UNIVERSITÀ DI PARMA

UNIVERSITÀ DEGLI STUDI DI PARMA

DOTTORATO DI RICERCA IN
INGEGNERIA INDUSTRIALE
CICLO XXXIV

Towards the definition of an engineering method to support the design for assembly and installation of aircraft systems at the conceptual design stage

Coordinatore:
Chiar.mo Prof. Royer Carfagni Gianni

Tutore:
Chiar.mo Prof. Favi Claudio

Dottorando:
Formentini Giovanni

Anni Accademici 2018/2019 – 2020/2021

Università degli Studi di Parma
Dipartimento di Ingegneria e Architettura
Parco Area delle Scienze 181/A – 43124 – Parma, Italy

ACKNOWLEDGEMENTS

During my Ph.D. path, I have received a great deal of support and assistance and now it is time to thank the people who made this accomplishment possible.

First, I would like to express my deepest appreciation to my supervisor Prof. Claudio Favi. You provided me with encouragement and patience throughout these years, being an outstanding mentor to me.

Claudio, Thank you.

I would like to extend my sincere thanks to Eng. Francois Bouissiere, Eng. Claude Cuiller and Eng. Pierre-Eric Dereux for their help. Your extensive knowledge of the aeronautical field and your practical suggestions have been always an inspiration for my research.

Many thanks should also go to my colleagues in Parma. You guys made my days less foggy!

To all my friends spread around the globe, even if times has not allowed us to see much, you have played a major role in these years, always supporting me whenever I needed it. Cheers!

To my sister Laura; To my brothers Federico and Pierluigi; To my whole family, you played an important role during these years. This achievement is also yours.

Infine, a mio padre e mia madre che avete reso possible tutto questo.

Grazie Mamma, Grazie Papà.

[page intentionally left blank]

When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind.

— Lord Kelvin

[page intentionally left blank]

LIST OF CONTENTS

LIST OF ABBREVIATION

Boothroyd & Dewhurst	B&D
colour vision deficiency	CVD
conceptual Bill of Material	cBoM
Conceptual Design for Assembly	CDfA
Concurrent Engineer	CE
Design Correlation Matrix	DCM
Design for Assembly	DFA
Design for Installation	DfI
Design for Manufacturing	DFM
Design for Manufacturing and Assembly	DFMA
Design for X	DfX
Design Guideline	DG
Element Vector	EV
Final Assembly Line	FAL
Functional Efficiency	FE
impact vector	IVec
Knowledge Engineering	KE
Multi-Attribute-Decision-Making	MADM
New Product Development Process	nPDP
One-Factor-At-Time	OFTA
Requirement, Functional, Logical, and Physical	RFLP

root mean square	RMS
sensitivity analysis	SA
simplified Digital Mock Up	sDMU
Systems Engineering	SE
Technique for Order of Preference by Similarity to Ideal Solution	TOPSIS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The design of new products requires a complex set of activities which need to be carried out in a systematic manner. In the past years, great focus has been put on optimizing the process in terms of cost, time, and quality. Several tools and methods have been developed to support designers and engineers during the whole product development process. Moreover, it is well-know that optimizing products during early design phases lead to a cost reduction of around 70% of the overall product cost.

Nowadays, a great number of methodologies are available in the literature to optimize the design of products. Among all, Design for Manufacturing and Assembly methodologies aim at optimizing the product manufacturing and assembly phase, which can impact up to 40% of the overall product cost. For complex engineering systems, such as aircraft and aerospace equipment, this influence is even more significant.

To date, the design of aerospace civil products is consolidated, and the optimization process is required to meet market demands. The design of these products is tackled from a system engineering point of view, assuming the aircraft product architecture is already given. This approach limits the potential benefits that can be introduced by developing novel methods at the conceptual design phase.

The aim of this research work is to provide a method, called Conceptual Design for Assembly, to assess the assembly and installation of complex systems into a product and to provide redesign guidelines to help designers and engineers reduce its complexity. The approach is based on a specific framework that allows for the gathering of product architecture data. Then, using a mathematical model, assembly indexes are derived. The approach also allows for the collection of technical information from the production and assembly departments using structured tables, facilitating the transition from implicit to explicit knowledge.

The method is applicable at early design phases, such as the Conceptual Design phase. The CDfA identifies major issues in the system under investigation and provides a practical tool for implementing redesign actions in structured steps. Starting from the functional analysis of the product, the method guides the user through the definition of the mathematical model to assess the product architecture of the system analysed. Once the assessment is completed, it is possible to obtain design guidelines to improve the product architecture.

The Conceptual Design for Assembly method was tested on two real industrial case studies: the Nose-Fuselage and the Cabin systems of a commercial aircraft. Results showed that improvements were obtained for both architectures applying redesign guidelines.

Finally, a software mock-up that can be coupled to an existing CAD program is proposed. The software seeks to make the approach more suitable to be adopted in daily engineering practices avoiding time-consuming and prone to error activities.

The developed method and the integrated platform enable designers and engineers to evaluate and improve the installation of complex systems into a product architecture, reducing the intrinsic uncertainty of early design phases. The Conceptual Design for Assembly method can be extended to investigate the product architecture of complex products in other industrial fields.

1. INTRODUCTION

In the industrial world, innovations and improvements are the key aspect to remain competitive in business. One of the main goals when improving products is to reduce their overall costs. This can be accomplished by modifying the product itself, for instance reducing the number of parts, or changing the manufacturing process to achieve better performances.

It is well known that for big and complex products, the assembly phase may account for more than 40% of the total cost. Furthermore, design decisions performed at the Conceptual and Embodiment Design phases influence around the 70% of the entire product cost. However, the information detail at these phases is very low, therefore the level of uncertainty is high.

Novel methodologies and tools are required to give designers a basis for guiding them through modifications during the early stages of design, decreasing the total risk of making trivial changes.

Whereas already affirmed design methods and software tools exist to support designers during the improvement of product assemblability, none of them are applicable for studying complex products at early design phases.

The proposed research thesis aims at filling this gap, providing a systematic method to assess and modify complex product architecture during early design phases. The purpose is to provide a comprehensive framework for performing assembly analysis with a low level of detail. The proposed approach was tested in the aeronautical field; however, it can be easily extended to other products.

The method is composed of seven steps for the assessment of complex product architecture and five steps to provide re-design suggestions. In addition, a software prototype is offered to assist users with little or no prior knowledge in applying the procedure and performing analysis while minimizing potential mistakes.

1.1 CONTEXT OF THE RESEARCH WORK

The aerospace industry experienced fast innovation in the last century, especially for military reasons, while the commercial sector is tackling the market with consolidated solutions. Indeed, modifications and improvements are limited from a product point of view due to several reasons such as:

- **Regulation**
  any modification needs to undergo through a long and complex acceptance process
- **Safety**
  it is necessary to guarantee and be compliant to strict safety requirements
- **Cost**
  aircraft is characterized by long lead time. Any modification will further increase the lead time, increasing costs

On the other hand, the aerospace commercial sector is expanding, and despite the recent outbreak of COVID-19 pandemic situation, the global demand of aircraft is increasing. New requirements such as the need to reduce the overall aircraft lead-time, the necessity to improve assembly features reducing manpower use and the demand of solutions which can be personalized according to airline company requirements, are challenging the aircraft concept from the root.

It is required to provide solutions which substantially modify the aircraft concept. Nowadays, there is a limited number of tools and methods developed with specific optimization (i.e., the assembly line). However, these solutions are at the end of the pipeline and do not allow a definition of new concepts with very important benefits.

Methods called Design for Manufacturing and Assembly (DFMA) are born with the aim to seek simplification of the manufacturing and assembly process for cost reduction of a given component. The DFMA is a family of methods belonging to the Design for X (DfX) category. DfX methodologies are used to improve specific aspects of the product under development. The X is generally substituted with the optimization goal, and these methodologies are used to support the new Product Development Process (nPDP) which relies on a Concurrent Engineer (CE) approach (Kusar, et al., 2004). Among the several methods created for this purpose, two approaches have been mainly used both in academia and industry:

i) Boothroyd & Dewhurst (B&D) (Boothroyd, 1987) and ii) Lucas method (Ltd, 1993).

DFMA methods have been widely used in several fields, among which the aerospace sector. Assemblability of aircraft can be tackled from two different point of view: i) aircraft component assembly or ii) Final Assembly Line optimization. The former focuses on studying components required to create the aircraft and how they interact with the final structure, while the ladder studies the aircraft assembly sequence. The B&D DFMA approach was used to challenge aircraft components by several authors (Herrera, 1997) (Herrera, 1998) (Swift & Brown, 2003) (Barbosa & Carvalho, 2014); while other authors (Mas, et al., 2013) (Gomez, et al., 2016) focused their studies on the optimization of the aircraft Final Assembly Line.

Most of the research done in this field does not question the product architecture of the aircraft system. In fact, all the optimization analysis need data gathered during the late stages of design. As a result, the hypothetical benefit in terms of reduction the overall product cost is highly affected by the need of a redesign of the product.

To overcome this problem, it is necessary to move the analysis from late design phases to early ones. According to Pahl and Beitz (Pahl & Beitz, 2013) the product development process can be divided into three main phases: i) Conceptual design phase, ii) Embodiment design phase, and iii) Detail design phase. At each phase, information availability and uncertainty vary. At the Conceptual design phase, the information granularity is very low (Figure 1) and classical DFMA approaches cannot be applied, since they require information available at late design phase (e.g., number of screws, etc.).

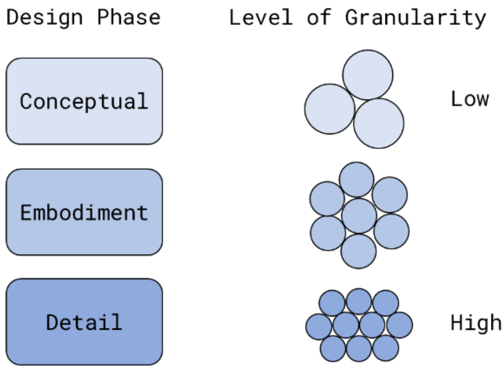


Figure 1 - Correlation among Design Phases and Level of Granularity

Some methods are available in literature to perform analysis with conceptual information. For example, the method of the functional basis and module heuristics (Stone, et al., 2004) can be used to support designers during the early stages of the design process. The functional basis is used to derive a functional model of a product in a standard formalism and the module heuristics are applied to the functional model to identify a modular product architecture (Dahmus, et al., 2000). These methods allow creating products with great assembly performances, starting from the identified modular structure. However, they seem not to be applicable to big and complex products, since are not able to handle the several constraints which characterize these types of products. To the best of the author knowledge, to date there are no methods available in literature aiming at optimizing the assembly phase of complex products at the conceptual design phase.

The aim of this research work is to provide a structured method to tackle the assemblability of complex products at the conceptual design phase. The approach may be used to offer a numerical evaluation of product architectures and to help designers through the redesign process by identifying changes that need be made to enhance the product architecture's assemblability.

1.2 TECHNICAL AND SCIENTIFIC RESEARCH OBJECTIVES

The proposed research work provides a method called Conceptual Design for Assembly, which aims to provide a structured methodology to numerically assess assembly aspects of complex products at the conceptual design phase. The development of the method required the accomplishment of the following challenges:

Engineering knowledge DFA formalization

The main challenge working at the conceptual phase is the quantity of information available which is low and characterized by low granularity.

In the conceptual phase, collected information is represented using schemes that encapsulate functions which are required by the product to operate. Low granularity information does not provide numerical data to perform mathematical analysis. To numerically assess assembly complexities at the conceptual phase, it is necessary to develop a technique to parameterize this low information, translating it into numerical values. In other words, it is necessary to find suitable parameters which can be computed with conceptual information and used to estimate assembly complexities.

During the conceptual phase, engineering knowledge may be gathered in the form of requirements and standards. Indeed, the formalization of the knowledge available at this phase, especially for assembly aspects, is demanding. The aim is to provide a method to: i) capture the engineering knowledge of the manufacturing and assembly department, in the aeronautical field, and ii) translate the acquired knowledge into numerical parameters which can be assessed with conceptual information.

The proposed methodology makes use of a modified Functional Derivation approach that allows users to deeply map functions inside complex systems and obtain conceptual parameters. Moreover, engineering knowledge is acquired and used by the adoption of *Scoring Matrices*, which are matrices collecting manufacturing and assembly knowledge in the form of strings and numbers.

Data Collection Framework formalization

In order to provide numerical evaluations, it is necessary to switch from a functional representation to a numerical one. Furthermore, the numerical data must be organized within a specific framework. The framework must account for the relative relevance of each parameter in the overall assembly rating.

The goal is to find a way to visualize functional information and obtain assembly parameters (i.e., numerical parameters). Moreover, it is necessary to design a method for formalizing a framework to collect these numerical parameters. This objective was accomplished through the development of two tools: i) a simplified digital mock-up and ii) a hierarchical structure (i.e., *conceptual* Bill of Material). The simplified Digital Mock Up (sDMU) is a graphical and geometrical representation of a specific product architecture related to the system of interest (i.e. aircraft). The sDMU is composed of 3D geometrical items such as boxes, cylinder, etc. that describe how functions and interfaces are connected. It enriches the product architecture by adding geometrical information. The *conceptual* Bill of Material is the core of the methodology. It is composed of Attributes, Domains and Levels. It is used to divide the product analysed into sub-systems. The sub-division provides two advantages:

- It allows analysing assembly properties of sub-systems inside the product
- It enables for the analysis of data from various design phases. Indeed, information from different design phases (e.g., conceptual, embodiment and detailed) can be mixed using levels

Using these two tools, it is possible to obtain numerical parameters from conceptual information and collect them following a specific structure.

Definition of the mathematical model to perform assembly/installation analysis

Numerical data require the definition of a mathematical model to be processed. The assembly performance of the product analysed is computed with the help of assembly indices. The computation of these indices necessitates the development of a mathematical model capable of processing numerical data gathered within the hierarchical structure.

The aim is to provide a mathematical model based on the hierarchical structure that can make use of data present in levels, domains and attributes to provide single assembly indices. The model created enable users to obtain indices assessing the assemblability of modules, interfaces and interfaces on modules. These indices are used to evaluate the goodness of the product architecture in terms of assembly complexity. Since the mathematical model is strictly related to the hierarchical structure, if the hierarchical structure is changed, the mathematical model needs to be changed as well. To further support the definition of the mathematical model, sensitivity analysis approaches are used to determine the effect of the hierarchical structure on the mathematical model created.

Definition of a method to collect and identify redesign guidelines to improve the product architecture

The assessment of product architecture requires to be supported by methods which guide designers through the modification of the architecture to improve its performance. The assessment phase can spot assembly criticalities, indicating where problems lie. It can provide both general and specific information, through the identification of a critical aspect (i.e., a domain of interest) or a precise task/activity (i.e., a module or an interface). However, it does not provide suggestions about the design changes that can be implemented. The definition of redesign guidelines can support designers through the modification of the product architecture. Redesign guidelines are necessary to understand which action should be implemented to reduce the complexity of a given component.

The derivation and selection of redesign guidelines is accomplished defining a mathematical model which make use of CDfA assessment results. Redesign guidelines are derived from senior expertise and industrial databases. Then, the impact that each design guideline has on the CDfA attributes is assessed. The outcome is a matrix, called Design Correlation Matrix, which allows linking guidelines impacts with CDfA data. Finally, using the mathematical algorithm, a single vector called Impact Vector is obtained for each design guideline. The redesign guideline with the highest impact represents the one that need to be implemented first to reduce the assembly complexity of the product architecture.

1.3 PROGRESS BEYOND THE STATE OF THE ART

Despite the extended literature about DFMA methods in product design available, almost all works focus on the optimization of simple products, meaning easy-to-handle with a short lead-time product. Few attempts have been done to adapt already existing DFMA methods to deal with complex products (Retolaza, et al., 2021) or to provide a framework to assess assembly complexities (Samy & ElMaraghy, 2010) (Alkan, et al., 2017). The proposed methods lack tools supporting the use-phase and real industrial case-study to support their application. Moreover, they cannot be applied on the study of aerospace products. The main reason is the impossibility to provide assessment at the conceptual design phase. Complex products are characterized by long lead times and a redesign of them will lead to a great increase of the overall product cost.

The shift of DFMA approaches from detailed to conceptual design phases requires a change in the DFMA paradigm, shifting from a systematic approach across the whole product development process to an early design optimization procedure. Few papers are presented in literature which tried to tackle this aspect. Stone (Stone, et al., 2000) proposed a method to support designers during the early phases of the design process. The approach is based on the use of two concepts: functional basis and heuristic methods. Functional basis is used to derive the functional model of the product in a standard language. The model heuristics are applied to the functional model to derive the modular product architecture. These methods can provide products with a minimum number of parts that might be less that typical DFMA methods (e.g., B&D DFMA). However, the method is not able to handle an elevate number of data, which is the typical case for aerospace products. A more advance approach was proposed by the author of this thesis (Formentini, et al., 2022.a). The method is composed of several steps which are used to guide the user through the definition and analysis of product architectures. The method was specifically developed for the aerospace sector; indeed, the proposed case study is the nose-fuselage of a civil aircraft. However, it is a preliminary work, and it lacks several aspects such as a structured method to provide design guidelines to users, a rigorous definition of the framework used to collect and process data, etc. Other works of the same authors are available in literature to overcome these drawbacks. For instance, Favi (Favi, et al., 2019) presented an improved version of the approach proposed by the author of this thesis. Formentini (Formentini, et al., 2021.a) proposed a method to derive design guidelines at the conceptual design phase to support the re-design of product architecture and the same author (Formentini, et al., 2021.b) performed a sensitivity analysis on the framework provided to study the assemblability of the cabin of a civil aircraft. However, a complete method to assess assembly complexity of complex products at the conceptual design phase and guide users through the redesign process is still missing. Summarizing, from the literature study, the current gaps identified are:

- Lack of methods to numerically assess assemblability of complex products at the conceptual design phase
- Lack of methods to guide designers through the optimization of product architecture
- Lack of tools to support the application of the method

Industrial Progress

To date, the application of methodologies for assessing complex product architecture has been limited in technical aircraft development departments. The effects of a poor product design are identified when the product reaches the manufacturing and assembly department, which most likely will require a redesign of the product, increasing the overall lead-time, hence costs. Moreover, today industries are facing a great challenge with knowledge management. Key information is held by few people inside the company, which increases the risk of losing it (e.g., changing job, retirement, etc.). This information is, moreover, hardly transferable due to the lack of structured methods applicable.

The possibility to have methods and tools which can consider manufacturing and assembly problems at the very beginning of the product development process lead to several industrial advantages:

- To collect manufacturing and assembly industrial key knowledge, which is the main asset for industries, making it available to anyone
- To remove barriers between manufacturing and assembly departments, in accordance with concurrent engineering practice
- To reduce developing and production costs, hence lead-time
- To optimize specific design aspects linked to assembly operation, such as ergonomic aspects
- To provide a software tool, implementable in programs already used by the firm (i.e., CAD software), to support the product development process

Scientific Progress

The proposed work placed itself in contrast with the current research trends in the industrial field, especially in the aircraft systems optimization Many authors are tackling the problem of optimizing product architectures with respect to several parameters, however they are looking at the problem as a mathematical optimization problem (i.e., definition of optimization function), approaching it with well-known optimization techniques, such as ant-colonies, genetic algorithms, etc. Indeed, these works aims at providing an automatic system to obtain optimized system architectures (Ölvander, et al., 2009) (Gavel, et al., 2007) (Judt & Lawson, 2016). The Conceptual Design for Assembly method developed in this work finds its roots in the idea that complex systems must not be studied with *only* complex techniques. Indeed, these systems are characterized by an elevated number of parameters, that are strictly related to each other at different design phases. To handle this complexity, it is required to use both human judgment and mathematical methods. The lack of one of them will inevitably lead to non-satisfactory solutions.

The literature analysis presents the need to extend Design for Assembly methods to complex products. The preliminary works available in literature showed how this goal can be reached by moving DFA analysis to early design phases (conceptual design). The proposed research thesis aims at pushing forward the current state of the art by:

- Providing a hierarchical framework to collect conceptual data and perform analysis to study different figure of merits (Design for X) and to apply the method in other fields (i.e., oil & gas, off-highway equipment, etc.). The CDFA method currently developed can be used to study other figure of merits other than assembly performances. For instance, sustainability analysis, disassembly, life cycle impact, can be performed using the same structure with different parameters.
- Developing a method to select design guidelines which generate the highest positive impact on the product architecture. The CDFA method currently available can assess product architectures, but it does not guide designers through the modification of them.

1.4 THESIS OVERVIEW

The proposed research work needed a comprehensive analysis of relevant literature in the domains of product architecture and system engineering, with a focus on the aviation field. The established methodology is structured into several phases that allow any user to successfully implement the method. The method is supported by two real-life case studies that include findings and comments. Finally, a software mock-up is suggested to enable a lean application of the method. The thesis is organized as follow:

Section 2 (State of art) discusses the state of art in the field of product development and systems engineering, focusing on the aerospace sector. Moreover, it delineates what it is currently missing.

