eight for the ES: four for manufacturing methods (two parts with two materials each), and four for assembly method (one per method). The following subsections describe these databases.

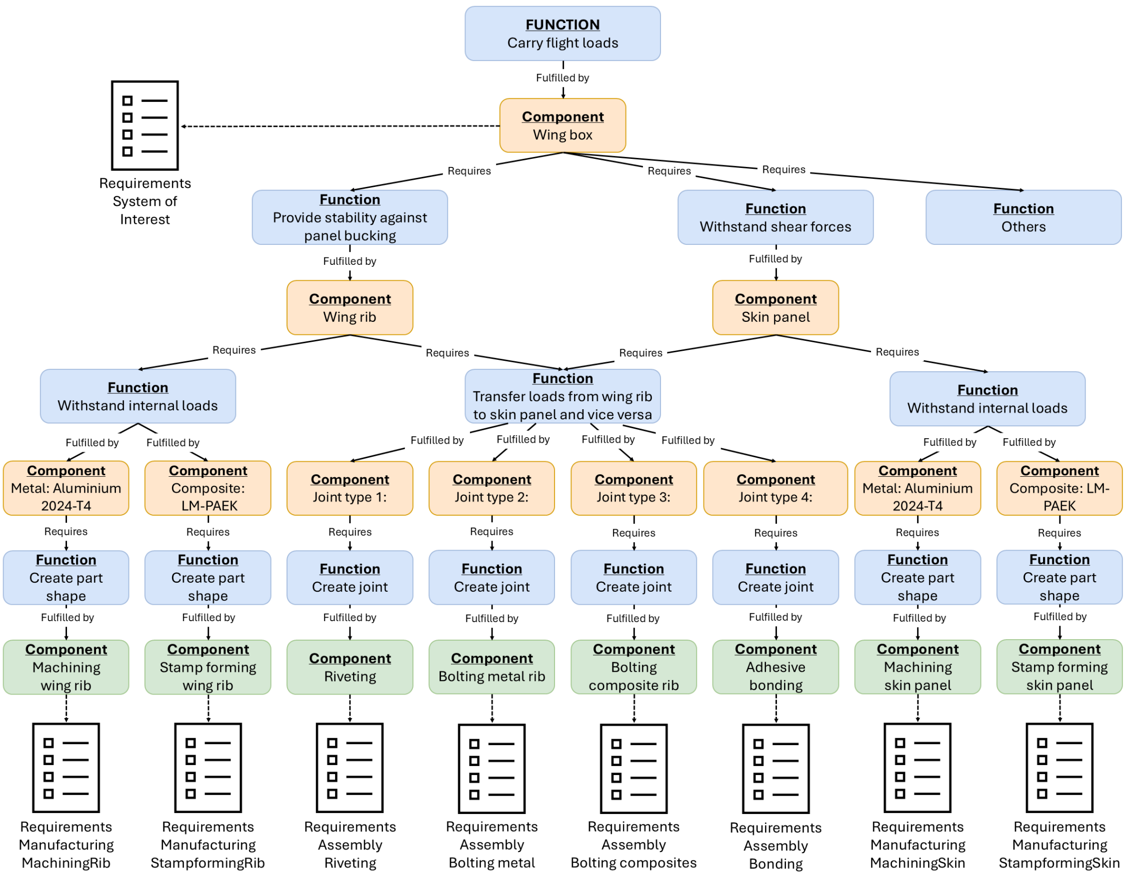


Figure 10. System Architectural Design Space model, including two part primitives, two manufacturing methods, and four methods for assembly.

1. Requirement inventarisation of System of Interest

At the SoI level, requirements are categorized into various types of requirements. In this research, *performance requirements* and *design requirements*[3] are used. Performance requirements (listed in Table 1) define how well a function, such as weight or loads, performs. Design requirements (listed in Table 2) specify physical parameters such as the width of the flange or the thickness of the rib [16, 17].

[3]Naming convention in accordance with Carson [16]

Table 1. Performance requirements of the use case, including problem roles (QoI, Objective and Constraint).

UID	System	Role	Description	Test Case Tool
RP-0001	Wingbox	Constraint	The wingbox must have a total weight of less than 70 kg.	GeometryTool
RP-0002	Wingbox	Objective	The wingbox must have a maximum cost of $10,000.	Total_cost
RP-0003	Rib	Constraint	The rib must have a critical buckling shear flow of at least 150 kN/m.	GeometryTool
RP-0004	Rib	Constraint	The rib must have a maximum cost of $5,000.	CATMAC_rib
RP-0005	Rib	Constraint	The rib must have a maximum weight of 20 kg.	GeometryTool
RP-0006	Skinpanel	Constraint	The skin panel must have a maximum cost of $5,000.	CATMAC_skin
RP-0007	Skinpanel	Constraint	The skin panel must have a maximum weight of 50 kg.	GeometryTool
RP-0008	Joint	Constraint	The joint must have a maximum cost of $500.	CATMAC_joint
RP-0009	Joint	Constraint	The joint must connect two materials that don't corrode.	ComplianceTool
RP-0010	Skinpanel	Constraint	The skin panel must have a critical buckling shear flow of at least 30 kN/m.	GeometryTool

Table 2. Design requirements used in the use case, including problem role Design Variable (Bound) (DV(B)), Where DV are discrete values and DVB are continues bounded values

UID	System	Role	Description	Test Case Tool
RD-0001	Rib	DVB	The rib must have a web thickness between 2 and 10 mm.	GeometryTool
RD-0002	Rib	DVB	The stiffener spacing must be between 0 and 300 mm.	GeometryTool
RD-0003	Rib	DV	The rib must have 1 to 4 holes.	GeometryTool
RD-0004	Rib	DVB	The hole in the rib must have a radius between 10 and 40 mm.	GeometryTool
RD-0005	Rib	DV	The rib material must be aluminum or composite.	GeometryTool
RD-0006	Skinpanel	DVB	The flange must have a width between 20 and 200 mm.	GeometryTool
RD-0007	Skinpanel	DVB	The skin panel must have a thickness between 2 and 10 mm.	GeometryTool
RD-0008	Skinpanel	DVB	The stringer spacing must be between 0 and 300 mm.	GeometryTool
RD-0009	Skinpanel	DV	The skin panel material must be aluminium or composites.	GeometryTool

2. Requirement inventarisation of Enabling Systems

ES requirements operate at a level lower than the SoI level. These ES are specified by the user in the RMM Excel tool as conditions as discussed in Section II.C. Three sets of condition are specified in the Excel tool, two for manufacturing and one for assembly. When a condition is met, the corresponding ES is activated, the condition and ES are shown in Table 3.

Table 3. Specified conditions of the use case in the RMM Excel. Each switch has multiple conditions existing of three parts, a design variable (red), a constraint (equal to, larger than) as entered in black, and a value (blue). Once a condition is true, the enabling system is activated in the workflow.

Switch	UID	Condition	Enabling system
Rib manufacturing	CR-0001	material_rib = aluminium	Machining system rib [9]
	CR-0002	material_rib = composite	Stampforming System Rib [9]
Skin panel manufacturing	CR-0003	material_skinpanel = aluminium	Machining system skin panel [9]
	CR-0004	material_skinpanel = composite	Stampforming System Skin panel [9]
Assembly method	CR-0005	material_rib = aluminium **and** material_skinpanel = aluminium	Riveting system Discussed in Section III.C.1
	CR-0006	material_rib = composite **and** material_skinpanel = composite	Adhesive bonding system Discussed in Section III.C.2
	CR-0007	material_rib = aluminium **and** material_skinpanel = composite	Bolting system aluminium rib Discussed in Section III.C.1
	CR-0008	material_rib = composites **and** material_skinpanel = aluminium	Bolting system composite rib Discussed in Section III.C.1

As shown in Table 3, each ES tool assesses manufacturability or assemblability depending on the selected materials. The manufacturing ES are integrated and used in the same manner as detailed in Nikitin [9]. The Assembly Method switch has four attached ES, one for each unique assembly method. These ES are discussed in more detail in the following subsection.

C. Integration of Geometry Depended Assembly requirements via Joint Assessment Methods

GDA requirements are integrated into the workflow through their corresponding ES. Each ES includes an JAM to evaluate the feasibility of the joint. Riveting (CR-0005) and bolting (CR-0007 and CR-0008) share a similar structure since they are both fastener-based. Their JAMs are discussed in Section III.C.1, while adhesive bonding is treated separately in Section III.C.2.

1. Fastener assessment tool

The fastener tool calculates (1) the number of fasteners required to ensure the joint is not a structural weak point and (2) the number of fasteners that can physically fit on the flange. These are used to assign an assembly grade between -1 and 1, where a grade ≥ 0 indicates feasibility. This is integrated within an optimization SWF, as shown in Fig. 11 where the design vector exists of the edge distance and the fastener spacing and the objective is to minimize the risk that occurs when assembling.

The optimization alters two design variables: edge_distance (e_f) and fastener spacing (S_f). Here, S_f is defined as the ratio between the center-to-center distance(s) of two fasteners divided over the diameter. e_f is the dimensionless distance between the center of the fastener and the edge divided over the diameter of the fastener, as shown in Fig. 12. Increasing e_f or S_f reduces the chance of complications (risk), since the margin of error is larger. The optimization subworkflow minimizes risk while maintaining a feasible assembly grade. Furthermore, it calculates the numbers of fasteners that are later used for the cost calculation.

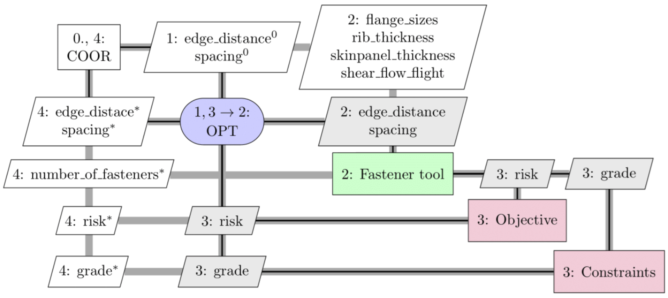


Figure 11. XDSM representation of the fastener JAM

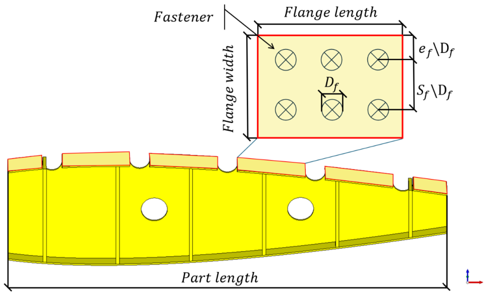


Figure 12. Flange layout

Table 4 lists the ES requirements. These requirements are listed in the Excel tool and used by the RMM to formalize the workflow.

Table 4. Product requirements of the fastener tool ES, the values for α, β, δ, and γ are given in Table 5

UID	System	Role	Description	Test Case Tool
RA-1100	Joint	Constraint	The joint grade must be above 0	Fastener tool
RA-1200	Joint	Objective	The assembly risk should be minimized	Fastener tool
RA-1300	Joint	DVB	The joint's edge distance must be between α and β	Fastener tool
RA-1400	Joint	DVB	The joint's fastener spacing must be between γ and δ	Fastener tool

Required number of fasteners

To prevent the joint from becoming the critical load case, the number of fasteners must ensure enough load transfer to outlast either the skin panel or rib. The required number of fasteners is calculated using Eq. (1), where τ_{max} is the maximum shear flow per unit length, which is provided by the GeometryTool as an input variable [9]. F_{max} is the maximum force transferred (equal to $\tau_{max} \cdot L_{joint}$). L_{joint} is the part length, as shown in Fig. 12. $F_{fastener}$ is the load that a single fastener can transfer, this equals the allowable shear flow $\tau_{allowable}$ multiplied by the area of a fastener A_f ($A_f = \frac{\pi D_f^2}{4}$).

$$n = \frac{F_{max}}{F_{fastener}} = \frac{\tau_{max} \cdot L_{joint}}{\tau_{allowable} \cdot A_f} = \frac{4 \cdot \tau_{max} \cdot L_{joint}}{\tau_{allowable} \cdot \pi D_f^2} \tag{1}$$

The fastener diameter D_f, is selected based on the maximum thickness of either the skin panel or wing rib [18, 19]. Using this data, Eq. (1) outputs the required number of fasteners.

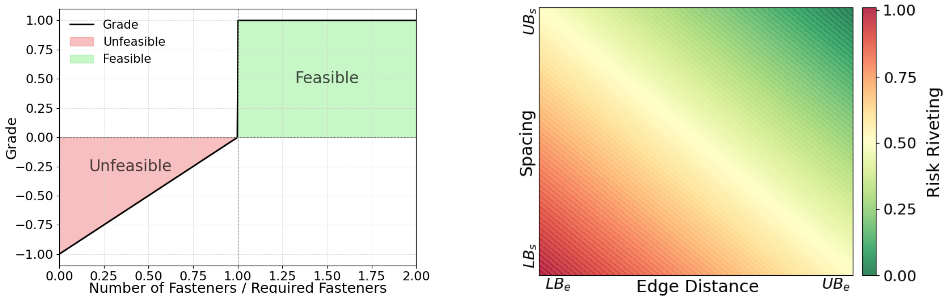
Actual number of fasteners placed

The number of fasteners that can be placed on the flange is based on the flange sizes. The total flange exists of multiple "flange segments" as shown in Fig. 12, these sizes are saved in a dictionary and used as input for the calculations of the number of fasteners. The number of fasteners placed on each individual flange segment is calculated using the edge_distance and fastener_spacing. Each of the values is added to find the actual number of fasteners placed on the total flange. The number of fasteners placed on a flange segment is calculated using Eq. (2):

$$N_{fasteners\_length} = \left|\frac{L_{flange} - 2e_f D_f}{S_f D_f}\right|, \qquad N_{fasteners\_width} = \left|\frac{W_{flange} - 2e_f D_f}{S_f D_f}\right|$$
$$N\_fasteners = N_{fasteners\_width} \cdot N_{fasteners\_length} \tag{2}$$

Grading and risk

The assembly grade depends on whether the flange can accommodate the required number of fasteners. The grade increases linearly from -1 (when no fasteners can fit) to 0. Stepping from 0 to 1 when the requirement is met, as shown in Fig. 13a. While additional fasteners still ensure feasibility, excess fasteners increase weight and cost, affecting other system requirements as shown in Table 1.



(a) Plot of the assemblability grade **(b)** Risk for fastener placement

Figure 13. Risk and grading of the fastener assembly method, both used for optimizing fastener placement

The risk, shown in Fig. 13b, evaluates the ease of fastener placement, based on the lower bound, upper bound, and value of the edge distance e_f and fastener spacing S_f [18, 19]. Risk is defined by Eq. (3).

$$Risk = \underbrace{\frac{e_f - lb_{e_f}}{2(ub_{e_f} - lb_{e_f})}}_{\text{Risk edge distance}}) + \underbrace{\frac{S_f - lb_{S_f}}{2(ub_{S_f} - lb_{S_f})}}_{\text{Risk spacing}} \tag{3}$$

A higher risk indicates lower margins for error, since the values of edge distance and fastener spacing are closer to their corresponding lower bound. This makes fastener installation more challenging. The optimizer minimizes this risk while ensuring feasibility, the risk is included as the objective as shown in Fig. 11 while the grade is included as a constraint since it should always be ≥ 0 to ensure assemblability.

