AN MBSE ENABLED MDAO APPROACH FOR THE CONCEPTUAL DEVELOPMENT OF COMPLEX SYSTEMS

Mahmoud Fouda[1], Lennart Willrodt[1], Henrique Almeida[1], Juliana Cortez[3], Omar Hussein[2], Fabian Castaneda[3], Hachem Dhouib[3], Chifaa Dahik[3], Vincent Drouet[3] & Ryan Brook[2]

[1]Digital Engineering and Manufacturing, Capgemini Engineering, Hamburg, Germany
[2]Model Based System Engineering, Capgemini Engineering, Munich, Germany
[3]Product and System Engineering, Capgemini Engineering, Toulouse, France

Abstract

In the dynamic field of aerospace engineering, the integration of various design and analysis methods has become essential to tackle the increasing product complexity. Model-Based Systems Engineering (MBSE) and Multidisciplinary Design Analysis and Optimization (MDAO) both aim to enhance and accelerate the product development process. MBSE aims to comprehensively describe the system of interest and its enabling systems, emphasizing various perspectives that include architectural, functional and behavioral aspects while MDAO helps to evaluate, explore and optimize pre-selected design solutions using mathematical tools.
The requirement for employing early-stage engineering analysis for system design exploration has increased the need for the combined application of MBSE and MDAO. This integration of both methods facilitates more efficient and informed decision-making processes, enhancing the overall effectiveness of system development from conception to implementation.
This paper addresses this need by presenting a use case for the conceptual design of the CPulse medical drone. It uses a novel MBSE driven approach for the design and implementation of MDAO processes where a wing design optimization problem is considered. The MDAO process definition is connected to the MBSE product model through an enabling process model allowing the MDAO problem specification to be automatically extracted from the MBSE model. The product requirements and model parameters' are shared between the product and the enabling process model, to ensure the digital continuity throughout the product development, from systems engineering analysis and requirements definition to evaluation and optimization.
This approach enhances the agility and reusability of MDAO systems, to enable rapid adaptation and reconfiguration to meet changing requirements and constraints through different design iterations. To demonstrate this, a change in requirements is introduced to show the improved traceability of design decisions to requirements' updates. The use case presented in this paper is extendable to other engineering projects seeking to harmonize MBSE and MDAO.

Keywords: Digital Continuity, MBSE, MDAO, Digital Engineering, MBSE-enabled MDAO

1. Introduction

The aerospace engineering landscape is continuously evolving, driven by advances in technology and the increasing complexity of systems. This evolution demands the integration of various design and analysis methods. Such integration is pivotal in addressing the intricacies and interconnected challenges at different design levels, a necessity highlighted by the increasing complexity of aerospace projects [1].
The concept of Model-Based Systems Engineering (MBSE) has emerged as a critical approach to managing this complexity. MBSE offers a structured method for capturing, analyzing, and communicating system requirements and architectures, enhancing understanding and collaboration across diverse engineering teams [2]. Complementing this, Multidisciplinary Design Analysis and Optimization (MDAO) provides a framework for computationally integrating and optimizing systems considering the interactions of all disciplines, a crucial factor in achieving high-performance aerospace systems [3].

Recently, there has been an increasing effort in combining the two methods for a streamlined product development process. Some of the contributions relevant to the work presented in this paper are discussed in the next sections.

1.1 Challenges In The Design of Complex Systems

Designing complex systems, particularly in aerospace, presents significant challenges. One major issue is the increasing system complexity, characterized by more components, extensive software, and intricate interfaces. This complexity results in non-linear relationships and feedback loops, making it difficult to predict interactions and potential failures [4]. In addition, it makes it more difficult to trace the implications of system requirements on design decisions.

Integration and coordination among multidisciplinary teams add to these challenges. The need for multiple design iterations, each requiring reintegration, complicates the process further, especially with global diverse teams and experts across different organizations and countries.

Communication issues also hinder the collaborative development of complex systems. Effective information exchange is essential, yet conflicts often arise from differing interpretations of requirements among cross-functional teams. Ensuring consistent information flow and alignment among stakeholders is critical for project success [5].

Therefore, there is a need to investigate more effective design methodologies that address these challenges, where stakeholder requirements are strongly connected to design iterations and design space exploration, while improving traceability in relation to the product model and enabling systems along the product development life-cycle.

1.2 Integrating MBSE and MDAO: State-of-the-art

MDAO processes enable the evaluation, exploration, and optimization of complex systems through an integrated approach. However, a significant challenge lies in ensuring that these MDAO processes produce solutions that satisfy system requirements while remaining cost-effective and computationally efficient. To address this challenge, considerable effort has been directed towards establishing digital continuity from high-level business and system requirements down to MDAO processes [6] by bridging the gap between MBSE and MDAO.

First, a trend in the literature involves enriching MBSE models with information pertinent to MDAO processes. Specifically, the MBSE product model is enhanced with an optimization context depicted by a parametric diagram [7, 8, 9]. This diagram connects objective functions, the necessary optimization models, and the system to be optimized. The product model captures architectural variability by introducing decision points related to software allocation and hardware redundancy, ultimately generating and solving a constraint satisfaction problem.

In addition, the AGILE 4.0 project [5, 10] proposed an MBSE framework for designing and modeling "MDAO Systems". This framework aims to address various MDAO challenges in the context of system design. One approach involves using parameter values extracted from a Capella[1] model of the system to feed the MDAO process, thereby enhancing numerical continuity between MBSE and MDAO.

To capture architectural variability into the design process, [11, 12, 2, 13] introduced a methodology to model the design space using Architecture Design Space Graph (ADSG) where product components are allocated to functional requirements. This methodology allows modeling the architecture design problem as an optimization problem where architectural decisions are taken into account to explore a large design space of various possible architectures.

Furthermore, [14] proposed a methodology based on the Requirements, Functional, Logical, and Physical (RFLP) framework to integrate MBSE and MDAO. The methodology's effectiveness is demonstrated through a test case involving the design of a single-aisle transport category aircraft. Then, the compliance with aircraft requirements is verified, and a scenario involving changes to the requirements is presented. Their long-term objective is to automate the methodology, enabling MBSE to drive the formulation and execution of MDAO.

[1]Capella is an open-source solution for MBSE `https://mbse-capella.org/`.

Finally, the project R-Evol, launched in mid-2020 at the Institute of Research Technology (IRT) Saint Exupéry explored MBSE-based MDAO [15], suggesting a method designing the MDAO process by using common MBSE concepts (functional, logical, and physical layers), mirroring the approach used in system design. Stereotyping properties are included to explicitly identify the MDAO-related elements within the model so that they can eventually be exported with all information regarding the MDAO process specifications to enable instantiating and running the MDAO problem. The method is tested as an initial application on an academic use case: the sizing of a floor-beam system [2].

1.3 Contribution

Most approaches found in the literature either enrich MBSE models with results from MDAO [7, 8, 9] or focus on achieving digital continuity between MBSE and MDAO through automated connections for highly simplified academic use cases [15] that assume the pre-existence of product models. Furthermore, the connection between the product model and the MDAO process is not adequately addressed and the complexity of the existing use cases does not sufficiently show the intricacies faced during the design problem modeling and execution.

This paper addresses these shortcomings by building upon the work presented in [15]. It uses the RFLP-based MBSE methodology discussed in [15] on an academic use case of floor-beam system for the conceptual design of a medical Unmanned Aerial Vehicle (UAV), the CPulse drone.

The approach involves designing the MDAO processes in relation to the product model during the MBSE modeling phase, which enables the automatic extraction of MDAO process definitions for early-stage engineering analysis. By providing a common formalism for both system engineers and MDAO engineers, it helps manage the complexity of the process, enhance the communication among different stakeholders and experts, in addition to capturing decision rationales through design iterations and improve the traceability of decision making.

Furthermore, the paper aims to investigate the connection between the product model and the MDAO process definition and execution where the requirements and parameters are connected between both models. Additionally, it addresses the various challenges involved in digital engineering for product development such as constructing a detailed MBSE model, connecting requirements to performance, and integrating various engineering disciplines.

The primary goal of this research is to tackle the often-overlooked issue of digital continuity in the early stages of design and development, particularly in the aerospace industry, and establish a streamlined development process from high-level business requirements and systems engineering analysis to evaluation and optimization. Lastly, the research shows the flexibility of the current approach in response to top level requirements changes which is very common during product development programs [5].

The paper first outlines the method used to build the MBSE product model, design the MDAO process and export it, all while detailing the overall workflow. Subsequently, we present the results of the multidisciplinary design optimization problem, concerning the drone's wing design, aimed at maximizing its flight range. Next, we delve into a discussion of the results of this case study. Afterwards, a change in performance requirements is introduced, altering the design objective to achieve maximum speed within a fixed flight range of 50 km. The flexibility of the method to the requirements' updates is discussed alongside the case study results. Furthermore, the traceability and implications of the new requirements on design decisions are analyzed. The final section contains conclusions and future work.

2. Methodology

The integrated approach combining MBSE and MDAO methodologies aims to build a streamlined development process that is flexible with respect to requirements changes. In this approach, MBSE provides the structured framework that is used to model the design and specification of two interconnected models, the CPulse product model and the MDAO process model. We use a common industry commercial software Cameo System Modeler (CSM) [16] to build the models since it is widely

[2]The sizing of a floor-beam system description can be found here: `https://openmdao.readthedocs.io/en/1.7.3/usr-guide/tutorials/beam-sizing.html`.

adapted in the industry, and store them in Teamwork Cloud (TwC) [17] for collaborative development, such that the two models are connected through a “Project Usage”. This approach allows for seamless data flow between these two models and ensures digital continuity from system requirements and system parameters of the CPulse product model to the MDAO process model. Subsequently, the MDAO process definition is exported and constructed in the MDAO framework Generic Engine for Multidisciplinary Scenarios Exploration and Optimization (GEMSEO) [18]. This framework employs optimization techniques and algorithms to run the disciplinary analysis including the performance evaluation, where the objectives such as range or flight time are computed for each pre-selected wing design candidate.

2.1 MBSE Product Model

The CPulse drone project serves as a case study for a medical delivery drone that is used as the storyline to explain the method and underlying activities such as the development of system architecture, design, verification and validation (V&V) and simulation [19]. It aims to showcase MBSE and digital continuity capabilities, serve as a dynamic platform for developing new capabilities and enable collaboration across various engineering disciplines.
The MBSE product model of CPulse is developed with regards to the three pillars of MBSE that are language, method, and tool [20]. The first pillar represents the modeling language that provides standardized syntax and semantics for representing model elements and relationships. The second pillar is the method that outlines the processes and techniques used to apply the modeling language to ensure model consistency. The third pillar focuses on the software tools that are used to create, manage, and analyze models.
In this approach we have chosen to design the CPulse model utilizing the Cameo Systems Modeler (CSM) that goes hand in hand with the most widely used modelling language SysML [21] in combination with the MagicGrid methodology [22] that was selected after a thorough evaluation of numerous other MBSE methodologies. An overview of the MBSE model of the CPulse drone is presented in Figure 1.