Section 3 (Material and methods) introduces the Conceptual Design for Assembly method developed. It explores every part of the process, including the assumptions made and the tools utilized to support it.

Section 4 (Case studies) presents the two case studies in which the Conceptual Design for Assembly method is applied. Two aircraft systems (nose-fuselage and cabin) were studied in order to analyse their present product architecture and provide design changes to reduce assembly complexity.

Section 5 (Results) examines and discusses differences between the developed approach and similar methods found in the literature. Advantages and drawbacks with respect to the Conceptual Design for Assembly are addressed.

Section 6 (Software prototype) presents the mock-up of the software tool developed in compliance with the CDfA methodology, describing the theory and assumptions made to generate the software concept.

Section 7 (Conclusion) concludes the thesis, summarizing the overall thesis and proposing further research activities.

2. STATE OF ART

In the following paragraphs, an overview of the literature related to the product development process and design for assembly is presented. Then, current research on early design phases and products conceptualization is introduced. Finally, a discussion about the system engineering is provided, with a focus on the aerospace sector.

2.1 PRODUCT DEVELOPMENT

Firms need to overcome several challenges to stay competitive in the market. Challenges come from the rapid change of need of demanding customers, who are seeking high quality and low-cost products (Gunasekaran, 1999). This pressure has led to the expansion of activities related to the development and improvement of products, in order to achieve innovations. Innovation is a key aspect to remain competitive in business. The term innovation indicates an improvement achieved by the company from the previous state of the art. Innovations can span several areas of a company, ranging from management to product improvements. In literature, it is possible to distinguish two types of innovations: incremental and radical (McDermott & O'connor, 2002). The former aims to improve specific aspects step by step, in an incremental manner. The latter seeks to introduce a completely new technology that, if successful, leads to a huge competitive advantage with respect to the competitors. The advantage may last for several years. Incremental innovations represent a company every-day-activity, while disruptive innovations are what companies compete for. Radical innovations are risky since if not successful, may lead to huge losses for the company. Industries compete over disruptive technologies, trying to achieve the most of them, reducing the risks associated. This type of innovation can be obtained through the introduction of new technologies, new methods, new products, or the improvement of existing products to achieve a considerable reduction of production per part (McDermott & O'connor, 2002). The realization of new products requires fulfilling varied functionalities, managing different expertise, in order to bring innovation inside the firm. The process of product creation is called Product Development Process (PDP). The PDP is a complex process articulated in several phases that requires various expertise to cooperate.

It collects different domains from design science to cognitive psychology. It requires the use of strategies, rules and principles to achieve goals and coercive working solutions. The main work on the formalization of the PDP (Figure 2) is proposed and summarized by Pahl and Beitz (Pahl & Beitz, 2013). They identified four (4) phases required to design a product

1. **Planning and Task Clarification**
   to collect information, in form of requirements that need to be fulfilled by the product and to identify existing constraints. The output obtained is the definition of design specifications.
2. **Conceptual design**
   to translate design specifications and constraints into functions. To search for suitable working principles and to provide a working structure, widening the solutions space. The output is the derivation of working solutions (i.e., concept).
3. **Embodiment design**
   to define the physical structure of the product, providing the overall layout considering technical and economic criteria. The output is the product layout.
4. **Detail design**
   to design the final product, fixing the shape, the dimensions, the surface, the material, etc. To generate all the drawings and the manufacturing documents. The output is the product documentation.

The main purpose of the **Planning and Task Clarification** phase is to generate new ideas and new concepts, gathering information in terms of requirements. The design task can be provided to the design departments in several forms, such as a development order, a defined order, or a request based on suggestions and criticism made by sales, research, test, etc. In this phase, engineers and designers need to identify the requirements that determine the solution, documenting them as much as possible. To support this process, several tools might be used (e.g., the Quality Function Deployment matrix). The result of this process is a requirements list. This document will serve as a basic document for any product development process.

During the **Conceptual Design** phase, the design process is rapid and interactive. It is the part of the design process in which the basic solution path is laid down through the elaboration of a solution principle. To do so, designers and engineers may use several tools such as moodboards to stimulate and contextualize the design and to create the initial idea of the product. Ideas are generated using individual and brainstorming group sessions. This design stage typically involves: the production of sketches, drawings, mock-ups or, models to test basic technical feasibility, assess proposed production methods, etc. (Eppinger & Ulrich, 2015).

The **Embodiment Design** phase consists of developing the product design, adding scope to the initial concepts. During this stage, designers investigate competitors' products, understanding pros, and cons from both manufacturing and assembly point of view. Concepts are embodied using 2D sketches, CAD models, layout drawings, schematics, and mock-ups. Mock-ups and/or prototypes are used to test technical principles such as users' needs, component configuration, and manufacturing capabilities and visualize layouts (Eppinger & Ulrich, 2015). Performance calculations and decisions on materials and finishes are made at this stage. For instance, the most important technical analysis performed at this stage is the refinement of cost-effectiveness.

At the **Detail Design** phase, designers and design engineers use manufacturing and material knowledge to ensure that designs are efficient and profitable to be produced, and issues such as safety and usability are tackled. The obtained working drawings that provide information on the materials selected, tolerances, and manufacturing processes, are passed onto the production elements of the product development phase. Finally, the product enters the manufacturing level, where design engineers liaise with the production team. In Figure 2 the overall Product Development Process is presented.

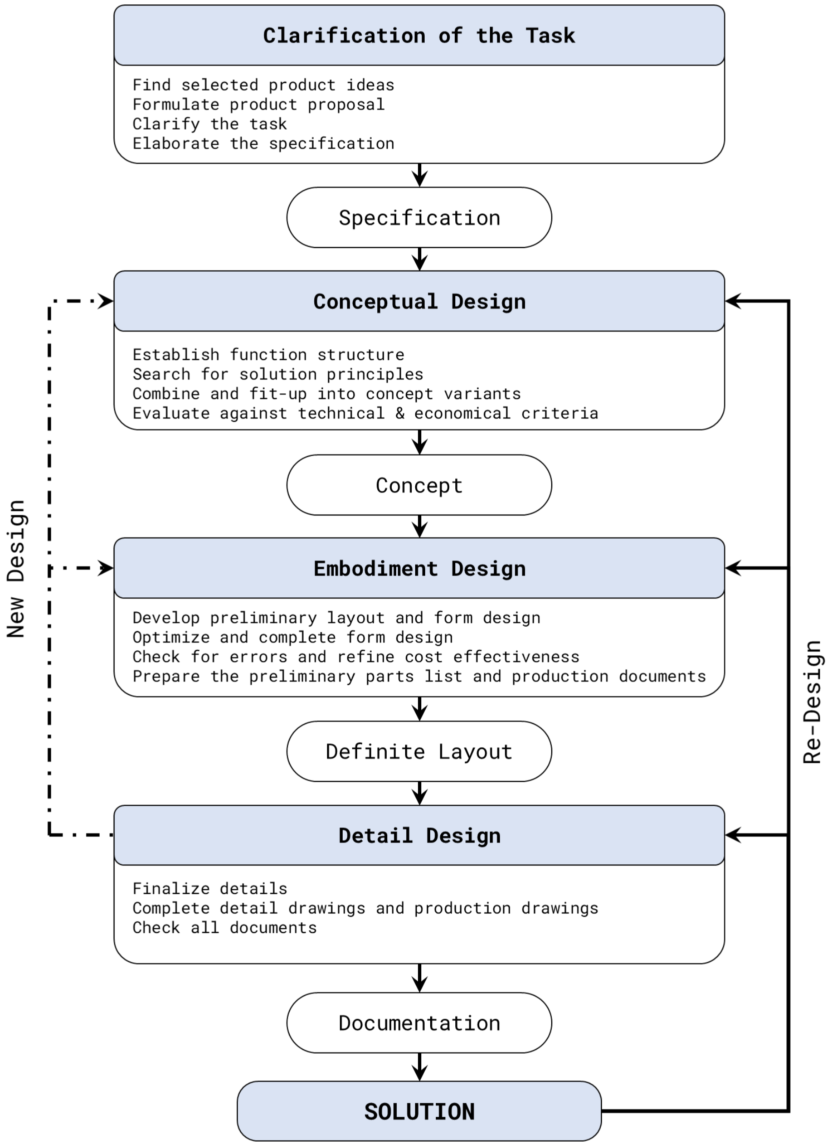


Figure 2 – Product Development Process design phases (Pahl & Beitz, 2013)

The PDP proposed by Pahl and Beitz (Pahl & Beitz, 2013) considers only the design process itself without considering the interaction between designers and other departments, such as the manufacturing department. Indeed, Boothroyd (Boothroyd, et al., 2010) proposed a New Product Development Process (nPDP) in which interactions between designers and other departments are supported since the initial phase of the design process. This approach is called Concurrent Design, also Concurrent Engineering (CE), and it aims at optimizing the design process by considering all product aspects from the beginning, breaking barriers between the different departments (Lyu & Chang, 2010). CE approaches consist of performing design tasks in parallel whenever possible, in contrast with the traditional sequential product development (Albin & Crefeld III, 1994). Concurrent engineering leads to a short lead time for both the development of new products and the re-design of existing ones. Advantages such as better product quality and the meet of customers' requirements with as low-cost level as possible (Tseng & Abdalla, 2006) (Sohlenius, 1992). However, nPDP is more difficult to manage due to the elevated number of stakeholders involved at the same time (Jun, et al., 2006). Concurrent engineering development and traditional approaches are represented in Figure 3.

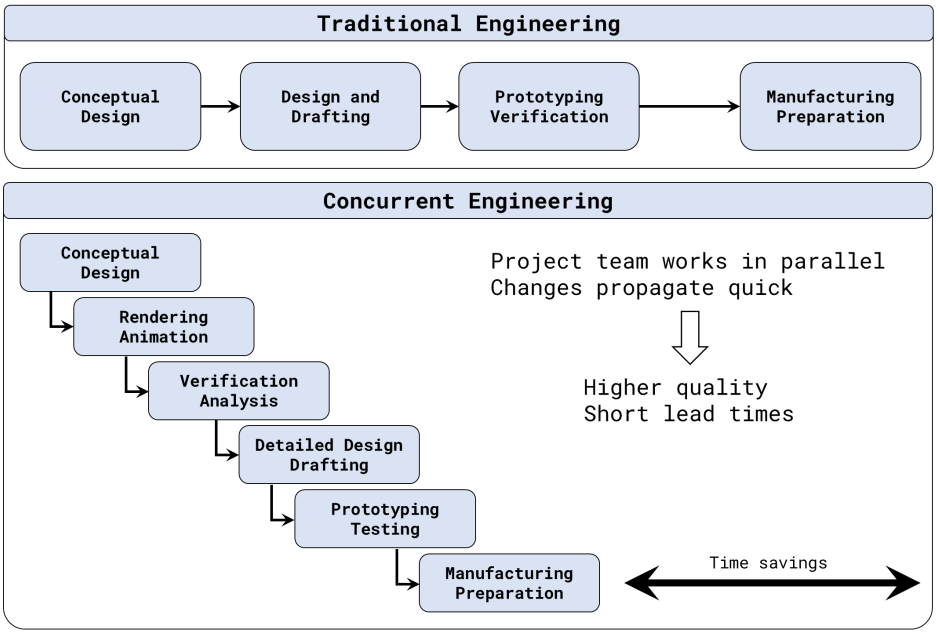


Figure 3 - Traditional vs Concurrent design process

In order to optimize the nPDP many methods have been developed. These methods are collected under the name of Design for X (DfX) methods (Huang, et al., 1999), where the *x* is replaced with the topic of the optimization (e.g., Design for Recycling, Design for Disassembly, etc.). DfX approaches optimize the product property identified by the *x* but they neglect all other aspects of the product lifecycle (Kuo, et al., 2001). An interesting review on DfX methods is proposed by (Benabdellah, et al., 2019) in which an overview of design for x techniques is proposed. In Table 1, the main DfX methods are collected.

Table 1 - Overview of design for X techniques (Benabdellah, et al., 2019)

Field	Scope	Design for	Design considerations
Economy	Product	Assembly (DFA)	Design to reduce the number of parts, tasks, and motions; design to reduce the difficulty of processes
		Manufacture (DFM)	Design to eliminate expensive manufacturing processes and materials
		Manufacture and Assembly (DFMA)	Design to address both DFM and DFA
		Variety (DFV)	Design to reduce design effort and time to market and to reduce the impact of variations in life cycle costs
		Six sigma (DFSS)	Design to reduce variation and defects; design to meet customers' requirements
		Safety (DFS)	Design to reduce risks of injury and to integrate hazard and risks of humans, materials, etc.
		Testability (DFTest)	Design to reduce failure modes
		Maintainability (DFMt)	Design to simplify repairs process; design to reduce repair time and to improve fault isolation
		Robustness (DFRb)	Design to decrease production costs
		Failure modes (DFMEA)	Design to reduce failure rate
		Supportability (DFSp)	Design to improve installation, user training, maintenance, customer support and product upgrades

Field	Scope	Design for	Design consideration
Economy	Product	Flexibility (DFF)	Design to consider changes in customer need/want; design to enable product reconfiguration
		Modularity (DFMod)	Design to have loosely coupled interfaces enabling module variation in products
		Miniaturization (DFMin)	Design to reduce production costs and to reduce barriers to innovation
		Serviceability (DFSv)	Design for compatibility with service and for streamlined service process, component and storage
	System	Supply chain (DFSC)	Design to address the performance of both logistics and reverse logistics benefits
		Logistics (DFL)	Design to decrease packaging and to reduce product size for storage and transportation
		Mass customization (DFMac)	Design to enable commonality and reusability between products parts and process
		Procurement (DFP)	Design to enable parts commonality and to leverage existing supplier relationship
	Both	Quality (DFQ)	Design to eliminate defects in production processes and to meet customers' requirements
		Life cycle (DFLC)	Design to reduce life cycle cost
		Cost (DFC)	Design to reduce life cycle cost

Field	Scope	Design for	Design consideration
Economy	Product	Recycle (DFR)	Design to increase recyclable material inputs and outputs and to minimize material variety
		Reuse (DFRu)	Design to standardize components and to enhance the durability of reuse targeted components
		End of life (DFEOL)	Design to ensure easy access to fasteners and joints and to lower destructiveness and selectiveness of disassembly process
		Remanufacture (DFRem)	Design to enable disassembly, assembly, cleaning, testing, repair, and replacement
		Reliability (DFRL)	Design to use proven components and to identify and eliminate critical failure modes
		Sustainability (DFSt)	Design to consider the three dimensions of sustainability: economy, ecology and equity
		Environment (DFE)	Systematic consideration of environmental safety and health
		Chronic risk reduction (DFCRR)	Design to reduce hazardous material and emissions or waste
		Energy conservation (DFEC)	Design to reduce energy consumption and to ensure rapid warm up and power down
		Material conservation (DFMC)	Design to reduce product dimensions and to utilize renewable, abundant, and recyclable resources

Field	Scope	Design for	Design consideration
		Waste minimization and recovery (DFWMR)	Design to reduce waste; design to increase use of biodegradable materials
		Reverse logistics (DFRL)	Design to enable customers to support preventing returns
		Disassembly (DFD)	Design to reduce environmental impact, to simplify repair time and to improve fault isolation
		Packaging (DFPk)	Design to reduce production costs, design to reduce environmental impact
	Product	Social responsibility (DFSR)	Design to enable linkages with society, design to consider non-traditional markets, design to eliminate social problems

2.2 DESIGN FOR MANUFACTURING AND ASSEMBLY METHODS

The Design for Manufacturing and Assembly method (DFMA) is the first family of DfX methods that was developed. DFMA methods aim at optimizing the manufacturing and the assembly process, hence the cost, of products. To improve these phases three aspects are analysed and optimized: (i) number of components, (ii) easiness of the assembly process, and (iii) components manufacturing cost, in terms of labour, material requirements, etc. DFMA methods are obtained through the implementation of Design for Assembly (DFA) and Design for Manufacturing (DFM) methodologies. Design for Assembly is a systematic procedure aiming at the reduction of assembly time through the reduction of the overall number of components in a given assembly, and the elimination of critical assembly tasks (Boothroyd, 1987). Design for Manufacturing is an engineering practice aiming at the simplification of the manufacturing process for cost reduction of a given component through: i) the selection of raw material type, ii) the selection of raw material geometry, iii) the definition of dimensional and geometrical tolerances, iv) the definition of roughness, v) the characterization of specific shape constraints based on the manufacturing process, and vi) the selection of secondary processing such as finishing (Favi, et al., 2016) (Srinivasan, et al., 1995). Several scholars have recognized the importance of DFMA in CE debating the relationship between the two domains. For some of them, DFMA is the basis for concurrent engineering, while others argued that concurrent engineering should frame the application of DFMA (Thompson, et al., 2018).

Two methods have been mainly used in academia and the industrial world: Boothroyd & Dewhurst (B&D) (Boothroyd, et al., 2010), and the Lucas method (Ltd, 1993).

The Boothroyd & Dewhurst method aims at optimizing the product assemblability by:

- Reducing the number of components
- Ensuring that parts are easy to assemble
- Increasing the use of standardized parts across the entire product range
- Designing with widest possible tolerances
- Selecting material following both manufacturing and functional considerations

The B&D approach provides a quantitative measure called Design Efficiency (DFA Index) based on the analysis of the product. The DFA Index is used to identify critical components in terms of manufacturing and assembly performances. When these components are identified, they can be optimized according to the following design guidelines:

- Reduce the number of parts and their type
- Design components easy to align
- Ensure accessibility and visibility during assembly operations
- Design components that cannot be installed if wrongly positioned
- Ensure easy component handling
- Maxime components symmetry

The Lucas Method is composed of three phases: i) functional analysis, ii) manipulation analysis and iii) insertion analysis.

The functional analysis aims at reducing the number of components necessary for the product to operate. The method defines two types of components: A groups components which are necessary to guarantee product functions, while B groups components which are not strictly necessary for the product functions. An example of B components are screws, junctions, etc. From these two components, the Functional Efficiency (FE) index is computed. The manipulation analysis consists of assessing the time to manipulate each component, computing the Handling Index. The Handling Index is obtained through empirical coefficients. Finally, the insertion analysis aims at assessing the easiness to assemble a component, through the computation of an index called Insertion Index. Like the Handling Index, the Insertion Index is computed using empirical coefficients. These methods are not the only available. A review presented by (Formentini, et al., 2022.b) looked at several publications in the literature and found many distinct DFMA techniques.

The benefits of using DFMA approaches are confirmed by multiple case studies, which suggest that using these methods can reduce assembly time (and thus cost) by 40% (Azevedo, et al., 2015) (Sudin, et al., 2016). These methods have been mainly applied on simple products or sub-assemblies, assembled manually with bolted joints made of less than 60 parts, in which all parts are manufactured with traditional production technologies (i.e., fusion, sheet metal stamping and bending, forging, etc.) (Formentini, et al., 2022.b).

Regarding the aerospace sector, several case studies are reported in literature where DFMA methods have been applied to reduce aircraft components' complexity. Eakin (Eakin, 2010) reports case studies in which the B&D DFMA approach was used to reduce the manufacturing and assembly cost, while Barbosa and Carvalho (Barbosa & Carvalho, 2014) collected a set of design guidelines to apply during the aircraft design to improve the assembly and manufacturing performance. This optimization is mainly done at late design phases where information of higher granularity is available and better predictions can be made. Moreover, none of them are done at the aircraft system level but only to small components. Nevertheless, few issues were highlighted in literature regarding the adoption of DFMA methods in the aerospace field. The first one is the product complexity: the elevated number of parts, assembly and installation operations required to create an aircraft do not allow applying state-of-art DFMA methods to the whole aircraft system, but only to small components. The existence of constraints such as safety or weight limits has a significant influence on the application of DFMA approaches since they do not take these factors into account. Finally, the elevated product cost and lead time do not allow performing a redesign of the product, as typically suggested by the DFMA methods.

These constraints require addressing the assembly and installation processes at a very early design phase. Typical DFMA techniques need information available at the late design phase when mainly all design decisions have been made. Working at the conceptual design phase allows tackling manufacturing, assembly, and installation problems earlier in the nPDP process, allowing to optimize the product without impacting the overall final costs.

2.3 EARLY DESIGN PHASE AND PRODUCT CONCEPTUALIZATION

Since the beginning of the advent of DFMA methods, some studies tried to move the analysis from the detail design phase to the conceptual design phase (Little, et al., 1997). Among these, the paper proposed by Rampersad (Rampersad, 1996) was one of the first to investigate DFMA methods from a relational point of view, to understand how design variables affect product assembly. A more recent attempt was performed by Emmatty and Sarmah (Emmatty & Sarmah, 2012) that tried to merge DFA and DFM techniques with product architectures analysis. The typical output of DFMA methods in the conceptual design phase is a product architecture with optimized performance in terms of assembly.