Optimization algorithm fastener tool

For the fastener tool, a study has been conducted to choose the most appropriate optimization algorithm. The *NLPQL algorithm* was selected due to its ability to deliver reasonable results in relatively short time frames. NLPQL has a balance between performance and efficiency, a more in-depth motivation is given in Appendix C.

Tool variants: Riveting, Bolting aluminium rib, and Bolting composite rib

Three key differences distinguish bolting and riveting tools:

1) **Row limitation** Composite flanges (used in bolting with composite ribs) allow a maximum of two rows of bolts due to local stress concerns (as shown on the left in Fig. 14). Aluminium flanges (used in Riveting and Bolting with aluminium rib) allow more rows [19].

2) **Web position**: In aluminium ribs, the web is centered, restricting the fastener layout. In composite flanges, the web is at the flange's end, offering a larger area where fasteners can be placed. This can be seen in Figure 14, where the left illustrates a composite flange, as used by bolting composite rib, and the right shows an aluminium flange, used within riveting and bolting aluminium rib.
3) **Design bounds**: The limits of the edge distance and fastener spacing vary by method, as given Table 5

Table 5. Edge distance and fastener spacing of riveting and bolting tools

Assembly method	Edge distance		Fastener spacing	
	Lower bound (α)	Upper bound (β)	Lower bound (γ)	Upper bound (δ)
Bolting aluminium rib	3	6	3	8
Bolting composite rib	3	6	3	8
Riveting	1.5	4	4	6

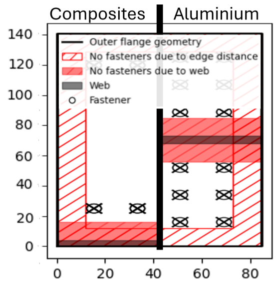


Figure 14. Flange segment of composites (left) and aluminium (right)

2. Adhesive bonding tool

The bonding tool developed in this research evaluates whether two composite components can be joined using adhesive bonding. For a structure where one side is fixed and the other side is free, such as the flange shown in Fig. 15, a specific bond width is required to ensure structural integrity. In such cases, the width of the bond (equal to the width of the flange, denoted as $\bar{w}$) should ideally be at least 30 times the thickness of the thickest plate in the joint. This thickness, t_{max}, is the greater of the skin panel thickness (t_{skin}) and the rib thickness (t_{rib}), as shown in Fig. 15. This ideal bond width is referred to as the target width, and a tolerance of ±10% is allowed around this value [19].

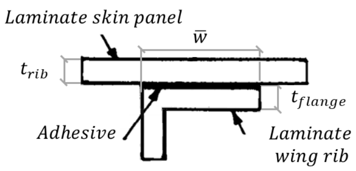


Figure 15. Overview of adhesive bond dimensions

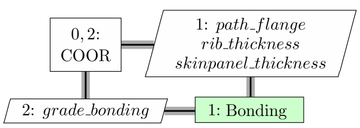


Figure 16. XDSM bonding tool

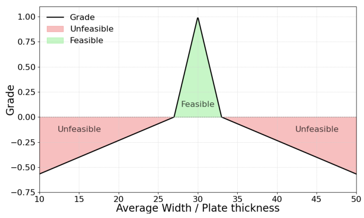


Figure 17. Grade plot bonding tool

As illustrated in the XDSM diagram in Fig. 16, the bonding tool uses the rib thickness and skin panel thicknesses as direct inputs. Furthermore, the flange_sizes dictionary is used to calculate the average flange width size ($\overline{w}$), enabling it to estimate the grade for adhesive bonding feasibility. The requirements applied in the RMM are outlined in Table 6. Since the bonding tool's grading relies solely on the geometry of the SoI, the bonding tool is implemented as a disciplinary tool within the workflow.

The only output of the bonding tool is the bond grade, as shown in Fig. 17. This is calculated by:

$$\text{Grade} = \begin{cases} 1 - \frac{|\overline{w} - 30 \cdot t_{\max}|}{0.1 \cdot 30 \cdot t_{\max}}, & \text{if } |\overline{w} - 30 \cdot t| \leq 10\% \text{ of } 30t_{\max} \\ -1 \cdot \left|\frac{\overline{w}}{30 \cdot t}\right| + 10\%, & \text{if } |\overline{w} - 30 \cdot t| > 10\% \text{ of } 30t_{\max} \end{cases} \tag{4}$$

Equation (4) indicates that the grade remains positive when the flange width is within 10% of the target width. If this condition is not met, the grade decreases linearly. The additional +0.1 ensures continuity in grading, as shown in Fig. 17.

Table 6. Product requirements of the bonding ES

UID	System	Role	Description	Test Case Tool
RA-1100	Joint	Constraint	The bonding grade must be equal to or above 0	bonding tool

D. MDAO workflow formulation

The use case in this research minimizes the cost of a wingbox. To fully define the workflow with this objective, a few remaining segments must be implemented. The GIA-requirements need to be integrated via a material compliance tool. Furthermore, a method to calculate the cost needs to be defined.

1. GIA Requirements evaluation tool: Material compliance

As elaborated in Section II.B.2, certain assembly requirements can be assessed before generating the geometric design. This study uses material compatibility as a proof of concept. According to C. Y. Niu [19], aluminium and composite materials should not be joined due to the risk of galvanic corrosion. Although real-world applications may have solutions to mitigate this risk, such as protective coatings, these solutions are beyond the scope of this research. The GIA tool checks for material compatibility and returns a grade of -1 when the materials are non-compliant (aluminium-composite combination) and 1 when the materials are compliant. This grade is included as a constraint in the workflow. This tool could be expanded in future research to include additional GIA requirement evaluations.

2. Cost Module

The total cost of a two-part primitive joint combination is determined using CATMAC (Cost Analysis Tool for Manufacturing of Aircraft Components) [20]. CATMAC calculates costs in three stages: first, it evaluates the total cost of both the skin panel and the rib, incorporating the labor of the manufacturing process, the layup labor (for composite parts), machine usage, and material costs. Secondly, it estimates the cost of the joint, including process labor, machine usage, and connection materials. And finally, these individual cost of each part primitive and of the joint are summed to calculate the total cost. By splitting out each part primitive and joint, the output can be used directly as a constraint, enhancing transparency.

The cost calculation of the part primitives is elaborated in Nikitin [9]. For the joint itself, CATMAC requires specific input parameters for cost estimation, such as the manufacturing method, adhesive / fastener type and the surface area, detailed in Appendix D. These values use empirical data to estimate costs, it is assumed that standard blind rivets are used for riveting and stainless steel hex bolts for bolting, since these do not corrode with the composite part [19].

E. Workflow implementation

To determine which MDAO formalizations are needed to address the research objective, it is necessary to analyze each research question discussed in Section I.A and define an execution strategy that answers the research questions. Research Question 1 (RQ-1) focuses on the development of tools that evaluate DfA considerations. This involves the creation of Joint Assessment Methods to assess GDA requirements and a material compatibility check to evaluate GIA requirements. RQ-2 explores how these GDA and GIA requirements are integrated and how the outcome is affected. To address this, the workflows with DfMA must be compared to a baseline workflow without DfMA, in order to evaluate their impact. RQ-3 requires a comparison between the static and dynamic workflow, to understand the effect on this research. To answer all three research questions, three workflows must be generated.

- Workflow 1: A static formalization based on the approach of Nikitin [9], extended with GIA and GDA requirements. This workflow assumes fixed material properties.
- Workflow 2: A dynamic workflow based on workflow 1, in which material selection is included as a design variables. This allows manufacturing and assembly considerations to be dynamically adapted according to material selection. Comparing this to workflow 1 answers RQ-3.
- Workflow 3: A dynamic workflow that changes DfMA considerations into QoI instead of constraints. Comparing this to workflow 2 evaluates the impact of the DfMA considerations, answering SQ-1 and SQ-2.

These workflows are detailed further in the following subsections.

1. [Wf1] workflow 1: Static workflow implementation

In the static workflow, the material of both the skin panel and the rib are pre-selected. Since each part primitive can be either aluminium or composite, four distinct static workflows exist, each representing one of the possible material combinations. The workflow shown in Fig. 18 shows the formalization for the case where both components are made of aluminium. A complete overview of the automatically generated XDSM by KADMOS and its implementation Optimus can be found in Appendix E.

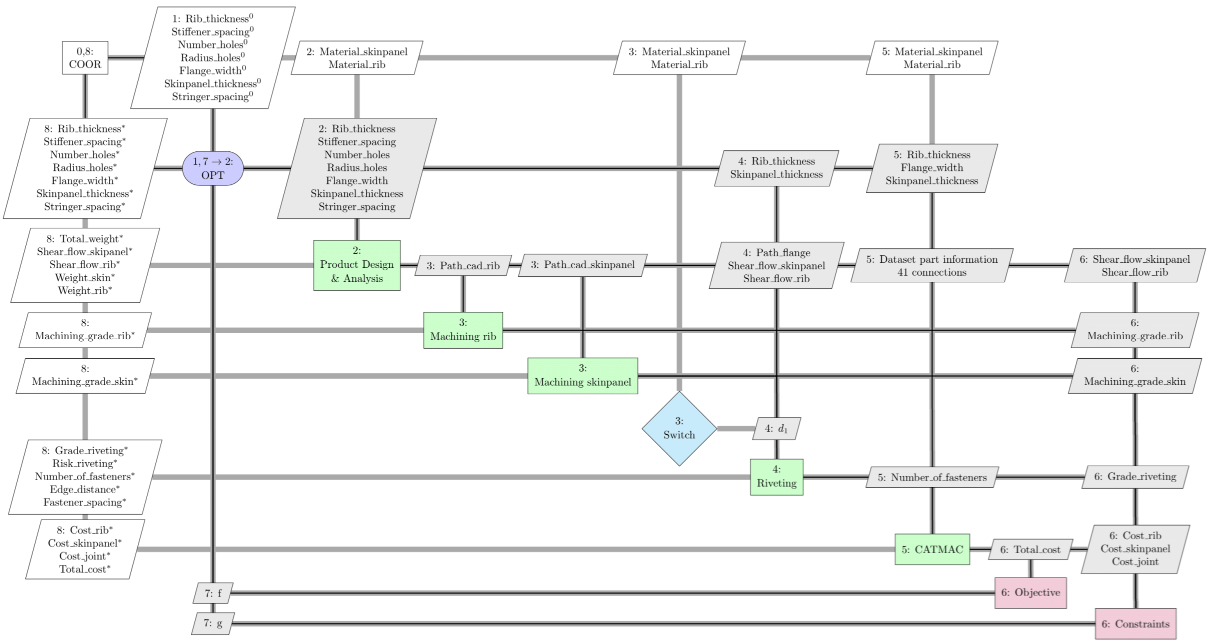


Figure 18. Workflow 1: XDSM formalization of the static optimization of a rib-skin panel combination made of aluminium. A full page version of this XDSM is shown in Appendix A

As shown in Fig. 18, the optimizer begins with the selection of the design vector, consisting of the variables specified in Section III.B. Next, the 2: Product Design & Analysis block generates the geometry of the skin panel and rib using ParaPy. Based on the generated geometry, both parts are evaluated for manufacturability, producing manufacturability grades that serve as constraints. At the same time, the JAM evaluates the joint feasibility using the riveting tool, which returns a risk, grade, and number of fasteners placed on the flange.

Once the manufacturability and assemblability assessments are completed, cost estimation is performed using CATMAC, using input from both the geometry model and the riveting tool. The resulting total cost is used as the objective function and is minimized using the NAVIRUN optimization algorithm provided by Optimus [15]. NAVIRUN is a random sample optimization method developed by NOESIS, it is a black box algorithm with its own termination criteria, meaning that the exact path to convergence is unknown [15].

One notable implementation detail is shown just before the riveting tool: the inclusion of a switch with a single, always-true condition. In the RMM excel tool, each ES must be connected via a condition. Unlike the manufacturing requirements, which can be directly integrated into the list of SoI requirements, the riveting tool represents a separate optimization workflow. As such, it cannot be embedded directly and must instead by triggered by a condition. To enable this, a switch is inserted to activate the SWF riveting. Although the switch does not affect the workflow, it is still required by the RMM to correctly formalize the workflow.

2. [Wf2] workflow 2: Dynamic workflow implementation

The dynamic MDAO workflow builds on wf1 by including material selection as a design variable. Figure 19 presents a summarized version, while the complete version generated by KADMOS is provided in Appendix F.

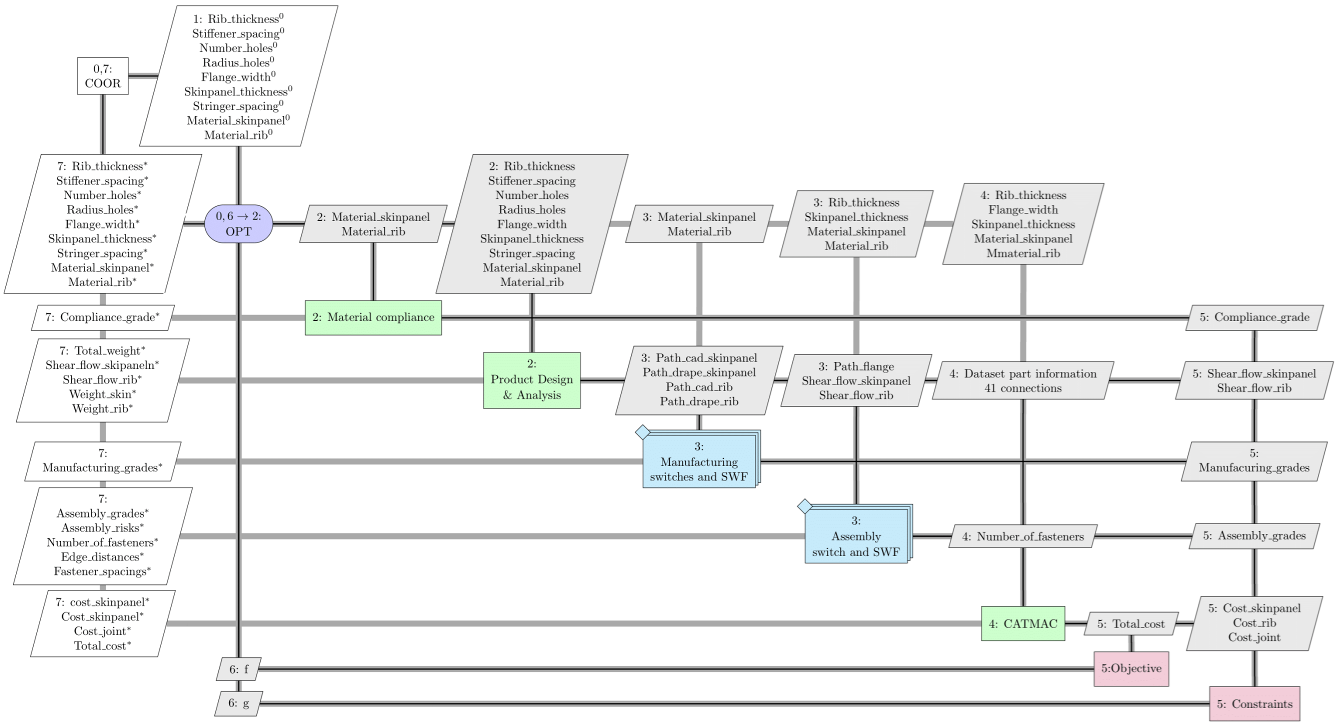


Figure 19. WF2 summarized XDSM formalization of the dynamic optimization of a rib-skin panel combination. The *Blue switch blocks* represents the switch system, including switches and corresponding branches. A full page version of this XDSM is shown in Appendix A

Although the structure remains similar to the static workflow, the manufacturing and assembly assessment tools must dynamically adapt depending on the design vector. Material compliance is checked within the optimization and is used as a constraint. After the Product Design & Analysis phase, manufacturability is assessed in the 3: Manufacturing switches and SWF block, which selects the appropriate manufacturing method and provides a grade. The same process applis to 4: Assembly switch and SWF block. Finally, CATMAC calculates the cost, which is used as the objective.