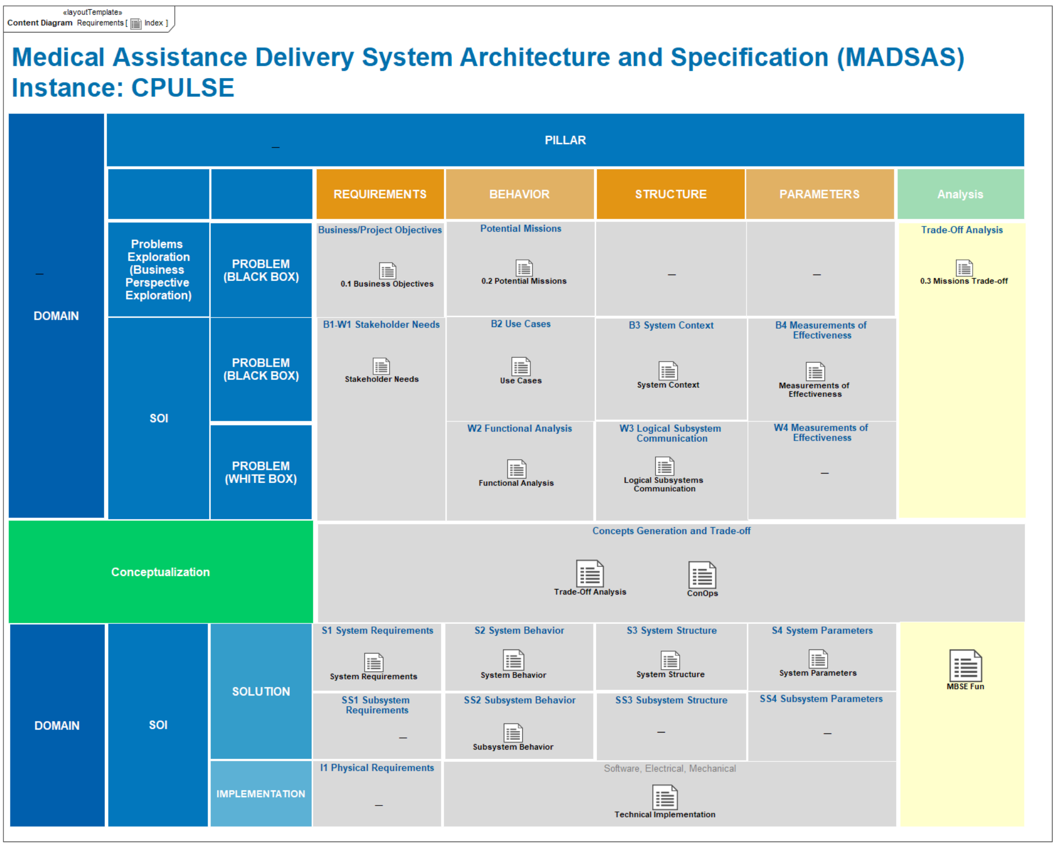


Figure 1 – MagicGrid framework applied to the CPulse MBSE model (adapted from [19]).

MagicGrid provides a structured framework for integrating multiple aspects of the product design into the MBSE model. It comes with templates that facilitates the modeling activities and provides a comprehensive documentation. MagicGrid breaks down the modelling process to three main domains, Problem Domain, Solution Domain, and Implementation Domain (see Figure 1) that aims at assisting the systems architect to model novel systems without having a specific solution selected in the early design phases. For the System of Interest (SoI), each domain definition includes different aspects referring to the four pillars of the SysML, that is, Requirements, Structure, Behavior, and Parametrics. Each cell of the intersection of rows and columns allow to represent different system views that consists of diagrams including model elements to specify the System of Interest (SoI) [22].
We have adjusted MagicGrid to bridge the aspects that we needed to address, such as trade-off analysis and showing the sync between the structure and behavior modelling views. The trade-off analysis serves in the systematic comparison between various potential missions. The behavior-structure sync shows various diagrams that depict both the structure and the behavior of the system under development. Once the model is executed the behavior view with its corresponding tokens reflects in the structural elements passing the same tokens.
Another aspect was the conceptualization part that bridges the problem domain and the solution domain. Carrying out activities such as devising a morphological matrix, structured selection matrix, Concept of Operations (ConOps) and capturing them in the MBSE model is an essential part of the product development process. For more details on this adjustment implementation, the readers are referred to [19].

2.2 MDAO Process Model

Instead of immediately diving into the MDAO workflow architecture, we use common MBSE abstraction layers (requirements, functional, logical, physical) to model and define the MDAO process, following the methodology proposed by [15]. In this approach, the MDAO process is treated as an enabling system, facilitating the development of a rationalized process that aligns with the product model. This process is represented in a dedicated MBSE model, integrated alongside the existing MBSE product model.
The MDAO process modeling consists of several steps that are closely aligned with MBSE concepts and are summarized below:

1. We define the requirements to formalize the problem that we aim to solve with the MDAO process, along with associated metrics, such as computational cost and execution time. Based on these requirements, the functional architecture of the MDAO process is then established.

2. We formalise the problem we seek to address with the desired inputs and outputs within the system view and functional architecture. Specifically, we elaborate a thorough understanding of the problem that our process is intended to solve by identifying the design variables, parameters, objectives, and constraints.

3. We define the process functions required to evaluate the objective and constraints, focusing on the functional architecture breakdown (see figure 7) and data flow. This resulting functional architecture is crucial for capturing the knowledge embedded in the designed process, as it encompasses expert insights, physical phenomena, expected fidelity and the considered couplings.

4. We identify the logical disciplines by appropriately grouping the computational steps (the functions), linking the functional layer to the logical layer by allocating the functions to the logical components. This grouping is based on various criteria, such as minimizing couplings between disciplines to simplify the sequencing of their execution, or clustering functions that belong to the same physical domains (e.g., aerodynamics, structure, etc.).

5. We specify the details of the implementation of each discipline in the physical layer by allocating the logical components to the physical ones.

It is crucial to maintain continuity between the MDAO process model and the product model in order to ensure coherence and efficiency and to minimize errors and redundant work. Several elements in the process model directly correspond to those in the product model. Thus, early detection of inconsistencies between these models can prevent additional MDAO executions or system design loops. For instance, product requirements can be translated into constraints within the MDAO problem, ensuring that updates to requirements highlight necessary constraint adjustments. In addition, design variables and parameters in the MDAO process should align with the product model parameters, with constant parameters matching the product definition and design variables using the product model values as initial inputs.

After fully modeling the different layers of the MDAO process, we add stereotyping properties, i.e. ('labels') to explicitly identify MDAO-related elements in the model. These properties include design variables, objectives, constraints, functions, disciplines, and fixed parameters. A meta-model of the MDAO process model is presented in Figure 2, where the stereotypes are clustered and allocated to the dedicated layers of the process model. The use of stereotyping properties provides several benefits, allowing model reviewers to quickly identify main MDAO elements, defining specific properties for each MDAO element type, and enabling an automated process specification export.

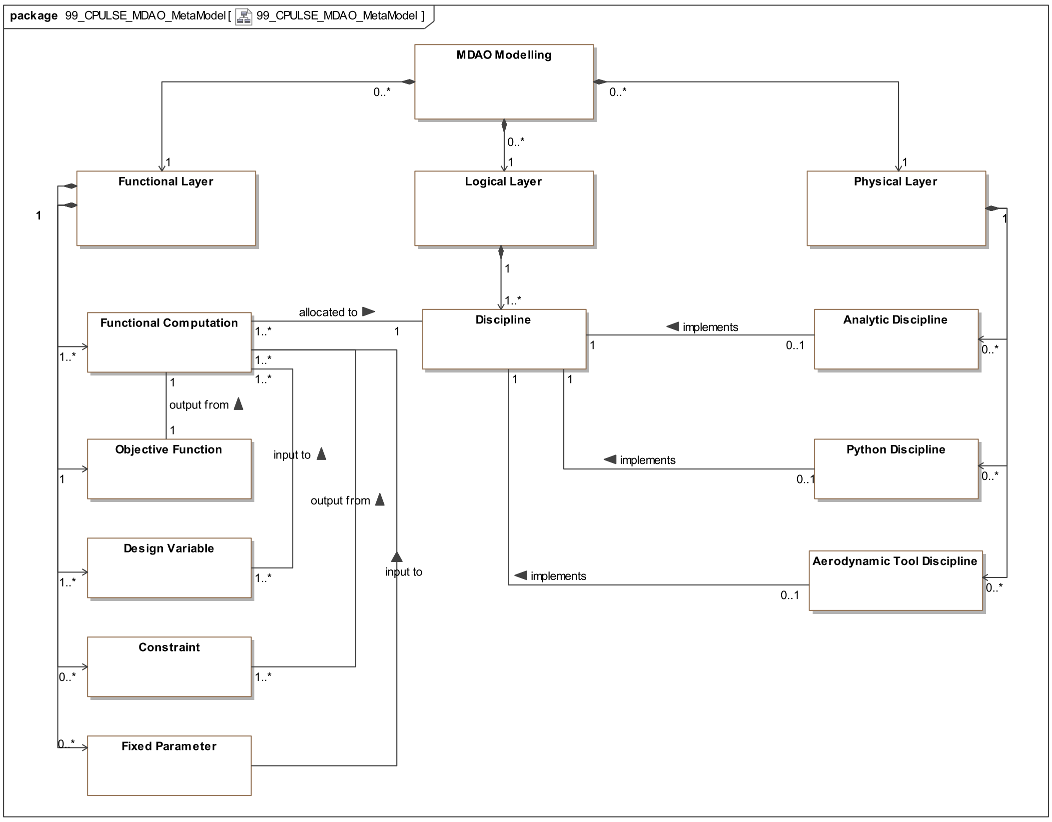


Figure 2 – MDAO process meta-model.

2.3 Extracting the MDAO Process Definition and Overall Workflow

To allow collaboration on building the model and to ensure digital continuity from system requirements, constraints and system parameters of the CPulse product model to the MDAO process model, both models are connected in CSM and stored in the Teamwork Cloud (TwC) platform [17]. The MDAO process model uses the CPulse product model through a "Project Usage" to retrieve the necessary data. A custom Python tool, based on the work presented in [15], interacts with TwC Application Programming Interface (API), to retrieve the necessary model's data and MDAO process definition, in order to prepare the optimization case that shall be run with GEMSEO. The MDAO process specification is automatically extracted from the MBSE model and the enabling models, generating the eXtended Design Structure Matrix (XDSM) diagram, providing a complete description of the MDAO problem. The overall workflow that describes the CPulse conceptual development process is shown in Figure 3.

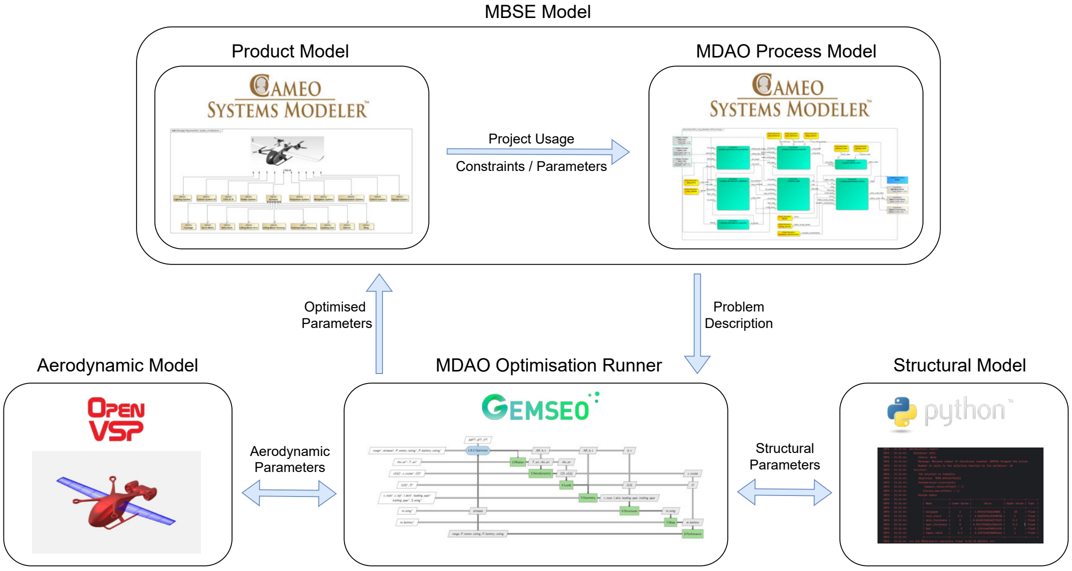


Figure 3 – Workflow diagram of the CPulse drone conceptual development focused on wing design optimization case study.