The possibility to perform analysis at early design phases with information of lower granularity enables further reduce the production cost since no redesign of the product is required. However, moving from late design phases to early design phases requires the use of tools that can deal with different information granularity and uncertainties.

According to the Product Development Process proposed by Pahl and Beitz (Pahl & Beitz, 2013) the first phase to develop a product is the creation of requirements' list to understand and collect features required by the client. The definition of the requirements' list is a collaborative phase in which both the firm and the client are actively involved. The conceptual design phase starts when the requirements' list is obtained. In this phase, using the requirements' list, product functions are derived to obtain the product architecture. The process can be divided into the following sub-steps:

i. **Creation of the functional scheme**
   to derive the list of basic and auxiliary functions that the product needs to perform in order to work successfully
ii. **Definition of functional modules**
   to cluster functions into functional groups (i.e., modules) to be able to achieve the desired functions
iii. **Definition of product architecture**
   to link modules through interfaces and to provide the product layout

The creation of a functional scheme consists of the translation of requirements into functions and the identification of the linkage among them. A function represents an action that the product is required to perform in order to fulfil its aim. For instance, the function of the product "electric screwdriver" is "Loosen/Tighten Screws".

The functional scheme can be accomplished through decomposition techniques. Decomposition is the process by which complex design problems are simplified to facilitate the decision-making process. Decomposition can be hierarchical or non-hierarchical. The former is obtained when:

- Interactions between various levels of subsystems are present. This interaction can go one way or both ways.
- Interactions between subsystems at the same level of the same parent or different parents are present. This interaction can go one way or both ways (Shupe, et al., 1987).

A typical hierarchical system is presented in Figure 4.

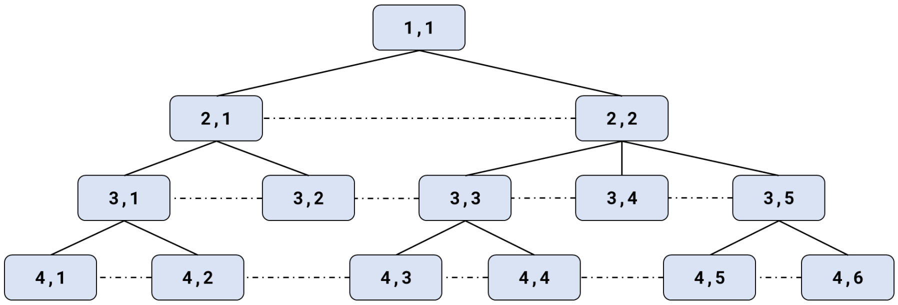


Figure 4 - General Hierarchical System decomposition (Shupe et al., 1987)

Hierarchical decomposition is used for many functional design schemes, in fact, functions tend to move from the most abstract (i.e., higher levels) to the most concrete (i.e., lower levels) (Kirschman & Fadel, 1998). Pahl and Beitz (Pahl & Beitz, 2013) proposed one of the most famous hierarchical decomposition techniques based on functions. The method called "Functional Decomposition" consists of a representation obtained with the *black-box* model. Functions are represented with boxes, in which basic flows are represented as input/output of each box. The identified basic flows are Energy, Signal, and Mass (Figure 5).

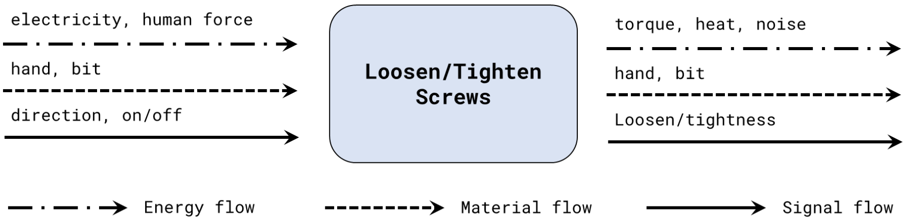


Figure 5 - Black Box model

The first level presents only the main function. The second level divides the main function into sub-functions that are easier to accomplish and handle. According to the knowledge and the complexity of the system under study, several levels can be accomplished.

To date, this approach is still widely used in academia and industry since it allows obtaining a wide solution space, increasing the possibility to find innovative solutions to achieve functions. However, it presents two main drawbacks: i) layers can be obtained only through a 2D representation, making the representation difficult to read when the system analysed is hierarchical, and

ii) it does not provide information about how a function should be decomposed into sub-functions.

Another method called "Function Means Trees" is available to decompose systems and obtain functional representations. The Function Means Trees is based on the work of Pahl and Beitz (Pahl & Beitz, 2013), which overcomes the identified limitations by combining functional decomposition and working solution in a single graph (Bracewell, 2002). A function means tree allows representing both alternative decompositions and solutions for a given function. It is achieved through a parent-child relation. Functions (i.e., parents) represent what the system must achieve, while means (i.e., child) represents how the system should achieve them. Even though the Function Means Trees provides an interesting way to derive functions, it limits the solution space since it is required to provide already working solutions.

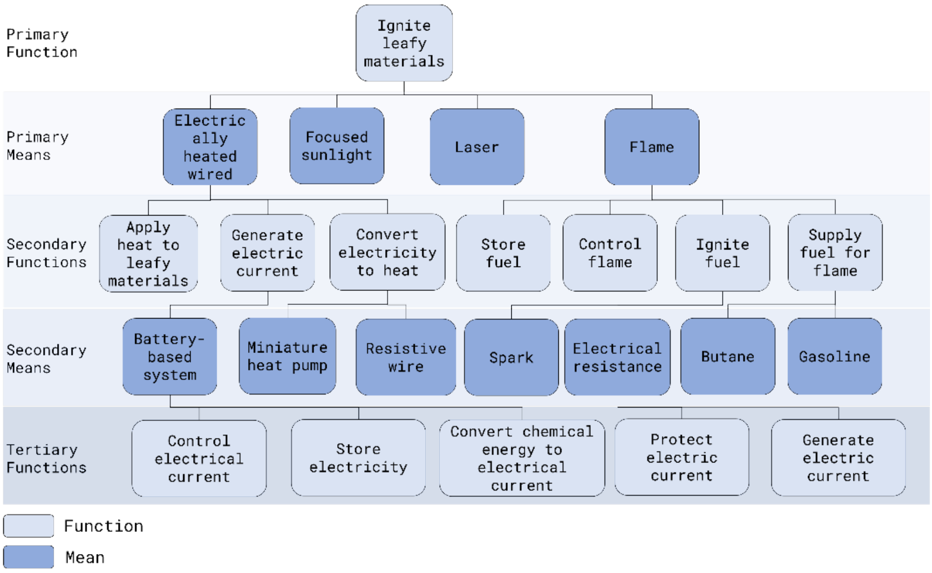


Figure 6 - Function Means Tree example

The Function Description Language proposed by Lai and Wilson (Lai & Wilson, 1989) is a system which uses functions to analyse a design. Different from the methods outlined above, all functions are decomposed based on their forms. Indeed, the decomposition starts with an exploded view of the item and, subsequently, a function is assigned to each part. The system proposed, called Function Rationalization System is represented in Figure 7.

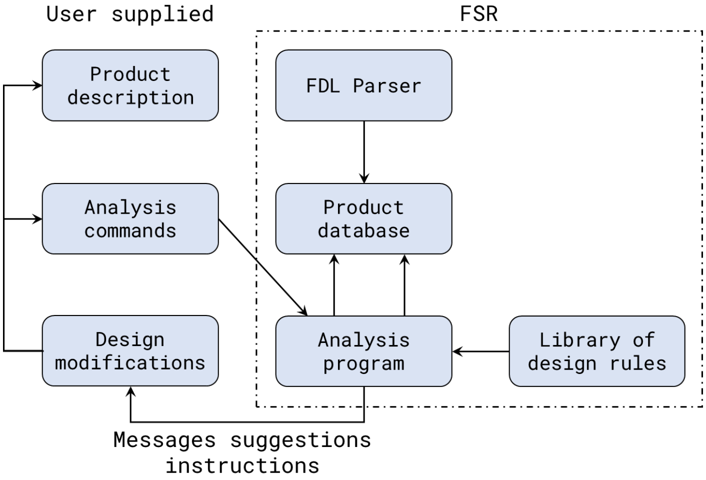


Figure 7 - Function Rationalization System proposed by Lai and Wilson (Lai & Wilson, 1989)

The choice of which decomposition method should be used depends on several factors. A system's decomposition is a difficult process that can result in a variety of outcomes depending on the method used. For instance, the Function Rationalization System can be used to generate a functional scheme starting from an already available product, in a structured manner. However, even though this method may provide repeatable results, it is difficult to use and require a great effort in term of time to generate a suitable output. The Functional-Means Tree and the Functional Analysis can both be used when a wider solution space is desired. These methods are easy to apply and understand, however, results may change according to the user who performed the analysis. In particular, the Functional Analysis provides the greater degrees of freedom to designers, since it is not required to provide a working solution along with the function. In other words, the Functional Analysis is more suitable when complex products are analysed, or completely new concepts need to be generated since the solution space provided by the method remains as wide as possible. A comparison among the three decomposition methods is provided in Table 2.

Table 2 - Decomposition Methods comparison

Decomposition method	Solution space	Required time	Level detail	Solution required	Product needed
Functional Decomposition	Wide	Medium	Fair	No	No
Function-Means Tree	Medium	Long	High	Yes	No

Decomposition method	Solution space	Required time	Level detail	Solution required	Product needed
Function Rationalization System	Small	Very Long	High	Yes	Yes

Once the functional schemes are obtained, they can be supported by modular analysis techniques. Indeed, by clustering functions together the "function modules" are obtained. A function module is a group of functions that can be achieved by the same working principle. For instance, the function module A identified in Figure 8 clusters the following functions: i) Convert Electricity to Torque, ii) Change Torque, and iii) Transmit Torque. The three functions can all be satisfied by using an Electric Motor (i.e., working principle).

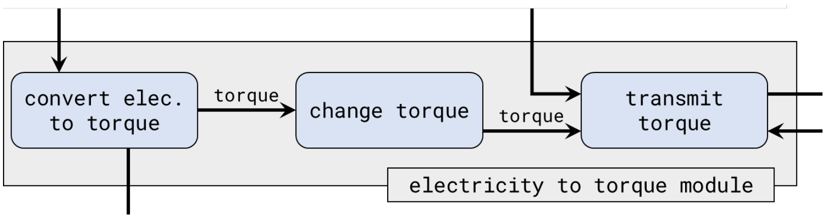


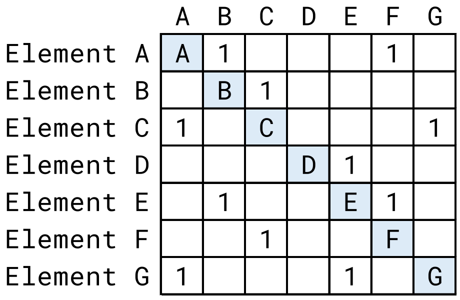
Figure 8 - Functions Module A

The creation of a function module is the first step towards a modular product. However, it is a complex task since many parameters must be considered to try to reach the optimum cluster. To perform this task, several methodologies and tools are available in the literature. Methods can be divided into qualitative and quantitative (Li, et al., 2013). The most famous qualitative method was proposed by Stone (Stone, et al., 2000). The authors propose three heuristics to derive modules in a systematic manner:

- Dominant Flow
- Branching Flow
- Conversion-Transmission Flow

Using this formalism, the functional scheme is analysed, and functions are clustered together to obtain function modules. Gao (Gao, et al., 2010) extended the heuristics proposed by Stone et al., to be applicable on the generalized direct graph. The approach allowed to reduce the necessary human judgment by providing a computer-aided tool for identifying functional modules.

Quantitative methods are generally based on Design Structure Matrix (DSM) formalization. DSM is a tool used to model a project or a system. DSM is represented by a NxN square matrix that maps interaction between systems' elements. It is a flexible tool that can be used to map different problems from organizational problems to complex systems interactions. When DSM is applied to system levels, functions are listed in the first row and the first column of the matrix and off-diagonal elements indicate system interactions (Figure 9).



	A	B	C	D	E	F	G
Element A	A	1				1	
Element B		B	1				
Element C	1		C				1
Element D				D	1		
Element E		1			E	1	
Element F			1			F	
Element G	1				1		G

Figure 9 - Design Structure Matrix

Several modularization techniques (i.e., clustering techniques) have been developed to work with DSM representation. For instance, Browning (Browning, 2015) proposes a component-based DSM to describe interactions among elements in a system. Four types of interactions were identified, and a quantification scheme was proposed to weigh each interaction (Table 3).

Table 3 - Interactions' type and Quantification scheme

Taxonomy	Description
Spatial	Associations of physical space and alignment, needs for adjacency or orientation between two elements
Energy	Needs for energy transfer/exchange between two elements (e.g., power supply)
Information	Needs for data or signal exchange between two elements
Material	Needs for material exchange between two elements

Interaction	Score	Description
Required	+2	Physical adjacency is necessary for functionality
Desired	+1	Physical adjacency is beneficial, but not necessary for functionality
Indifferent	0	Physical adjacency does not affect functionality
Undesired	-1	Physical adjacency causes negative effects but does not prevent functionality
Detrimental	-2	Physical adjacency must be prevented to achieve functionality

Van Beek (van Beek, et al., 2010) tried to avoid the filling of the DSM through experts' knowledge by applying modularization techniques based on function-behave-state models. The clustering process is performed adopting k-means algorithm to keep the computational effort manageable.

The main difference between qualitative and quantitative modularization techniques is the solution space they can handle and provide. Qualitative techniques, due to the need for human judgment provide greater solution space and can cope with several constraints. On the other hand, they do not guarantee the reach of the optimal function clusters, meaning modules. Quantitative techniques work with optimization functions; therefore, they guarantee the reach of an optimum result (or sub-optimum), but they fail to handle several constraints. Indeed, the optimization problem might be too complex to solve.

By switching from functional modules to physical components, it is possible to derive the product architecture. According to Eppinger and Ulrich (Eppinger & Ulrich, 2015), product architecture is the scheme where functional modules are translated into physical components and linked through physical interfaces. Product architectures can be divided into two types: i) modular architectures and

ii) integrated architectures. Modular architecture consists of an architecture in which a loose relation between functions and physical components is present, meaning that a change in one architectural component will not require a change in another one. On the other hand, integrated architecture is a fixed architecture which is typically optimized for one aspect. Both product architectures present advantages and drawbacks. An interesting overview of these trade-offs was proposed by Jose and Tollenaere (Jose & Tollenaere, 2005) and reported herein Table 4.

Table 4 - Trade-off between modular and integral product architectures (Jose & Tollenaere, 2005)

Modular design Benefits	Integral design Benefits
• Module task specialization	• Interactive learning
• Increased number of product variants	• High levels of performance through special technologies
• Economies of scale in component commonality	• Systematic innovations
• Costs savings in inventory and logistics	• Superior access to information
• Lower life cycle costs through easy maintenance	• Protection of innovation from imitation
• Shorter product life cycle through incremental improvements such as upgrade, add-on and adaptations	• High entry barriers for component and module suppliers
• Flexibility in component reuse	
• Outsourcing	
• System reliability due to high production volume and experience curve	
• Faster assembly and less production time	
• Postponement of operations of differentiation for fast reaction of the market	
• Parallel manufacture of modules	
• Fast development of products	

Product architecture enables the study of products from a conceptual point of view, before proceeding with the creation of them. It helps engineers and designers to argue the product's functional point of view, spotting possible errors that may affect the cost, time, and quality of the product itself. The main components of product architectures are modules and interfaces. By changing their position and their relation, different product architectures can be obtained. Again, product architecture may be integral or modular. According to the product aim, one architecture may present better performances with respect to another. To date, there is a continuously growing interest in modular product architectures, since they present advantages in terms of assemblability, upgradability, and sustainability. However, even though modularity has been used on a wide range of applications and numerous modularity metrics have been considered, there is a concrete lack of evidence of potential benefits, especially to big and complex products (Bonvoisin, et al., 2016). Currently, research works in the field of product architecture are trying to tackle these aspects.

2.4 SYSTEM ENGINEERING IN AIRCRAFT INDUSTRIES

System engineering is a relatively new branch of the engineering field. It is a multi-disciplinary area in which engineering, and engineering management are strictly correlated. It focuses to design, integrate, and optimize complex systems over their life cycle. The official definition provided by INCOSE is

> *"Systems Engineering (SE) is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs."*
> *(INCOSE 2010)*

Several tools and methods have been developed to support SE studies. Some of the most used tools to represent complex systems in accordance with SE studies are:

- Design Structure Matrix (also known as N^2 chart)
- Data flow diagram
- Functional Flow block diagram
- Model based system engineering

The following levels can be described in the aerospace area from the perspective of a System Engineer:

- Level 1 – System of Systems
  It clusters all systems used in aerospace, such as the air transportation/defence system which includes missiles, airports, etc.
- Level 2 – System Level
  It considers the single system and related ones, such as the aircraft, manufacturing plant, etc.
- Level 3 – Subsystem Level
  It collects the main aircraft subparts, such as hydraulic, electric, avionic, etc.
- Level 4 – Component Level
  It clusters components such as landing gear, auxiliary power units (APU), nacelles, etc.
- Level 5 – Part Level
  It considers the elementary part of the product, such as screws, frames, wires, etc.

Each level presents different granularity and, thus, different information. Optimization can be performed on different levels or considering a single level. Systems can be defined as:

> *"An assemblage or combination of elements, members, components, and parts forming a complex or unitary whole."* (Sadraey, 2012)

A system can be defined as a combination of elements, components, and parts, clustered together to form a complex unit. According to this definition, an aircraft is a complex system composed of several interrelated components working together to achieve several functions.

Aircraft can be divided into 5 complex sub-systems: i) nose-fuselage, ii) wings, iii) cabin, iv) tail and v) engines. Sub-systems are shown in Figure 10.

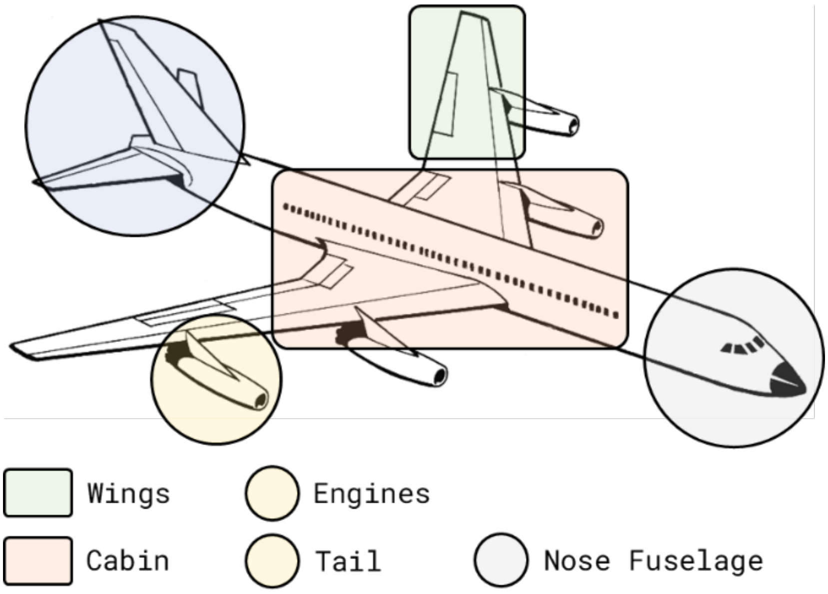


Figure 10 - Aircraft sub-systems

Due to its complexity, the optimization of this product is a challenging task and requires the definition of *ad-hoc* models.

Model-based systems engineering (MBSE) is a method to formalize requirements, design, development, and validation of complex systems (Wymore, 2018). This method is based on three concepts:

- **A model**
  it is required to provide a simplification of the system analysed. It can be represented through graphs, schemes, product architectures, or other tools. It should be less complex than the overall analysed problem, but it must represent the reality correctly.
- **System thinking**
  it requires looking at the system as a part of a bigger system and not as a self-sufficient entity. System thinking emphasises the interconnection among parts belonging to a system, meaning parts do not only connect to each other but are interdependent. It allows spotting patterns that might not appear at first.
- **System engineering**
  it collects several engineering disciplines to allow an optimized development process and reduce overall risks.

All the techniques mentioned above play an important role during the development of complex systems.

Research on System Engineering studies showed that manufacturing problems have been tackled mainly from the management and production system design (Milner, et al., 2013) (Sage, 1996). Moreover, even if the assembly phase of complex systems can impact over 40% of the final cost (Bullen, 1999), the assembly phase is not considered by any of the proposed methods.

Only a few works have been found in literature proposing the study of the assembly phase of complex and large systems. For instance, Yuan (Yuan, et al., 2018) proposed a DFMA method to be applied on prefabricated buildings, while Remirez (Remirez, et al., 2019) adapted the well-known Boothroyd and Dewhurst (B&D) and Lucas/Hull methodologies to optimize the assembly phase of a solar tracker.