3. [Wf3] Workflow 3: Dynamic workflow without DfMA

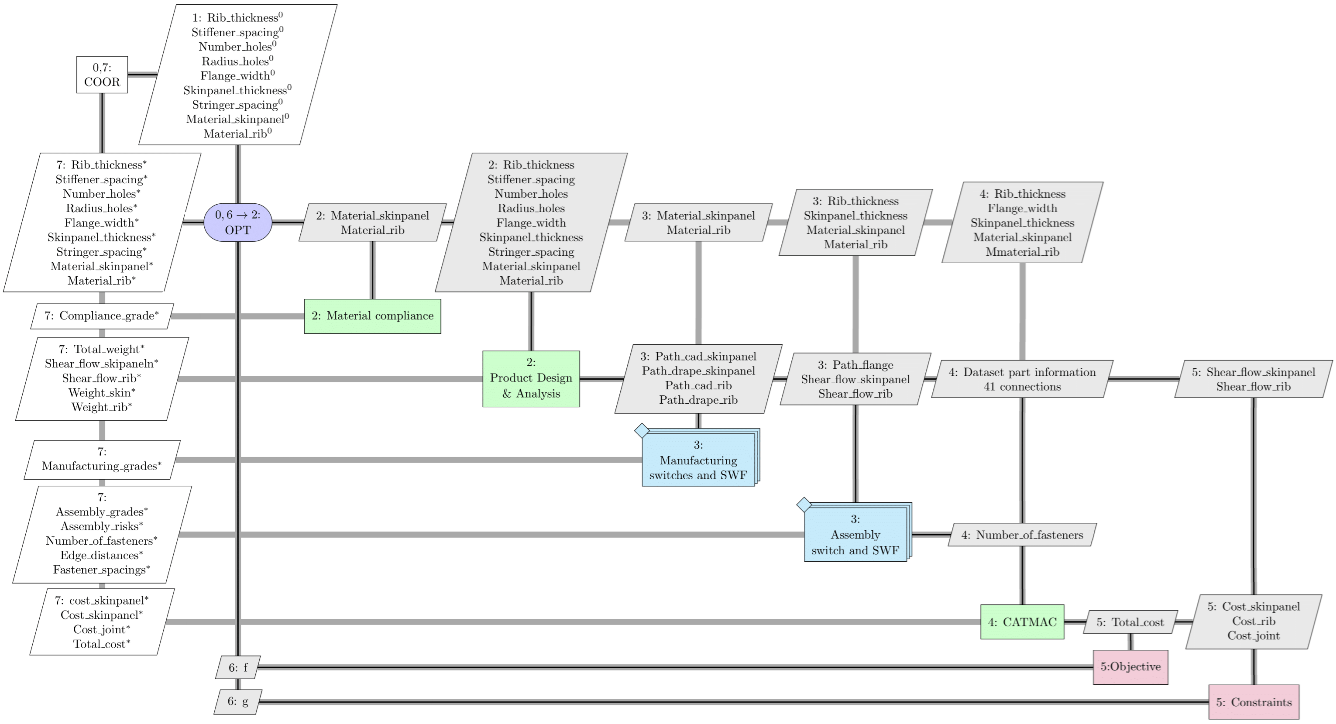


Figure 20. WF3 summarized XDSM formalization of the dynamic optimization where DfMA considerations are excluded as constraints while still evaluated. A full page version of this XDSM is shown in Appendix A

This workflow builds on [wf2] with the key difference that DfMA are considered QoI rather than constraints as shown in Fig. 20. While they are accounted for, they do not impact the optimizer, and thus do not change the physical design itself. The cost is still computed with the assembly and manufacturing data included.

Each workflow optimizes the cost of the rib-skin panel combination, assessing the following key performance indicators (KPIs): iteration count, runtime, cost, weight, and manufacturing/assembly grades.

IV. Results use case

During the use case, three different workflows were formalized: (wf1) Static MDAO formalization, (wf2) Dynamic MDAO formalization, and (wf3) Dynamic MDAO workflow without DfMA. Each of those formalizations are optimized for cost minimization. This section presents the results, starting with the relevant results of each workflow in Section IV.A. Section IV.B discusses the static workflows, followed by Section IV.C, which evaluates the dynamic workflow, including the optimal design and all KPIs, along with an analysis of JAMs. Next, Section IV.D compares static and dynamic workflows, and Section IV.E examines the effect of assembly inclusion.

A. Outcomes of test runs

As stated in Section III.E, four runs were performed, generating values for the KPIs. Table 7 provides a summary of the optimized design variables and the results of wf1, wf2, and wf3. The full list of variables is available in Appendix G. The data presented in Table 7 is used throughout the result section.

Table 7. Optimized design for both static executed workflows, dynamic workflow with and without DfMA. The full data is given in Appendix G

Run	Static workflows		Dynamic workflows	
Formalized workflow	Workflow 1 **Aluminium**	Workflow 1 **Composites**	Workflow 2 **With DfMA**	Workflow 3 **Without DfMA**
Optimized DV				
Rib thickness	6.7 mm	7.9 mm	6.7 mm	10 mm
Skin panel thickness	5.3 mm	6.0 mm	5.3 mm	5.3 mm
Flange width	65.8 mm	193.87 mm	65.8 mm	20 mm
Outputs				
Shear flow rib	150 kN/m	192.50 kN/m	150 kN/m	498.7 kN/m
Shear flow skin	30 kN/m	33 kN/m	30 kN/m	30 kN/m
Total weight	23.55 kg	20.43 kg	23.55 kg	21.39 kg
Assembly grade	1	0.20	1	-1
Assembly risk	0.83	-	0.83	0.65
Objective				
Total cost	€2968.29	€7498.95	€2968.29	€2017.77
Statistics				
Feasible iterations	90 (16%)	95 (17%)	82 (10%)	232 (33%)
Infeasible iterations	477 (84%)	463 (83%)	746 (90%)	462 (67%)
Total iterations	567	558	828	694
Estimated run time	27.5 hours	10.5 hours	39 hours	29 hours

B. Workflow 1: Static workflow results

The first workflows discussed are static workflows as introduced in Section III.E.1. In total, four workflows were considered, corresponding to different material selections: aluminium-aluminium, composite-composite, aluminium-composite, and composites-aluminium. However, the literature suggests that aluminium-composites and composite-aluminium combinations are infeasible due to galvanic corrosion, making optimization impossible, as the material compatibility constraint (Requirement RP-0009, Table 1) is always violated. Consequently, only two static runs are performed in which the aluminium-aluminium and composite-composite are evaluated. This results in two separately optimized wingbox designs, each optimized for cost.

Optimized aluminium wingbox design

The optimized aluminium wingbox design is cost-driven, resulting in a total estimated assembly cost of approximated €3000 (wingrib ≈ €1650, skin panel ≈ €1150, and joint ≈ €200). The optimization process reached convergence after 567 iterations (27.5 hours), with about 16% of the design iterations considered feasible. to reach convergence, with 16% of iterations deemed feasible. The final optimized design parameters are summarized in Table 7 with all KPIs given in Appendix G.

In this design, the thickness of the skin panel is 5.3 mm and of the wing rib is 6.7 mm, these values are lowered by the optimizer since this both lowers the manufacturing cost and assembly cost. Reduced rib thicknesses lead to narrower flange dimensions. A thinner structure proportionally lowers the diameter of the fastener, which increases the number of fasteners that can be positioned along the flange. The optimizer thus reduces both the rib and skin panel thickness to the minimum values allowable by shear flow constraints, as thickness is directly proportional to shear flow capacity [9].

Since machining costs are estimated based on the bounding box volume of the part, minimizing flange width significantly reduces the rib's manufacturing cost. The optimized flange width settles at 65.8 mm, narrow enough to reduce cost while accommodating the required fasteners and maintaining structural integrity. The flange houses 50 fasteners, although only 27 are structurally required to carry the expected shear load. These fasteners are symmetrically distributed in two rows, as described later in Fig. 24a. Further reduction in flange width would prevent placing the necessary number of fasteners, thereby violating design constraints.

Optimized composite wingrib design

The static optimization of the composite wingrib results in an optimized assembly cost of ≈€7500, where the wingrib is ≈€3250, the skin panel is ≈€4200, and the joining cost is≈ €50. The optimization of the composite wingbox converged after 558 iterations, taking a total of 10.5 hours. This duration is significantly faster than the optimization of the aluminium assembly. This is caused by the bonding tool that takes ≈ 5 seconds compared to the ≈ 120 seconds required for the riveting JAM, these times are elaborated more in Appendix I. In this static optimization, 17% of the iterations were deemed feasible, with the majority filtered out due to a negative assemblability grade (78% of joints had a negative bonding grade).

The thickness of the composite rib is 7.9 mm and the skin panel thickness is 6.0 mm, both minimized to reduce the shear flows. The total weight is 20.5 kg, slightly lower than the aluminium assembly. The skin panel weight is 45% lower compared to the aluminium counterpart, as expected. However, the rib is slightly heavier due to the adhesive bonding method that require a larger flange, this increases the rib's volume by 40%, hence counteracting the composites weight advantage.

C. Workflow 2: Dynamic workflows

The dynamic workflow produces the same optimized design as the static aluminium-aluminium workflow. This result is expected, as the MDAO problem is solved using identical objective, requirements, and initial design point.

1. Optimized design

The optimization required 828 iterations to converge to the optimal design, as shown in Fig. 13. The design complies with all predefined requirements, as detailed in the compliance report (Appendix H).

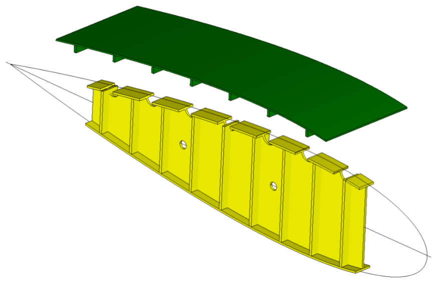


Figure 21. Optimized design of the wingbox section.

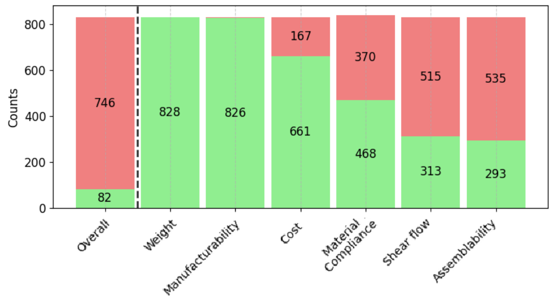


Figure 22. Feasibility of individual constraints, showing the number of designs assessed (counts)

In this workflow, six criteria, described in Section III.B and illustrated in Fig. 22, must all be met for a design iteration to be considered feasible. This occurs in only 10% of the design iterations (counts in Fig. 22). Constraints related to rib and skin panel weight, as defined in the performance requirements (RP-0005 and RP-0007 in Table 1), as well as manufacturability (CR-0001 through CR-0004), are almost always satisfied. These findings align with Nikitin [9], which also discussed the limited impact of manufacturability in a similar use case. The cost feasibility requirements (RP-0004, RP-0006, and RP-0008) are met in 80% of the cases, although high-thickness composite designs do exceed the cost requirements. Material compatibility (RP-0009) is achieved in 55% of the designs. For structural performance, only 40% of the iterations meet the shear flow criteria for the rib and the skin panel (RP-0003 and RP-0010). The assemblability shows the lowest compliance, with only 35% of the configuration passing the joint feasibility check of the JAM (CR-0005 through CR-0008), indicating that most of the proposed design iterations are not practically viable.

Although material compatibility is evaluated early, the rest of the workflow (manufacturability, assemblability, and cost) is still fully evaluated, resulting in increased computational overhead. 45% of the total optimization time is spent evaluating designs that do not meet the GIA requirements.

Outcomes Joint Assessment Methods

During the execution of the dynamic workflow, 828 iterations were evaluated, each assessed for assemblability using one of four assembly methods, as illustrated in Fig. 23. It is important to note that a design marked as "feasible" for assembly may still be infeasible overall due to failure in criteria such as material compatibility. Both bolting methods were never feasible in this workflow due to the consistent material incompatibility, however, this does not imply that bolting itself is inherently infeasible.

Riveting was the method most frequently selected, applied in 35% of the iterations. The other three methods were selected roughly equally (≈ 22% of the cases), reflecting the preference of the optimizer for riveting. In 60% of the riveting cases, the edge distance and the fastener spacing were modified to accommodate a sufficient number of rivets. In the remaining cases, the flange width was too narrow to place enough fasteners.

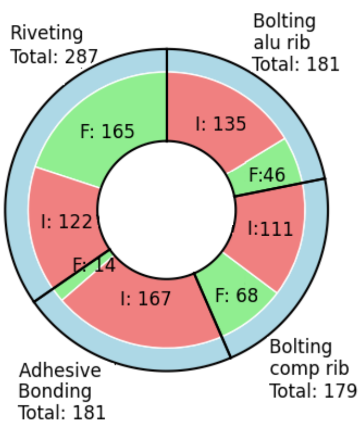


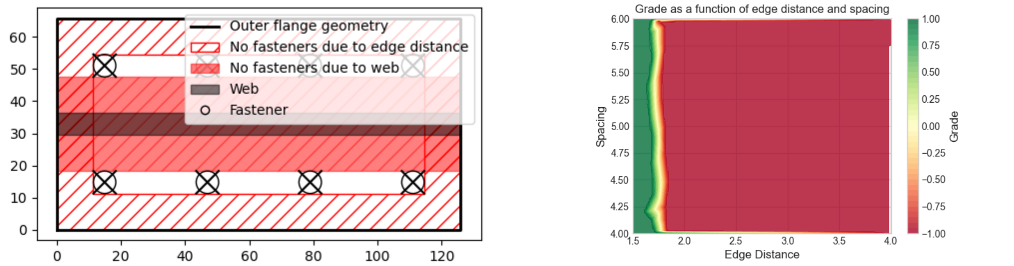
Figure 23. Feasibility of assembly, including number of feasible (F) and infeasible (I) joints

Bolting showed a lower success rate rate, of which bolting with an aluminium rib was less successful (25% succession rate compared to 40% for bolting with a composite rib). This lower feasibility is due to the fact that the web is placed in the middle for the aluminium ribs, leaving less room for bolts. Bonding showed the lowest feasibility rate, at just 8%. This is caused by the strict requirements as discussed in Section III.C.2.

Both the fastening tool and bonding tools are discussed below.

Fastening tool

The fastener tool calculates the number of fasteners that can be placed on a structure, as described in Section III.C.1. The optimized structure consists of eight individual flange pieces, as shown in Fig. 21 which in total houses 50 rivets. The tool optimizes the edge distance and spacing so that enough fasteners can be placed on the flange. This optimization takes on average $\approx$ 120 seconds.