The MDAO framework GEMSEO facilitates the evaluation of various performance scenarios and optimizes the product's design for its intended operational environment, mission and objective using the model parameters and specifications of the MBSE and MDAO models. This stage is crucial to ensure that the product meets its performance targets and requirements. The resulting optimized parameters can be passed back to the MBSE model to update the product model specifications.
Before running GEMSEO, several optimization parameters, such as selecting the MDAO architecture, the optimization or Design of Experiment (DOE) algorithms, and the maximum number of iterations, need to be defined. These choices should be determined by the MDAO architect, rather than included in the enabling system.
This approach allows us to execute multiple experiments with different numerical configurations without modifying the model. However, it requires the addition of a few manual steps in the process, such as instantiating the GEMSEO optimization case and updating the MBSE model data after the optimization, particularly if a user needs to run a new execution from the optimized design. The goal is to fully automate the workflow in upcoming future work, possibly through the addition of multiple enabling MDAO systems that uses different pre-defined configurations.

3. Implementation

We use Cameo Systems Modeler (CSM) [23] for MBSE modeling and the MDAO framework GEMSEO [18] to run the multidisciplinary analysis and optimization problem. The disciplinary analysis includes a mix of custom Python scripts, analytic expressions and existing tools. The aerodynamics tool VSPAERO [24] which leverages OpenVSP geometries is used to estimate the aerodynamics performance of the drone. The structure analysis and performance evaluation are computed using Python scripts. The precision of the selected tools is sufficient for the present publication, which aims at demonstrating the methodology on an aerospace related use-case, but each of them could be replaced by a higher-fidelity one if required by the scope of the work.

3.1 Designing the CPulse Product Model

The CPulse product model was developed utilizing CSM and the MagicGrid methodology applying the SysML language as outlined in chapter 2.1. Here we will focus on the solution domain of the model as this part contains the system requirements, the system structure, and parameters crucial for executing the MDAO process in a later step. Referring to the four pillars of SysML, we describe the most relevant parts of this model in the upcoming sections.

3.1.1 Requirements

For the development of a UAV, requirements play an important role to ensure that the product meet the stakeholder needs and satisfy the system requirements. Therefore, the requirements in the CPulse product model are grouped into categories: stakeholder requirements, which reflect the needs of stakeholders, and system requirements, which are derived from the stakeholder requirements. As system requirements define the design and functionality of the product, they are essential for the MDAO process. In this context, performance requirements are particularly important because they relate to the objective function and constraints of the MDAO process. Additionally, system requirements for the wings are critical, as they constrain the wing geometry which is the subject of this study. Both performance requirements and system requirements for the wing are shown in Figure 4. Each requirement is refined with at least one constraint that contains the limits or ranges of values defined by that requirement. For instance, Figure 5 shows how the wingspan is constrained between two values. These constraints serve as inputs for the MDAO process model, as shown in Section3.2.

#	△ Name	Text
1	SR System Requirements	
2	SR.2 Mission	
3	SR.2.4 Performance	
4	SR.2.4.1 Range	The system shall have a range of at least 50 km.
5	SR.2.4.5 Mass	The take-off mass of the UAV shall not exceed 25 kgs.
6	SR.2.4.6 Payload	The UAV shall be capable of carrying a payload up to 5 kg.
7	SR.3 UAV	
8	SR.3.1 Structural	
9	SR.3.1.2 Wings	
10	SR.3.1.2.2 wing_span	The wing span shall be in a range between 1.8 m to 3.5m
11	SR.3.1.2.4 wing_taper_ratio	the wing taper ratio shall be in a range between 0.5 and 1
12	SR.3.1.2.5 wing_aspect_ratio	The wing aspect ratio shall be in a range between 5 and 10
13	SR.3.1.2.13 wing_material	The wing material shall be carbon fiber with a minimum tensile strength of 210 MPa

Figure 4 – Excerpt of relevant performance requirements and wing requirements formulated for the CPulse drone.

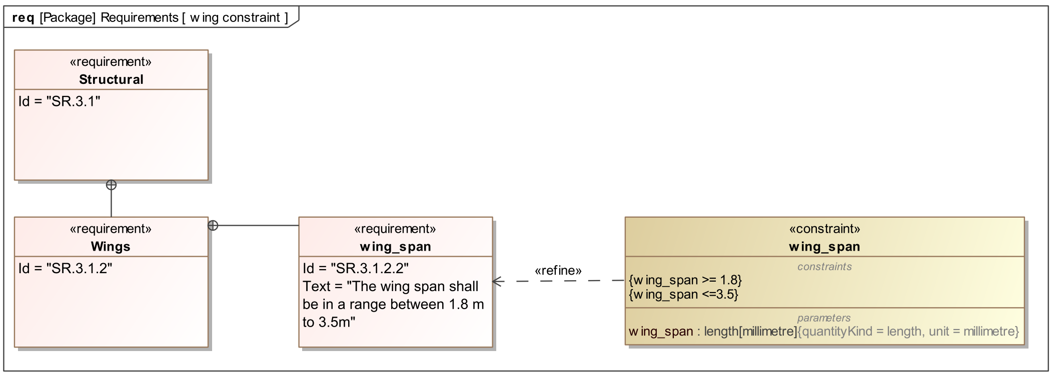


Figure 5 – Example of how a constraint refines the requirement, as shown for the wing span.

3.1.2 Structure (i.e. System Architecture)

The CPulse product model provides a system structure where the System of Interest (SoI) is decomposed into several subsystems that are broken down even further into more specific components as illustrated in Figure 6. These subsystems and components are represented as blocks within SysML block definition diagrams (bdd). Since subsystems have interfaces to other subsystems, their interconnections are modeled using internal block diagrams (ibd). The block of a subsystem contains a set of defining parameters, which can be utilized in parametric diagrams to calculate related parameters. In this study, the performance parameters, and the parameters of the subsystem wing are used in the MDAO process definition. Some fixed wing parameters like *skin_thickness* or *spar_thickness*

serve as inputs for the process model in order to allow certain computation steps (see Section 3.2). Other parameters like *wing_mass*, *wing_span*, *aspect_ratio*, *taper_ratio*, as well as the performance parameters like *range* and *speed* are subject to updates through the MDAO process.
Finally, all parameters are linked to the specified system requirements and must fulfill them, which can be verified through requirements checks in CSM.

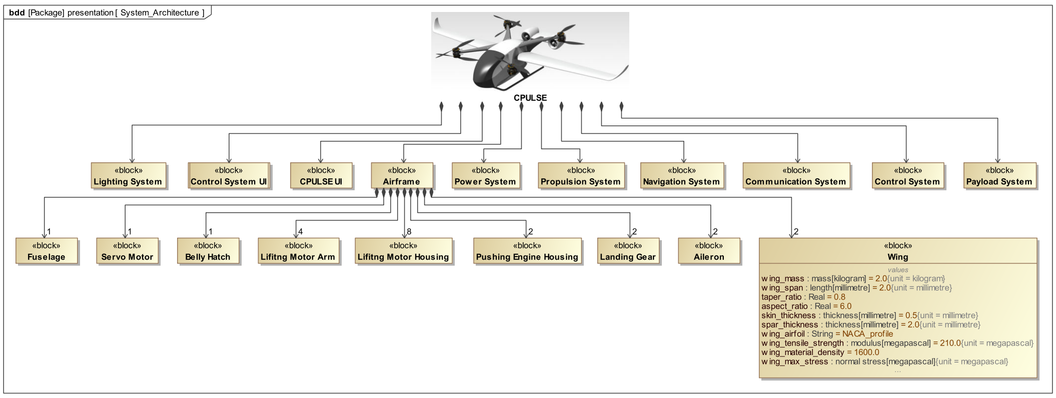


Figure 6 – CPulse system structure and breakdown of subsystems and components, with the wing parameters as an example.

3.1.3 Parametrics

The parameters of the model elements in the CPulse product model define the characteristics of the system. These parameters are used in parametric diagrams to model the relationship between various parameters of subsystems and components, which helps to understand how changes in one component affect other subsystems. In the CPulse product model, parametric diagrams were used accordingly and extended to build a parametric mission feasibility study that is analyzing the system's behavior under different scenarios.
In this work, all the computation efforts are executed externally by GEMSEO due to the higher complexity of computations and the use of external tools such as OpenVSP and VSPAERO. Once the results of this analysis are available, it is intended to implement them in the product model and to further process them using parametric diagrams. This integration aims to refine the model by enhancing its capabilities to simulate the system performance.

3.1.4 Behavior

For modeling the system's behavior, SysML provides several types of diagrams such as activity diagrams, sequence diagrams, state machine diagrams, and use case diagrams. These have been used for modeling the drone's behavior across diverse missions and operations. In this study, mission analysis computations are executed externally. In future work, state machines could be used to model the different behaviors for certain flight modes such as climbing, horizontal flight, and landing. This requires a clear understanding of various parameters that specify the flight modes, e.g. aerodynamic properties, performance properties, and mission parameters.
To summarize, the CPulse demonstrator serves as a platform for collaborative engineering. The MBSE product model utilizes the SysML language and the MagicGrid framework to define requirements, structure, parametrics, and behavior into the model. The CPulse's requirements, system structure, and parameters provide essential inputs for the MDAO process as discussed in the next section.

3.2 Designing the MDAO Process Model

The design and specification of the MDAO process architecture is modeled in an enabling model - the MDAO process model. Utilizing the SysML language within CSM, this model is structured into three layers: Functional, Logical, and Physical (FLP). The functional layer describes the MDAO process by breaking down computation steps into discrete functions, the logical layer organizes these functions into specific disciplines, the physical layer is tailored to specifying the implementation of the disciplines, while the execution of the MDAO workflow is described in the generated eXtended Design Structure Matrix (XDSM). These elements provide all necessary information to process each discipline and fully define the optimization process.

3.2.1 Functional Layer

The functional layer represents the functional architecture of the MDAO process, where the process is broken down into it's main computation steps, modeled as activities in a SysML activity diagram, presented in Figure 7. These activities are called functions, each of which having its own inputs and outputs. The output of a function is always a result (e.g. *lift_coefficient*) produced by the function. The output of a function can also be used as a constraint, such as the maximum wing stress. The input of a function can be:

- **A design variable:** The parameters of the product on which the designer can act to achieve an optimal design. Imported from the product model and derived from the appropriate system requirement of the product (e.g. “The wing span shall be in a range of 1.8 m to 3.5 m”).
- **A fixed parameter:** It represents an assumption or a model parameter used in the computation. It is imported from the product model (e.g. cruise_altitude = 250 m).
- **A result of another function** (e.g. wing_mass).

Each model element in the functional layer is typed by a specific stereotype. The Functions are typed by “Functional”, whereas the incoming constraints or functions outputs that are used as constraints are flagged as "Constraint", design variables are typed by “Design Variable”, fixed variables as “Fixed Parameters” and the objective function as “Objective Function”. This differentiation ensures that each element can be identified correctly when the model is exported to GEMSEO.

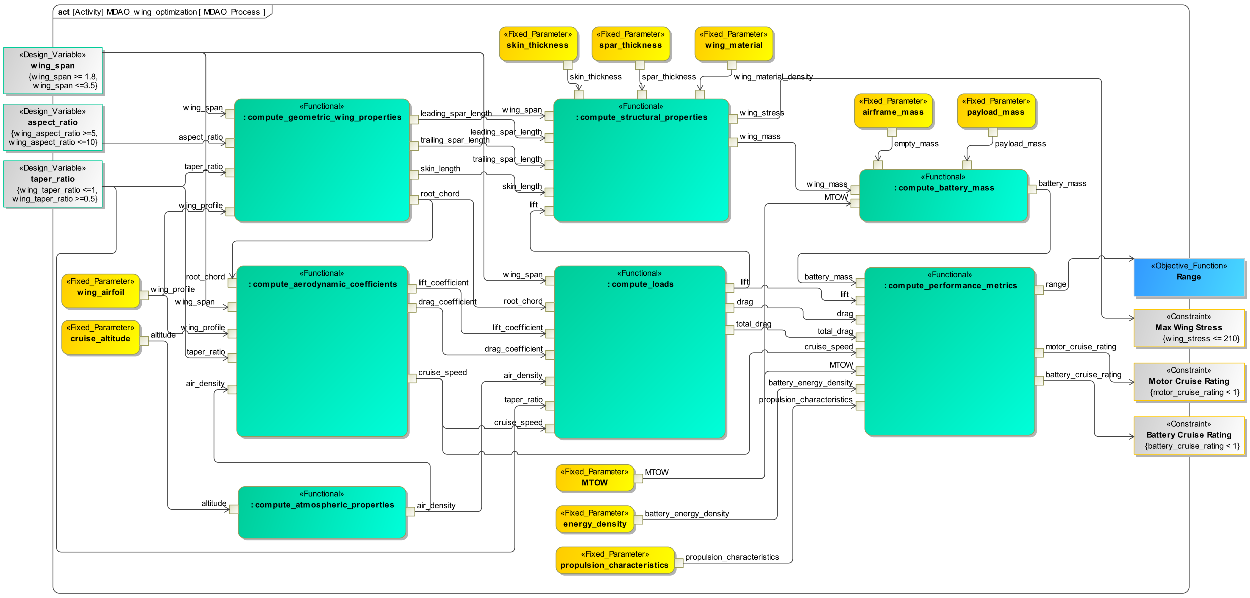


Figure 7 – Functional architecture of the MDAO process modeled in a SysML activity diagram.