In the aerospace field, product assembly is frequently overlapping with product installation. In literature, the two terms are often used as synonyms, even though they are used in different contexts. Assembly is the process by which components are brought together to obtain the final working product (i.e., once components of an Auxiliar Power Unit are assembled; they create the working product). On the other hand, installation is the process by which harnesses, cables, pipes, etc. are fixed and connected to the main components (Lockett, et al., 2014). Assembly processes are performed on components before the installation processes. According to this definition, Design for Installation (DfI) is a part of the Design for Assembly methods. DfI refers to the process of optimizing the assembly phase of components themselves, while DfA refers to the process of optimizing the process of fixation of cables, harnesses, ducts, etc. for the components. In general, assembly processes are characterized by specific assembly sequences, that are often the object of optimization studies, which is not the case for the installation processes (Hermansson, et al., 2013) (Paik & Thayamballi, 2007).

Optimizing the manufacturing and assembly aspects of complex systems such as aircraft means to reduce the overall manufacturing and assembly process to, eventually, reduce costs. Altfeld stated in his book that "*analysis of assembly and integration process may well change the original layout of the product architecture*" (Altfeld, 2016). The optimization of manufacturing and assembly features of aircraft can be challenged from two different points of view: i) Final Assembly Line (FAL) optimization and ii) product optimization.

The FAL is the final stage of the assembly life cycle of aircraft. It is an industrial installation that involves assembly processes, tools, jigs, machines, human resources, and industrial means. The design process of an aircraft assembly line is similar to product design. It is composed of three main steps: i) Concept, ii) Definition and iii) Development.

The first phase consists of creating a feasible concept of the assembly line. Some figures of merit that are defined at this phase are the capacity of the line, the number of stations, basic technology to be used, station order, and so on (Mas, et al., 2016). The second phase involves the definition of the assembly task that must be carried out in each assembly station. This phase requires a deep knowledge of the product functionality to be able to provide the right assembly sequence both from the manufacturing and the system point of view. In this phase, balancing virtual workstations and FAL is required. 3D representation (i.e., virtual FAL) might be used to make the overall process leaner. The final phase seeks to define assembly tasks in depth, breaking them down into elementary assembly tasks. The main challenge at this stage is documenting the elementary assembly tasks to create work instructions. Information such as process times and personnel allocation are refined and used to provide a precise balancing of the virtual FAL. In Figure 11 the product lifecycle and development milestones of the company AIRBUS S.A.S. are represented.

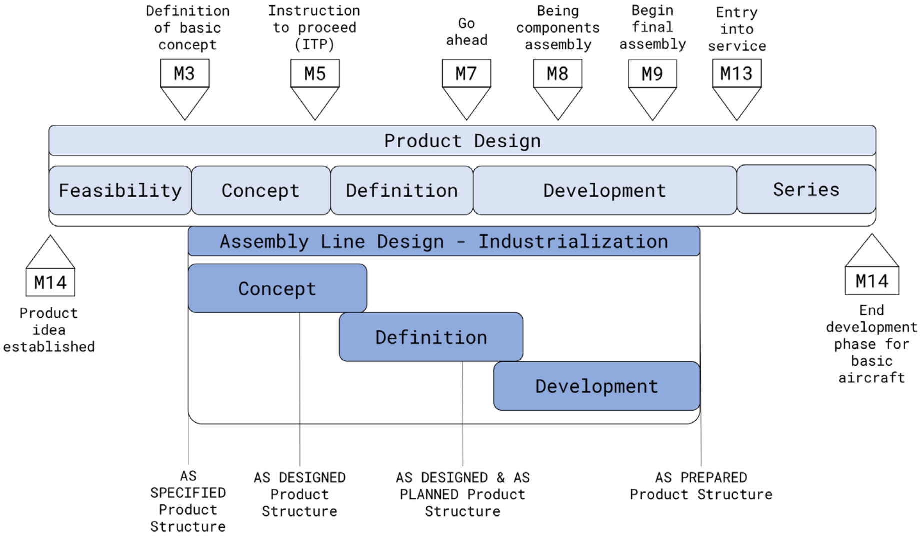


Figure 11 - AIRBUS product lifecycle and development milestones (Mas et al., 2015)

Since the FALs are complex systems, their development needs to be supported by *ad-hoc* methods to try to optimize them. Several authors tackled the FAL optimization (Mas, et al., 2016). For instance, Butterfield (Butterfield, et al., 2007) proposed a method to make use of digital manufacturing techniques to evaluate the assembly process of an aircraft fuselage in the final assembly line. While Li (Li, et al., 2020) appointed the problem from a SE point of view, proposing to use the Requirement, Functional, Logical, and Physical (RFLP) principles to derive its approach. The approach is divided into four (4) steps:

1. Connect assembly integration activities with SE principles
2. Decompose aircraft requirements to obtain system functions and 3D physical design, and allocate to assembly sequence
3. Use the RFLP approach to provide traceability from design requirements to assembly activities
4. Generate an initial feasible assembly sequence

The approach has been tested using an RFLP modelling platform integrated with the software CATIA V6. The assembly line optimization is an important part, although it can only result in minor improvements in assembly performance. In fact, without modifying the product design and optimizing solely the assembly line, there is a lack of concurrent design preventing cooperation among different departments. Furthermore, optimization made at the plant level is generally valid only for the plant analysed, or a limited type of plant. This is not the case for optimizations performed at the product design level.

In terms of product optimization, the concept of modularity has recently progressed into the aerospace industry. In the beginning, modularity was applied due to the need of creating aircraft sub-parts at different geographical locations (Monnoyer & Zuliani, 2007). With the introduction of electronic components, it was possible to create independent sub-parts or modules such as wings, cockpit, cabin, etc., and assemble at a later assembly phase (Frigant & Talbot, 2005). Moreover, product modularity allowed the creation of aircraft product families (Erens & Verhulst, 1997). The sub-systems that have been most studied in terms of modularity are: i) cabin interior monument and ii) aircraft engines (Brusoni & Prencipe, 2001) (Farid, 2008) (Jung & Simpson, 2017). Even if modularization has been applied successfully for the mentioned sub-systems, they can be considered stand-alone products that do not directly interact with the aircraft. In other words, the modularization of these systems does not have a direct impact on the aircraft. The main problem applying modularity to the overall aircraft is the elevated quantity of information that needs to be handled, making it difficult to create product architecture. Helmer (Helmer, et al., 2010) proposed a method to acquire and cluster information focusing on “assembly modules” as opposed to “design dependency modules”. Some studies showed that although modularity brings many advantages to, virtually, any product it might not be of benefit for products such as aircraft with respect to specific product performances. Hölttä (Hölttä, et al., 2005) states that there is an inverse correlation between modularity and performances for complex products, indeed fully integrated products such as wing or fuselage have better performance (i.e., drag, fuel efficiency, etc.) than modular ones. However, from a lifecycle point of view, modularity mainly leads to an advantage, enhancing the product assemblability, disassemblability, and upgradability performances.

Working with the product architecture at the conceptual phase allows investigating other solutions without impacting the overall product costs while having higher degrees of freedom to innovate. The definition and the use of a product architecture to tackle assembly and installation aspects is a complex task for products such as aircraft.

2.5 STATE OF THE ART SUMMARY

The literature analysis highlighted how the conceptual design phase is the most critical phase to challenge installation and assembly issues. Aerospace industries suffer this aspect since an elevated number of operations are in the critical path for cost and resource minimization, due to product complexity. Some studies attempted to address assembly and manufacturing aspects within the conceptual framework of product development with the main limitations highlighted above (i.e., modularity, DFMA, installation, etc.) missing a general approach to couple these aspects.

Design for Manufacturing and Assembly methods aim at providing a systematic method to optimize product manufacturing and assembly aspects. This is accomplished by iterating the approach several times through the creation of prototypes. The optimization of complex products' manufacturing and assembly aspects necessitates a shift in perspective from iterative techniques used throughout the whole product development process to iterative approaches used in the early design phases. In fact, redesigning items with complex features and large sizes necessitates a significant investment of both money and time, which is not feasible.

Working at the product architecture level will allow overcoming these drawbacks, tackling them at the very early design phases, reducing the overall impact.

3. MATERIAL AND METHODS

The analysis of complex product architectures requires the definition of methods and tools able to cope with information granularity and uncertainties typical of the conceptual design. In this chapter, the framework to perform the Conceptual Design for Assembly analysis is presented. The proposed method is a systematic approach that can be performed several times to optimize the assemblability of the analysed system architecture. The method is composed of seven steps required to assess the product architecture of the system analysed, and five steps to generate and select the more suitable redesign guidelines to improve the architecture assemblability. Tools and models required to perform the analysis successfully are explained in detail, with an extensive mathematical description.

3.1 OVERVIEW OF CONCEPTUAL DESIGN FOR ASSEMBLY METHODOLOGY

The proposed methodology, called Conceptual Design for Assembly (CDfA) is composed of two parts, the assessment phase where a numerical evaluation of the assembly complexity is performed, and the redesign phase, where the selection of the most impacting design guidelines is done. The two parts are divided into four main phases: (i) Product Functional Decomposition, (ii) Architecture Geometrical Definition, (iii) Conceptual Design for Assembly Assessment, and (iv) Product Architecture Redesign. Each phase is characterized by different steps and design tools as reported in Figure 12. The following paragraphs describe in detail each phase of the CDfA methodology.

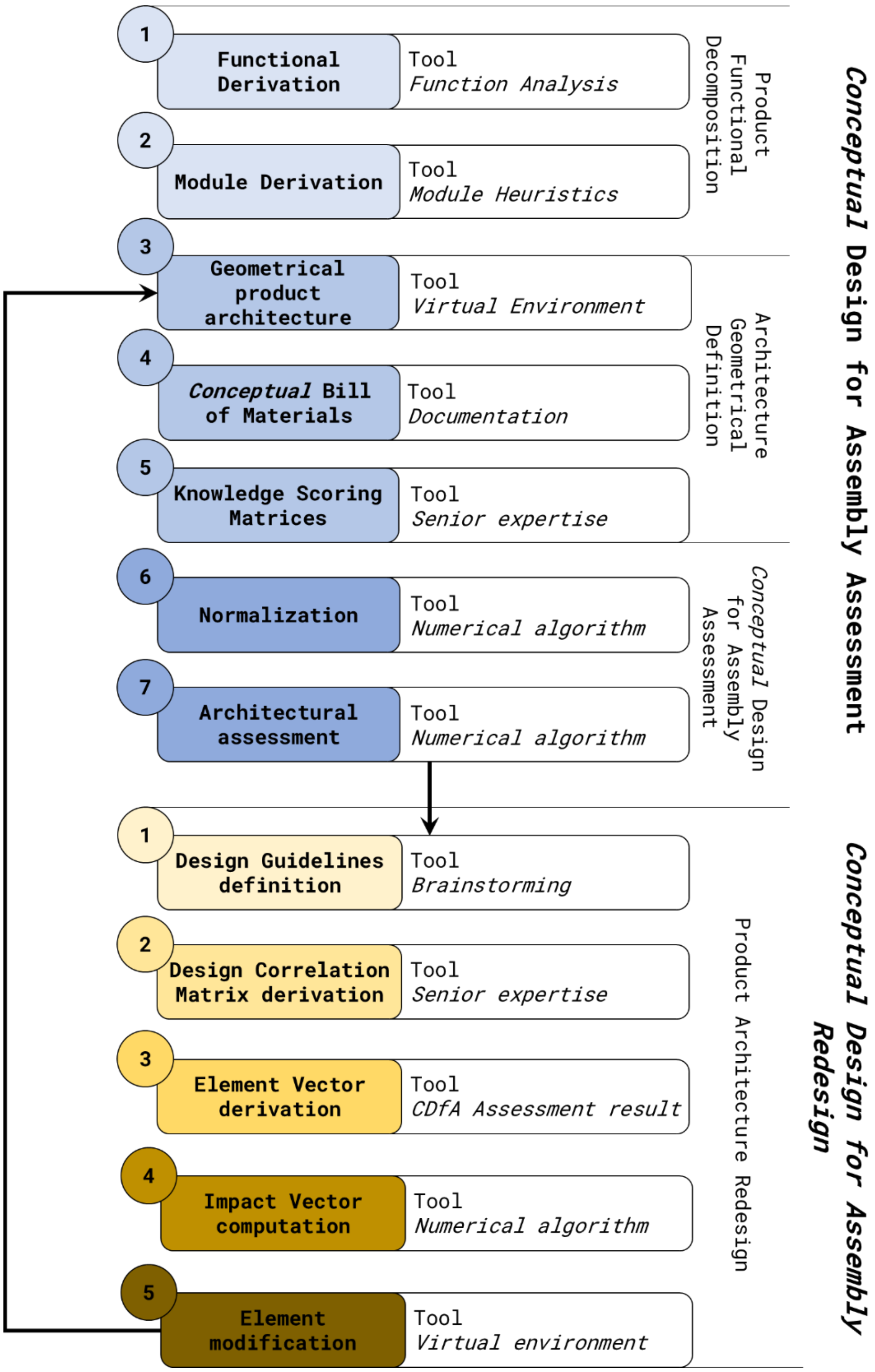


Figure 12 - Conceptual Design for Assembly methodology

3.2 CDFA ASSESSMENT PHASE

The CDfA assessment phase is composed of three phases (i.e., phase 1, phase 2 and phase 3) which are required to guide the users through the definition and assessment of the product architecture of the system analysed. The CDfA assessment allows obtaining DFA indices to understand the assembly complexity of elements composing the product architecture. In the following sub-paragraphs, each phase is described in detail, explaining how to perform the assessment phase.

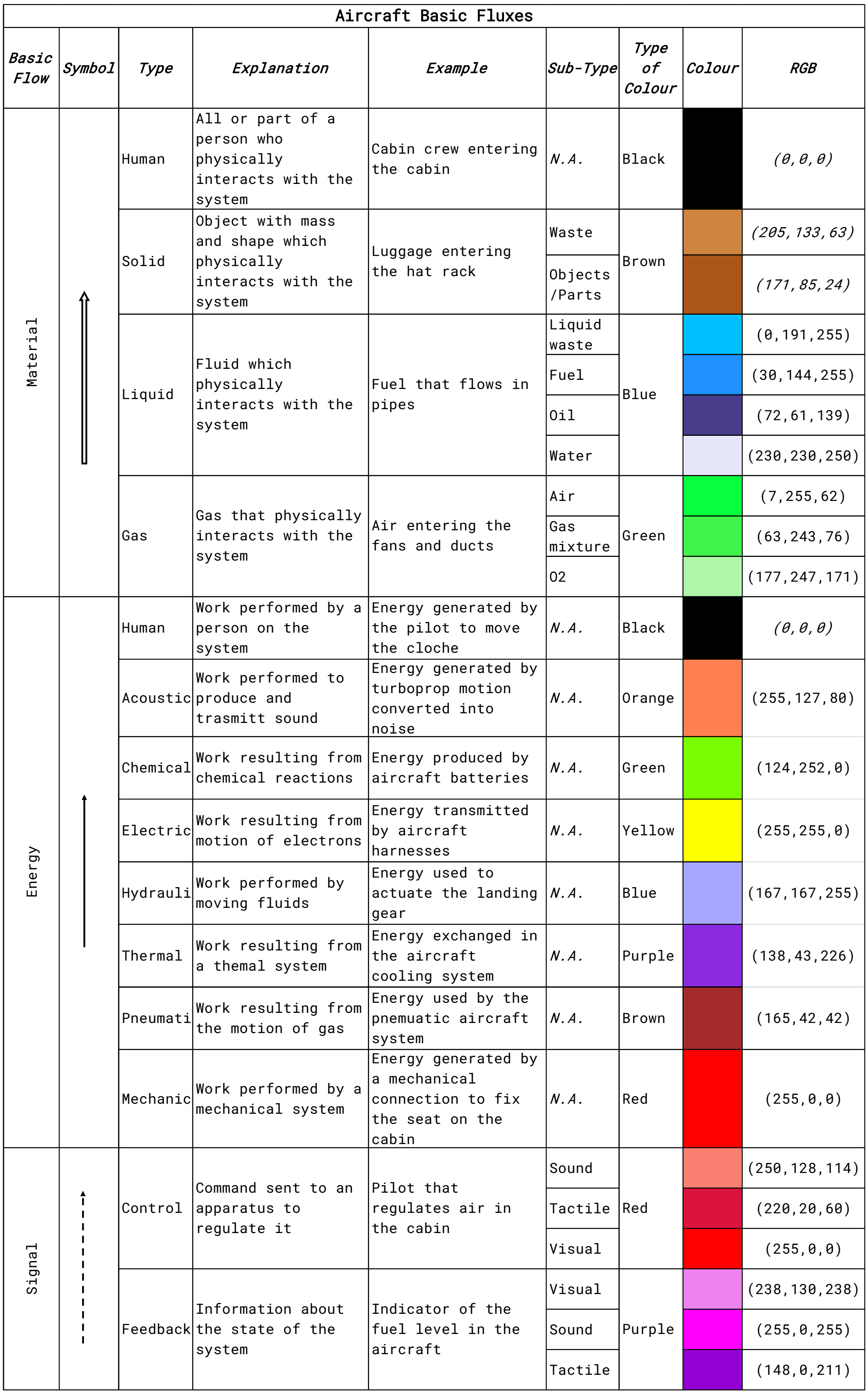
PHASE 1: PRODUCT FUNCTIONAL DECOMPOSITION

The Product functional decomposition is the starting point of the analysis. It enables the identification of functional modules and their functional interconnections that will then be used to define the physical modules and their physical interconnections. Modules and their interconnections will be used to perform the assembly assessment. Product functional decomposition is of great importance in this field which is characterized by consolidated design solutions and technologies. While keeping the compliance with stringent requirements for this type of product (i.e., complex products), the design of new product architecture at the conceptual design phase (e.g., module layout/arrangement, module position, module integration/decoupling, module assembly/installation, module fixation, interface routing, interface installation, etc.) is allowed. For this reason, the Product functional decomposition, even if limited to initial choices about technology and design solutions, still provides important information regarding possible changes to perform at the conceptual level. The Product functional decomposition is divided into two (2) steps: (i) Functional Derivation, and (ii) Module Derivation. The two steps have been performed with the help of customized literature approaches.

Functional Derivation

The functional derivation applies the functional analysis Pahl and Beitz (Pahl & Beitz, 2013) to the product under study to obtain the product functions. The functional analysis consists of defining functions and sub-function (main and auxiliary) through a hierarchical scheme and basic fluxes that link these functions (Kroll, 2013). Basic fluxes identified by Pahl and Beitz (Pahl & Beitz, 2013) are material, energy, and signal. In the proposed methodology a modified version of the original functional analysis is used to characterize the given product with the support of a dedicated tool. Indeed, within the same functional flow (e.g., material flow) different types of fluxes are determined and associated to a given colour (Figure 13). For instance, assuming that two generic functions are connected by a flux of material, it is possible to further specify the type of material (e.g., gas) and the sub-type (e.g., air). A given colour is assigned to the type of material (e.g., green for gas type) and a unique RGB code is assigned to the sub-type (e.g., (7, 255, 62) for the air). The outcome of the modified approach is a graph presenting as many colours as different fluxes available for the total functions. The colour assignment is important to address the issue related to people with colour vision deficiency (CVD) (Nuñez, et al., 2018). Indeed, a general colour was assigned for each type of flow, then a gradation of the colour is proposed for sub-flows to provide a CVD-safe colour map. Once colours are assigned, they must not be changed to avoid inconsistencies during the application of the CDfA approach on other case studies. The presented functional derivation enables users to: (i) improve readability of the functional representation, (ii) increase the level of detail of the functional analysis without requiring data from a lower design phase, (iii) provide better understanding of the implication of each requirement, and (iv) facilitate the switch from fluxes (functional representation) to physical interconnections (physical representation).

Figure 13 shows all basic fluxes identified within an aircraft. These fluxes were defined by the authors following the previous work of Hirtz (Hirtz, et al., 2002). Figure 14 shows an example of the functional decomposition graph obtained. Fluxes are then used to derive modules interfaces and subsequently their physical interconnections to create modules.