(a) Flange layout of optimized design, where the flange segment houses 8 fasteners.

(b) Influence of edge distance and spacing on the fastener grade

Figure 24. data Analysis of the Joint Assessment Method

Figure 24a illustrates the optimized aluminium wingrib flange, which accommodates eight rivets with an edge distance of 1.6, fastener spacing of 4.6 and a total risk of 0.83. As shown, the rivets are placed precisely so that as little space as possible is wasted. Figure 24b shows the flange grade as a function of the edge distance and fastener spacing. As shown, the edge distance significantly influences the grade, whereas the spacing has minimal impact.

Bonding tool

The bonding tool has a success rate of just 8%, mainly due to strict flange requirements, with an allowable offset of 10% from the optimal width. This means that both a flange which is too wide and too narrow can be negatively graded. The average duration of the bonding tool is 5 seconds to calculate the grade.

Evaluation manufacturing tools

The manufacturing tools used in this study are evaluated in Nikitin [9]. Within this use case, both the skin panel and wing rib designs are assessed for manufacturability. All skin panel designs are assessed to be manufacturable. However, two of the wing rib designs are negatively assessed. One example of such an infeasible configuration is shown in Fig. 25, where a lightning hole (red outline) overlaps the region between two stiffeners. The manufacturing tool attempts to remove the material between the lightning hole and the stiffener, but the remaining distance is so minimal that it results in a negative manufacturability grade.

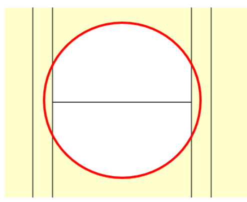


Figure 25. Exp 9, grade = -0.96

D. Static VS Dynamic

To address the third research question from Section I.A, the dynamic and static workflows are compared based on various KPIs such as computation time and iteration count. Both static workflows minimize thickness while satisfying strength requirements. The dynamic workflow converges to the same design as the static aluminium workflow, which is the most cost-optimal of the two static optimal designs. The number of iterations and corresponding time estimates are provided in Table 8

Table 8. Dynamic and static workflows comparison

	Workflow 2 Dynamic	Workflow 1 Static Aluminium	Workflow 1 Static composite	Workflows combined
Number of iterations:	828	567	558	1302
Estimated time:	39 hours	27.5 hours	10.5 hours	38 hours
Average time per iteration:	170 seconds	175 seconds	67 seconds	105 seconds
Estimated time difference:		-30%	-73%	-3%

The total run time of the dynamic workflow equals 39 hours, this is 3% longer compared to 38 hours the static workflow stook as shown in Table 8. This increase in time is not in line with previous research [9] but is caused by two reasons. 1.): On average, each iteration in the dynamic workflow takes 20 seconds longer than in the static workflows due to the additional material compatibility check (+5 seconds) and increased data processing from added switches. 2.): In this use case, the dynamic workflow spends over 350 iterations (18 hours) evaluating aluminium-composite and composite-aluminium configurations, which are not considered in the static workflows. These unnecessary evaluations of infeasible configurations add to the computational time, making it in general less effective.

The fastener tools require on average 120 seconds per iteration (175 seconds for a single iteration), while the bonding tool takes approximately 5 seconds (65 seconds for a single iteration). The average time is elaborated in more detail in Appendix I.

E. Impact of DfMA inclusion on dynamic workflows

Workflows Wf2 and wf3 are compared to evaluate the effect of including DfMA considerations as constraints during optimization. In Wf2, DfMA considerations are used as constraints within the optimization process, In Wf3, DfMA aspects are taken into account as QoI, and thus do not influence the design. This reflects the scenario introduced in the introduction, where additional design iterations are required after optimization to ensure manufacturability and assemblability.

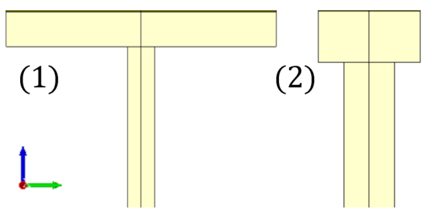


Figure 26. front view of the optimized rib flange with DfMA (left) and without DfMA (right)

In practice, flange width must be sufficient to allow for joining. When assembly constraints are excluded, the design tends to minimize flange width to reduce weight and cost, as one of the main functions of the flange is to house the joint. This is illustrated in Fig. 26: The left design includes DfMA constraints, while the right included these as QoI. The rib without DfMA reduced cost by over 50%, since the required material is based on the bounding box of the rib. To minimize machining waste, rib thickness is maximized. Thinner ribs require more material removal, increasing the process cost as evaluated by CATMAC.

With DfMA constraints (WF2), flange width scales with rib thickness because fastener size increases accordingly (as elaborated in Section IV.C). This results in a smaller bounding box and lower material costs, making thinner ribs more advantageous. The rib without DfMA constraints cost €800, compared to €1650 with DfMA. The skin panel remains largely unaffected. In addition, smaller flanges reduce joining costs. The total cost of the optimized design with DfMA is €2950, whereas the cost without DfMA constraints is €2000, a 47.5% reduction.

While excluding manufacturability constraints has minimal impact, including assembly considerations is critical. The assembly grade of the optimized design without DfMA is negative (-1), indicating that the design can not be physically assembled.

V. Discussion

The findings in Section IV demonstrate the impact of incorporating Design for Assembly considerations into MDAO workflows. In a dynamic workflow that optimizes the cost of a wing rib-skin panel assembly, only 10% of proposed design iterations were found to be feasible. The optimization process evaluated six design criteria, all of which had to be satisfied for a design iteration to be deemed feasible. Convergence was reached in all three workflows using the NAVIRUN algorithm. This algorithm, created by Optimus itself is a black-box optimizer that relies on random sampling of the design variables. Optimus has its own build-in termination criteria based on the . show the impact of DfA incorporation into MDAO workflows. Analyzing a dynamic workflow that optimizes the cost of a rib-skin panel combination showed that only 10% of proposed design iterations were feasible. This optimization was checked for six different design criteria which all had to be met for the design iteration to be deemed feasible. In all three workflows, the stopping criteria (for convergence) defined by Optimus was reached. The NAVIRUN algorithm was used, this black-box algorithm uses random sampling of the design variables to find the most optimal solution.

Among the six evaluated criteria, *assembly feasibility* and *part strength* were the main limiting factors. Specifically, just under half of the proposed design iterations did not meet the material compatibility requirement (GIA), while 65% of these designs could not be joined via the chosen assembly method (GDA). Among GDA failures, riveting was the most optimal assembly method within the dynamic workflow. While most (60%) of aluminium-aluminium combinations were feasible, composite part assembly had lower succession rates due to stricter requirements and the need for a wider flange.

The optimal design generated by the dynamic optimizer without DfMA considerations was almost 50% cheaper compared to dynamic workflow with DfMA considerations. However, while both parts could be manufactured individually, they could not be assembled, this is caused by the fact that the flange is added as a design variable. In reality the flange often is pre-selected as the function is to house the fasteners or adhesive of the joint. In this case, where the flange is a design variable, the optimizer minimizes the flange width as its proportional to the cost, giving an unfair comparison. Incorporating joints into MDAO workflows introduces additional computational overhead, depending on the required tools. Bonding adds approx. 5 seconds per iteration, whereas fastener tools, which involve sub-optimization, increase iteration time by 120 seconds.

Both the aluminium static and dynamic workflows converge to the same optimized design. However, a dynamic workflow required 3% more computation time compared to multiple combined static workflows. This is due to the fact that the dynamic workflow evaluates all potential configurations, while static only evaluate configurations which are material compatible. This results in an additional 18 hours spent in the dynamic workflow evaluating infeasible designs. In addition, the additional material compatibility check within the dynamic workflow and the multiple added switches, which add $\approx$ 20 seconds per iteration.

VI. Conclusion

In the early design stages, the integration of manufacturing and assembly considerations significantly influences the design process. Neglecting these aspects may lead to a cheaper optimized design, but the design would require alterations to ensure producibility.

This paper presented a new methodology for integrating assembly considerations within MDAO workflows. The integration of assembly considerations expands the capabilities of the DEE, a framework which translates requirements into MDAO workflows. By incorporating assembly considerations alongside manufacturing considerations in the DEE, designs can consist of multiple part primitives, and joint feasibility can be assessed. This methodology successfully meets the research objective. The methodology was validated using a wing rib-skin panel assembly as a case study, addressing the following research questions:

(RQ-1) What assembly assessment methods can be developed to assess the feasibility of assembly/joining of parts?

To address this, assembly requirements were split based on when they are evaluated in the workflow: GIA requirements are imposed before the geometry creation, and GDA requirements are evaluated afterwards using geometric data. Joint Assembly Methods (JAM) are used to evaluate GDA requirements, these methods were developed for mechanical fastening (riveting and bolting) and adhesive bonding. The integration of these tools is done in such a manner that new assembly methods can be added. Each JAM assigns a grade to the joint between -1 and 1, where a negative grades indicates an unfeasible joint. While the bonding tool performs a calculation, the fastener tool is an sub-workflow containing an optimization. This optimization changes the edge distance and fastener spacing to minimize the risk, increasing ease of assembly.

GIA requirements were implemented using a separate tool, within this tool, various types of geometry independent assembly requirements can be evaluated. Currently, material compatibility is used as a proof of concept, evaluating if galvanic corrosion occurs. The GIA tool is implemented as a constraint, meaning that designs that are negatively assessed by the GIA tool do continue through the workflow, adding extra computational cost.

(RQ-2) Would the incorporation of the **assembly assessment methods** and the **manufacturing assessment methods** result into challenges within the design and engineering engine?

During the integration of JAMs, challenges arose due to the *multi-conditional switch* definition in the RMM, as they could not be automatically generated from Excel. To address this, the RMM was updated to be able to identify and generate multi-conditional switches, this allows users to specify more complex conditions in the RMM Excel tool. Furthermore, within the Excel tool, each enabling system must be connected to a condition in order to be activated. The fastening tools (riveting and bolting) are an entire optimization by itself and must therefore be specified as an enabling system. For static workflows, this means that there is an integrated switch which always evaluates true.

The GIA requirements are integrated as constraints, this means that all tools will always be evaluated, independent on the GIA tools outcomes. One of the main advantages of the distinguishment based on "when" requirements are evaluated was that GIA requirements could potentially be used to break-off the workflow before the geometry was created. This lowers computational overhead. This is not currently possible within the DEE as a hard-stop mechanism is not supported. The last research question discusses the dynamic and static workflows

(RQ-3) What are the advantages and disadvantages of a dynamic MDAO workflow compared to a static one when manufacturing and assembly considerations are included?

The dynamic and static workflows were compared in the use case. The result shows that dynamic workflows require 3% more computational time than combined static workflows. This is a result of the fact that a large portion of iterations evaluated infeasible designs that did not meet material compatibility, which were not considered by the static workflows. Furthermore the dynamic workflows has the material incompatibility tool included and switches which increases computational overhead. As stated before, GIA requirements could be used to assess unfeasible joints before geometrical creation, this could potentially lower the time of the dynamic workflows, but this yet have to be researched and is therefore recommended.

VII. Recommendations

1. Research extra capabilities of the Excel tool as the RMM GUI
The RMM is a user-friendly tool that allows users to specify an MDAO problem within Excel, which KADMOS then transforms into an Optimus-compatible workflow. During this research, the switch mechanism, defined in the Excel tool using conditions, was expanded to accommodate multiple conditions. While this increases functionality, it also makes the Excel tool more complex and reduces its transparency. It is recommended to evaluate the Excel tool's scalability as the RMM increases in complexity. If additional capabilities are required, it should be assessed whether these can be effectively implemented within the Excel tool or if an alternative tool should replace Excel as the primary interface.

2. Expand complexity and realism
This research utilizes a simplified use case that involves a rib-skin panel assembly, evaluated using manufacturing and assembly tools. However, both geometric representation and assessment methods lack accuracy. The realism of the manufacturing and assembly assessment tools should be enhanced. To further improve realism, it is recommended to increase the number of part primitives to achieve a more accurate representation.

3. Enable workflows to break off infeasible iterations earlier
A part of this research investigated the possibility of evaluating GIA requirements early in the workflow to filter out infeasible designs before unnecessary computations were performed. However, this system could not be fully implemented. It is recommended to explore methods for terminating design iterations early to reduce computational time. An approach is to implement a "hard-stop" mechanism within a disciplinary tool, allowing it to return directly to the optimizer when a constraint is violated. Another option is to introduce an additional level within the workflow to prevent infeasible designs from proceeding to full evaluation (as shown in wf4 in this research). Additionally, the impact of such a mechanism on the optimization process should be further investigated.

4. Incorporate one single source of truth / database
Currently, data such as material properties and method-specific parameters are retrieved from various sources in the workflow. As the scope of a project expands, this approach can lead to inefficiencies and reduced transparency. It is recommended to explore the feasibility of maintaining a centralized database where all relevant data are stored for the MDAO project. This database should also include material properties, manufacturing method data, and assembly method data. One possible solution is the Manufacturing Information Model developed by the TU Delft [21], which could serve as a structured data repository for MDAO projects.

References

[1] Dostaler, I., "Avoiding rework in product design: evidence from the aerospace industry," *International journal Quality and Reliability Management*, Vol. 27, No. 1, 2010, pp. 5–26. doi: 10.1108/02656711011009281, URL `https://www.emerald.com/insight/content/doi/10.1108/02656711011009281/full/html`.

[2] Ciampa, P. D., and Nagel, B., "AGILE Paradigm: The next generation collaborative MDO for the development of aeronautical systems," *Progress in Aerospace Sciences*, Vol. 119, 2020. doi: 10.1016/j.paerosci.2020.100643, URL `https://linkinghub.elsevier.com/retrieve/pii/S0376042120300555`.

[3] Shahpar, S., "Challenges to overcome for routine usage of automatic optimisation in the propulsion industry," *Aeronautical Journal*, Vol. 115, 2011, pp. 615–625. doi: 10.1017/S0001924000006308.

[4] Sobieszczanski-Sobieski, J., Morris, A., Tooren, M. v., Rocca, G. L., and Yao, W., 2015, pp. 1–378. doi: 10.1002/9781118897072.

[5] Haymaker, J., and Lee Flager, F., "A comparison of Multidisciplinary Design, Analysis and Optimization Processes in the Building Construction and Aerospace Industries," *International Conference on Information Technology in Construction*, Vol. 84, 2007.

[6] van Gent, I., Aigner, B., Beijer, B., Jepsen, J., and La Rocca, G., "Knowledge architecture supporting the next generation of MDO in the AGILE paradigm," *Progress in Aerospace Sciences*, Vol. 119, 2020, p. 100642. doi: https://doi.org/10.1016/j.paerosci.2020.100642.

[7] Van Der Laan, T., and Hootsmans, L., "Consequences of Multi or Single disciplinary optimization results on aircraft component cost and weight," *2018 Aviation Technology, Integration, and Operations Conference*, 2018. doi: 10.2514/6.2018-3026, URL `https://arc.aiaa.org/doi/10.2514/6.2018-3026`.

[8] Bruggeman, A.-L., Nikitin, M., La Rocca, G., and Bergsma, O., "Model-Based Approach for the Simultaneous Design of Airframe Components and their Production Process Using Dynamic MDAO Workflows," *AIAA SCITECH 2024 Forum*, 2023. doi: 10.2514/6.2024-1530.