3.2.2 Logical Layer

The logical layer connects the functional layer and the physical layer through an abstraction, clustering certain functions into specific disciplines. Each discipline is modeled as a block element in a SysML block definition diagram (bdd), as shown in Figure 8. Functions belonging to a specific discipline are linked to it through "allocate" links (e.g. function "compute_structural_properties" is allocated to the discipline “Structure”). Multiple functions can be allocated to the same discipline. The scope of decomposition is flexible and can be adapted according to the user's needs. The method of linking logical elements to the elements of the physical layer is described in the next section. Similar to the stereotyping of elements in the functional layer, all block elements of the logical layer are typed by the specific stereotype “Discipline”.

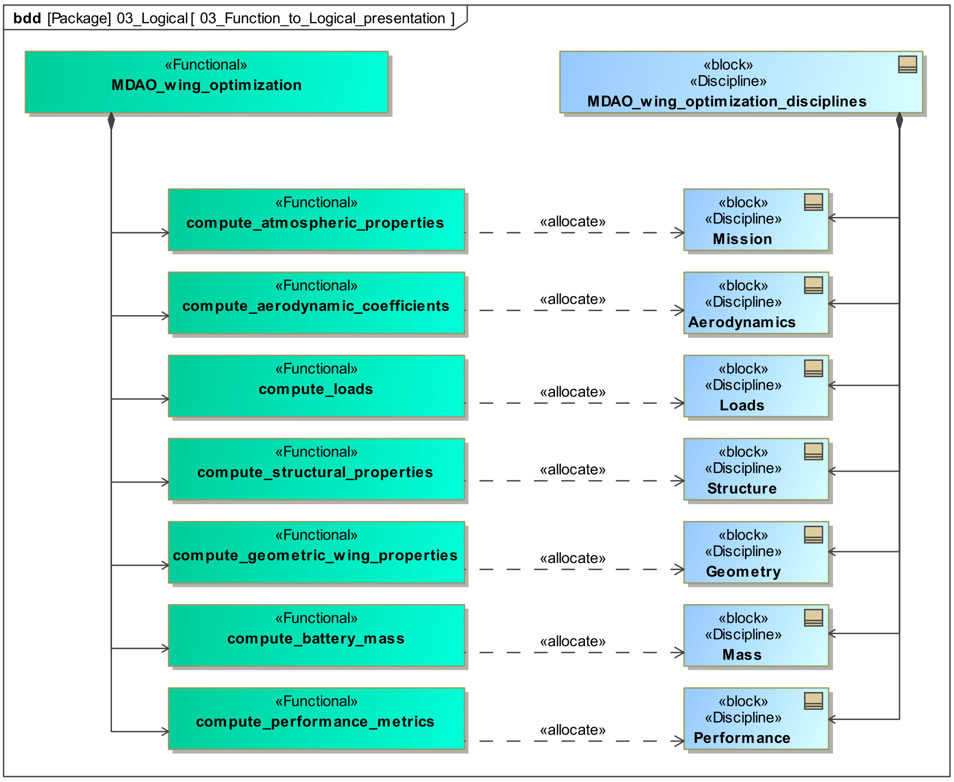


Figure 8 – Functions are allocated to the disciplines of the Logical layer modeled in a SysML block definition diagram.

3.2.3 Physical Layer

The physical layer provides all information about the computation procedures for each discipline identified in the logical layer. Unlike the logical layer, where disciplines are abstract groupings of functions, each discipline in the physical layer corresponds to specific computation methods and tools. In this context, the physical layer is tailored to GEMSEO, providing all necessary information to execute the computations for each discipline.
The disciplines of the physical layer are modeled as blocks in a SysML block definition diagram as presented in Figure 9. Each block is linked to the corresponding discipline block in the logical layer. Similar to the logical layer, each block in the physical layer is typed by a specific stereotype. In this layer, stereotypes indicate the computation method that will be used by GEMSEO. Additionally, each block inherits suitable attributes required for GEMSEO. For instance:

- **Analytic_Discipline:** The block includes analytic expressions needed for computation.
- **Tool_Discipline:** The block contains the definition of the specific tool to perform the disciplinary analysis in addition to a description or a script to run it.
- **Python_Discipline:** The block includes the names of the Python scripts needed for execution.

3.3 MDAO Framework and Disciplinary Analysis

The MDAO framework GEMSEO is used to run the multidisciplinary analysis and optimization problem. It facilitates the definition of an MDAO scenario in terms of design space, disciplines, objectives, and constraints, allowing for the selection of a formulation and the resolution of the related optimization problem.

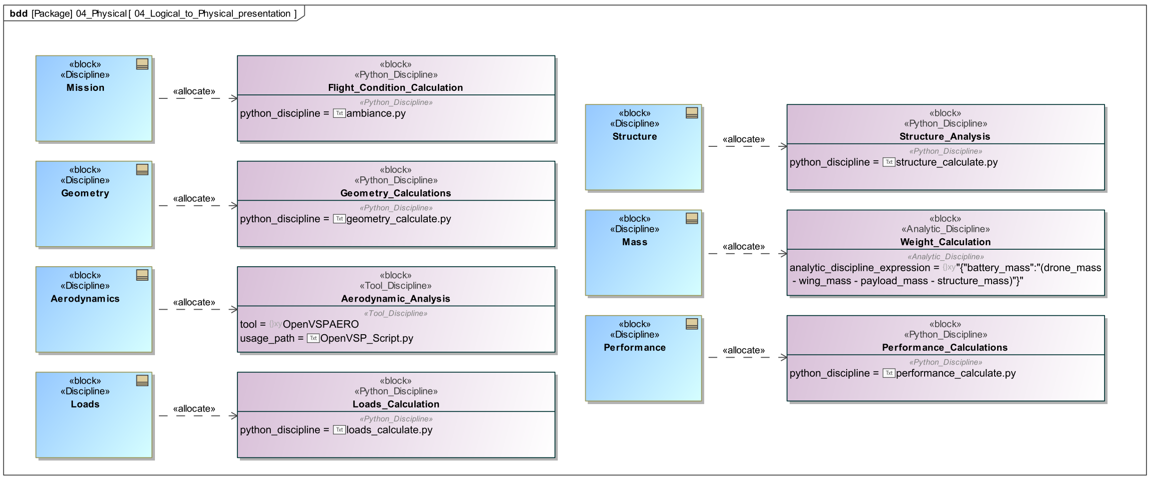


Figure 9 – Disciplines in the physical layer are specific to GEMSEO. They are modeled in a SysML block definition diagram and contain specific information on how the Disciplines will be executed.

The use case involves the wing design of the drone. Low fidelity simplified disciplinary models are used to analyse typical trade-offs during the design process. The existing geometry of the CPulse drone's fuselage is kept constant, while the wing parameters are varied to compute the objective.
The disciplinary analysis include 7 disciplines which consist of a mix of Python scripts, complete analysis tools and simple analytic expressions. Figure 10 illustrates the computational workflow with the eXtended Design Structure Matrix (XDSM), which provides a detailed representation of the data exchanges between the disciplines, inputs and outputs. In this analysis, we rely on the "frozen loads" methodology by assuming that the aerodynamic loads remain relatively constant and the structures displacements are negligible. This approach is well known for the conceptual design phase, as it removes the coupling of the computations [25] and is deemed sufficient within the scope of the present study. Therefore, the resulting optimization problem is not an MDO problem *per se*, as it has no strong coupling between the disciplines. The presented methodology could nevertheless be applied to more complex multidisciplinary problems with strong coupling.
Next, we provide a brief description of the internal computations of a few disciplines to detail the assumptions and the approach used to run the multidisciplinary analysis.

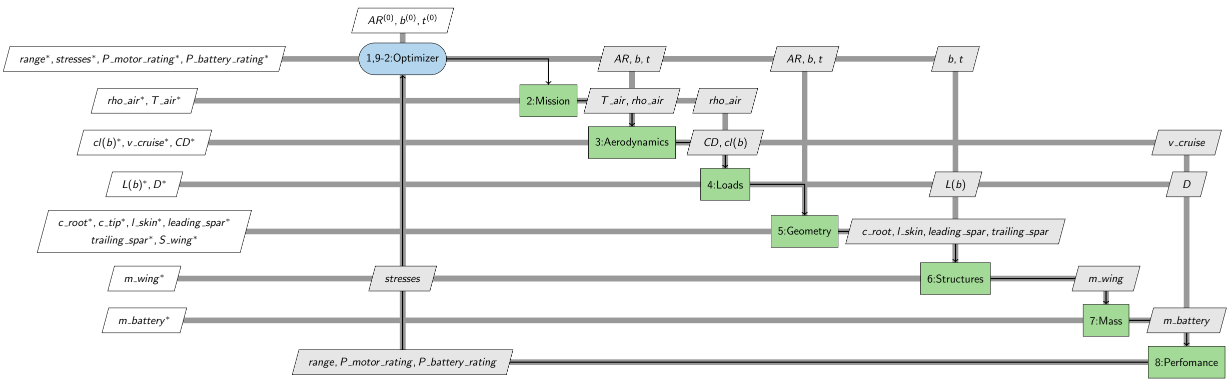
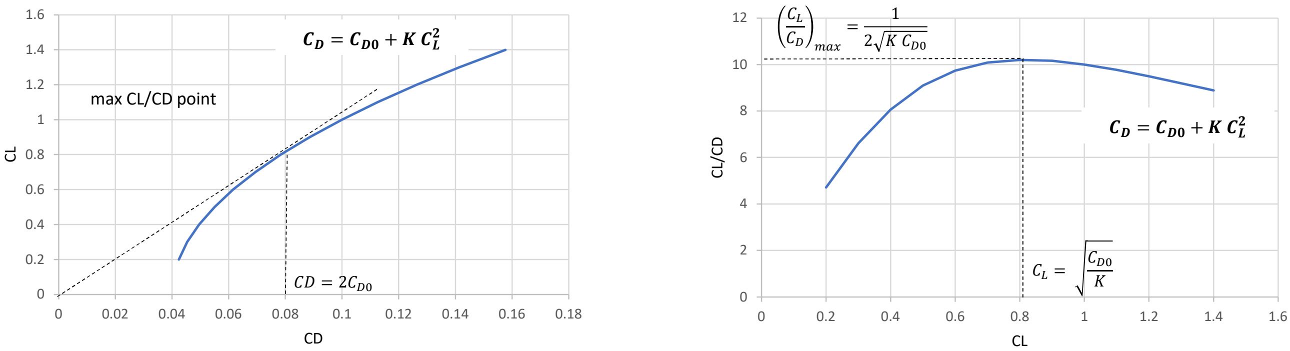


Figure 10 – The XDSM diagram to show the computational workflow of the wing design optimization for maximum range problem.