Aircraft Basic Fluxes								
Basic Flow	*Symbol*	*Type*	*Explanation*	*Example*	*Sub-Type*	*Type of Colour*	*Colour*	*RGB*
Material	⇧	Human	All or part of a person who physically interacts with the system	Cabin crew entering the cabin	*N.A.*	Black	⬛	*(0,0,0)*
		Solid	Object with mass and shape which physically interacts with the system	Luggage entering the hat rack	Waste	Brown	🟫	*(205,133,63)*
					Objects /Parts		🟫	*(171,85,24)*
		Liquid	Fluid which physically interacts with the system	Fuel that flows in pipes	Liquid waste	Blue	🟦	(0,191,255)
					Fuel		🟦	(30,144,255)
					Oil		🟦	(72,61,139)
					Water		⬜	(230,230,250)
		Gas	Gas that physically interacts with the system	Air entering the fans and ducts	Air	Green	🟩	(7,255,62)
					Gas mixture		🟩	(63,243,76)
					O2		🟩	(177,247,171)
Energy	↑	Human	Work performed by a person on the system	Energy generated by the pilot to move the cloche	*N.A.*	Black	⬛	*(0,0,0)*
		Acoustic	Work performed to produce and trasmitt sound	Energy generated by turboprop motion converted into noise	*N.A.*	Orange	🟧	(255,127,80)
		Chemical	Work resulting from chemical reactions	Energy produced by aircraft batteries	*N.A.*	Green	🟩	(124,252,0)
		Electric	Work resulting from motion of electrons	Energy transmitted by aircraft harnesses	*N.A.*	Yellow	🟨	(255,255,0)
		Hydrauli	Work performed by moving fluids	Energy used to actuate the landing gear	*N.A.*	Blue	🟦	(167,167,255)
		Thermal	Work resulting from a themal system	Energy exchanged in the aircraft cooling system	*N.A.*	Purple	🟪	(138,43,226)
		Pneumati	Work resulting from the motion of gas	Energy used by the pnemuatic aircraft system	*N.A.*	Brown	🟫	(165,42,42)
		Mechanic	Work performed by a mechanical system	Energy generated by a mechanical connection to fix the seat on the cabin	*N.A.*	Red	🟥	(255,0,0)
Signal	⇡	Control	Command sent to an apparatus to regulate it	Pilot that regulates air in the cabin	Sound	Red	🟥	(250,128,114)
					Tactile		🟥	(220,20,60)
					Visual		🟥	(255,0,0)
		Feedback	Information about the state of the system	Indicator of the fuel level in the aircraft	Visual	Purple	🟪	(238,130,238)
					Sound		🟪	(255,0,255)
					Tactile		🟪	(148,0,211)

Figure 13 - Basic fluxes for aircraft; extension of the basic fluxes provided by Hirtz (Hirtz, et al., 2002)

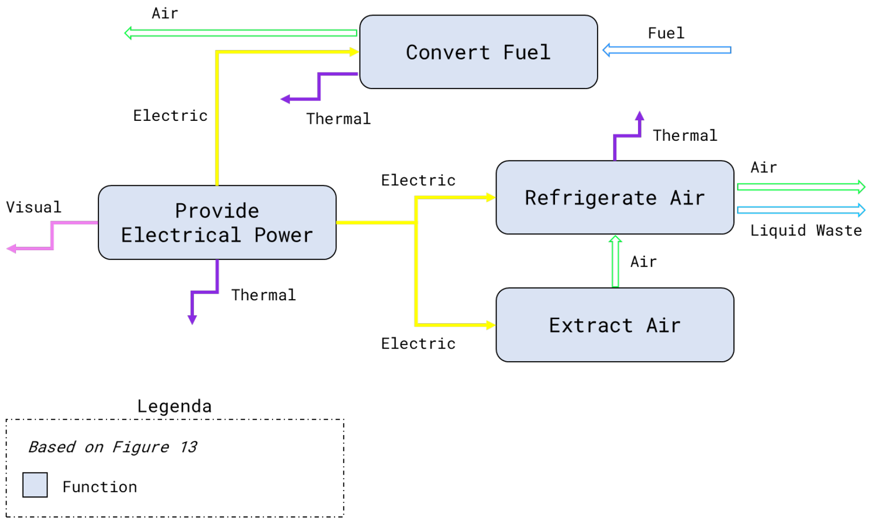
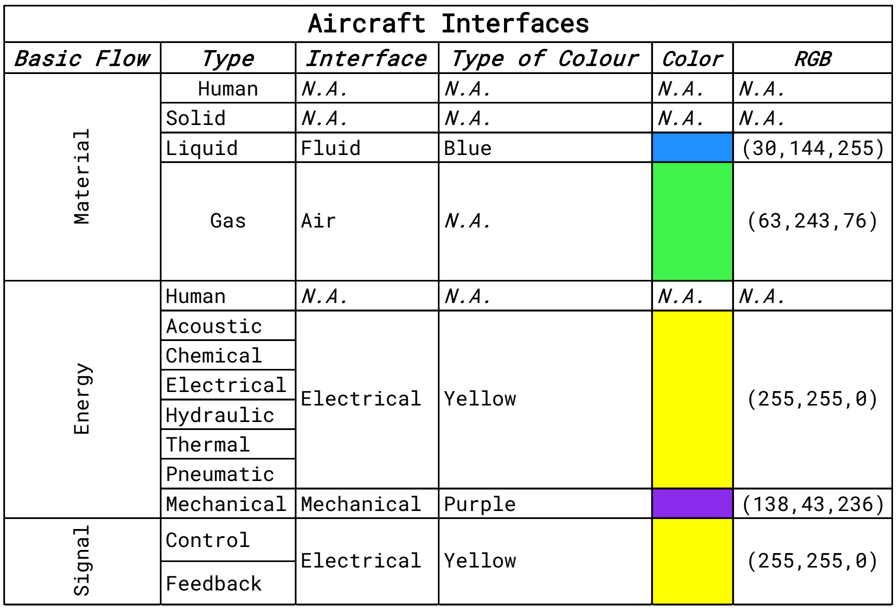


Figure 14 - Functional Decomposition Graph obtained using the proposed approach.

The functional decomposition allows obtaining several benefits such as product abstraction, functions characterization and fluxes identification. Nevertheless, it presents some shortcomings, in fact the process is laborious, time-consuming and requires collaborative sessions to successfully define the functional scheme. Furthermore, some functions can only be analysed in the context of a particular solution, limiting the design space and the ability of designers to think in abstract terms (Kroll, 2013). As well as, when the level of details in functional decomposition is too deep, it may lead to a lack of freedom for the designer adversely affecting innovation and creative performance (Leenders, et al., 2007). Due to the specificity of the aeronautical field and aircraft product development, which is mainly characterized by very stringent regulations about safety requirements and high cost of changes, the conceptualization of new ideas requires a long time to be formalized, discussed, accepted, validated and tested. Within this framework, the criticisms highlighted for the method proposed by Pahl and Beitz (Pahl & Beitz, 2013), although partially restrict the potential of this method, do not negatively affect the development of the CDfA methodology which focuses on the assessment of product architectures in terms of manufacturing and assembly. Besides, functional representation is a powerful tool to develop new module concepts and proceed towards the design of new architectures.

Module Derivation

Once the functional scheme is obtained, it is necessary to derive functional modules. The derivation of modules requires moving from functions to interfaces. Interfaces represent how functions physically interact with the system of interest. Interfaces are derived considering the basic flows that interact with the system and their type. The colour is inherited by the basic flux colour type (i.e., general colour) while, at this stage, the arrow's type becomes meaningless. Thus, for the presented CDfA methodology, all interfaces are represented with solid arrows. It is worth noting that, while the functional representation identifies all the interactions that are present in the system of interest, some of them might not need an interface to connect with the system. For instance, the cabin crew interacts with the system to store luggage within the cabin (usually called hat-rack), and, in this case, the interface necessary to connect these two modules is the human interface. Since the purpose of the CDfA methodology is to analyse product assembly and system installation, the human interface is meaningless because the action to store luggage within the cabin is performed by the human. Following this principle, for aircraft systems only four interfaces are derived starting from the identified fluxes: i) electrical, ii) mechanical, iii) fluid, and iv) air. Figure 15 reports the four interfaces considered for the aircraft systems and the related matching with the basic flow types.



Aircraft Interfaces

Basic Flow	*Type*	*Interface*	*Type of Colour*	*Color*	*RGB*
Material	Human	*N.A.*	*N.A.*	*N.A.*	*N.A.*
	Solid	*N.A.*	*N.A.*	*N.A.*	*N.A.*
	Liquid	Fluid	Blue	🟦	(30,144,255)
	Gas	Air	*N.A.*	🟩	(63,243,76)
Energy	Human	*N.A.*	*N.A.*	*N.A.*	*N.A.*
	Acoustic Chemical Electrical Hydraulic Thermal Pneumatic	Electrical	Yellow	🟨	(255,255,0)
	Mechanical	Mechanical	Purple	🟪	(138,43,236)
Signal	Control Feedback	Electrical	Yellow	🟨	(255,255,0)

Figure 15 - Aircraft interfaces

Once interfaces are defined, it is possible to proceed with the creation of modules. Modularisation aims at clustering the functions into modules with specific interfaces (Ericsson & Erixon, 1999). Module derivation from the functional analysis is a key step at the conceptual design stage, whether novel design or redesign is desired. It provides an engineering view of how the sub-functions work together to achieve the desired functional requirements, independently of how the function is performed. The engineering definition of modules in aircraft systems presents many concerns related to the huge number of constraints that need to be satisfied (e.g., the presence of redundant elements placed in different areas for safety reasons). The method proposed by Stone (Stone, et al., 2004) is adopted within this methodology with the aim to consider all the constraints required for the development of modules. This method is based on three heuristics (dominant flow, branching flow, and transmission/conversion) and it allows identifying product modules by grouping sub-functions together. The list of modules obtained by using this method can be used to generate concepts. Among the different methods developed for module derivation (e.g., module heuristics, design structure matrix, modular function deployment, etc.), the module heuristics is the most suitable for the scope of this work, since it shows important features which fit with the type of product under analysis and the level of confidence required to develop the modules. Module heuristics can capture flows describing the underlying physics of the product and it is more flexible than other methods that require a matrix or mathematical description (Borjesson, 2010). The method presents a high repeatability on a given function structure diagram, which is the outcome of the functional derivation phase, demonstrated by Hölttä-Otto (Hölttä-Otto, 2005). The module heuristics is well supported by empirical research on hundreds of real products; however, it requires an engineering judgement (theoretical foundation is less scientific than the other methods) to be operated and it does not guarantee that the identified modules represent the optimum clusters. Nevertheless, it provides a consistent and repeatable product structure breakdown. The application of the module heuristics suffers of software integration/implementation which would be beneficial for large projects as in the case of aircraft. This drawback is counterbalanced by flexibility; indeed, it is worth noting that to increase the level of confidence in the definition of suitable modules in such product, a mapping with existing modules for a given product is possible. In this case, the module heuristics method (top-down approach) is coupled with the analysis of available product structure (bottom-up approach) with the aim to match the existing modules with the ones retrieved by module heuristics. If the goal of the CDfA method is the optimization of a given architecture, this task is necessary and it will lead to two results: (i) maintain the level of confidence about modules derivation in relation to aircraft products that are characterized by many design constraints and, (ii) identify possible alternative solutions for modules definition (i.e., module splitting/merge). On the other hand, if the goal is to assess a new concept (i.e., concepts that are newly developed), module breakdown is fully based on module heuristics (top-down approach) increasing the design solution space but downgrading the level of confidence in the module definition. A one-to-one mapping between modules and functions is the easiest way to consider aspects such as safety or operability requirements. However, this option can bring to product architectures with more modules (more options and a higher level of modularity) and give more importance to component interfaces assemblability (Engel, et al., 2017). Hence, having more modules is not always the right way to proceed in this phase and the use of module heuristics allow an engineering assessment of feasible modules based on initial requirements and given design decisions (e.g., combustion of fuel for power generation). Once modules are defined, it is possible to create the product physical architecture by linking modules with interfaces derived in the previous step (Figure 16).

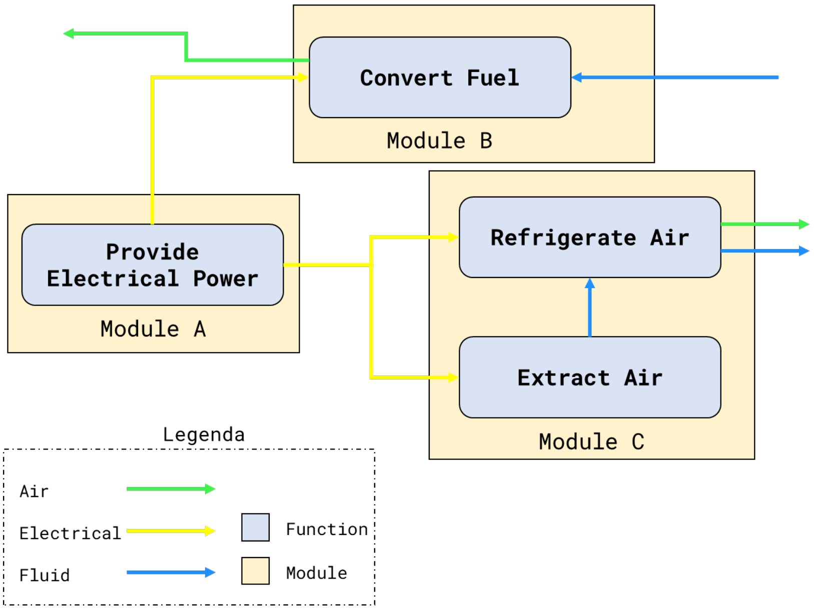


Figure 16 - Example of module derivation

PHASE 2: ARCHITECTURE GEOMETRICAL DEFINITION

The Architecture Geometrical Definition consists in reading the data available derived in the previous phase and translating them into numerical format. This phase is composed of three (3) steps: (i) Definition of geometrical product architecture, (ii) Definition of c*onceptual* Bill of Materials, and (iii) Definition of knowledge scoring matrices (knowledge-based).

Definition of geometrical product architecture

The assembly assessment at the conceptual level requires the translation of conceptual features into parameters that can be visualized and measured. The information derived in the previous steps (i.e., functional modules and functional schemes) are used to create a virtual representation of the product under study. The use of a virtual environment (i.e., CAD tool) enables to represent data available at the conceptual phase into elementary geometries. The *simplified* Digital Mock Up (sDMU) is a graphical and geometrical representation of a specific product architecture showing modules shape and interfaces among modules in a three-dimensional space. The sDMU is composed of 3D geometrical items such as boxes, cylinders, etc. (Figure 17). The sDMU provides a visual representation of the product architecture obtained in the previous steps, enriching the product architecture itself with a new set of information. It presents more detailed information such as the distribution of modules inside the system of interest (module position), the overall module shape and volume (module bounding box) and many others. Information represented in the sDMU is enclosed in the functional scheme, for example by knowing how many functions and the type of functions collected in a functional module, parameters such as module position, module bounding box, length of interfaces, etc. can be estimated. The level of detail and the granularity of the sDMU evolves during the application of the methodology. If the methodology is used to analyse new products, then the sDMU starts with a low level of detail (i.e., low granularity) and it is enriched with more detailed information when later design phases are approached. On the other hand, if the methodology is based on an existing product of which a CAD file is available, it is possible to simplify the CAD file by neglecting all details and representing modules as boxes with their bounding box and interfaces as cylinders whose diameters reflect the real dimension. However, the geometrical product architecture cannot be derived solely from functional information; expert design involvement is required. Expert designers, for example, must support decisions about the physical location of items and the size of the bounding box.

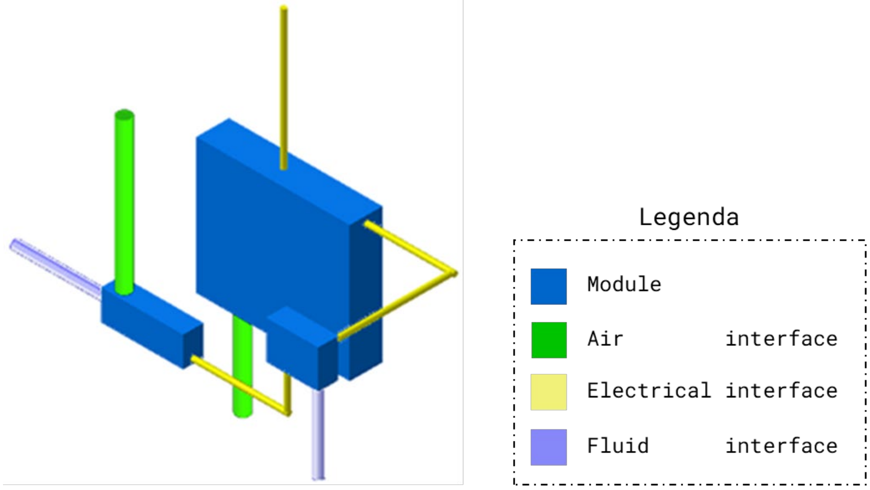


Figure 17 - Example of a *simplified* Digital Mock Up

The sDMU is providing the minimum set of data enabling to perform the assembly assessment. The required set of data necessary to build a sDMU is reported in Table 5.

Table 5 - required data provided by sDMU

sDMU item	Information collected in sDMU
Module	Position (x, y, z) based on a given reference Shape (i.e., rectangular box, cylinder) Bounding Box Colour (i.e., blue)
Interface	Position (x, y, z) based on a module in/out reference Path Overall length Shape (i.e., cylinder) Size (i.e., diameter) Colour (based on interface type)

Definition of conceptual Bill of Materials

The *conceptual* Bill of Materials (cBoM) is a document aiming at capturing the functional and geometrical data of the previous steps to enable computation of the assembly scoring. The cBoM presents a table-form in which each row represents an interface between modules and each column represent an attribute that is characterizing a particular interface. The cBoM is characterized by a hierarchical structure, subdivided into levels (layers), domains, and attributes. The hierarchical structure is the methodology framework that allows combining attributes for a given system (i.e., the overall assembly or a sub-assembly). The cBoM structure enables a decomposition of the problem in sub-problems allowing to incorporate aspects from different level of granularity that otherwise might be discharged. The proposed framework is a description of the product enabling to identify the impact of each attribute on the assembly process. Table 6 and Table 7 list, respectively, the information for interfaces and modules that must be included in the fixed information section according to the analysed element.

Table 6 - Fixed Information for the cBoM document

Element	**Name**	**Type**	**Description**
Interface	Interface Type	*string*	it identifies the type of interface (i.e., fluid, air, electrical, and mechanical); it is compliant with interfaces identified in the functional scheme
	Name	*String*	it is the name associated to the interface under investigation (i.e., F for fluid, A for air, E for electric, and M for mechanical)
	ID	*integer*	it describes the ID of the interface under study. It can be generated according to a specific rule (progressive number)
	Module IN	*string/integer*	it represents the module where the interface starts
	Module OUT	*string/integer*	it represents the module where the interface ends

Table 7 - Module fixed information for the cBoM document

Element	Name	Type	Description
Module	Module Type	*string*	it identifies the type of module (i.e., equipment, valve, filter, etc.)
	Name	*string*	it is the name associated to the module under investigation. It can be chosen arbitrarily by the user
	ID	*integer*	it describes the ID of the element under study. It can be generated according to a specific rule (progressive number), or it can be chosen arbitrarily by the user

If necessary, other information can be added within this framework, depending on the level and the type of product under study.

Levels definition (layers)

A level is defined as a group of data which is modelling the main feature characterizing a specific sub-problem of the overall product assembly. Different levels can be defined according to the available information. To switch from a level to another one is necessary to identify the product invariant. A product invariant is a design feature that does not change and cannot be changed within the product under study. The definition of the invariant allows specifying information with respect to it. For each defined level, the following actions are necessary: (i) to identify possible invariants that link two neighbouring levels, and (ii) to express the relation that exists between two neighbouring levels using invariants.

For example, if the "space distribution" (i.e., product areas) in a product is fixed and cannot be changed; then the "space distribution" can be considered as an *invariant*. The identified invariant allows splitting the global analysis into sub-problems that are limited in terms of complexity (problem discretization).

Attributes and Domains definition

Attributes and Domains that characterize the main criteria of the assembly performances are defined on a knowledge basis, with a concurrent engineering approach (i.e., involvement of manufacturing department, architecture designers, operators, etc.). A mathematical model is then created to operate on these criteria providing the assembly assessment.

Attributes Class definition

An attribute (A) is a key feature that influences assembly operations. Giving a generic attribute A [1 to x; x ∈ N>0], it describes a specific aspect related to the assembly operation. To define an attribute, it is necessary to indicate: (i) the name of the attribute, (ii) the dimension (i.e., the unit measure or the quantity), and (iii) the level in which it is available. To define an attribute, it is required a deep study of the product under development. A list of key attributes reflecting the assembly complexity can be obtained knowing the design phase in which the methodology is applied (e.g., conceptual phase, detail phase, etc.). Attributes might be of interest or not, according to the level in which they are placed. Attributes places on lower levels (i.e., level 2, level 3, etc.) are less important than attributes on main levels.

Domain Class definition

A domain (D) is a cluster of one or more attributes (A) that address the same assembly aspect and provides the same meaning for each designer/engineer. Giving a generic domain D [1 to t; t ≤ x; t ∈ N>0], the domain D is a vector characterized by *n* attributes [n ≥ 1; n ∈ N>0]. For each level of the cBoM framework, domains can have a different number of attributes. By clustering attributes into domains, and placing domains into levels, the hierarchical structure is obtained (Figure 18).