[9] Nikitin, M., "Dynamic MDAO workflows to Enable Design for Manufacturing of Aircraft Structural Components," Msc thesis, Delft University of Technology, 2024. URL `http://resolver.tudelft.nl/uuid:e35cdf66-8921-42e3-a84e-b3976ab42d8d`.

[10] van den Berg, T., and van der Laan, T., "A multidisciplinary modeling system for structural design applied to aircraft moveables," 2021. doi: 10.2514/6.2021-3079, URL `https://arc.aiaa.org/doi/abs/10.2514/6.2021-3079`.

[11] Bruggeman, A.-L., and La Rocca, G., "From Requirements to Product: an MBSE Approach for the Digitalization of the Aircraft Design Process," *INCOSE International Symposium*, Vol. 33, No. 1, 2023, pp. 1688–1703. doi: 10.1002/iis2.13107, URL `https://incose.onlinelibrary.wiley.com/doi/10.1002/iis2.13107`.

[12] Pernstål, J., Magazinius, A., and Gorschek, T., "A study investigating challenges in the interface between product development and manufacturing in the development of software-intensive automotive systems," *International Journal of Software Engineering and Knowledge Engineering*, Vol. 22, 2013. doi: 10.1142/S0218194012500271.

[13] Bruggeman, A.-L., Van Manen, B., Van Der Laan, T., Van Den Berg, T., and La Rocca, G., "An MBSE-Based Requirement Verification Framework to support the MDAO process," 2022. doi: 10.2514/6.2022-3722, URL `https://arc.aiaa.org/doi/10.2514/6.2022-3722`.

[14] "ISO/IEC/IEEE International Standard - Systems and software engineering–Vocabulary," *ISO/IEC/IEEE 24765:2017(E)*, 2017, pp. 1–541. doi: 10.1109/IEEESTD.2017.8016712.

[15] *Optimus 2021.1 User manual*, Noesis Solutions, 3rd ed., October 2021. Available at `https://example.com/manual.pdf`.

[16] Carson, R. S., "Implementing Structured Requirements to Improve Requirements Quality," *INCOSE International Symposium*, Vol. 25, No. 1, 2015, pp. 54–67. doi: https://doi.org/10.1002/j.2334-5837.2015.00048.x, URL `https://incose.onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2015.00048.x`.

[17] Boggero, L., Ciampa, P. D., and Nagel, B., "An MBSE Architectural Framework for the Agile Definition of System Stakeholders, Needs and Requirements," 2021. doi: 10.2514/6.2021-3076, URL `https://arc.aiaa.org/doi/abs/10.2514/6.2021-3076`.

[18] C. Y. Niu, M., *Airframe Stress Analysis and Sizing*, 3rd ed., Hong Kong Conmilit Press LT., Hong Kong, 2016.

[19] C. Y. Niu, M., *Composite Airframe Structures*, 3rd ed., Hong Kong Conmilit Press LT., Hong Kong, 1996.

[20] Van Der Laan, T., and Van Den Berg, T., "An open source part cost estimation tool for MDO purposes," *AIAA AVIATION 2021 FORUM*, 2021. doi: 10.2514/6.2021-3058, URL `https://arc.aiaa.org/doi/10.2514/6.2021-3058`.

[21] Bansal, D., "Automating Inclusion of Production Considerations in the Conceptual Design of Aircraft Structures," Msc thesis, Delft, universtiy of technology, 2022.

A. Full page versions of XDSMs

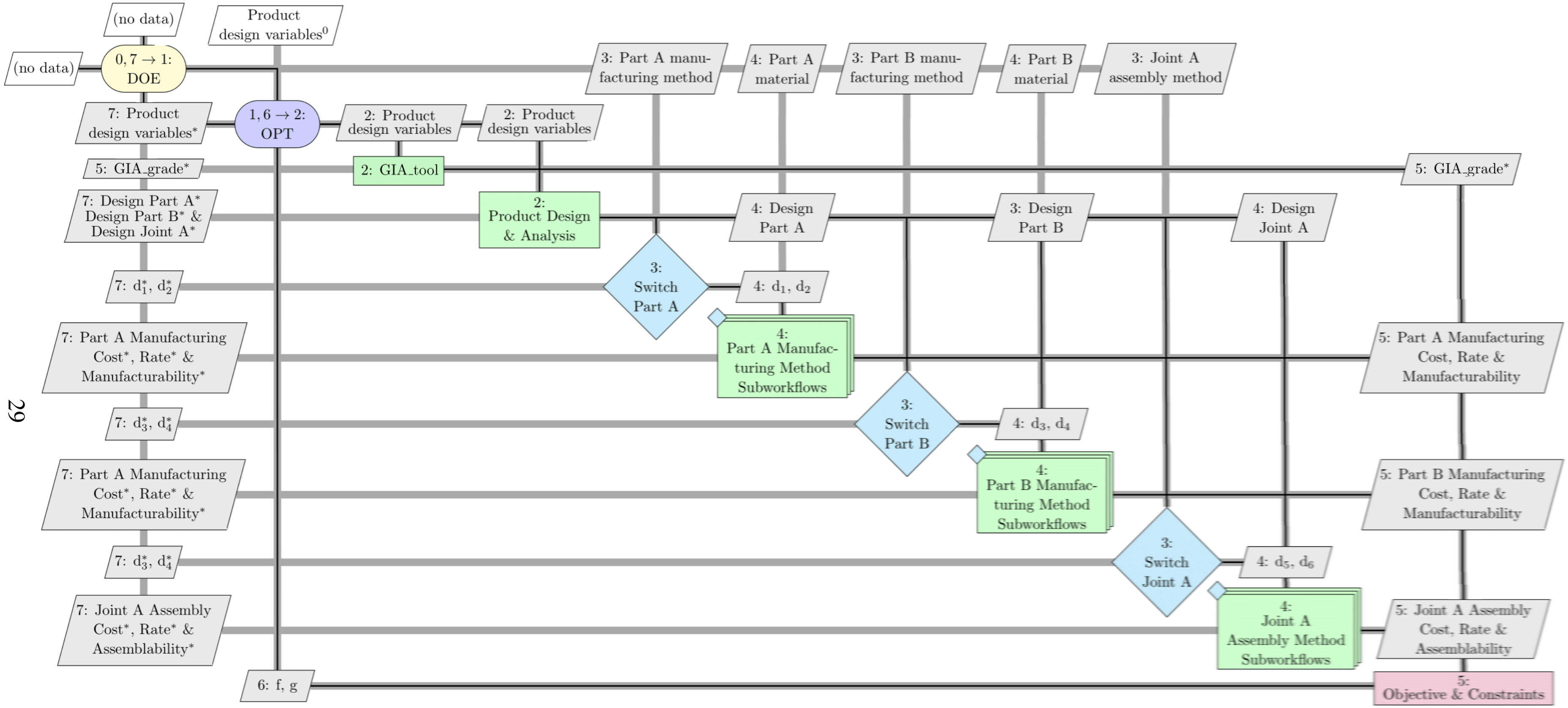


Figure 6 XDSM formalization of dynamic workflow for the optimization of a two-part primitive design. Both part primitives are tested for manufacturing and assembly considerations. After each switch, a stacked green block with a small blue diamond is illustrated, this indicates multiple possible branches [8].

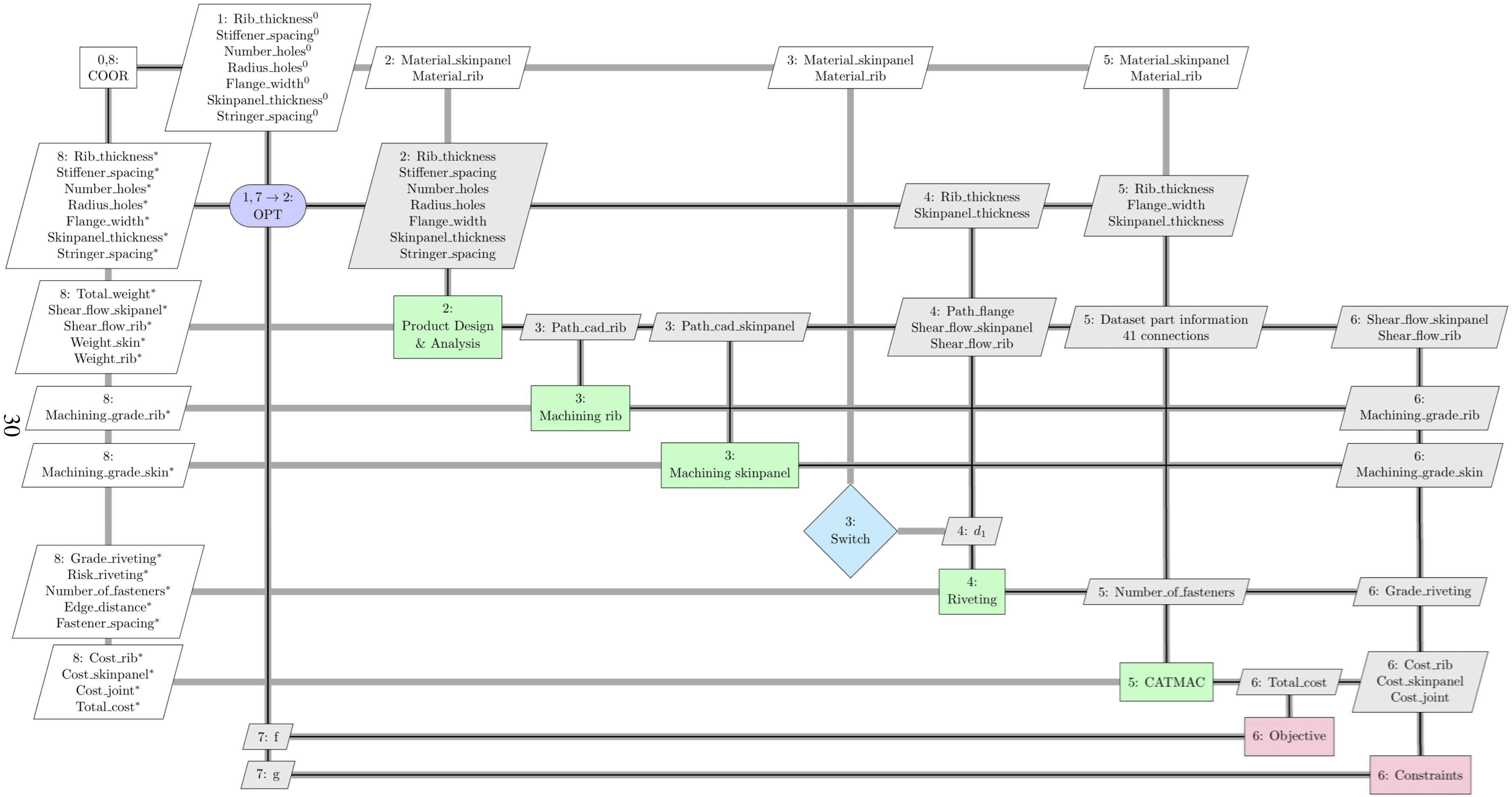


Figure 18 Workflow 1: XDSM formalization of the static optimization of a rib-skin panel combination made of aluminium.

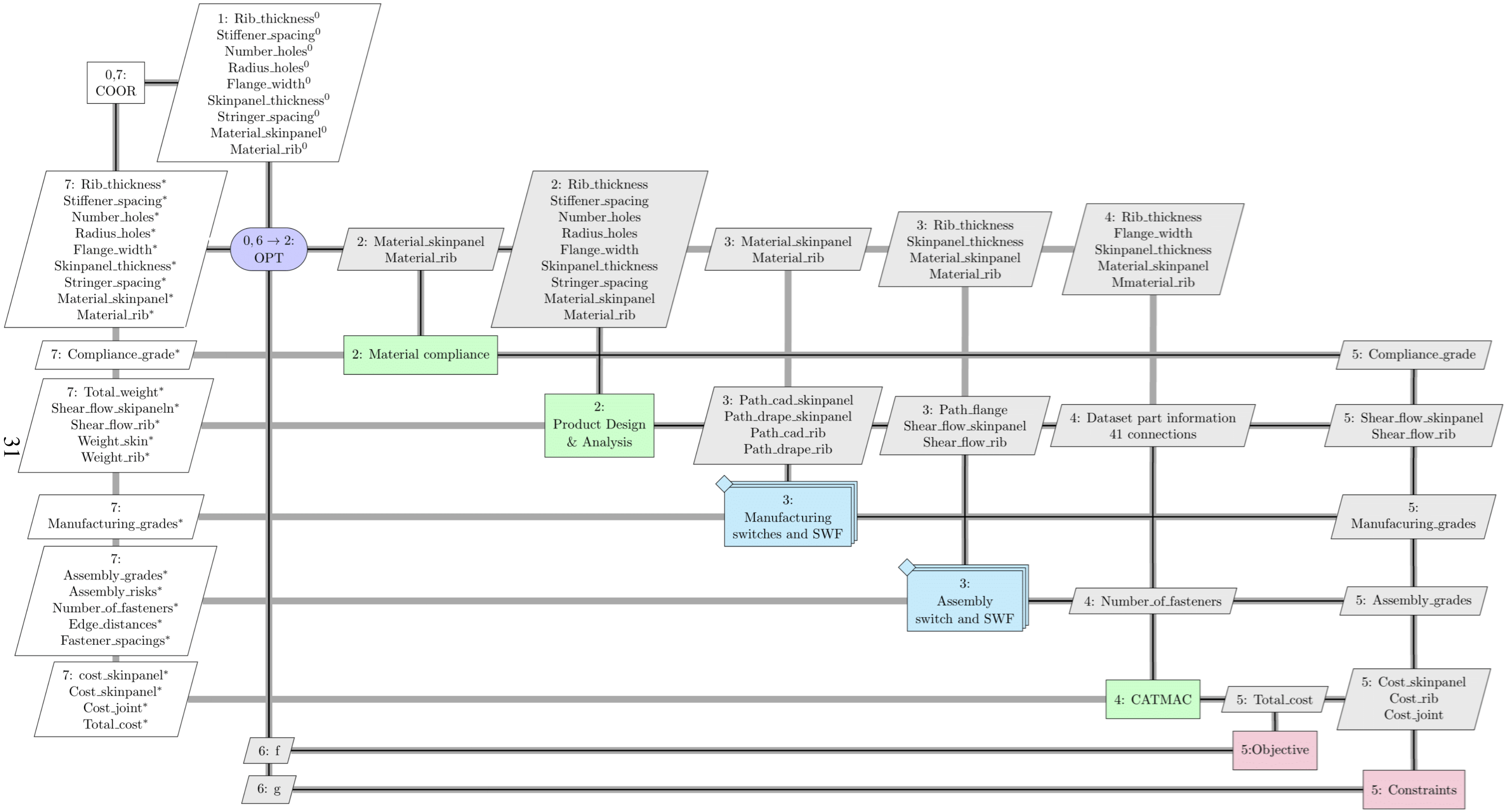


Figure 19 WF2 summarized XDSM formalization of the dynamic optimization of a rib-skin panel combination. The *Blue switch blocks* represent the switch system, including switches and corresponding branches.