3.3.1 Aerodynamics Discipline

To model the aerodynamic performance of the varied wing geometry, the open source tool Open Vehicle Sketch Pad (OpenVSP) [24] is selected to run the analysis. It is largely used in academic studies for overall aircraft design since it is easy to use and integrates with Python or Matlab codes [26] [25]. It has an interface with the aerodynamic tool VSPAero which utilizes different methods such

as the Vortex Lattice Method (VLM) and empirical parasitic drag models that leverage existing geometric model components from OpenVSP with a variety of atmospheric, skin friction, and form factor equations for aerodynamic simulation. The lift induced drag is computed based on the OpenVSP panel VLM method for the designed 3D finite wing. The wing parasite drag is computed based on empirical formulas for 3D bodies considering a flat plate drag adapted to the wing wetted area, form factor and skin friction coefficient. Fuselage drag is estimated based on the equivalent Skin-Friction method [27], considering turbulent flow due to the fuselage length and the presence of the vertical motors arms.
The wing geometry, flow conditions, and fuselage parameters are used as inputs. A custom python script is developed to integrate OpenVSP into GEMSEO and automatically run the software in the loop. The lift distribution over the wing and the three main drag contributions (wing zero-lift drag, wing induced drag and fuselage zero-lift drag) are calculated to obtain the complete drag polar curves. The resulting zero-lift drag coefficient (C_{D0}) and induced drag coefficient (K) characteristic of a parabolic drag polar are then estimated for the specified wing parameters and passed as an output. An exemplary drag polar curve output from the aerodynamics discipline analysis in this study is presented in Figure 11. This curve is used to find the suitable flight point to fly a mission to minimize the drag, as will be explained in Section 4.2.1.



(a) Drag Polar - Max Range

(b) $C_{\mathrm{L}}/C_{\mathrm{D}}$ - Max Range

Figure 11 – An example for maximum range flight point (C_{L}) obtained from the drag polar curve output of the aerodynamics discipline in the current study.

3.3.2 Structure Discipline

The wing is structurally modeled as a mechanical beam composed of two spars that will withstand the stress it is subject to due to the bending caused by the lift force. The bending stress is given by $\sigma = \frac{c}{I}\int L\,\mathrm{d}x$, where L represents the lift force, c the spar length and I the moment of inertia around the neutral axis. With the wing's structural beam geometry, its weight can be estimated. The structural beam is considered as a straight beam, the dimensions of which are defined by the wing's tip chord. The weight of the drone's body including the propulsion system components is kept constant at 12 kg based on reference drones [28]. Therefore, only the wing sizing is performed. After the wing sizing for the structure beam and skin, to ensure that the designed wing can resist the applied loads, the total wing mass is calculated and passed as an output.

3.3.3 Performance Discipline

The CPulse is a vertical take-off and landing fixed wing UAV. Its flight mission includes the segments: vertical take-off, transition (from vertical to horizontal flight), climb, cruise, descent, transition (from horizontal to vertical flight) and landing. This mission is typical for the delivery of medical related items, such as blood samples, vaccines or critical medications, over remote or highly congested areas.
An energy-based approach is used to calculate the performance metrics such as the range. Given the required power for each flight segment, along with the time spent, we can obtain the energy consumed per segment. The battery is the only energy source available which is assumed to be at

full charge at the beginning of the flight. A a typical 20% reserve margin is selected for the battery load.
For the take-off segment, the double transition and landing, the power computation is based on the disk loading method [29] where 30 seconds per mission segments are assumed. The size and number of vertical propellers size are taken as inputs based on the drone's design. The Propulsive efficiency for a vertical flight is assumed to be equal to 40% due to the propeller size and the low vertical speed [30]. The Climb power computation is based on fixed propulsive characteristics assumptions (i.e. efficiency and maximum available power), in addition to the drag for the selected climbing speed. The rate of climb and time to climb to cruise altitude can then be computed using basic flight mechanics equations, allowing to finally compute the energy consumption in this segment. During the descent, the electrical motors can be turned off and no energy is assumed to be lost.
The remaining available energy is consumed during the cruise segment. The required power for a steady-level flight is computed to counter the drag for the specified cruise speed at the cruise altitude. The Horizontal flight propulsive efficiency is assumed to be 65% [30]. Provided the efficiency for the horizontal flight and the available battery energy, the cruise time and range are computed and passed as outputs.
The other disciplines include the *Mission* discipline, where the atmospheric conditions are computed at the cruise altitude, the *Loads* discipline, where the Lift and Drag forces are computed from the coefficients, the *Geometry* discipline, where several wing geometry parameters are calculated, in addition to the *Mass* discipline, where the battery mass is computed. In the next section, the optimization problems and results are discussed.

4. Results and Discussion

Two optimization problems for the wing design of the drone are considered to demonstrate an application case of the approach presented in Sections 2and 3. First we start by testing our performance module on a reference UAV WingCopter [28]. Next, we introduce the first optimization case study of wing design for maximum range. Then we check the performance of this design at off-design points by considering a post optimality study of range, paylaod and cruise speeds. The flexibility of the proposed approach is demonstrated through introducing new stakeholder requirements that will affect the design. The traceability of the implications of the requirements change on the MBSE model, the MDAO process definition, the implementation of the new problem and the resulting design are presented. Lastly, a few remarks on the use case are discussed.
The authors would like to point out that *the aim of the present study is not to present a use case on advanced drone design* but rather to demonstrate an aerospace related application case on the methodology introduced to bridge the gap between MBSE and MDAO.

4.1 Performance of a Reference Medical UAV WingCopter

To test our evaluation model and assumptions, we first calculate the performance metrics of a reference drone and compare the actual values to our model predictions. The Wingcopter UAV is also a vertical take-off and landing fixed wing UAV that performs similar missions to those considered for the CPulse drone. The drone's dimensions and performance values are obtained from [28]. Cruise drag, induced drag and zero-lift drag are estimated using the formulation in this study and presented in Section 3.3.1. The performance is computed using the method explained in section 3.3.3.
The aim of this validation is to ensure that the flight point (i.e. the flight at a given speed with a specific lift coefficient) and payload impacts are well captured in the range results.
The range for a given payload is computed with the performance model, along with the speed at which the range reaches its maximum (speed at maximum range). Initial runs indicate that the resulting range and flight time are closely aligned with the published values, as shown in Table 1. The performance model estimates similar ranges for the different payload values and captures the increase in selected cruise speed for higher payloads.

4.2 Use Case: Wing Design for Maximum Range

In this use case, the goal is to optimize the wing design to maximize flight range of the drone. The optimization problem involves varying wing design parameters such as wingspan, aspect ratio, and

Payload (kg)	Wingcopter (published) Time (mins)	Range (km)	Speed (m/s)	Performance model Time (mins)	Range (km)	Speed (m/s)
1	60	95	26	57	93	27
2	55	90	27	52	88	28
3	50	85	28	49	83	28
4	45	80	30	45	79	29
5	40	75	31	41	74	30

Table 1 – Wingcopter published vs performance model range values for a given payload.

taper ratio. These changes significantly affect the aerodynamics and structural weight (including battery weight), ultimately influencing the flight range. In the next section, the optimization problem is formulated, followed by a discussion of the optimization results and their rationale.

4.2.1 Optimization Problem Formulation

The design variables describe the geometric parameters of the wing, which employs a NACA 2412 airfoil. Constraints are imposed to maintain the wing spar stress below the material's tensile strength, and to ensure that the motor power rating during cruise (i.e. the motor power output divided by the maximum motor power output) and battery discharge power do not exceed their rated power outputs. Battery specific power is set to 340 W/kg and battery specific energy is selected as 160 Wh/kg with 20% reserve [31]. The horizontal flight motor maximum power rating is set to 1500 W [30].
The objective is to maximize the range while carrying a 5 kg payload, as specified in the system requirements in Figure 4. We use the Constrained Optimization by Linear Approximation (COBYLA) algorithm for the optimization. Convergence criteria are set with a threshold of 0.1% relative delta between iterations. The optimization problem is formulated as follows:

$$
\begin{array}{ll}
\text{maximize} & \text{Cruise range} \\
\text{with respect to :} & \\
& \text{Wing span (b)} \in [1.8, 3.5], \\
& \text{Aspect ratio (AR)} \in [5, 10], \\
& \text{Taper ratio} \in [0.5, 1], \\
\text{subjected to :} & \\
& \text{Tensile strength } \leq \text{Ultimate tensile strength (210 MPa)}, \\
& \text{Motor cruise rating } \leq 1, \\
& \text{Battery cruise rating } \leq 1.
\end{array}
\tag{1}
$$

The XDSM depicting the computational workflow for solving this optimization problem is presented in Figure 10. An additional note on the flying conditions that allow to maximize the range for a given wing is further discussed. The range of a propeller aircraft is maximized when the required thrust is minimized under steady-level flight conditions. Therefore, at any given moment where the weight equals the lift component in the vertical direction, the range is maximized by flying at maximum L/D. By describing C_D as a function of C_L using the drag polar output from the aerodynamic discipline (as shown in Figure 11, where $C_\mathrm{D} = C_\mathrm{D0} + KC_\mathrm{L}^2$), we can analytically determine the corresponding flying conditions (C_L and the speed). Hence, equation 2 is used to find the cruise speed:

$$V = \sqrt{\frac{2W}{\rho S C_\mathrm{L}}} = \sqrt{\frac{2W}{\rho S}\sqrt{\frac{K}{C_\mathrm{D0}}}}. \tag{2}$$

The lift and drag forces can then be computed and passed to the next discipline.

4.2.2 Optimization Results

The main factors influencing the optimization results are those affecting the flight range, i.e. the drag, the cruise speed and the battery weight. Theoretically, the range, flight time, cruise speed and battery

capacity should be maximized while drag is minimized. However, optimizing these variables simultaneously presents challenges. Increasing the speed typically increases the parasitic drag component, necessitating a balanced compromise to achieve optimal performance. Decreasing the speed without the penalty of increased induced drag requires a higher aspect ratio wing and a higher wingspan. These factors will contribute to a higher wing weight resulting in less battery weight. Since the parasitic drag increases with the square of the velocity, it is expected that the optimizer will move towards higher wingspan values in this case, due to the high drag penalty associated with higher velocity.
The optimization results are presented in Table 2. The optimizer has selected design variables that closely approach the upper limits for wingspan and aspect ratio, while selecting a lower taper ratio.

Variables	Initial	Optimized
Wingspan (m)	2	3.43
Aspect ratio	6	10
Taper ratio	0.8	0.5
L/D	9.4	13.4
Cruise speed (m/s)	26.9	19.3
Battery mass (kg)	7.1	6.7
Motor cruise rating	0.36	0.18
Battery cruise rating	0.45	0.23
Objective function: Range (km)	65.24	88.82

Table 2 – Comparison between initial and optimized values (first optimization problem).

Figure 12 illustrates the sensitivity of the design variables with respect to the objective function in scatter distribution plot. We see that initially the optimizer tested higher speed values, but quickly recognized the substantial drag penalty associated with increased cruise speed. Consequently, it adjusted towards lower speeds, settling around 20 m/s. Similarly, there was initial testing of low aspect ratios values near 6 and low wingspans values around 2 m. However, after a few iterations, the optimizer moved towards higher values for both variables to capitalize on improved aerodynamic efficiency at lower speeds. This strategic selection coupled with a low taper ratio aimed to optimize lift-to-drag ratio to achieve a longer range. Ultimately, the optimization converged at a range of 88 km.

4.2.3 Post Optimality Study: Payload, Range vs Cruise Speed

The wing design was developed through a single point optimization, i.e. it is designed to fly optimally under a specific flight condition with a defined payload. This particular scenario was chosen to demonstrate the integration of MBSE and MDAO for this design problem. However, in practical operations, drones do not always carry the same payload, or fly at the same speed (i.e. fly the same mission at the same flight point). Therefore, we conducted a post optimality study using the optimized wing parameters to evaluate the drone's performance under various off-design conditions, including different cruise speeds and payloads. The trade-off between range, cruise speed and payload is illustrated in Figure 13.
As expected from the single point optimization, the performance of the optimized wing decreases significantly at other operational points. For instance, the range declines sharply at higher cruise speeds (above 25m/s) across all payloads. Specifically, for a 5 kg payload, the range drops from an optimal 88 km to nearly 40 km (less than half of its optimal value). Additionally, with payloads under 5 kg, the optimal cruise speed is also lower, leading to longer flight times.