The definition of the cBoM may be supported by sensitivity analysis (SA) approaches. Indeed, the use of SA may help designers and engineers to better understand the effects that the defined domains and levels have on the mathematical model.

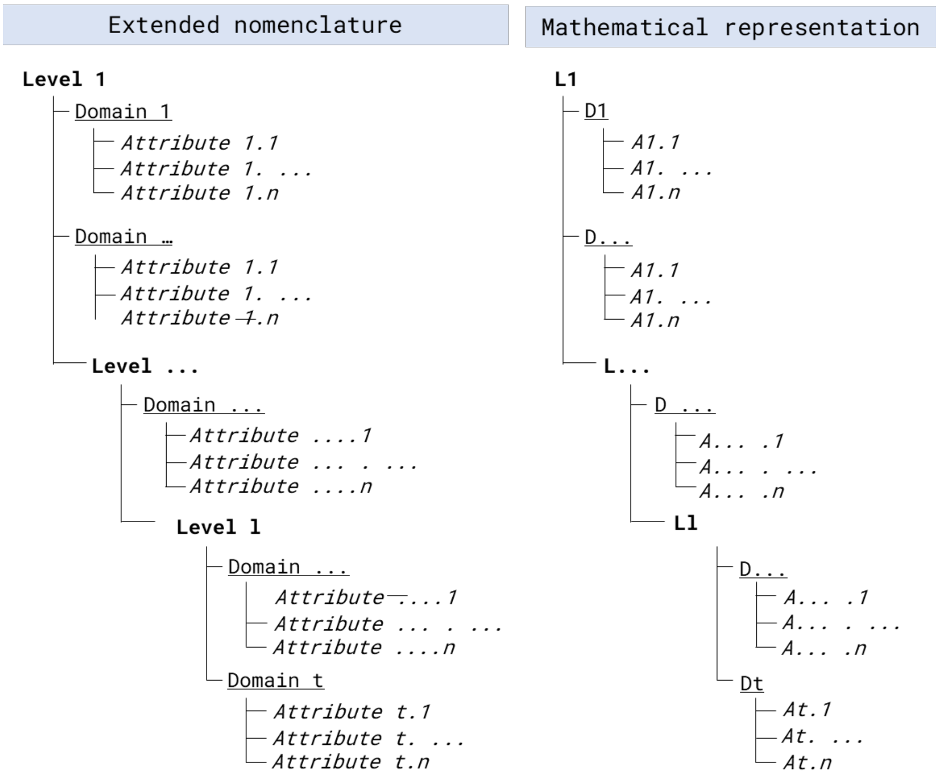
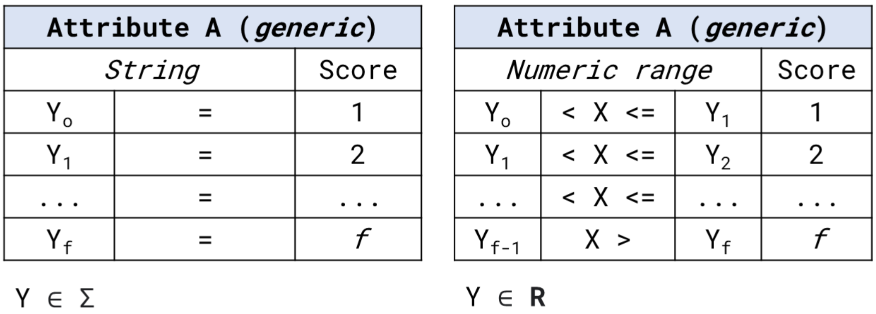


Figure 18 - cBoM framework (hierarchical structure)

Knowledge scoring matrices definition

The formalization of the knowledge is a great challenge and the research field associated to it is called Knowledge Engineering (KE). Relevant knowledge might appear in different forms (e.g., technical drawings, spreadsheets, etc.) and in unstructured manners. The main challenge is how to deal with it in terms of collections, formalization, and utilization (Staab, et al., 2001) (Ahmed, 2005) (Reed, et al., 2011). The approach proposed within this methodology to formalize the knowledge focuses on two aspects: knowledge acquisition and ontology definition (Guarino, 1995). The knowledge acquisition relies on a knowledge-based concurrent approach (Favi, et al., 2019). The method enables to retrieve the knowledge through the definition of scoring matrices. The structure and the vocabulary (i.e., ontology) of the knowledge must follow three principles: (i) role-limiting, (ii) knowledge typing and (iii) reusability (Musen & Schreiber, 1995). The ontology proposed within this methodology is based on the definition of knowledge scoring matrix (kSM). The knowledge scoring matrix is table of Y_f rows, where each row corresponds to a score (i.e., from 1 to m). The kSM form is different according to the information it processes. For example, if the kSM translates strings information (i.e., Y ∈ **Σ** with **Σ** be the set of alpha-numerical strings without the null element), then to each string a score is associated (Figure 19 - sx). On the other hand, if the kSM translates numerical data (i.e., Y ∈ **R** with **R** be the set of real numbers), a score is associated to a given range of values a (Figure 19 - dx). The CDfA approach was defined using a scale of 1 to 5 (Likert scale), as it was deemed the most appropriate for the specific problem.



Attribute A (*generic*)		
String		Score
Y_o	=	1
Y_1	=	2
...	=	...
Y_f	=	*f*

$Y \in \Sigma$

Attribute A (*generic*)			
Numeric range			Score
Y_o	< X <=	Y_1	1
Y_1	< X <=	Y_2	2
...	< X <=	...	...
Y_{f-1}	X >	Y_f	*f*

$Y \in \mathbf{R}$

Figure 19 - kMS definition for string (sx) and numeric data (dx)

The kSM is necessary to translate the collected information into dimensionless values and perform mathematical computations. The kSM shall be created for each attribute in the cBoM and is composed of: (i) name of the attribute, (ii) numerical range or string, and (iii) score. It is important to notice the maximum score available for all the knowledge scoring matrix must be the same. Each kSM is defined considering the available industrial capabilities in terms of assembly technologies for the analysed product. Whenever a novel assembly process is implemented and/or the production facility is upgraded, then the kSM must be updated accordingly. Since the goal of the kSM is to collect and translate the tacit knowledge into explicit knowledge, the validation of the data collected inside the kSM may be performed only empirically, using surveys. Indeed, surveys must be submitted to all people involved in the assembly process of the system of interest (i.e., engineers, blue collars, technicians, etc.). In fact, by increasing the number of people and the type of audience, the kSMs are validated at their best.

PHASE 3: CONCEPTUAL DESIGN FOR ASSEMBLY ASSESSMENT

The architectural analysis is divided into two phases: (i) Normalization process, and (ii) Architectural assessment.

Normalization process

The normalization process consists in transforming the data inside each attribute into dimensionless value switching from heterogeneous data (i.e., string, number, etc.) to homogeneous data (i.e., dimensionless scores) to perform mathematical operations. The normalization process requires two inputs: (i) the knowledge scoring matrices, and (ii) data for each attribute. For example, if the interface 1 presents a generic value for the attribute "length" (i.e., 3,5 [m]) then the normalization process will translate the value into a dimensionless score according to the knowledge scoring matrix associated to the attribute "length" (i.e., 1). Once the normalization process is completed for all attributes, all data in the cBoM are dimensionless, enabling to proceed further with the architecture assessment.

The mathematical model used for the normalization task is presented here below. Starting with the cBoM framework, it is possible to identify four variables:

- *Level l* with $l \in [1, L]$ where L is the overall number of levels
- *Domain d* with $d \in [1, D]$ where $D = D(l)$ indicates the overall number of domains belonging to level l
- *Type t* with $t \in [1, T]$ with t representing the element's type (e.g., interface, module, etc.) and T is the total number of available types
- *Element e* with $e \in [1, E]$ where $E = E(l,t)$ indicates the overall number of elements of type t collected at the level l
- *Attribute a with* $a \in [1, A]$ *where* $A = A(l,t)$ indicates the overall number of attributes identified for the product analysed

Following the definition above, it is possible to define the generic qualitative kSM $\underline{\underline{Q}}(a,t)$, which is used for converting strings into scores, and the generic quantitative kSM $\underline{\underline{P}}(a,t)$, which is used for converting numerical values into scores, as follow:

$$\underline{\underline{\mathrm{Q}}}(a,t) = \begin{bmatrix} o_1 & v_1 \\ o_2 & v_2 \\ \vdots & \vdots \\ o_\mathrm{n} & v_\mathrm{n} \end{bmatrix}$$

(I)

$$\underline{\underline{\mathrm{P}}}(a,t) = \begin{bmatrix} r_1 & R_1 & w_1 \\ r_2 & R_2 & w_2 \\ \vdots & \vdots & \vdots \\ r_{\mathrm{n}} & R_{\mathrm{n}} & w_{\mathrm{n}} \end{bmatrix}$$

(II)

Where:

- $o_i = o_i(a,t)$ and $v_i = v_i(a,t)$ with $i \in [1,n]$ are, respectively, a unique numerical value that identifies one of the n possible values of the qualitative kSM and, $v_i = v_i(a,t)$ represents the associated normalized value.
- $r_i = r_i(a,t)$ and $R_i = R_i(a,t)$ with $i \in [1,n]$ identify, respectively, the lower and the upper limit of the ranges for the quantitative kSM as $[\mathrm{r_i}, \mathrm{R_i}[$, and $w_i = w_i(a,t)$ represents the associated normalized value.

Thus, the set of kSMs $\underline{\underline{S}}(a,t)$ is defined as:

$$\underline{\underline{S}}(a,t) = \underline{\underline{\mathrm{Q}}}(a,t), \underline{\underline{\mathrm{P}}}(a,t) \mid q \in [1, \underline{\underline{\mathrm{Q}}}(a,t)], \mathrm{p} \in [1, \underline{\underline{\mathrm{P}}}(a,t)]$$

Considering E = E(l,t) the total number of elements of *type t* in the *level l*, it is possible to define the attribute's vectors $\underline{a}(e,l,t,d)$ with e ∈ [1,E] as:

$$\underline{a}(e,l,t,d) = \left(\mathrm{a}_1, \dots, \mathrm{a}_0, \mathrm{a}_{(\mathrm{Q}+1)}, \dots, \mathrm{a}_{(\mathrm{Q}+\mathrm{P})}\right)$$

with $a_1, \dots, a_0$ represent qualitative attributes that require normalization (i.e., the o_i elements in the matrix (I)), while $\mathrm{a}_{\mathrm{Q}+1}$, …, $\mathrm{a}_{\mathrm{Q}+\mathrm{P}}$ indicates quantitative attributes that require normalization (i.e., the r_i elements in the matrix (III)).

Each level l is characterized by $E^*(l,t) = \mathrm{E}(l-1,t)$ elements of type t which are inherited from the level above. If *l=1*, it is assumed that $E^*(1,t) = \mathrm{E}(1)$

The matrix of attributes $\underline{\underline{\mathrm{A}}}(e^*,d)$ with e^* ∈ [1, E^*] is defined as:

$$\underline{\underline{\mathrm{A}}}(e^*,d) = \begin{bmatrix} \mathrm{a}\left(e_{s_1}, d\right) \\ \mathrm{a}\left(e_{s_2}, d\right) \\ \vdots \\ \mathrm{a}\left(e_{s_m}, d\right) \end{bmatrix} \text{with } \left[\mathrm{e}_{\mathrm{s}_1}(\mathrm{e}^*), \ \dots, \mathrm{e}_{\mathrm{s}_m}(\mathrm{e}^*)\right] \subset [1, E(l,t)]$$

(III)

which is the mathematical representation of the element subdivision according to the *invariant*. Indeed, the overall number of elements is always the same (i.e., E(*l,t*)) but, as the level increases, elements can be subdivided into sub-elements (i.e., e_{s1}, …, e_{sm}).

To obtain dimensionless values (i.e., scores), it is necessary to normalize them using kSM (I) (II). A generic attributes' vector $\underline{a}(e, l, t, d)$ is composed of Q qualitative attributes and P quantitative attributes, then it is possible to define $\underline{a}_{norm}(e, l, t, d)$ the vector of normalized attributes as

$$\underline{a}_{norm}(e, l, t, d) = \left(\mathrm{a}_{\mathrm{norm}_1}, \dots, \mathrm{a}_{\mathrm{norm}_\mathrm{k}}, \mathrm{a}_{\mathrm{norm}_\mathrm{k}+1}, \dots, \mathrm{a}_{\mathrm{norm}_\mathrm{k}+\mathrm{w}}\right)$$

which is composed of a_k elements deriving from quantitative kSMs (I), and a_{k+w} elements deriving from qualitative kSMs (II).

Substituting the vector of normalized attribute inside the matrix of attribute $\underline{\underline{A}}(e^*, d)$ (III) is possible to obtain the normalized matrix of attributes $\underline{\underline{\mathrm{A}}}_{\mathrm{norm}}(e^*, d)$

$$\underline{\underline{A}}_{norm}(e^*, d) = \begin{bmatrix} \mathrm{a}_{\mathrm{norm}}\left(\mathrm{e}_{\mathrm{s}_1}, \mathrm{d}\right) \\ \mathrm{a}_{\mathrm{norm}}\left(\mathrm{e}_{\mathrm{s}_2}, \mathrm{d}\right) \\ \vdots \\ \mathrm{a}_{\mathrm{norm}}\left(\mathrm{e}_{\mathrm{s}_\mathrm{m}}, \mathrm{d}\right) \end{bmatrix}$$

(IV)

Once the normalization process is completed for all attributes, it is possible to proceed further with the architecture assessment.

Architectural assessment

The architectural assessment task consists of several mathematical steps which allows obtaining, from information collected inside the cBoM framework, one single score for each analysed element (module or interface). The score for each element represents the fit for assembly analysis and provides a ranking of critical modules/interfaces.

Starting with the normalized matrix of attributes, it is possible to defined the function H(·) = H(A$_{\mathrm{norm}}$,l,t,d) as $\mathrm{H}: \mathbb{R}^{(Ax(P+Q))} \rightarrow \mathbb{R}^{(Ax1)}$ with A = dim(A,1) which transforms the normalized attributes matrix (IV) into the domain vector $\underline{d}(e^*, d)$:

$$\mathrm{H}(\mathrm{A}_{\mathrm{norm}},\mathrm{l},\mathrm{t},\mathrm{d}) = \begin{bmatrix} h\left(\mathrm{a}_{\mathrm{norm}}(\mathrm{e}_{\mathrm{s}_1},\mathrm{d})\right) \\ h\left(\mathrm{a}_{\mathrm{norm}}(\mathrm{e}_{\mathrm{s}_2},\mathrm{d})\right) \\ \vdots \\ h\left(a_{norm}(e_{s_m},d)\right) \end{bmatrix} \rightarrow \begin{bmatrix} d_1 \\ \vdots \\ d_A \end{bmatrix} = \underline{d}$$

(V)

The function $H(\cdot)$ is applied for each element of *type t,* belonging to the *level l* and *domain d* to obtain one score for each element for each domain.

The function $H(\cdot)$ is a general function which has the following characteristic:

$$\frac{dh}{d_{a_i}} \geq 0 \quad \forall i \,\epsilon\, [1, Q+P]$$

and it is a positive function.

To move inside levels, it is required to define the function $G(\cdot) = G(\underline{d}, l, t, d)$ as $\mathrm{G}: \mathbb{R}^{(Ax1)} \rightarrow \mathbb{R}$ that takes as input the generic domains' vector $\underline{d}(e^*, d)$ and provides as output the domain score *D(e*(l-1),t,d)*:

$$G(\underline{d}, l, t, d) = \begin{bmatrix} G(d_1) \\ \vdots \\ G(d_A) \end{bmatrix} \rightarrow D(e * (l-1), t, d)$$

(VI)

Where the function $G(\cdot)$ has the general characteristics:

$$\frac{dg}{d_{d_i}} \geq 0 \quad \forall \mathrm{i} \,\epsilon\, [1, \mathrm{A}]$$

and it is a positive function.

Assuming all normalized attributes' matrix has been obtained for each value of *l, d, t* and *e,* by fixing the variable *l, d* and *t* it is possible to obtain the domain's vectors for each value of $e^* \epsilon\ [1, E^*(l,t)]$ with the function $H(\cdot)$ (V).

Now, performing two operations iteratively for *(l-1)* times:

1. Computation of scores D (VI) using the function $G(\cdot)$ for each element $e^* \epsilon [1, E^*(l-1, t)]$
2. Identification of domain's vector for each element $e^* \epsilon\ [1, E^*(l-1, t)]$ as:

$$\underline{\mathrm{d}}(\mathrm{e}^*, \mathrm{l}-1, \mathrm{t}, \mathrm{d}) = \begin{bmatrix} G\left(e_{s_1}(e^*, l^*-1, d, t)\right) \\ G\left(e_{s_2}(e^*, l^*-1, d, t)\right) \\ \vdots \\ G\left(e_{s_m}(e^*, l^*-1, d, t)\right) \end{bmatrix}$$

It is possible to obtain the domain's vector $\underline{D}(0, t, d) = \underline{D}(t, d)$ at the main level.

Performing the same operation keeping the domain fixed but changing the type, it is possible to obtain the vector *type* at the main level $\underline{T}(0, t, d) = \underline{T}(t, d)$ defined as:

$$\underline{\mathrm{T}}(0, t, d) = \begin{bmatrix} t(0,1, d) \\ \vdots \\ t(0, T, d) \end{bmatrix} = \begin{bmatrix} t(1, d) \\ \vdots \\ t(T, d) \end{bmatrix} = \underline{T}(t, d)$$

Keeping the level fixed and changing the domains, it is possible to obtain the level $\underline{\underline{L}}(l)$ matrix, which is defined as:

$$\underline{\underline{L}}(l) = \left[\underline{T}(l, 1) \ldots \underline{T}(l, T)\right]$$

By extending the process to all the L levels, it is possible to obtain the matrix of the main level $\underline{\underline{M}}$ defined as:

$$\underline{\underline{M}} = \left[\underline{\underline{L}}(1) \ldots \underline{\underline{L}}(L)\right]$$

The matrix related to the main level is the mathematical representation of the main level where each element for each domain has a single score associated.

In the last step is necessary to apply the function $F(\cdot)$ such as $\mathrm{F(M)}: R^{(E_{tot}\ x\ D)} \rightarrow R^{(E_{tot}\ x\ 1)}$ which translates the domains' matrix at the main level, into the final score vector $\underline{C}$:

$$\mathrm{F(M)} = \begin{bmatrix} f(m_1) \\ \vdots \\ f\left(\mathrm{m}_{\mathrm{E}_{\mathrm{tot}}}\right) \end{bmatrix} = \begin{bmatrix} C_1 \\ \vdots \\ C_{E_{tot}} \end{bmatrix} = \underline{C}$$

where:

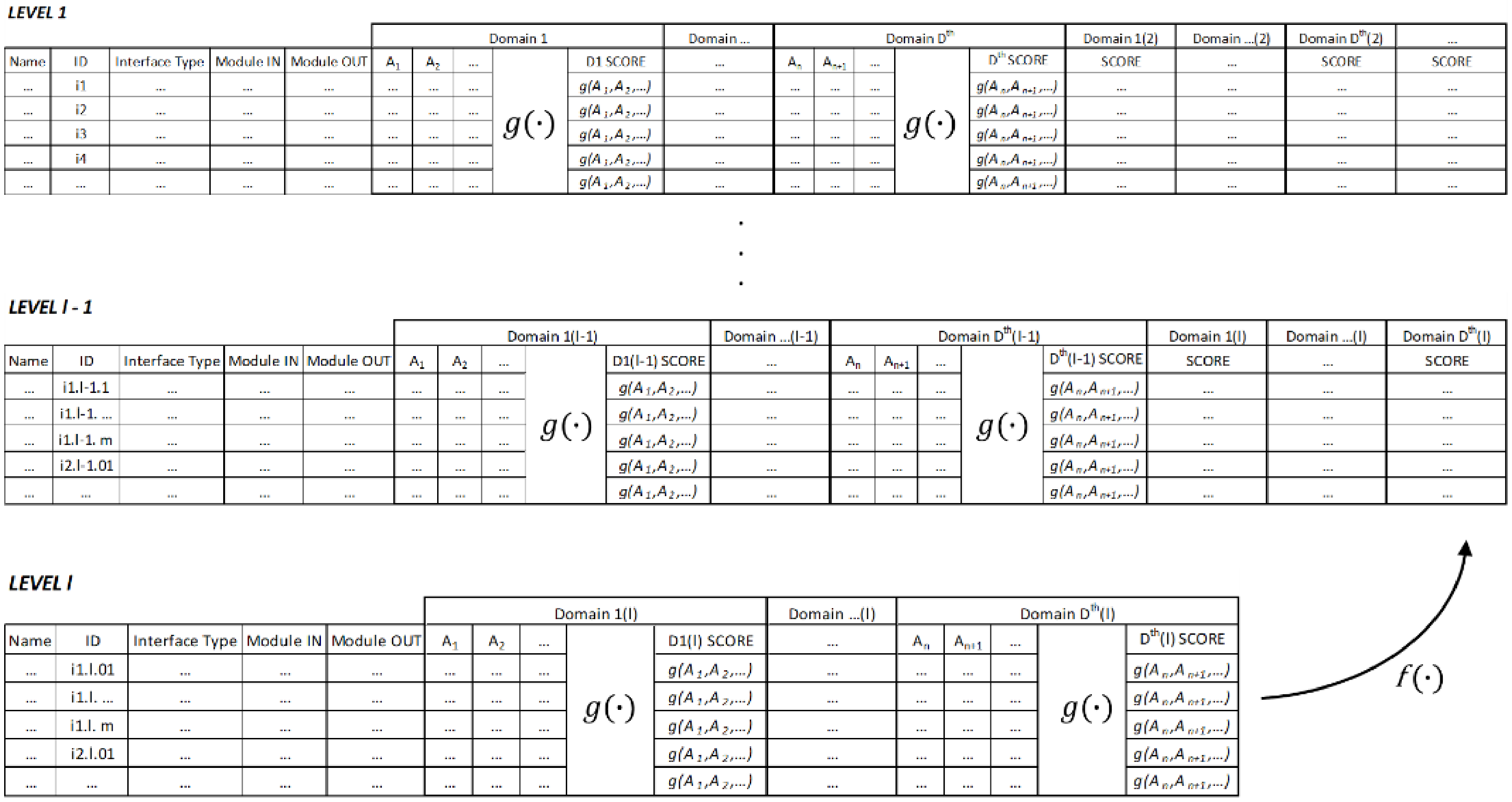
- $\mathrm{E}_{\mathrm{tot}}$ indicates the sum of all elements at the main levels, for each type T:

$$E_{tot} = \sum_{t=1}^{T} E(l = 1, t)$$

- m$_i$ with $i \epsilon [1, E_{tot}]$ represents the generic row vector of the main level matrix.