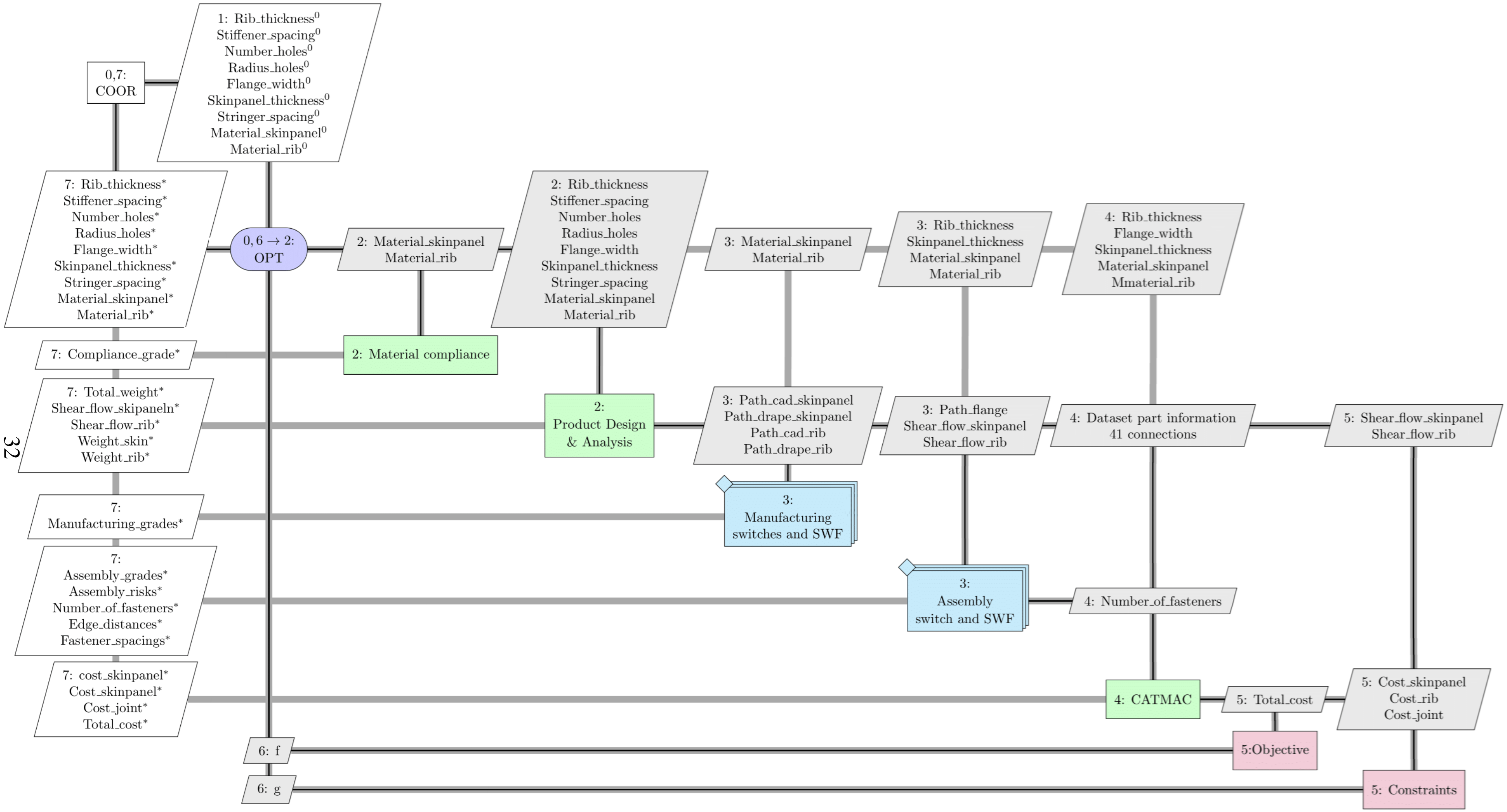
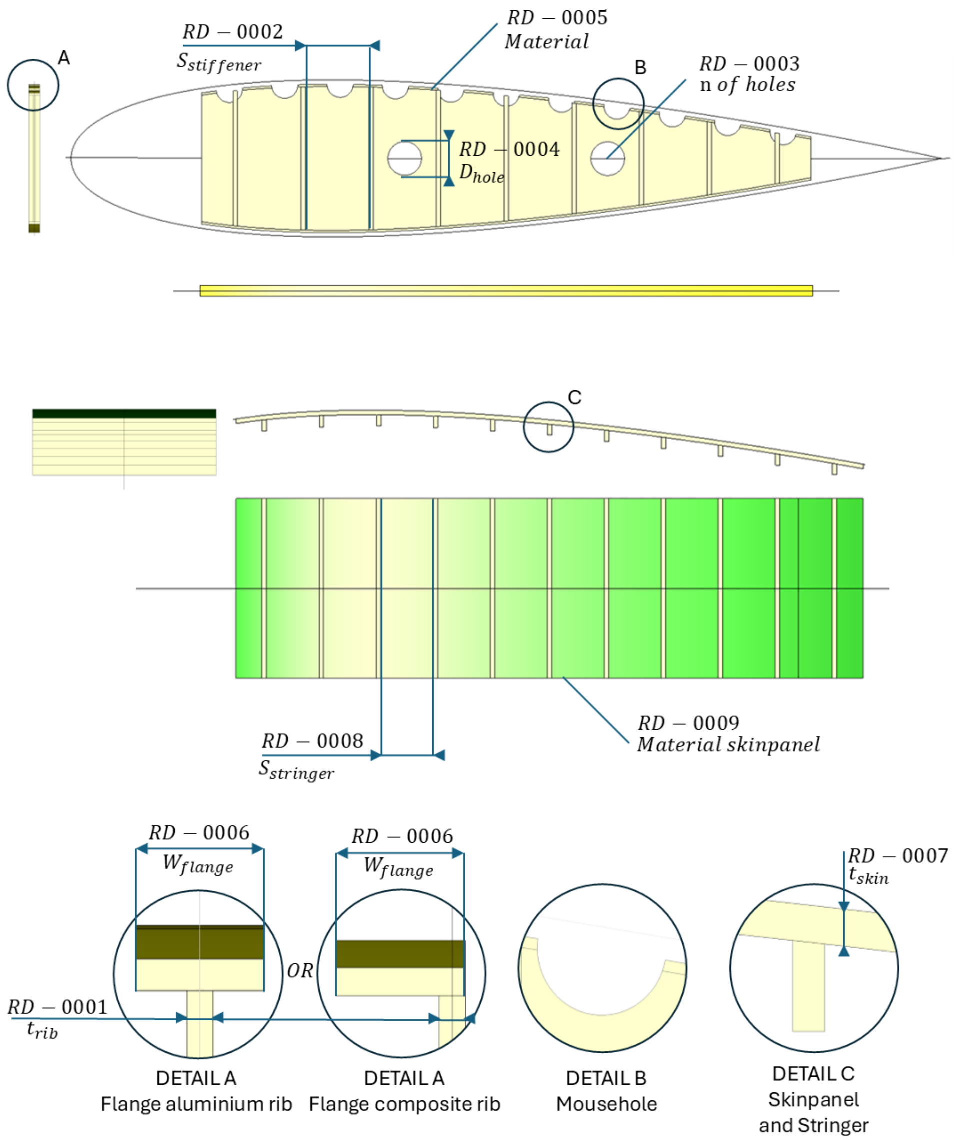


Figure 20 WF3 summarized XDSM formalization of the dynamic optimization where DfMA considerations are excluded as constraints while still evaluated.

B. Geometrical definition of wingrib-skinpanel combination

This appendix present all relevant design variables related to the wingbox geometry. The upper section illustrates the wingrib, and the lower section provides a detailed view of the skin panel. Both the wingrib and the skinpanel include specific detail, marked as A, B, or C, which are magnified below the pictures.
It is important to note that the wingrib geometry is dependent on the material selection. For aluminium, the wingrib features a T-flange, while composite wingribs have a L-flange, as illustrated in detail A.



Overview of geometrical definition of wingrib (top) and skinpanel (bottom) with the connected relevant design variables, specified using the unique design variable ID and name

.

C. Motivation selected optimization scheme for fastener optimization tool

This appendix explains the selection of the optimization algorithm NLPQL (Non-Linear Programming Quadratic Line Search algorithm) for the optimization of the fastener tool. NLPQL is a gradient-based algorithm capable of solving problems with continuous variables [15]. The primary advantage of NLPQL over NAVIRUN is its computational speed combined relative quick convergence. Alternative optimization algorithms, such as Suquential Quadratic Programming (SQP) and Generalized Reduced Gradient (GRC), were also tested. However, SQP failed to achieve convergence, while GRC and Differential Evolution required significantly longer computation times compared to NLPQL.

Although NLPQL may be less acurate and may not fully converge to the optimum (as demonstrated in Table 9), this limitation is found acceptable. On average, the joint cost is about 2% of the total cost. For instance, the incomplete optimization might result in an extra 20% of fasteners (as shown in Table 9), leading to an overall cost discrepancy of approximately 0.4%. Given that the uncertainty in CATMAC is generally around 50% [20], this discrepancy is considered neglected.

Selecting NAVIRUN as the performing algorithm instead of NLPQL for the full runs would increase computational time by approximately 130 hours.

Table 9. Algorithm analysis, comparing NAVIRUN and NLPQL for each of the fasteners tools

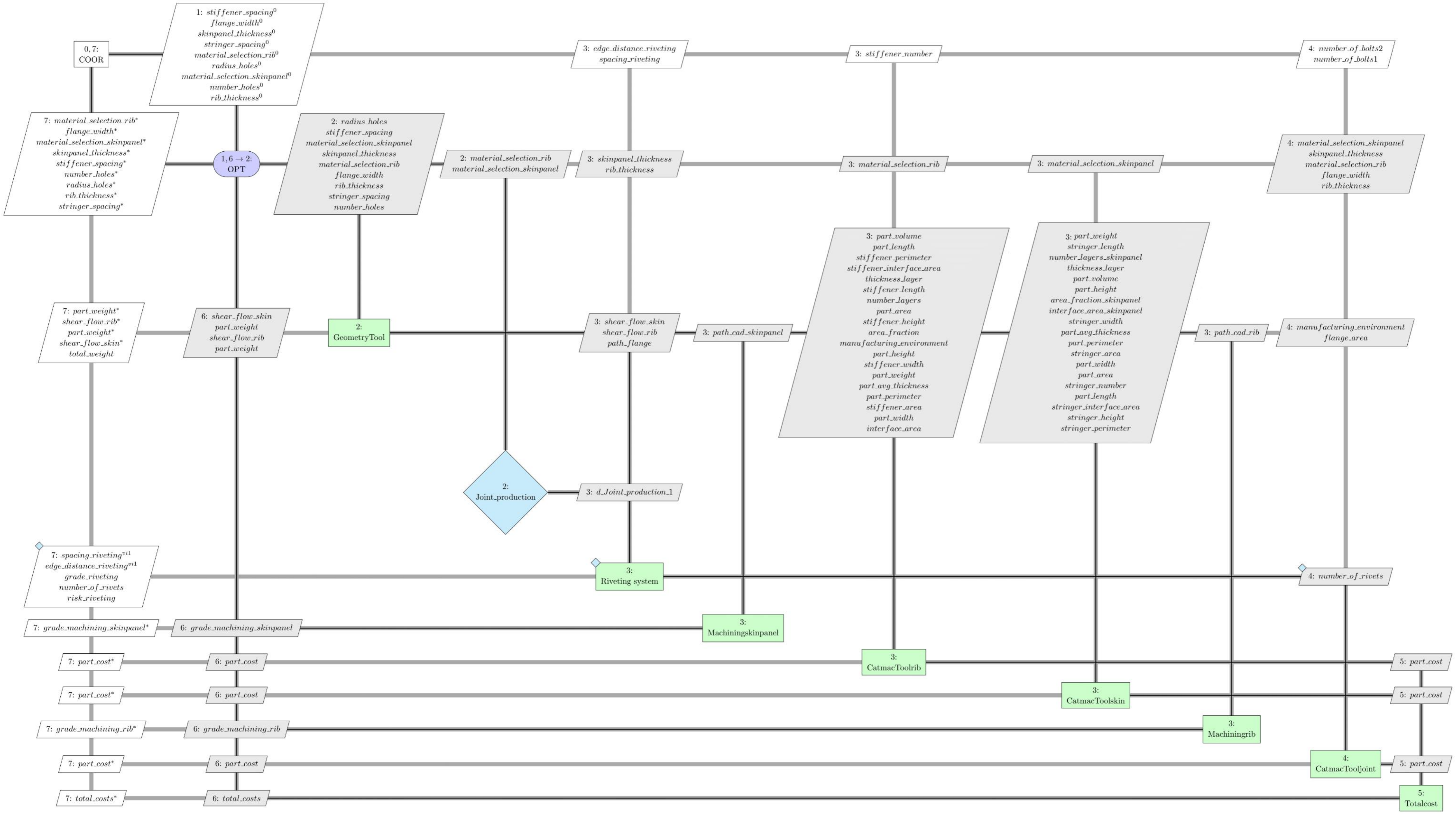
Assembly method	Riveting		Bolting 1		Bolting 2	
Optimization algorithm	NAVIRUN	NLPQL	NAVIRUN	NLPQL	NAVIRUN	NLPQL
Number of iterations	303	50	307	58	91	49
Total time	850s	100s	750s	150s	300s	130s
Results:						
Risk	0.45	0.79	0.92	0.92	0.32	0.39
Grade	1	1	1	1	1	1
Edge distance	1.8	3 (LB)	3.3	3.2	4.52	4.84
Fastener spacing	6 (UB)	5.9	3.7	3.4	8.00	6.05
Number of fasteners	40	46	11	11	24	24

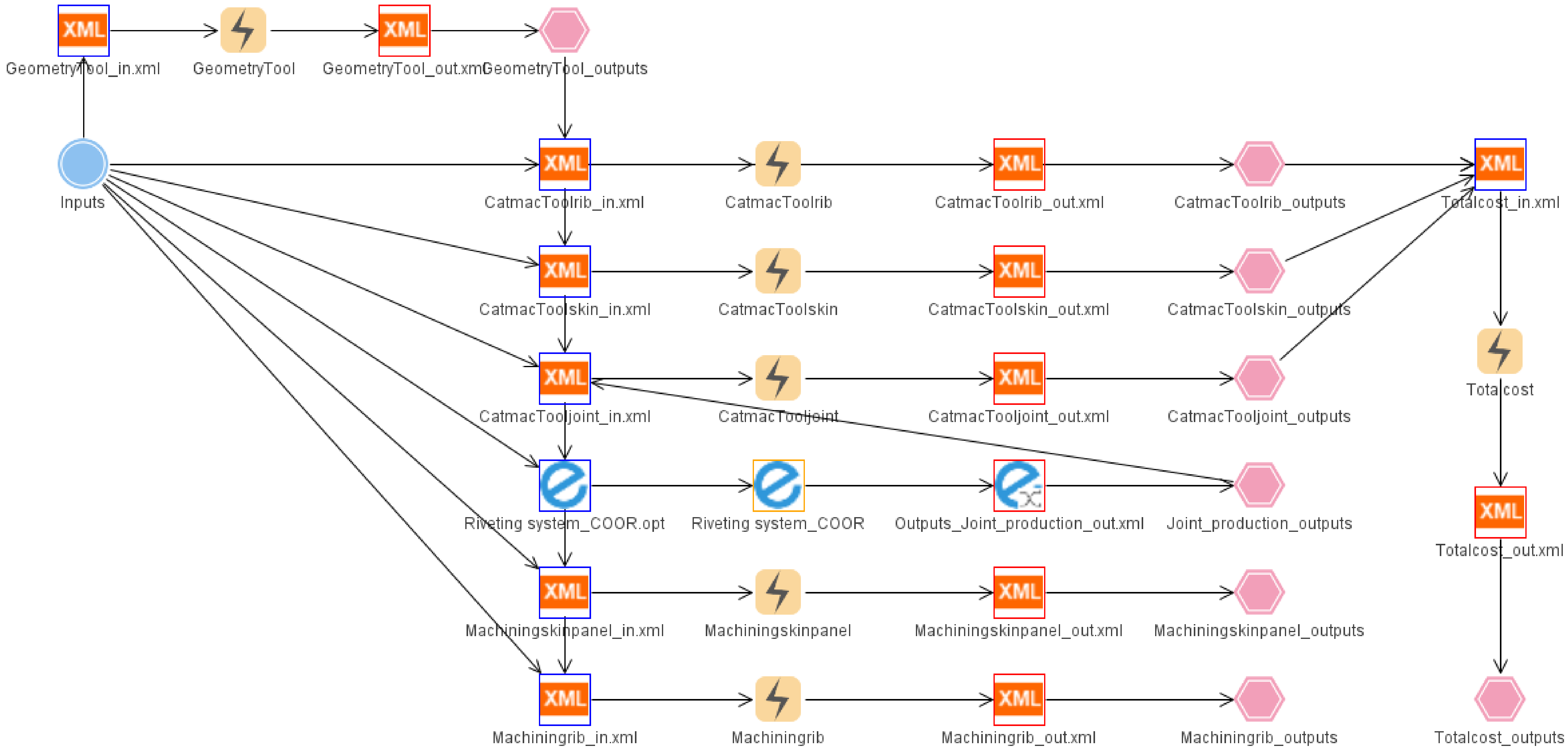
D. CATMAC required inputs for connection