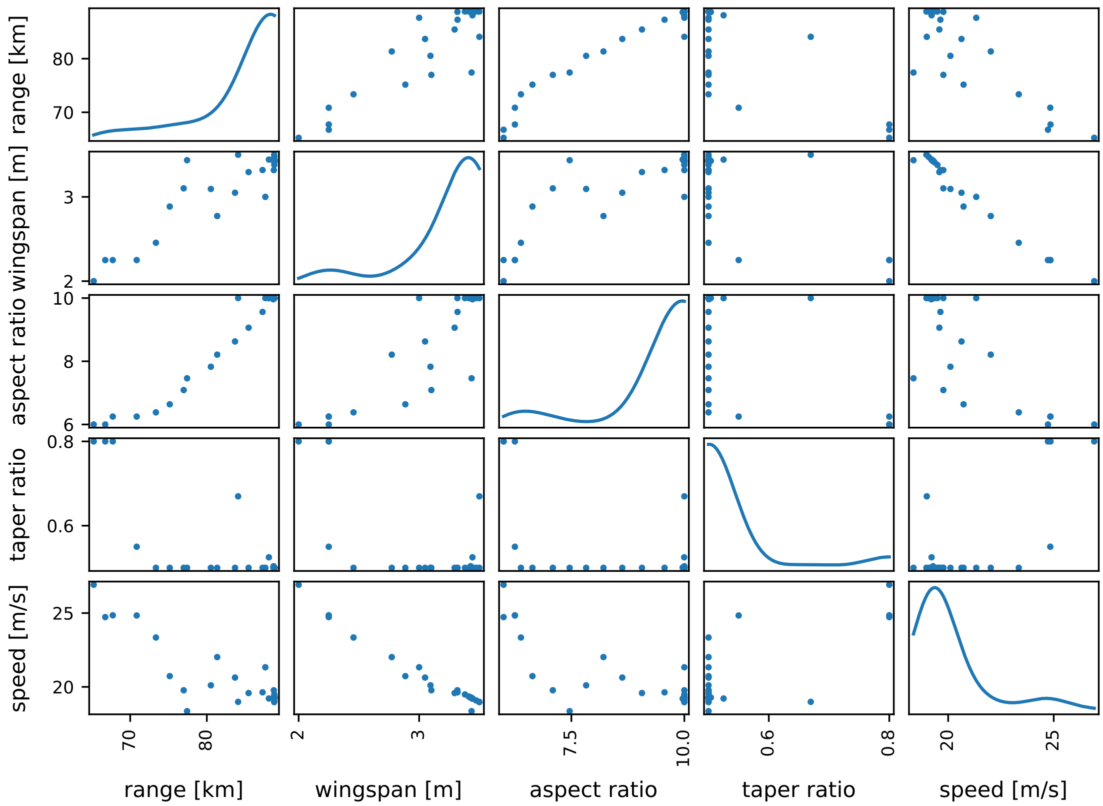


Figure 12 – Scatter matrix of variables of interest (cruise speed), design variables (aspect ratio, wing span and taper ratio) and objective function (range) for the wing design for the maximum range problem.

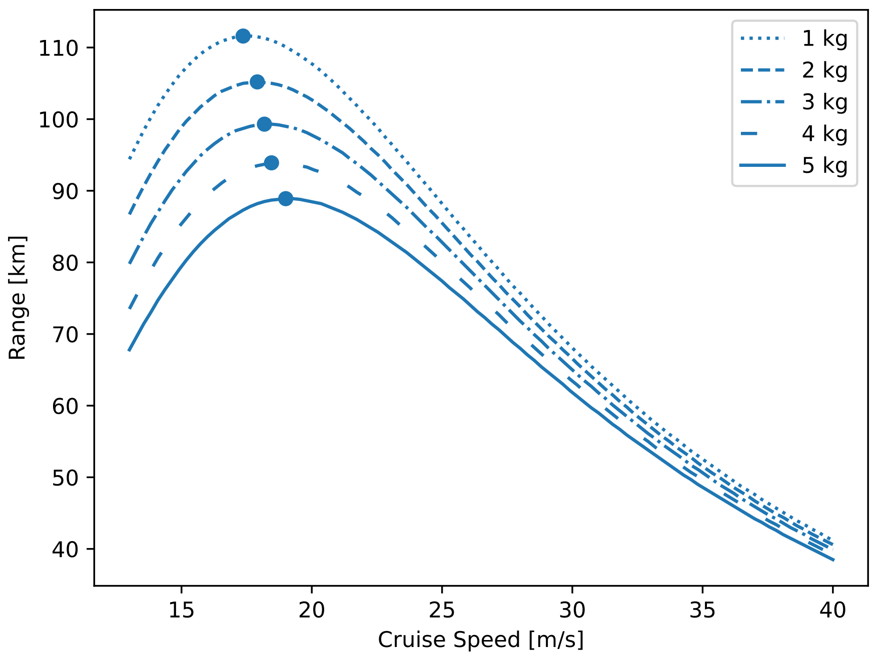


Figure 13 – Range as a function of cruise speed for various payloads.

4.3 Adaptation to Requirements Change: Wing Design for Minimum Flight Time

A key aspect of the proposed methodology is its flexibility with respect to model updates. In addition, it aims to enhance the traceability of the implications of requirements changes on the product design, facilitating an understanding of the trade-offs in relation to stakeholder needs.

For instance, consider a scenario where a requirement update is added to the product MBSE model. The objective for this scenario is to deliver an urgent item as fast as possible over a minimum distance within a maximum of 20 minutes. This reflects a common real-world scenario, such as transporting urgent samples in crowded cities or remote areas. Therefore, the goal now is to find the optimal wing and flying conditions that maximize the speed while ensuring a minimum range of 50 km. The updated requirements table is presented in Figure 14.

#	△ Name	Text
1	⊟ R SR System Requirements	
2	⊟ R SR.2 Mission	
3	⊟ R SR.2.4 Performance	
4	E SR.2.4.1 Range	The system shall have a range of at least 50 km.
5	E SR.2.4.5 Mass	The take-off mass of the UAV shall not exceed 25 kgs.
6	E SR.2.4.6 Payload	The UAV shall be capable of carrying a payload up to 5 kg.
7	R SR.2.4.8 Flight time	The payload shall be delivered in no longer than 20 minutes

Figure 14 – Updated performance requirements documented in the MBSE model of the drone.

4.3.1 Optimization Problem Formulation

Based on the updated requirements, a different optimization problem needs to be formulated, which is reflected in the MDAO process definition model. To deliver an item in less than 20 minutes, the average cruise speed must be at least 41.5 m/s for a distance of 50 km. Instead of constraining the speed, we formulate the problem to minimize the flight time by maximizing the cruise speed.

The new optimization problem shares some similarities with the previous one. However, unlike the previous problem, the flying conditions (C_L and speed) that maximize the speed over a 50 km range for a given wing can no longer be analytically defined. Therefore, C_L will be set as a design variable so that the optimizer can simultaneously find the best wing design and flight point (on the drag polar). This approach helps with the convergence behavior of the optimizer due to the direct relation between C_L and the flight performance equations. The resulting C_L will correspond to the aerodynamic performance of the wing at a specific angle of attack, which is checked during the aerodynamic analysis. In addition, a constraint on the minimum range is introduced to ensure that the 50 km distance requirement is satisfied. The new optimization problem is presented as follow:

$$
\begin{array}{ll}
\text{maximize} & \text{Cruise speed} \\
\text{with respect to :} & \\
 & \text{Wing span (b)} \in [1.8, 3.5], \\
 & \text{Aspect ratio (AR)} \in [5, 10], \\
 & \text{Taper ratio} \in [0.5, 1], \\
 & \text{Lift Coefficient } (C_L) \in [0.4, 1], \ \text{(find operational point)} \\
\text{subjected to :} & \\
 & \text{Tensile strength } \leq \text{Ultimate tensile strength (210 MPa)}, \\
 & \text{Range } \geq 50 \text{ km}, \\
 & \text{Motor cruise rating } \leq 1, \\
 & \text{Battery cruise rating } \leq 1.
\end{array}
\tag{3}
$$

The MDAO process model is updated to capture the new requirements over the flight time. The updated functional layer (see Section 3.2.1) is presented in Figure 15.

The technology assumptions (i.e. regarding battery capacity and specific power outputs of the motor and the battery), the selected optimization algorithm and the convergence criteria for this problem are the same as those previously described in Section 4.2.1 .The modified XDSM diagram is presented in Figure 18.

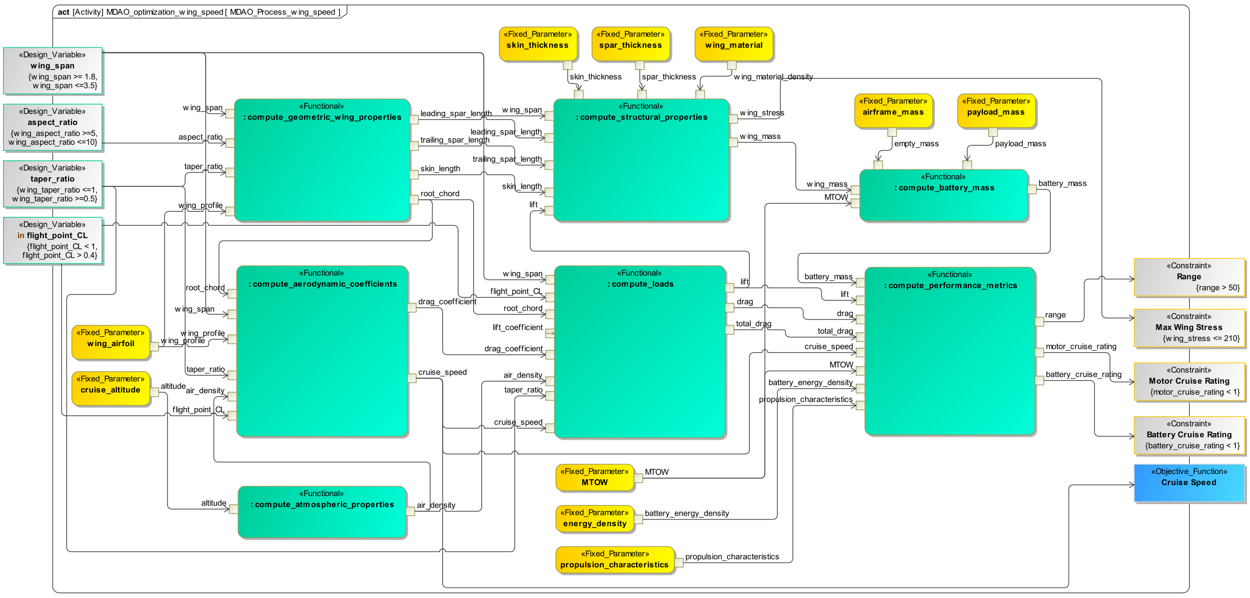


Figure 15 – Functional architecture of the MDAO process modeled in a SysML activity diagram for the second optimization case, wing design to maximize cruise speed.

4.3.2 Optimization Results

In this design problem, the objective is to maximize the cruise speed to deliver the payload as fast as possible. However, flying at high speeds results in increased parasite drag. Thus, given these cruise requirements and the high drag penalty associated with higher speeds, the optimal wing design will tend to be the one with least wetted area, to reduce the drag penalty.

The flying conditions (C_L and speed) that guarantee the maximum speed while ensuring a minimum range of 50 km are determined by setting the lift coefficient as design variables. This variable is controlled by the optimizer within a reasonable range, allowing the computation of the corresponding drag coefficient from the drag polar. The convergence behavior of the optimization process is shown in Figure 17b and the optimization results are presented in Table 3. As expected, the optimizer selects design variables near the lower bound for wingspan and taper ratio, and near the upper bound for the aspect ratio to reduce the wetted area.