The vector of final score $\underline{C}$ is composed of one score for each element, for each element type, at the main level. Analysing the vector $\underline{C}$ is possible to understand which element type (interface or module) is the most impacting from the assembly point of view. According to the mathematical function chosen, it might be the one with the highest or the lowest score. The output of this process is represented in Figure 20.

The choice of mathematical operator to use for each function (i.e., $H(\cdot)$, $G(\cdot)$, $F(\cdot)$) is made according to different aspects. In the literature there are several mathematical operators that can be used to collect scores. For the proposed methodology, functions can be classified into two types: weighted operators and weight-less operators. The former allows the application of weight to the obtained results. Multi-Attribute-Decision-Making (MADM) techniques are included in this category (i.e., Technique for Order of Preference by Similarity to Ideal Solution - TOPSIS). The mean operator, the root mean square (RMS), and the average square, on the other hand, are all weight-less operators. The mathematical operator is determined by the following factors: (i) the invariant selected in the analysis, (ii) the uncertainty influencing the input data, and (iii) the weight assigned to outliner/inliner data. The mean operator is used to collect data that do not contain discrepancy, for instance domains' scores are collected using the mean operator since, through the data normalization and score collection, domains' scores do not contain discrepancy. On the other hand, the RMS is used to collect scores when data might present discrepancy due to i) errors in the data collection process, ii) the creation of clusters of information which are not strictly related (e.g., *total length of harness* and *number of connections* belong to the same domain, but these attributes are not directly affecting each other). Thus, RMS is usually adopted to cluster scores of different attributes within a domain.



LEVEL 1

					Domain 1					Domain ...	Domain D^{th}					Domain 1(2)	Domain ...(2)	Domain D^{th}(2)	...
Name	ID	Interface Type	Module IN	Module OUT	A_1	A_2	...		D1 SCORE	...	A_n	A_{n+1}	...		D^{th} SCORE	SCORE	...	SCORE	SCORE
...	i1	...	...	...	...	...	...	$g(\cdot)$	$g(A_1,A_2,...)$	...	...	...	...	$g(\cdot)$	$g(A_n,A_{n+1},...)$	...	...	...	...
...	i2	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$	...	...	...	...
...	i3	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$	...	...	...	...
...	i4	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$	...	...	...	...
...	...	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$	...	...	...	...

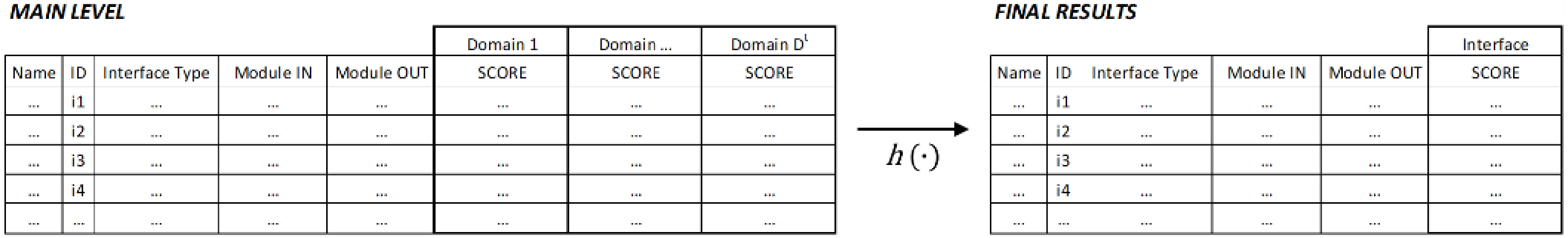
LEVEL l - 1

					Domain 1(l-1)					Domain ...(l-1)	Domain D^{th}(l-1)					Domain 1(l)	Domain ...(l)	Domain D^{th}(l)
Name	ID	Interface Type	Module IN	Module OUT	A_1	A_2	...		D1(l-1) SCORE	...	A_n	A_{n+1}	...		D^{th}(l-1) SCORE	SCORE	...	SCORE
...	i1.l-1.1	...	...	...	...	...	...	$g(\cdot)$	$g(A_1,A_2,...)$	...	...	...	...	$g(\cdot)$	$g(A_n,A_{n+1},...)$	...	...	...
...	i1.l-1. ...	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$	...	...	...
...	i1.l-1. m	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$	...	...	...
...	i2.l-1.01	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$	...	...	...
...	...	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$	...	...	...

LEVEL l

					Domain 1(l)					Domain ...(l)	Domain D^{th}(l)				
Name	ID	Interface Type	Module IN	Module OUT	A_1	A_2	...		D1(l) SCORE	...	A_n	A_{n+1}	...		D^{th}(l) SCORE
...	i1.l.01	...	...	...	...	...	...	$g(\cdot)$	$g(A_1,A_2,...)$	...	...	...	...	$g(\cdot)$	$g(A_n,A_{n+1},...)$
...	i1.l. ...	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$
...	i1.l. m	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$
...	i2.l.01	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$
...	...	...	...	...	...	...	...		$g(A_1,A_2,...)$	...	...	...	...		$g(A_n,A_{n+1},...)$

Figure 20 - Mathematical model used collect information inside a level and to switch from one level to another one



MAIN LEVEL

					Domain 1	Domain ...	Domain D^L
Name	ID	Interface Type	Module IN	Module OUT	SCORE	SCORE	SCORE
...	i1	...	...	...	...	...	...
...	i2	...	...	...	...	...	...
...	i3	...	...	...	...	...	...
...	i4	...	...	...	...	...	...
...	...	...	...	...	...	...	...

$\xrightarrow{h(\cdot)}$

FINAL RESULTS

					Interface
Name	ID	Interface Type	Module IN	Module OUT	SCORE
...	i1	...	...	...	...
...	i2	...	...	...	...
...	i3	...	...	...	...
...	i4	...	...	...	...
...	...	...	...	...	...

Figure 21 - Output of the CDfA assessment (results)

The process is repeated for all levels, clustering the results of all interfaces, with the aim to obtain a single score for each domain. Interfaces and scores are collected in the final domain called "Main Level" which presents scores for all domains defined in the cBoM and all interfaces of the product under study. The final output is presented in Figure 21. Once final scores have been computed, it is necessary to set a threshold and check those elements that lie above or below the threshold, according to the function used.

The scores describe the component assembly complexity considering all the parameters involved in the assembly and installation processes. Scores can be sorted out from lowest to highest to identify elements (interfaces and modules) that are the most complex to install inside the aircraft and that are required to be redesigned.

3.3 CDFA REDESIGN PHASE

The CDfA Redesign phase allows performing the redesign of the product architecture in a systematic manner. The redesign is based on results obtained from the CDfA assessment phase. Using a structured method, it is possible to identify the most impacting design guidelines that must be implemented to improve the assembly property of the identified critical elements. In the following sub-paragraphs, the CDfA Redesign phase is described in detail.

PHASE 4: PRODUCT ARCHITECTURE REDESIGN

The redesign of the product architecture is composed of four steps, that guide the user through the modification of elements identified as critical. In the following paragraphs, each phase is described in detail.

Definition of Design Guidelines

The first step consists of deriving a list of design guidelines (DGs). A design guideline is a design activity that can be performed to modify the design of given architecture (i.e., the position of a module, the shape/geometry of the module, the routing of an interface). A design guideline may improve or reduce the score of a module or an interface, changing the performances of parameters defined in the CDfA model. The framework to collect design guidelines is composed of:

- **ID** – it is an incremental number which identify the design guideline uniquely.
- **Source** – it represents the source of the design guideline. The source can be engineering knowledge or knowledge scoring matrices.
- **Domain of interest** – it describes the domain which is mainly affected by the design guideline.
- **Affected attributes** – it identifies the attribute that is mainly affected by the design guideline.
- **Design guideline** – it explains the design guideline. It must be in the form of verb + object.
- **Explanation** – it details the design guideline proving more information.

The list of design guidelines can be obtained in two different ways. The first one is based on the scoring matrices associated to the CDfA method. Scoring matrices represent the way in which implicit knowledge is translated into explicit knowledge. They represent critical aspects observed during the assembly phase providing rationale behind the issues observed in the assembly line. For this reason, the lowest score of the scoring matrix associated with a given attribute represents the best design option to implement. The other method consists of collecting the engineering knowledge and developing new innovative solutions using brainstorming sessions. To generate the greatest number of design guidelines, brainstorming sessions should be conducted concurrently. In fact, different specialists from various departments should be involved in the formulation of design guidelines. This approach allows achieving completely new design solutions from different point of view. However, some of the design guidelines obtained might not be feasible or implementable yet. In fact, they might represent a technology which is still under development and not yet implemented inside the firm. The design guideline must still be gathered in this situation, but a remark should be included to indicate that it will not be considered. The main goal of the brainstorming session is to wider the solution space previously obtained with the scoring matrix approach.

Design Correlation Matrix derivation

Once the list of design guidelines is derived, the Design Correlation Matrix (DCM) is created. The DCM is a matrix [t x n] where rows (t) represent design guidelines while columns (n) represent the attributes identified within the CDfA methodology. The goal of the DCM is to determine the influence of each design guideline on the cBoM attributes. The DCM aggregates design guideline impact, which can vary from -2 to +2 with negative values indicating an increase in assembly complexity for the considered attributes, and positive values indicating an improvement (Figure 22).

It is worth noting that for the sake of creating the DCM, all attributes are considered on the same level. Indeed, no hierarchical structure is created.

ID	*Attribute 1*	*Attribute 2*	*Attribute 3*	*Attribute 4*	*Attribute 5*
1	0	2	2	0	0
2	0	0	0	1	2
3	-1	-1	0	0	0

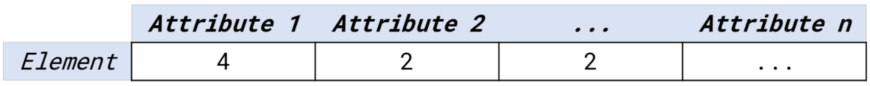
-2 Elevate negative impact
-1 Moderate negative impact
0 No impact
1 Moderate positive impact
2 Elevate positive impact

Figure 22 - Design Correlation Matrix [3 x 5]

Elements' vector derivation

The third step focuses on the analysis of the CDfA hierarchical structure to obtain the normalized attributes' scores of each element, and thus, to create the element vector (EV).

The third step is the derivation of the element vector (EV) for a given component. The element vector is a vector [1, 2, …, n] composed of *n* items, where *n* is the overall number of attributes in the cBoM (Figure 23). Each item contains the score of the associated attribute. For instance, if the element vector of module A is desired, then item 1 represents the score of attribute 1 for module A; item 2 represents the score of attribute 2 for module A; and so on. The element vector can be derived for all elements analysed in the CDfA methodology (i.e., modules or interfaces), but only the most critical component's element vector can be derived to keep the analysis simpler.



	Attribute 1	*Attribute 2*	...	*Attribute n*
Element	4	2	2	...

Figure 23 - Element Vector [1, 2, ..., n]

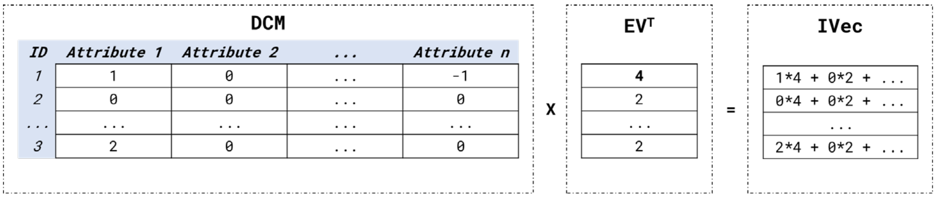
Impact vector computation

The Design Correlation Matrix and the element vector derived in the previous phase are combined to provide a vector, known as impact vector (IVec). The impact vector is a vector [1, 2, …, t] representing the impact of each design guideline on the analysed component (Figure 24).

The impact vector is calculated by multiplying the two quantities in equation (VII):

$$\underline{IVec} = \underline{\underline{\mathrm{DCM}}} * \underline{EV}^{t}$$

(VII)



ID	*Attribute 1*	*Attribute 2*	...	*Attribute n*
1	1	0	...	-1
2	0	0	...	0
...	...	...	...	...
3	2	0	...	0

EV^T
4
2
...
2

IVec
1*4 + 0*2 + ...
0*4 + 0*2 + ...
...
2*4 + 0*2 + ...

Figure 24 - Design guideline score vector

Finally, by selecting the item with the maximum value of the impact vector, the most impacting design guideline for the analysed elements is obtained. In other words, the greatest improvement in the assembly property of the studied component will be achieved by implementing the modification indicated by the design guideline with the highest score (i.e., Ivec).

If elements can be divided into families, meaning they can be clustered into groups of elements which share same features (e.g., seats modules in the cabin of an aircraft, etc.), data can be visualized in three different ways:

- **Globally** – The overall impact vector matrix is considered. This visualization is useful to have a rough idea of which design guideline might provide the best improvement overall.
- **Family related** – It consists of dividing the results per category. For instance, clustering all impact vectors assessing modules belonging to the category "Toilets". This visualization is useful to provide an estimation of what design guideline provides the best improvement for a given element family.
- **Element related** – It consists of analysing the single element's impact vector. This visualization helps focus on a single element and suggests the best design guideline to improve the element's assemblability.

Elements' Modification

Once the preferred design guideline (i.e., the one with the highest score) is selected, it can either be implemented or not. In fact, some design principles might point to a solution that is not yet available within the organization. It could, for example, be in the process of being developed or awaiting certification. If the first DG with the highest IVec score cannot be performed, the second or third DG shall be chosen.

When the chosen design guideline is implemented, a new product architecture is obtained. After that, a new CDfA analysis should be made to see if the critical elements' scores have changed. Indeed, the proposed workflow in combination with the CDfA assessment would be used in an iterative manner to reduce the product architecture score. Indeed, a sub-optimal product architecture can be achieved by reducing the assembly complexity of the new most critical element.

4 CASE STUDIES

The Conceptual Design for Assembly methodology was applied to assess the assemblability of complex systems, such as aircraft systems. The assembly process can be divided into two main areas: i) assemblability of interfaces and ii) assemblability of modules. For each area, the CDfA method requires the definition of a conceptual Bill of Material (i.e., hierarchical model). The assemblability of interfaces can be divided into two processes, the installation of interfaces on the structure and the installation of interfaces on modules (Figure 25).

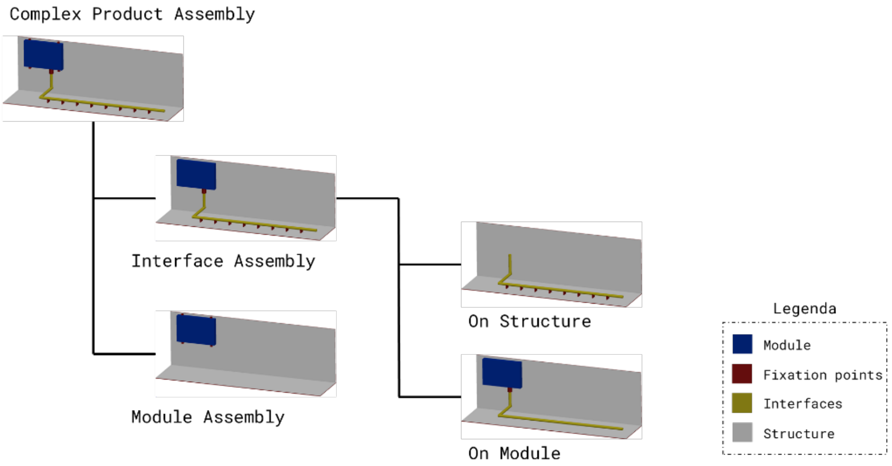


Figure 25 - Complex product assembly decomposition

The CDfA method can be used to assess the assembly of each area for complex products. The hierarchical model associated with the assessment of the assemblability of interfaces can be analysed with different mathematical models, to obtain different scores and, indeed, information. In the following sections, the method was applied to assess two aircraft complex systems: i) the nose-fuselage and ii) the cabin.

Among all aircraft sub-systems, the nose-fuselage is the most complex since it collects several modules and interfaces. In fact, the main functions of the aircraft are performed by the nose-fuselage. The nose-fuselage case study aims at studying the system architectures for both modules and interfaces. Assemblability complexity of all three types of processes is considered. In other words, the assembly complexity refers to the installation process of interfaces on the skin, interfaces on modules, and modules on the skin are considered. To do so, two different hierarchical structures had to be developed. Finally, an improved nose-fuselage architecture created following the redesign phase is proposed and analysed with respect to the original one.

The cabin sub-system presents several functional constraints. For instance, the need of accommodating a given number of people limits the design choice regarding the number and position of seats. The same can be said for other elements such as toilets, galley, etc. As a result, the cabin system was examined just from the perspective of a module assembly, meaning only the assembly complexity of modules on the skin is considered. Only one hierarchical structure had to be created in order to accomplish this. Finally, modifications suggested from the redesign phase were implemented and differences between initial and optimized product architectures were discussed.

4.1 NOSE-FUSELAGE SYSTEM

The proposed methodology was applied to assess the assembly performances of the nose-fuselage of a civil aircraft (Figure 26). Within the nose-fuselage, many modules and connections are present (e.g., electrical cables, air ducts, hydraulic pipes, etc.) and they need to be installed and assembled within very confined areas. Furthermore, the nose-fuselage architecture is limited by several factors, including the need to place elements inside an already-existing structure (i.e., the aircraft skeleton), the need for redundancy for safety reasons, and so on.

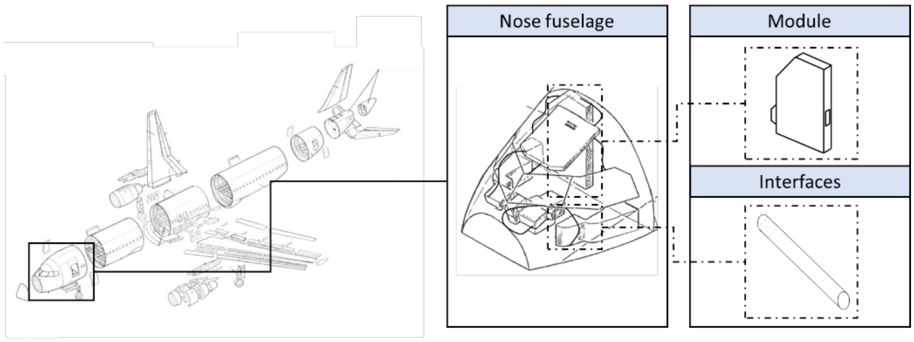


Figure 26 - Aircraft Nose-Fuselage

The goal of the presented case study is to numerically assess the nose-fuselage architecture with the aim to identify interfaces and modules considered critical for the installation process, providing a list of the most critical items to install. Finally, design improvements are identified within the product re-design phase. In the following paragraphs, each phase of the CDfA method is presented and described with respect to the considered case study.