CATMAC needs several inputs to assess the cost of a joint. These inputs are:

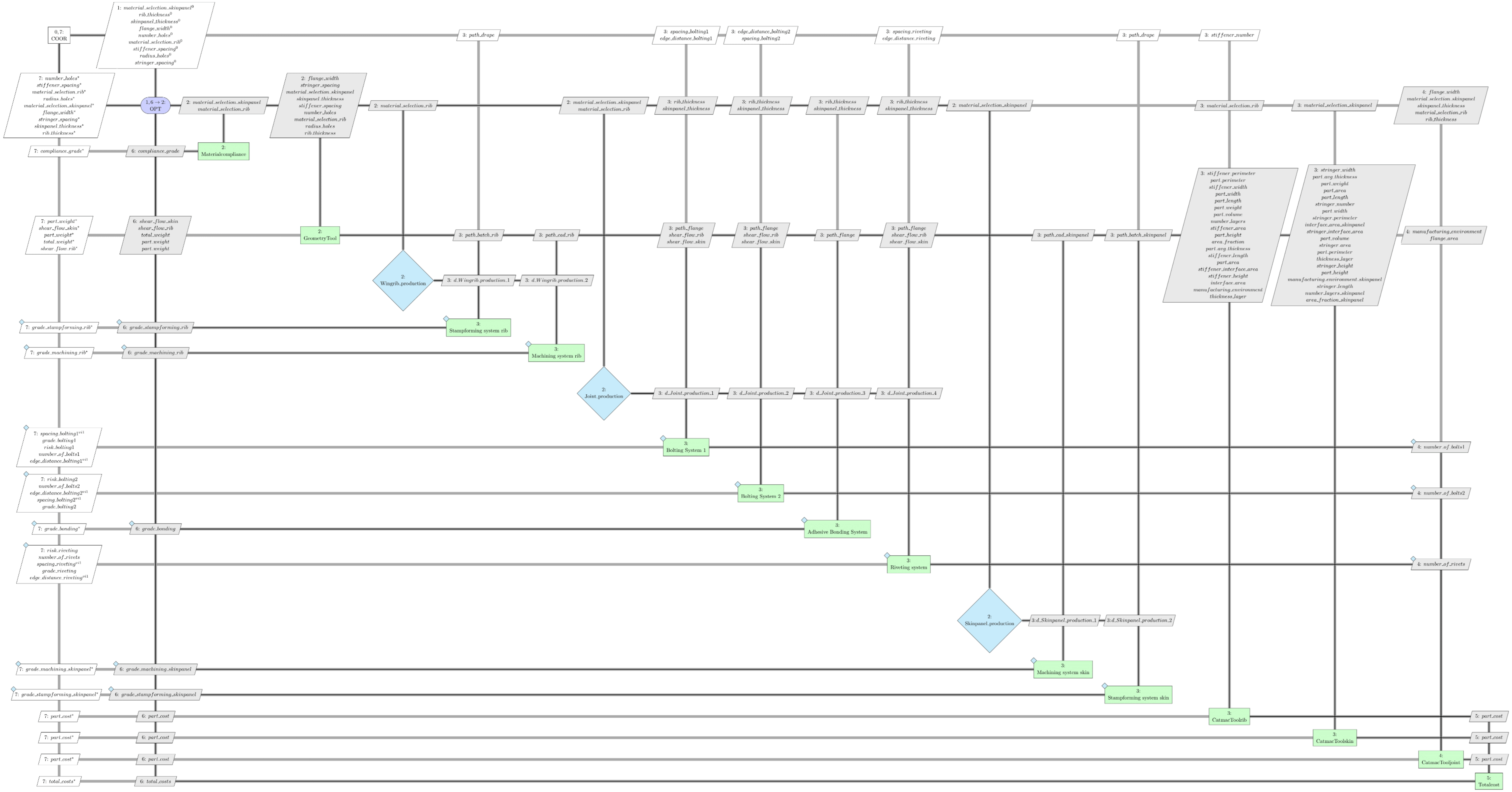
- manufacturing_environment
- manufacturing_process (either bonding or mechanical assembly)
- interface_area
- interface_width
- interface_length
- hole_diameter (if manufacturing_process is mechanical assembly)
- fastener_pitch (if manufacturing_process is mechanical assembly)
- fastener_type (if manufacturing_process is mechanical assembly)
- adhesive_type (if manufacturing_process is bonding

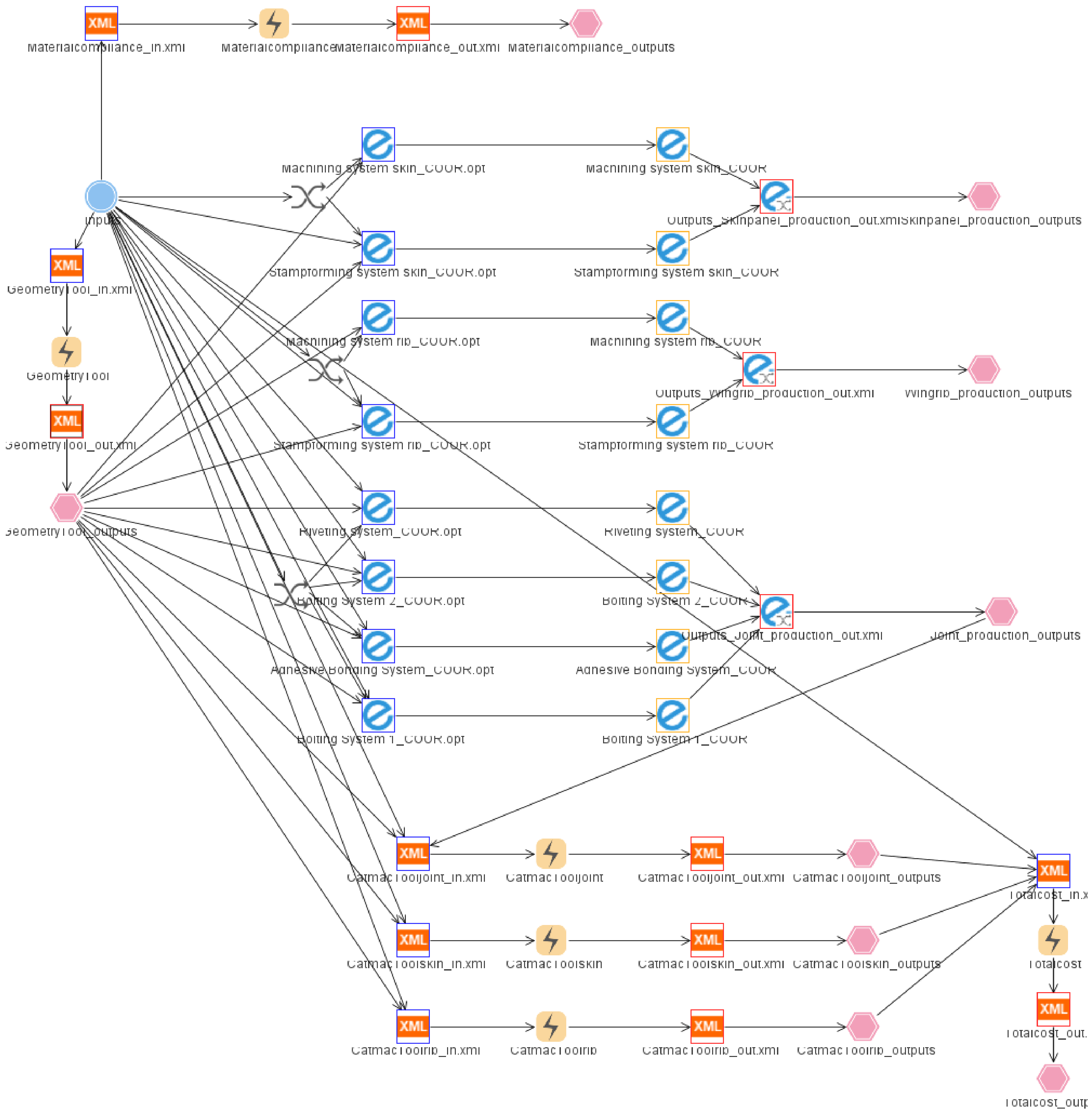
E. Optimus representation of static workflow (workflow 1)


0, 7:
COOR
1,6 → 2:
OPT
2:
GeometryTool
2:
Joint_production
3:
Riveting system
3:
Machiningskinpanel
3:
CatmacToolrib
3:
CatmacToolskin
3:
Machiningrib
4:
CatmacTooljoint
5:
Totalcost


GeometryTool_in.xml
GeometryTool
GeometryTool_out.xml
GeometryTool_outputs
Inputs
CatmacToolrib_in.xml
CatmacToolrib
CatmacToolrib_out.xml
CatmacToolrib_outputs
Totalcost_in.xml
CatmacToolskin_in.xml
CatmacToolskin
CatmacToolskin_out.xml
CatmacToolskin_outputs
Totalcost
CatmacTooljoint_in.xml
CatmacTooljoint
CatmacTooljoint_out.xml
CatmacTooljoint_outputs
Riveting system_COOR.opt
Riveting system_COOR
Outputs_Joint_production_out.xml
Joint_production_outputs
Totalcost_out.xml
Machiningskinpanel_in.xml
Machiningskinpanel
Machiningskinpanel_out.xml
Machiningskinpanel_outputs
Machiningrib_in.xml
Machiningrib
Machiningrib_out.xml
Machiningrib_outputs
Totalcost_outputs

F. Optimus representation of dynamic workflow (workflow 2)


0, 7: COOR
1, 6 → 2: OPT
2: Materialcompliance
2: GeometryTool
2: Wingrib_production
2: Joint_production
2: Skinpanel_production
3: Stampforming system rib
3: Machining system rib
3: Bolting System 1
3: Bolting System 2
3: Adhesive Bonding System
3: Riveting system
3: Machining system skin
3: Stampforming system skin
3: CatmacToolrib
3: CatmacToolskin
4: CatmacTooljoint
5: Totalcost
3: path_drape
3: stiffener_number
3: path_batch_rib
3: path_cad_rib
3: path_cad_skinpanel
3: path_batch_skinpanel
6: compliance_grade
7: compliance_grade*
6: grade_stampforming_rib
6: grade_machining_rib
6: grade_bonding
6: grade_machining_skinpanel
6: grade_stampforming_skinpanel
6: part_cost
6: total_costs
7: total_costs*
5: part_cost
4: number_of_bolts1
4: number_of_bolts2
4: number_of_rivets
4: manufacturing_environment
flange_area


MaterialCompliance_in.xml
MaterialCompliance
MaterialCompliance_out.xml
MaterialCompliance_outputs
Inputs
GeometryTool_in.xml
GeometryTool
GeometryTool_out.xml
GeometryTool_outputs
Machining system skin_COOR.opt
Machining system skin_COOR
Stampforming system skin_COOR.opt
Stampforming system skin_COOR
Outputs_Skinpanel_production_out.xml
Skinpanel_production_outputs
Machining system rib_COOR.opt
Machining system rib_COOR
Stampforming system rib_COOR.opt
Stampforming system rib_COOR
Outputs_Wingrib_production_out.xml
Wingrib_production_outputs
Riveting system_COOR.opt
Riveting system_COOR
Bolting System 2_COOR.opt
Bolting System 2_COOR
Adhesive Bonding System_COOR.opt
Adhesive Bonding System_COOR
Outputs_Joint_production_out.xml
Joint_production_outputs
Bolting System 1_COOR.opt
Bolting System 1_COOR
CatmacToolJoint_in.xml
CatmacToolJoint
CatmacToolJoint_out.xml
CatmacToolJoint_outputs
CatmacToolSkin_in.xml
CatmacToolSkin
CatmacToolSkin_out.xml
CatmacToolSkin_outputs
CatmacToolRib_in.xml
CatmacToolRib
CatmacToolRib_out.xml
CatmacToolRib_outputs
TotalCost
XML

G. Design variables and outcomes of test case

Optimization results of static and dynamic workflows (design variables and statistics)

	LB	UB	Workflow 1, Static	Workflow 1, Static	Workflow 2, Dynamic	Workflow 3, Dynamic without DfMA
Design variables:						
Material rib	0	1	Aluminium	Composites	Aluminium	Aluminium
Material skinpanel	0	1	Aluminium	Composites	Aluminium	Aluminium
Rib thickness	5 mm	10 mm	6.70 mm	7.93 mm	6.70 mm	10 mm
Flange width	20 mm	200 mm	65.80 mm	193.87 mm	65.80 mm	20 mm
Number of holes	1	4	1	2	1	1
Radius holes	10 mm	40 mm	25.00 mm	14.30 mm	25.00 mm	39.37 mm
Stiffener spacing	75 mm	300 mm	190.00 mm	204.04 mm	190.00 mm	205.42 mm
Skinpanel thickness	5 mm	10 mm	5.31 mm	5.99 mm	5.31 mm	5.31 mm
Stringer spacing	100 mm	300 mm	200.00 mm	277.47 mm	200.00 mm	274.09 mm
Statistics						
Number of iterations			567	558	828	694
Estimated run time			27.5 hours	10.5 hours	39 hours	29 hours

Optimization results of static and dynamic workflows (results)