Variables	Initial	Optimized
Wingspan (m)	2	1.8
Aspect ratio	6	10
Taper ratio	0.8	0.5
Flight point CL	0.7	0.61
L/D	9.2	6.6
Battery mass (kg)	7.1	7.7
Motor cruise rating	0.41	0.87
Battery cruise rating	0.51	1.0
Range (km)	64.0	50.3
Objective function: Cruise Speed (m/s)	29.65	45.53

Table 3 – Comparison between initial and optimized values (second optimization problem).

The scatter plot showing the sensitivity of the design variables with respect to the objective function is presented in Figure 19. We see that the optimizer tests a few wings with high span values, resulting in a high drag penalty from attempting to fly fast. It then moves towards wing geometries with a lower span value, close to the lower bound of 1.8m, and a higher aspect ratio. The resulting optimized cruise speed is more than twice the cruise speed in the previous case with a value of 45.5 m/s. As we discussed in Section 4.2.3, this is only a single point optimization, which means the benefit gained will come with a penalty on the overall performance of the drone at different design points that are not considered in this optimization. The trade off post optimality study is presented in Figure 20 where we see the that the maximum range that can be obtained for this design is close to 65km at a payload of 5 kg, which is significantly less (around 26%) than the 88km range obtained for the range optimization case. As the focus of the current study is to demonstrate bridging the gap between MBSE and MDAO, a multi objective optimization case that includes the different operation points is not considered since the use cases presented are sufficient to demonstrate the proposed methodology.

4.3.3 Requirements Traceability: Implications on Design Decisions

The need for employing early-stage engineering analysis for system design exploration in connection to scenarios and stakeholder needs is one of the key motivations for connecting MBSE and MDAO. In addition, the traceability of requirements' changes over the product development cycle and their impact on the design iterations are important for understanding improvements implemented over each iteration.

The use case presented in Section 4.3 in relation to Figure 16 shows how such traceability can be implemented using the proposed methodology. In Figure 14, the introduced new requirements are captured in the MBSE product model, where the requirements are categorized for each relevant subsystem and at the system level. Since the MDAO process definition model is connected to the product model, the requirements update are reflected on the MDAO model, which captures the new optimization objective along with relevant changes. This is implemented in the functional layer (see Section 3.2.1), logical layer (see Section 3.2.2), and physical layer (see Section 3.2.3), previously presented.

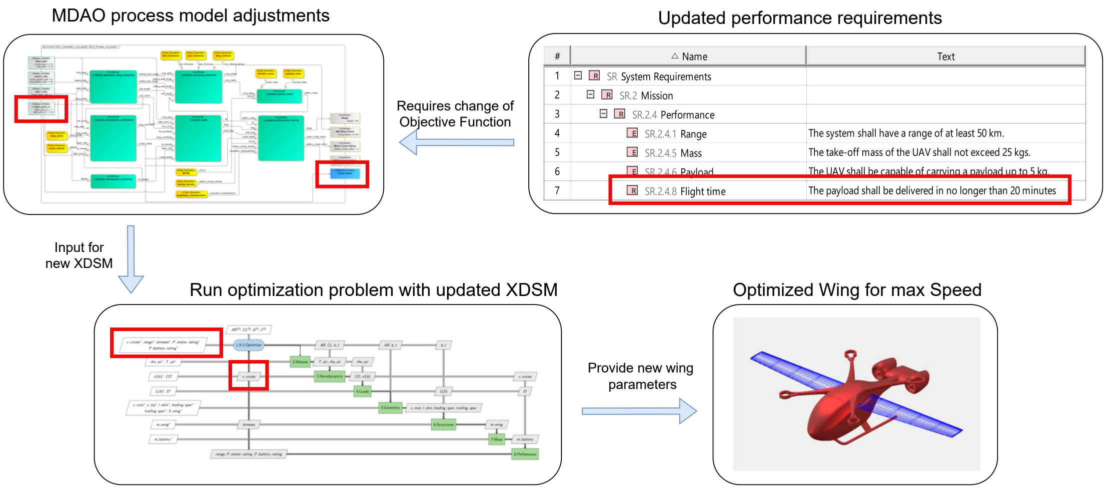


Figure 16 – Traceability of the implications of requirements on design decisions for early engineering analysis at the conceptual level.

The changes in the MBSE product model and its enabled MDAO process model drive the definition of the new computational workflow presented in the XDSM diagram (see figure 18) to solve the new optimization problem. Each design iteration can thus be traced back to the associated requirements update. For example, in the case of the extended requirement on the flight time introduced in Section 4.3, we can see that such update has resulted in different design decisions, such as reducing the wing span to achieve a higher speed. However, the new design also has a worse performance in terms of the range covered: even when flying at a lower speed, it could only reach a maximum range of around 65km, as shown in Figure 20. The resulting values can also be passed back to the MBSE model such that the information are captured and shared among development teams, to connect the product conception to the analysis and the implementation. By capturing the stakeholder needs in relation to early engineering analysis, the proposed methodology enables streamlined collaborative development with a consistent exchange of information flow among stakeholders, which is a major challenge in collaborative developments [5], at each iteration and over the product development life-cycle.

5. Conclusion

This research presents an application case of a streamlined MBSE-based methodology for the development of complex systems. It helps to bridge the gap between MBSE and MDAO allowing the integration of early-stage engineering analysis. The approach facilitates seamless digital continuity, ensuring that system requirements and design parameters are consistently linked throughout the product development cycle (see Sections 2.and 3.), using a single source of truth (i.e. the MBSE model), from conception to implementation.

One key benefit of this approach is enhanced communication among stakeholders and engineers as it establishes a common formalism for systems engineers, disciplinary domain experts and MDAO engineers to ensure consistent information flow among them.

Another key strength of this methodology is its flexibility in adapting to requirements' changes. The approach enables rapid adjustments to design process definition in response to updated requirements (see Section 4.3.3). This flexibility was demonstrated through the case study of the conceptual design of the CPulse medical drone, where the design objectives were shifted from maximizing cruise range to maximizing cruise speed (see Section 4.3), effectively showcasing the system's ability to adapt to new performance criteria.

Additionally, the methodology enhances the traceability of design decisions across the product development life-cycle. The connection between the MBSE product model and the MDAO process model allows for detailed tracking of how requirements' changes impact design iterations (Section 4.3.3). This traceability ensures that all design decisions are well-documented and can be reviewed in the context of their originating requirements, improving transparency and accountability in the development process.

This approach is poised to significantly enhance collaborative development efforts, particularly in complex engineering domains such as aerospace, by ensuring that all stakeholders have access to consistent, up-to-date information throughout the development cycle using a single source of truth. Future work will aim to further enhance the automation of this workflow and explore additional use cases to validate the methodology's broader applicability.

6. Acknowledgements

The authors would like to express their deepest gratitude to their colleagues at the Institute of Research Technology (IRT) Saint Exupéry for their invaluable support and collaboration throughout this research which is built upon the R-Evol project funded in the PIA framework (SGPI, ANR) and by the industrial members Airbus, Airbus Defence and Space, Liebherr, Capgemini, Expleo, CENAERO and CERFACS.

Special thanks to Michael Gerkens, Jörn Mühlenbrink and Klaas Birkenfeld from Capgemini Engineering, Germany, for their extended support. The authors also thank Mathieu Doucet and Julien Deborde from Capgemini Engineering, France, for their invaluable encouragement and support of this work. Additionally, the authors thank Jasper Bussemaker from the German Aerospace Center DLR Institute of Systems Architecture in Hamburg for his kind review of this article.

7. Contact Author Email Address

The authors would like to invite interested potential collaborators to contact the corresponding authors, mahmoud.fouda@capgemini.com and chifaa.dahik@capgemini.com.

8. Copyright Statement

References

[1] Ciampa, P. D. and Nagel, B., "Accelerating the Development of Complex Systems in Aeronautics via MBSE and MDAO: a Roadmap to Agility," *AIAA AVIATION 2021 FORUM*, 2021, p. 3056.

[2] Bussemaker, J., Boggero, L., and Nagel, B., "The AGILE 4.0 Project: MBSE to Support Cyber-Physical Collaborative Aircraft Development," *INCOSE International Symposium*, Vol. 33, 2023, pp. 163–182.

[3] Baalbergen, E., Vankan, J., Boggero, L., Bussemaker, J. H., Lefèbvre, T., Beijer, B., Bruggeman, A.-L., and Mandorino, M., "Advancing cross-organizational collaboration in aircraft development," *AIAA AVIATION 2022 Forum*, 2022, p. 4052.

[4] "Adaptive Vehicle Make," `https://www.darpa.mil/program/adaptive-vehicle-make/`, 2010, Accessed: 2024-06-02.

[5] Ciampa, P. D., La Rocca, G., and Nagel, B., "A MBSE approach to mdao systems for the development of complex products," *AIAA Aviation 2020 Forum*, 2020, p. 3150.

[6] Chaudemar, J.-C. and de Saqui-Sannes, P., "MBSE and MDAO for early validation of design decisions: a bibliography survey," *2021 IEEE International Systems Conference (SysCon)*, IEEE, 2021, pp. 1–8.

[7] Leserf, P., de Saqui-Sannes, P., Hugues, J., and Chaaban, K., "Architecture Optimization with SysML Modeling: A Case Study Using Variability," *Model-Driven Engineering and Software Development: Third International Conference, MODELSWARD 2015, Angers, France, February 9-11, 2015, Revised Selected Papers 3*, Vol. 580, 01 2016, pp. 311–327.

[8] Leserf, P., de Saqui-Sannes, P., and Hugues, J., "Trade-off analysis for SysML models using decision points and CSPs," *Software and Systems Modeling*, Vol. 18, 12 2019.

[9] Aïello, O., Kandel, D. S. D. R., Chaudemar, J.-C., Poitou, O., and de Saqui-Sannes, P., "Populating MBSE models from MDAO analysis," *2021 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE, 2021, pp. 1–8.

[10] Jeyaraj, A. K., Tabesh, N., and Liscouet-Hanke, S., "Connecting Model-based Systems Engineering and Multidisciplinary Design Analysis and Optimization for Aircraft Systems Architecting," *AIAA AVIATION 2021 FORUM*, 2021, p. 3077.

[11] Bussemaker, J. H., Ciampa, P. D., and Nagel, B., "System architecture design space exploration: An approach to modeling and optimization," *AIAA Aviation 2020 Forum*, 2020.

[12] Bussemaker, J., Boggero, L., and Ciampa, P. D., "From system architecting to system design and optimization: A link between MBSE and MDAO," *INCOSE International Symposium*, Vol. 32, 2022, pp. 343–359.

[13] J.H. Bussemaker, L. B. and Nagel, B., "System Architecture Design Space Exploration: Integration with Computational Environments and Efficient Optimization," *AIAA AVIATION 2024 FORUM*, July 2024.

[14] Swaminathan, R., Sarojini, D., and Hwang, J. T., "Integrating MBSE and MDO through an extended requirements-functional-logical-physical (rflp) framework," *AIAA AVIATION 2023 Forum*, 2023, p. 3908.

[15] Demachy, R., Guillen, M., Gazaix, A., Ambert, V., Giret, C., Jayez, L., and Brezillon, J., "A Model-Based System Engineering approach for Multi-Disciplinary Analysis and Design Optimisation processes definition," *CSD&M Proceedings*, 2022.

[16] DassaultSystemes, "Cameo Systems Modeler," `https://www.3ds.com/products/catia/no-magic/cameo-systems-modeler`, 5 2024, Accessed: 2024-06-03.

[17] DassaultSystemes, "Teamwork Cloud," ://www.3ds.com/products/catia/no-magic/teamwork-cloud, 2024, Accessed: 2024-06-03.

[18] Gallard, F., Vanaret, C., Guénot, D., Gachelin, V., Lafage, R., Pauwels, B., Barjhoux, P.-J., and Gazaix, A., "GEMS: a Python library for automation of multidisciplinary design optimization process generation," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018, p. 0657.