	LB	UB	Workflow 1, Static	Workflow 1, Static	Workflow 2, Dynamic	Workflow 3, Dynamic without DfMA
Results:						
Material rib:			Aluminium	Composites	Aluminium	Aluminium
Material skinpanel:			Aluminium	Composites	Aluminium	Aluminium
Shearflow rib	150kN/m		150 kN/m	192.50 kN/m	150 kN/m	498.7 kNm
Shearflow skinpanel	30 kN/m		30.0 kN/m	33.18 kN/m	30.0 kN/m	30.0 kN/m
Manufacturability rib:	0		1	1	1	1
Manufacturability skin:	0		1	1	1	1
Assembly method			Riveting	Bonding	Riveting	Riveting
Assembly grade	0		1	0.20	1	-1
Assembly risk			0.83	-	0.83	0.65
Rib weight			13.70 kg	14.81 kg	13.70 kg	12.14 kg
Skinpanel weight:			9.85 kg	5.62 kg	9.85 kg	9.25 kg
Total weight:			23.55 kg	20.43 kg	23.55 kg	21.39 kg
Rib cost			€1653.05	€3237.40	€1653.05	€800.80
Skinpanel cost			€1156.97	€4185.62	€1156.97	€1152.39
Joint cost			€158.27	€75.93	€158.27	€64.58
Total cost			€2968.29	€7498.95	€2968.29	€2017.77

H. Compliance report generated for optimization

REPORT

Requirement Verification Output

Subject :		Issue date :	2025-03-21
Controlled :		Issue number :	
Copy to :		Report number :	
Company/Dept. :		Order number :	

Summary

This report provides an overview of the requirements compliance

Prepared/department :	Flight Performance and Propulsion	Tool :	Requirements Modeller
User :	keesv	Tool Version :	1.0
Checked/department :			

TU Delft

REQUIREMENTS

System: Wingrib

Result file: C:\Users\keesv\Thesis\GithubClones\repository-kees\FileUsedForComplianceReport.xml

UID	Name	Text	MoC	Test Case	Compl.	Value	Margin
RD-0001	Rib thickness	The rib must have a rib thickness between 5 and 10 mm.	MOCR-0001	TCR-0001	True	6.7	[235.0%, 33.0%]
RD-0002	Stiffener spacing	The rib must have a stiffener spacing between 75 and 300 mm.	MOCR-0001	TCR-0001	True	190.0	[153.33 %, 36.67%]
RD-0003	Number of holes	The rib must have 1 to 4 holes.	MOCR-0001	TCR-0001	True	1.0	n/a
RD-0004	Hole radii	The rib must have hole radii between 10 and 40 mm.	MOCR-0001	TCR-0001	True	25.0	[150.0%, 37.5%]
RD-0005	Material rib	The rib material should be aluminium (0) or composites (1)	MOCR-0001	TCR-0001	True	0.0	n/a
RP-0003	Critical buckling shear flow	The rib must have a critical buckling shear flow of 150 kN/m.	MOCR-0002	TCR-0001	True	150.0	0.0%
RP-0004	Cost rib	The rib must have a maximum cost of €5,000.	MOCR-0006	TCR-0004	True	1653.05	66.94%
RP-0005	Weight rib	The rib must have a maximum weight of 20 kg.	MOCR-0003	TCR-0001	True	13.7	31.5%
RP-0010	Critical buckling shear skin	The skinpanel must have a critical buckling shear flow of at least 30 kN/m	MOCR-0002	TCR-0001	True	30.0	0.0%

System: Wingbox

Result file: C:\Users\keesv\Thesis\GithubClones\repository-kees\FileUsedForComplianceReport.xml

UID	Name	Text	MoC	Test Case	Compl.	Value	Margin
RP-0001	Total weight	The wingbox must have a total weight of less than 70 kg.	MOCR-0003	TCR-0001	True	23.55	66.36%
RP-0002	Total cost	The wingbox must have a maximum cost of €10,000.	MOCR-0008	TCR-0006	True	2968.29	70.32%
RP-0009	Material Compliance	The jount must connect two materials that don't corrode.	MOCR-0004	TCR-0002	True	1.0	1.0%

System: Joint

Result file: C:\Users\keesv\Thesis\GithubClones\repository-kees\FileUsedForComplianceReport.xml

UID	Name	Text	MoC	Test Case	Compl.	Value	Margin

RP-0008	Cost joint	The joint must have a maximum cost of €500.	MOCR-0007	TCR-0005	True	158.27	68.35%

System: Skinpanel

Result file: C:\Users\keesv\Thesis\GithubClones\repository-kees\FileUsedForComplianceReport.xml

UID	Name	Text	MoC	Test Case	Compl.	Value	Margin
RD-0006	Flange width	The flange must have a width between 20 mm and 200 mm	MOCR-0001	TCR-0001	True	65.8	[229.0%, 67.1%]
RD-0007	Skinpanel thickness	The skin panel must have a thickness between 5 and 10 mm	MOCR-0001	TCR-0001	True	5.31	[165.5%, 46.9%]
RD-0008	Stringer spacing	The skin panel must have a stringer spacing between 100 mm and 300 mm	MOCR-0001	TCR-0001	True	200.0	[100.0%, 33.33%]
RD-0009	Material skinpanel	The skin panel material must be aluminium (0) or composite (1)	MOCR-0001	TCR-0001	True	0.0	n/a
RP-0006	Weight skinpanel	The skin panel must have a maximum cost of €5,000.	MOCR-0003	TCR-0001	True	9.85	80.3%
RP-0007	Cost skinpanel	The skin panel must have a maximum weight of 50 kg.	MOCR-0005	TCR-0003	True	1156.97	76.86%

VERIFICATION ELEMENTS

Means of Compliance

UID	Name	Text	Conditions	Process req.	Design stage
MOCR-0001	Inspection moc	Geometrical inspection of the CAD model	None	Future development	Not specified
MOCR-0002	Structural analysis moc	The structural analysis module of the rib tool has to be used	None	Future development	Not specified
MOCR-0003	Weight moc	The weight calculation module of the rib tool has to be used	None	Future development	Not specified
MOCR-0004	Compliance tool	Check if the parts comply with eachother	None	Future development	Not specified
MOCR-0005	Cost moc skin	A process-based cost estimation tool has to be used for the skin	None	Future development	Not specified
MOCR-0006	Cost moc rib	A process-based cost estimation tool has to be used for the rib	None	Future development	Not specified
MOCR-0007	Cost moc joint	A process-based cost estimation tool has to be used for the joint	None	Future development	Not specified
MOCR-0008	Total cost moc	A process-based cost estimation tool has to be used	None	Future development	Not specified

Test cases

UID	Name	Disciplinary tools	Design stage
TCR-0001	GeometryTool test case	{'DTR-0001'}	Not specified
TCR-0002	Material compliance tool	{'DTR-0002'}	Not specified
TCR-0003	Cost tool skin	{'DTR-0003', 'DTR-0001'}	Not specified
TCR-0004	Cost tool rib	{'DTR-0004', 'DTR-0001'}	Not specified
TCR-0005	Cost tool joint	{'DTR-0005', 'DTR-0001'}	Not specified
TCR-0006	Cost tool total cost	{'DTR-0006'}	Not specified

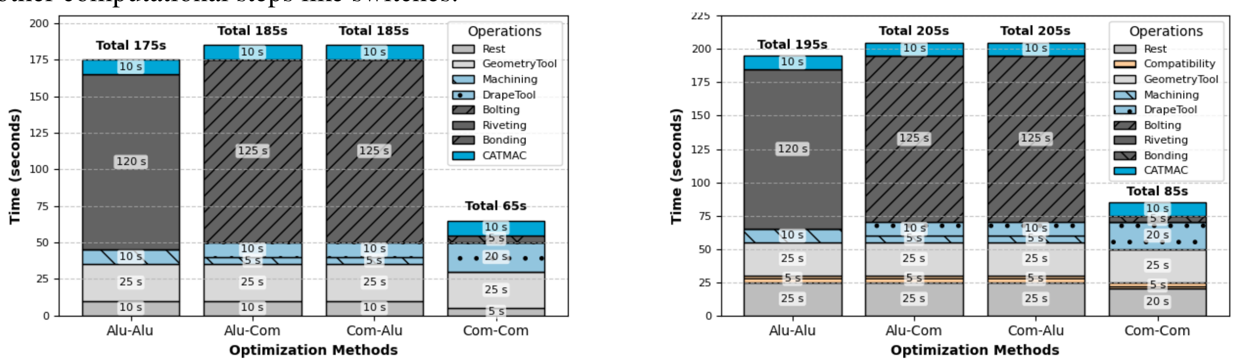
Disciplinary tools

UID	Name	Version
DTR-0001	GeometryTool	1.0
DTR-0002	Materialcompliance	1.0
DTR-0003	CatmacToolskin	1.0
DTR-0004	CatmacToolrib	1.0
DTR-0005	CatmacTooljoint	1.0
DTR-0006	Totalcost	1.0

I. Computational overhead caused by extra tools

Incorporating assembly considerations into the workflow increases computational time for both static and dynamic workflows. While the workflow of previous research only took ≈ 20 seconds per iteration [9], this research added an extra part-primitive, extra design variables, and a multi-conditional switch. A time-based evaluation was conducted for both the static and dynamic workflow. Although highly dependent on computational power and other variables, this appendix gives insight in the time durations used in this paper.

On average, the dynamic workflow requires an additional 20 seconds per iteration compared to the static workflow. This increase is primarily due to the material compatibility check, as shown in Fig. 27b, along with other computational steps like switches.



(a) Static workflows **(b)** Dynamic workflows

Figure 27. Breakdown of iteration time duration for static and dynamic workflows, divided by tool. Each material combination of aluminium (Alu) and composites (Com) is shown.

The time values shown in Fig. 27 are averaged results, but uncertainty remains, particularly for the optimization of riveting and bonding. These times depend on the convergence speed of the optimization process itself. To enhance accuracy, all workflows were executed within the same computational environment.

C

SUPPORTING DOCUMENTATION

C.1. CATMAC ANALYSIS

CATMAC is a tool created by GKN Fokker to estimate the total cost of a product based on empirical data. While it is included in this thesis as the way to calculate the cost, there are a few parts that needs further explanation. The CATMAC tool is used to calculate both the cost of the wing rib, skin panel and joint. For the joint, a workaround is performed to use the correct number of fasteners in the cost calculations, this workaround is explained below.

C.1.1. JOINT CALCULATION

CATMAC needs several inputs for the calculation of the cost, these are listed below:

- Manufacturing environment (material)
- Manufacturing process (either mechanical or bonding)
- Interface area
- Interface width
- Interface length
- Indexing hole diameter (for mechanical)
- Fastener pitch (for mechanical)
- Fastener type (for mechanical)
- Adhesive type (for bonding)

Within CATMAC, the number of fasteners are calculated by means of the fastener pitch. CATMAC uses the following equation to calculate the number of fasteners

$$number\_of\_fasteners = \frac{Interface\_area}{Interface\_width \cdot Fastener\_pitch} + 1 \tag{C.1}$$

meaning that it finds the total length of the flange pieces and multiplies this by the fastener pitch. This calculation is performed internally in CATMAC. Within this research, the number of fasteners is calculated by the fastener tools, this information is passed to CATMAC. In order for CATMAC to use the correct value of fasteners placed, the fastener pitch is tweaked to ensure the number of required fasteners according to CATMAC equals the number of fasteners placed on the structure according to the fastener tool.

The fastener pitch is calculated, this is done by rewriting eq. (C.1) as a function of the actual number of fasteners placed in the structure, as calculated by the JAM tools. The equation is rewritten as follows:

$$Fastener\_pitch = (number\_of\_fasteners - 1) \cdot (\frac{Interface\_width}{Interface\_width})$$

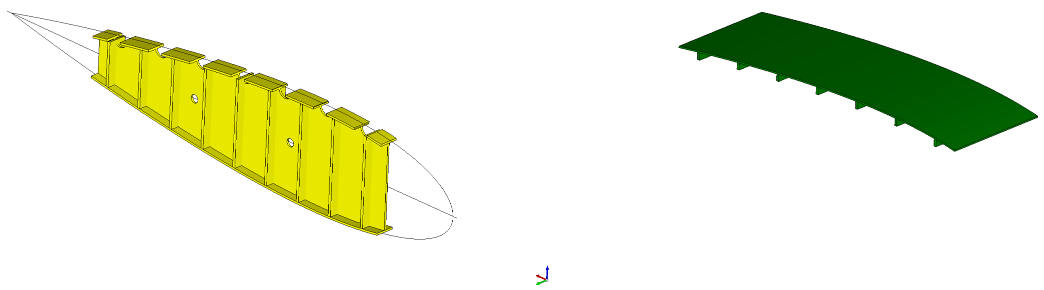
By calculating the fastener pitch in this way, and including it into the system, CATMAC calculates the cost with the correct amount of fasteners.

C.1.2. FASTENER COST

Two type of fasteners are used within this research, standard_blind_rivet_1433478 for the joint of two aluminium parts and hex_bolt_SS for connecting composite parts to aluminium parts. This stainless steel hex bolt is recommended. The cost of a riveting in €0.0412 a piece, independent on the diameter. For the hex_bolt_SS, its a function of the diameter D. cost= €0.002557 $\cdot D + 0.2096$

C.2. GEOMETRICAL DEFINITION

The general geometrical definition is defined in the Knowledge Based Engineering tool Parapy. The parametrized model created defines a part of the wingrib, including a skin panel and one wing rib as shown in fig. C.1b and fig. C.1a respectively.



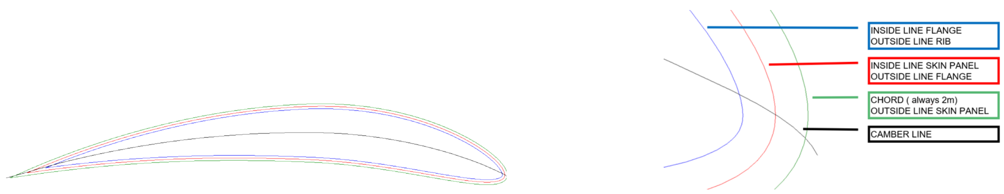
(a) Geometrical representation of wing rib

(b) Geometrical representation of skin panel

Figure C.1: Geometrical representation used within the use case, this representation is optimized for cost and is created in Parapy

C.2.1. DEFINED SECTIONAL SHAPES

The core of the geometrical definition is defined by three outlines, these lines are showed in fig. C.2. the outer line (outer mold line) (blue line) represents the outer surface of the skin panel where the length (trailing edge to leading edge) equals the set chord by the user. The red line represents the line where the skin panel meets the rib (the top surface of the rib and the lower surface of the skin panel). The blue line is the inner line, this is the bottom of the flange. Depending on the skinpanel thickness, the chord of the rib varies slightly.



(a) Airfoil representation with three lines

(b) Outlines of leading edge

Figure C.2: Outlines of airfoil shape used for geometric representation

C.2.2. DEFINED SKIN PANEL

The skin panel is created as a flat plate with stringers under it, this represents the normal conventional way how skin panels are usually made.

C.2.3. DEFINED RIB

The geometry of the rib, which was defined in the research of M. Nikitin **?**], stays similar. The rib is a web, strengthened with stiffeners and a flange at the top and bottom. A new addition in this research is the inclusion of the so-called *"Mouseholes"*, these mouseholes ensures that the stringers of the skin panel can be housed when the two parts are joined, such that these can be seamlessly joined. Figure C.3 shows two mouseholes with stringers.

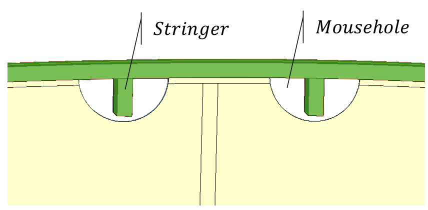


Figure C.3: side view of (wing) rib with two mouseholes included, these mouseholes are used to house the stringers.