[19] Mahboob, A., Hussein, O., Willrodt, L., Thanga Raj Ashok Raaj, V. R., Takim, M. B., Spaans, A., Hasib, H., Birkenfeld, K., and Ilmenau Scientific Colloquium. Technische Universität Ilmenau ; 60 (Ilmenau) : 2023.09.04-08, "C.Pulse: an industrial demonstrator for a digital twin powered by MBSE for achieving digital continuity during the complete development process," *Engineering for a Changing World: Proceedings; 60th ISC, Ilmenau Scientific Colloquium, Technische Universität Ilmenau, September 04-08, 2023*, Nov 2023.

[20] Delligatti, L., *SysML distilled: A brief guide to the systems modeling language*, Pearson Education, 2014.

[21] Object Management Group, *OMG Systems Modeling Language (OMG SysML) Version 1.6*, 2019.

[22] Aleksandravičienė, A. and Morkevičius, A., "MagicGrid Book of Knowledge: A Practical Guide to Systems Modeling using MagicGrid," 2021.

[23] Casse, O., "2 - About Cameo Systems Modeler," *SysML in Action with Cameo Systems Modeler*, edited

by O. Casse, Elsevier, 2017, pp. 65–82.

[24] McDonald, R. A. and Gloudemans, J. R., “Open vehicle sketch pad: An open source parametric geometry and analysis tool for conceptual aircraft design,” *AIAA SciTech 2022 Forum*, 2022, p. 0004.

[25] Alonso Castilla, R., Lutz, F., Jézégou, J., and Bénard, E., “Wing Structural Model for Overall Aircraft Design of Distributed Electric Propulsion General Aviation and Regional Aircraft,” *Aerospace*, Vol. 9, No. 1, 2022.

[26] Herbst, S. and Staudenmaier, A., “Using OpenVSP in a Conceptual Aircraft Design Environment in MATLAB,” *Congress of the International Council of the Aeronautical Sciences*, 2016.

[27] Raymer, D., *Aircraft Design: A Conceptual Approach*, AIAA, 2018.

[28] Wingcopter, “Technical Details : Wingcopter 198 Delivery Variant,” `https://wingcopter.com/wp-content/uploads/2022/06/Specsheet-Wingcopter-198.pdf`, 2021, [Accessed 11-06-2024].

[29] Melo, S. P., Cerdas, F., Barke, A., Thies, C., Spengler, T. S., and Herrmann, C., “Life Cycle Engineering of future aircraft systems: the case of eVTOL vehicles,” *Procedia CIRP*, Vol. 90, 2020, pp. 297–302, 27th CIRP Life Cycle Engineering Conference (LCE2020) Advancing Life Cycle Engineering : from technological eco-efficiency to technology that supports a world that meets the development goals and the absolute sustainability.

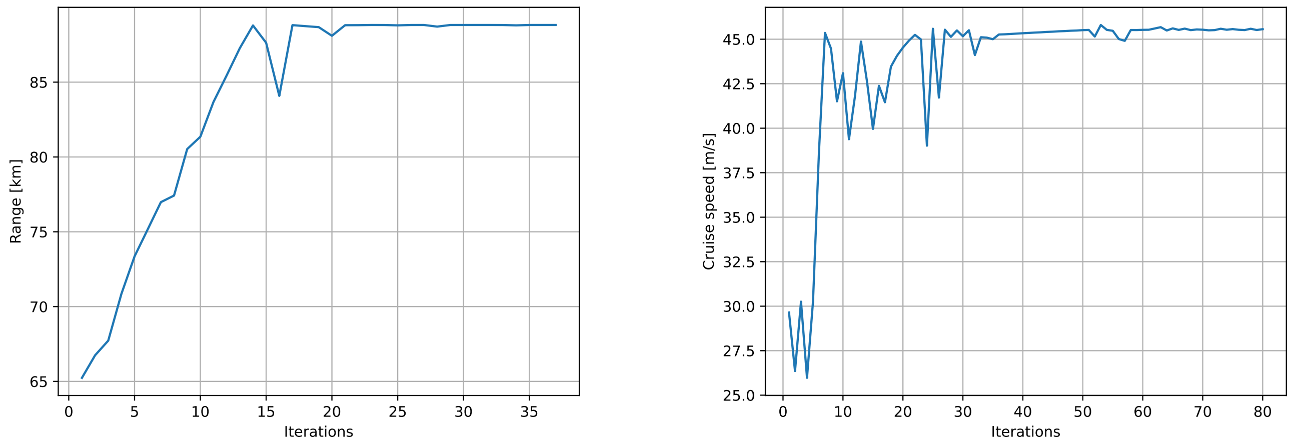
[30] Gong, A. and Verstraete, D., “Development of a dynamic propulsion model for electric UAVs,” *Proceedings of the 7th Asia-Pacific International Symposium on Aerospace Technology, Cairns, Australia*, 2015, pp. 25–27.

[31] Panasonic, “NCR-18650B : Lithium-ion / NNP + HRL technology,” `https://www.imrbatteries.com/content/panasonic_ncr18650b-2.pdf`, [Accessed 11-06-2024].

Appendix

8.1 Convergence Behavior

The convergence behavior is presented in Figure 17 for both optimization problems. For the first case of the wing design to maximize range, the solution converges after 37 iterations (37 minutes on typical notebook PC), as shown in Figure 17a). For the second case of the wing design to maximize cruise speed, the solution is reached after 80 iterations (1 hour and 26 minutes on typical notebook PC, as shown in Figure 17b).



(a) Evolution of the range (objective function) with respect to the number of iterations.

(b) Evolution of the cruise speed (objective function) with respect to the number of iterations.

Figure 17 – Convergence behavior for both optimization problems.

8.2 Requirements Change: Wing Design for Maximum Cruise Speed

8.2.1 Computational Workflow

The updated eXtended Design Structure Matrix (XDSM) diagram is presented in Figure 18. It describes the the computational workflow that is associated with the new optimization problem of the wing design to minimize the flight time by maximizing the cruise speed.

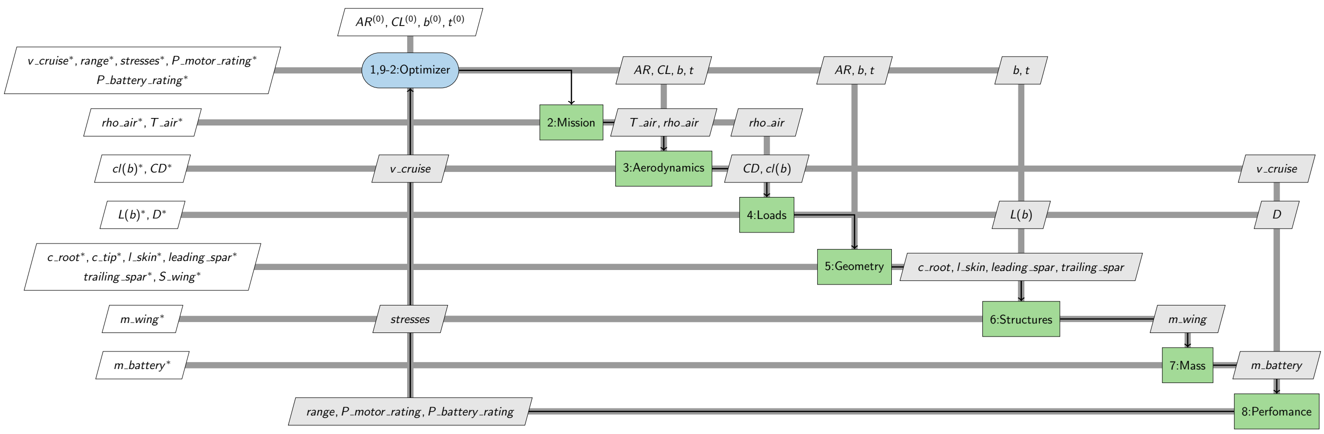


Figure 18 – XDSM diagram of the Wing design for maximum cruise speed optimization problem.

8.2.2 Optimization results: Sensitivity of Design Variables with respect to the Objective

The scatter plot presented in Figure 19 shows the sensitivity analysis between the design variables (wing span, aspect ratio and taper ratio) with respect to the objective function (i.e. the cruise speed) while ensuring that the flight range is at least 50 km. We can see that the optimizer moves towards a combination of design variables that results in a lower wet area of the wing to allow flying at higher speeds while minimizing the associated penalty of the increased drag.

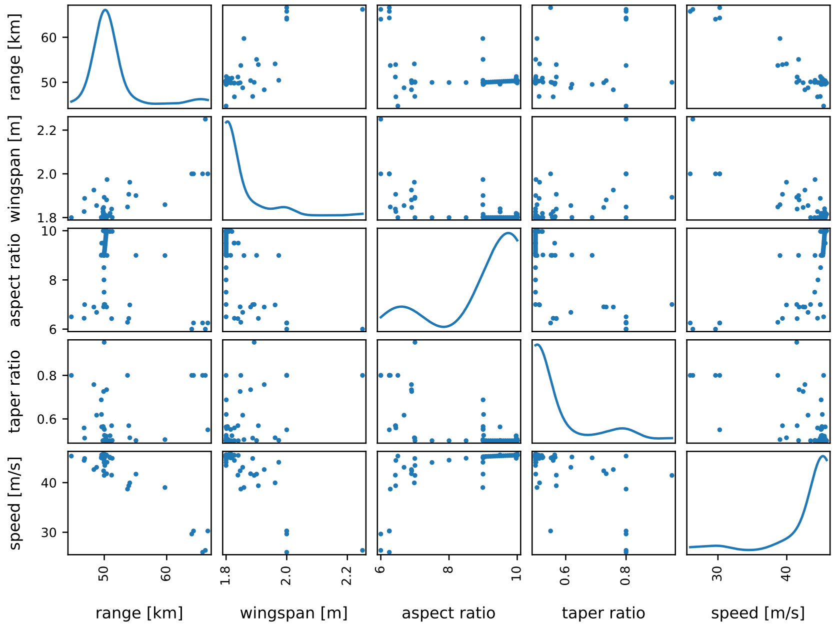


Figure 19 – Scatter matrix of variables of interest (range), design variables (aspect ratio, wing span and taper ratio) and objective (cruise speed) for the wing design for maximum speed problem.

8.2.3 Post Optimality Study: Range, Speed and Payload

The trade-off study between range, speed and payload is presented in Figure 20. The single point optimization performed allows flying at higher speeds to be able to deliver the payload in less than 20 minutes over a distance of 50 km. The gained benefit comes with penalty at other off-design points, as we see that flying at a lower speed (around 30 m/s) with this particular design, with a payload of 5 kg, can only result in a maximum range of 65 km, which is significantly lower than the 88 km range obtained in the previous design problem (see Section 4.2).

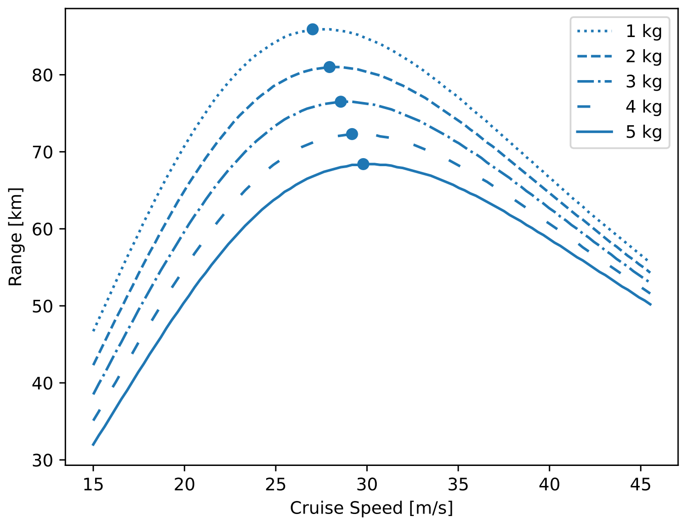


Figure 20 – Range as a function of cruise speed for various payloads.