PHASE 1: PRODUCT FUNCTIONAL DECOMPOSITION

The Nose-Fuselage Functional Decomposition is composed of: (1) Functional Derivation, and (2) Module Derivation. The Functional Analysis was performed to derive all the nose-fuselage functions and interfaces among them. Functions were derived according to Pahl and Beitz (Pahl & Beitz, 2013) using information available in literature (e.g., equipment functions, connection among equipment, etc.). The functional scheme was obtained considering a civil aircraft already available from the market. In fact, it is necessary to start from design actions already available to be able to derive a fully functional scheme. Functions were derived using several documents among which technical documents describing systems used in aircraft, technical drawing of the nose-fuselage, use and maintenance manual, accident, and malfunctions reports, etc. The analysis of these documents allowed obtaining information regarding systems inside the aircraft, how they interact, and how they are subdivided. The general function "*Fly, manage flight and allow passengers entrance and carry*" was considered, and then it was subdivided into more specific functions with a hierarchical structure. Four levels were obtained. Figure 27 shows an extract of the second level (the overall functional scheme is not provided due to confidentiality).

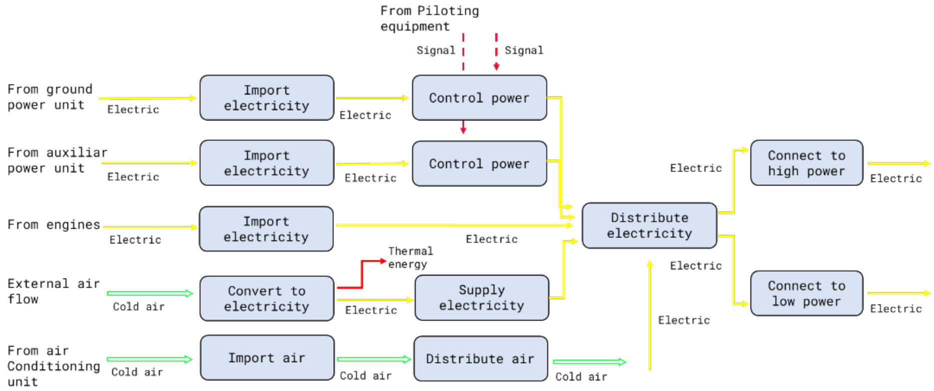


Figure 27 - extract of Functional Scheme (4th level)

Once the functional scheme was obtained, a top-down approach based on module heuristics (Stone, et al., 2004) was coupled with a bottom-up approach to derive modules. From basic fluxes identified in the functional scheme, it was necessary to switch to interfaces. To reduce the overall analysis perimeter, only *electrical* and *air* interfaces were considered within this work since from engineering experience they were considered the most complex to install (Hermansson, et al., 2013). In this specific case, since it is based on a real product and the scope of the analysis is the evaluation of a given product architecture through the mean of the CDfA, the module heuristics was coupled with the analysis of available product structure (overlapping existing modules with the ones retrieved by module heuristics). It is worth noticing that in the product architecture development of aircraft systems, the optimization phase is mainly characterized by the module re-arrangement which means the possibility to split/merge modules, module reallocation, interface routing, re-architecture of interconnection among modules. Thus, the list of modules is mainly driven by the available solutions, and modules derivation using module heuristics is an exercise done to investigate possible alternatives (split/merge modules) for module definition based on functional decomposition. An extract of the derived modules is shown in Figure 28.

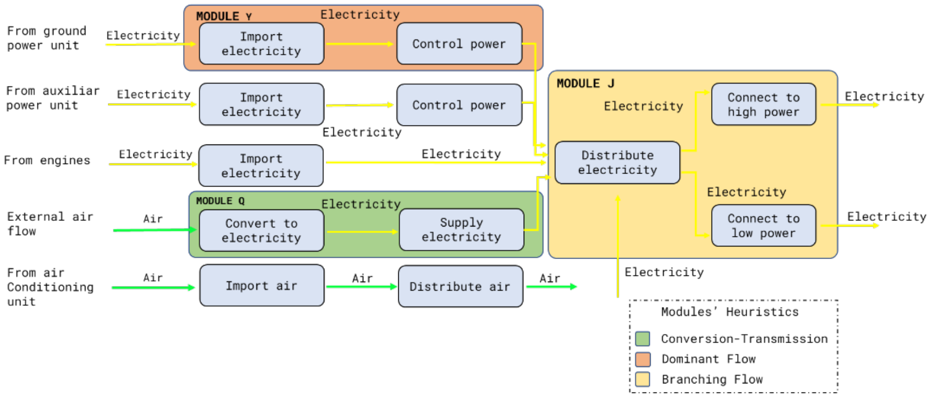


Figure 28 - Extract of Modules derivation

PHASE 2: ARCHITECTURE GEOMETRICAL DEFINITION

The *simplified* Digital Mock Up was created following the conceptual modules, and architectural data were created making use of existing product definition information, in particular: (i) the bounding box of each module, (ii) the length of connections (interfaces) among modules, (iii) the modules and connections position, and (iv) the connections diameters. Rectangular boxes were used to model modules while cylinders for interfaces. Interfaces colours are based on Figure 15. For the case study in exam, since 3D models were already available, with the help of virtual reality technologies it was possible to draw interfaces, modules connections, and bounding boxes easily and in a straightforward manner. The sDMU obtained is presented in Figure 29.

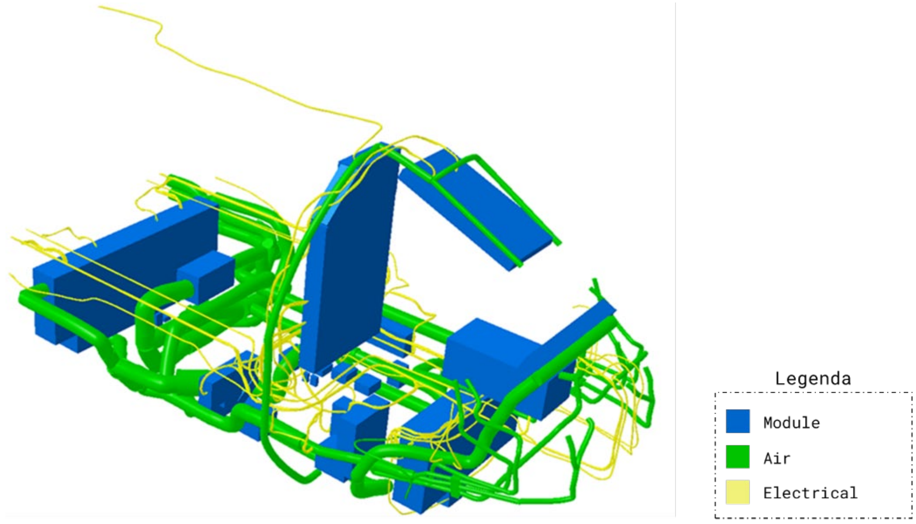


Figure 29 - Nose-Fuselage simplified Digital Mock Up

Two conceptual Bill of Materials were derived, one to perform interfaces assessment and one to perform module assessment. For the interface assessment, the cBoM is composed of three levels (layers) of data: Level 1, Level 2, and Level 3. Level 1 is composed of only one domain called "Interface Domain" which includes attributes referring to the overall interfaces among modules. To switch from Level 1 to Level 2 the invariant working area was defined. Working areas are pre-defined areas identifiable in the nose-fuselage. These areas cannot be changed due to structural reasons (i.e., design of beams, skin, and floor). The invariant working area is used also to define attributes' scores inside knowledge scoring matrices. Each attribute has a different score according to the working area in which is defined (Table 8). For instance, in the working area "Right Bay" the attribute "Zone" for the value "Middle" has a score of 4, while for the working area "Cockpit" the same attribute with the same value (i.e., Middle) has a score of 1.

Table 8 - Extract of kSM for attribute "Zone" (II level)

Invariant (Working Area)	Zone	Score
Right Bay	Upper	5
Right Bay	Middle	4
Right Bay	Lower	4
Cockpit	Upper	3

Invariant (Working Area)	Zone	Score
Cockpit	Middle	1
Cockpit	Lower	3

Level 2 is composed of two domains called "Ergonomic Domain" and "Assembly Domain". The first one includes four attributes referring to ergonomic aspects of the installation process, while the second one presents two attributes representing the complexity of the installation process itself.

To switch from Level 2 to Level 3 the invariant interface was defined. Interfaces are defined from the functional analysis and cannot be changed without changing the product. The invariant interface was used to define attributes' scores inside knowledge scoring matrices (Table 9).

Table 9 - Extract of kSM for Attribute "Length" (III Level)

Invariant (Interface)	Length	Score
Air	0 <x <= 2	1
Air	2 < x < 5	2
Electrical	0 < x <= 5	1
Electrical	5 < x < 10	2

Level 3 presents one domain called "Component Domain" which includes attributes referring to physical elements composing the interface.

The cBoM used to assess nose-fuselage modules is composed of only one level (Level 1) and two domains (i.e., Mechanical Domain and Handling Domain). The framework for the module assessment is presented in Figure 30.

Main level

Module Type	Name	ID	d1	d2
…	…	…	…	…

Level 1

			d1		d2		
Module Type	**Name**	**ID**	**a1(d1)**	**a2(d1)**	**a1(d2)**	**a2(d2)**	**a3(d2)**
…	…	…	…	…	…	…	…

Figure 30 - cBoM framework for modules assessment

The overall list of attributes defined for each cBoM framework is presented in Table 10.

Table 10 - List of attributes for interface and module assessment

Assessment	Domain	Domain ID	Attribute	Attribute ID	Explanation
Interface assessment	Interface domain	d1	Total Length of Ducts	a1(d1)	Air interface length
			Branches	a2(d1)	Number of times air interface branches out
			Total Length of Harness	a3(d1)	Electrical interface length
			Number of Connections	a4(d1)	Number of connections in the electrical interface
			Number of Straight Nodes	a5(d1)	Number of times electrical interface branches out

Assessment	Domain	Domain ID	Attribute	Attribute ID	Explanation
Interface assessment	Ergonomic domain	d2	Working Areas	a1(d2)	Area in which installation operations are performed
			Access	a2(d2)	Access used to bring the interface inside the working area
			Zone	a3(d2)	Zone in which interface is installed
			Working Space Size	a4(d2)	Available space during the installation operations
	Assembly domain	d3	Variety of Tools	a1(d3)	Number of tools necessary to perform the assembly
			Process	a2(d3)	Complexity of the installation process
	Component domain	d4	Air Bends	a1(d4)	Number of air ducts elbow
			Air Shape	a2(d4)	Shape of the air duct
			Air Weight	a3(d4)	Weight of the air duct
			Air Piece Length	a4(d4)	Length of the air duct

Assessment	Domain	Domain ID	Attribute	Attribute ID	Explanation
Interface assessment	Component domain	d4	Electrical Weight	a5(d4)	Weight of the electrical cable
			Electrical Piece Length	a6(d4)	Length of the electrical cable
			Fragility	a7(d4)	Breakability of the duct/cable material
	Mechanical Domain	d1	Number and Position of Mechanical Interfaces	a1(d1)	Number of module anchors' points and relative position
			Access	a2(d1)	Access used to bring the module inside the working area
	Handling Domain	d2	Tool/ Assistant	a1(d2)	Number of tools and operators to perform the assembly
			Weight	a2(d2)	Weight of the module
			Clearance	a3(d2)	Space available to perform assembly operations

The creation of the cBoM framework was supported by sensitivity analysis (SA) methods. SA allows understanding the relative importance of each attribute and each domain within the framework providing a tangible tool to support the framework modification towards more suitable and accurate results. The SA was performed on both cBoM frameworks. The method called "One-Factor-At-Time" was chosen to perform the analysis (Saltelli, et al., 2006). To understand how each parameter influences the overall output, the method consists of changing the value of one parameter while leaving the others constant. In Figure 31 an extract of the SA performed on the modules assessment framework (cBoM) is shown. Results exhibit attributes belonging to the Mechanical Domain have a higher impact on the overall result with respect to the Handling domain. The reason lies in the domain composition: the fewer is the number of attributes per domain, the higher is the impact of each domain on the final score (Formentini, et al., 2021.a). On the other hand, considering the framework for interfaces assessment, which is characterized by more than one level, attributes belonging to the lower levels (i.e., level 3) have less impact on the final score with respect to attributes belonging to higher levels (i.e., level 1).

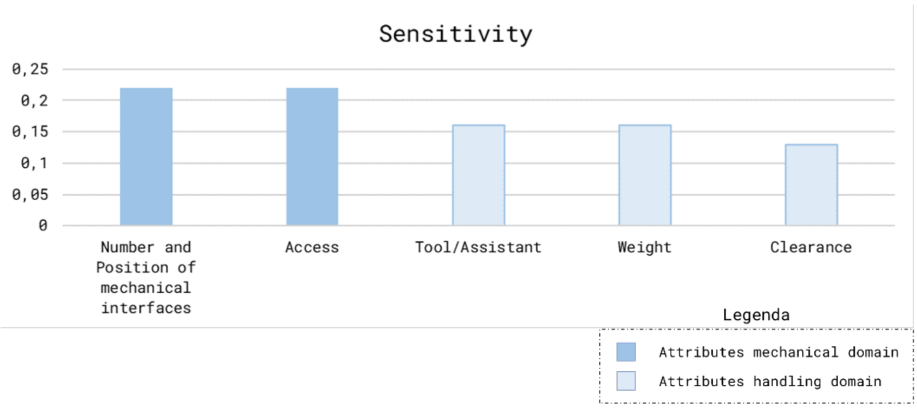


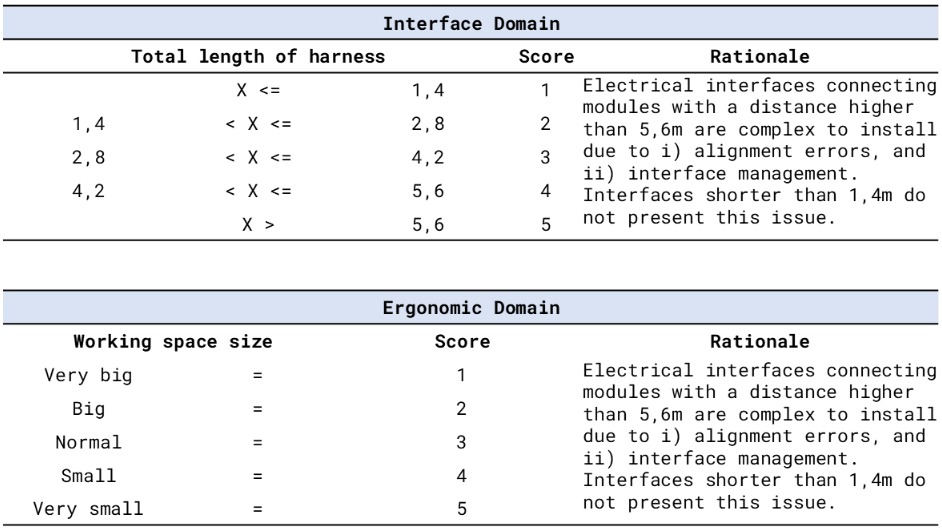
Figure 31 - Sensitivity Analysis for attributes inside a domain

After the definition of the cBoM structure, the knowledge scoring matrices for each attribute were defined according to the ontology developed. Four meetings were organized: (i) an initial in person meeting, (ii) two follow-up web meetings and (iii) a final in person review meeting. Industry departments involved in the meetings are collected in Table 11.

Table 11 - Meeting's participants

Department	Participants	Meeting			
		First	Second	Third	Fourth
Product Architecture & Design	Aircraft Architect	X		X	X
Product Architecture & Design	DMU operators	X		X	X
Product Architecture & Design	System installation designers	X		X	X
Manufacturing & Assembly	Industrial architect	X	X		X
Manufacturing & Assembly	Industrial routing designers	X	X		X
Manufacturing & Assembly	Manufacturing operators (blue-collar)	X	X		X
Manufacturing & Assembly	Ergonomic expert	X	X		X

During the initial meeting (first meeting), the methodology was presented. Then, in the first web meeting (second meeting) a survey was submitted to the manufacturing department to collect expertise from assembly operations and related tasks. Then, the results of the survey were analysed to obtain the first draft of the kSMs. The second web meeting (third meeting) was organized with the Product Architecture & Design department to show the kSMs obtained and a few modifications were suggested. In the final meeting, the latest version of kSMs was presented and finalized. An extract of the kSMs obtained is presented in (Figure 32).



Interface Domain				
Total length of harness			Score	Rationale
	X <=	1,4	1	Electrical interfaces connecting modules with a distance higher than 5,6m are complex to install due to i) alignment errors, and ii) interface management. Interfaces shorter than 1,4m do not present this issue.
1,4	< X <=	2,8	2	
2,8	< X <=	4,2	3	
4,2	< X <=	5,6	4	
	X >	5,6	5	

Ergonomic Domain			
Working space size		Score	Rationale
Very big	=	1	Electrical interfaces connecting modules with a distance higher than 5,6m are complex to install due to i) alignment errors, and ii) interface management. Interfaces shorter than 1,4m do not present this issue.
Big	=	2	
Normal	=	3	
Small	=	4	
Very small	=	5	

Figure 32 - Derived knowledge Scoring Matrix for two different attributes (total length of harness and working space size)

PHASE 3: CONCEPTUAL DESIGN FOR ASSEMBLY ASSESSMENT

To perform the final step, which is the Conceptual Design for Assembly Assessment, it was necessary to: i) normalize all information collected in the cBoM using the derived knowledge scoring matrices, and ii) provide the mathematical algorithm to collect the normalized data inside the cBoM to obtain a final global score for each interface and module.

The mathematical algorithm was defined using three different operators: i) Root Mean Square operator, ii) Mean operator and iii) TOPSIS method.

The RMS was chosen to collect attributes' scores for each domain. The reason lies in the need to consider possible errors, in fact, initial data present different roots (i.e., some data are measured, others are derived by engineering knowledge) and they might present some evaluation errors. The use of RMS allowed obtaining a more conservative result with respect to other mathematical operators.

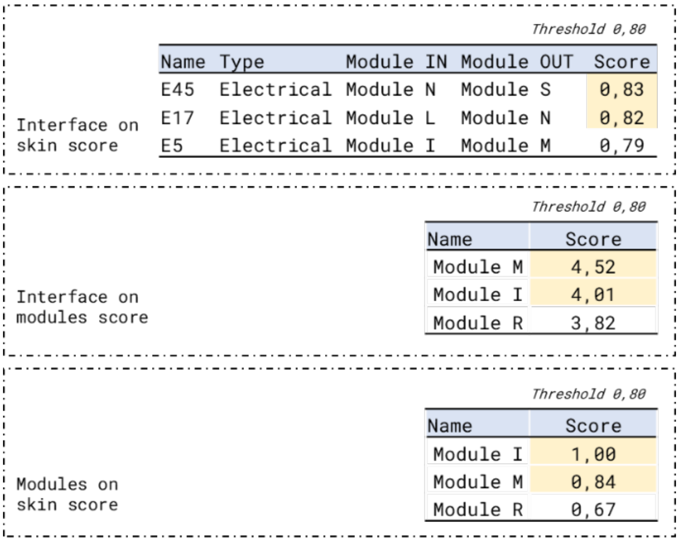
The Mean operator was chosen to collect scores at each level (i.e., from Level 3 to Level 2, from Level 2 to Level 1). In fact, data are initially collected in the cBoM, then, using the knowledge scoring matrices, they are normalized (i.e., data are translated into scores) and collected per domain.

After the normalization process and the domains' collection, scores have all the same roots, and no further source of error needs to be considered.

The TOPSIS was used to obtained one single score for assessing the assemblability of interfaces on the skin (i.e., final global score) for each interface from domains' scores. The TOPSIS method was chosen since it allows applying weights on each domain to tune the overall assessment. In fact, due to the nature of the methodology itself, in the model definition some attributes and domains might be underestimated or overestimated in terms of assembly complexity. This may lead to some shortfalls that can be recovered afterward making use of weights. Indeed, weights can be used to increase/decrease the importance of the final global score of each domain. However, for the specific case-study, no weights were added since results showed to be in line with engineering judgment.

The score to assess assemblability of interfaces on the module (i.e., Final interface-on-module score) was obtained by collecting the final global score of each air and electrical interface for each module through the mean operator. Finally, the obtained score was normalized with respect to the maximum value to get a score ranging from 0 to 1. In this way, it was possible to switch from an interface point of view to a module point of view.

The TOPSIS was not used for the module assessment, thus the Mechanical Domain scores and the Handling Domain scores were collected using the Mean operator. This choice reflects the fact that the framework for module assessment is simpler, characterized by only two domains and one level. An extract of the final scores obtained is shown in Figure 33, where the most critical interfaces and modules (i.e., most complex to install) are displayed and highlighted in the sDMU.



Threshold 0,80

Interface on skin score

Name	Type	Module IN	Module OUT	Score
E45	Electrical	Module N	Module S	0,83
E17	Electrical	Module L	Module N	0,82
E5	Electrical	Module I	Module M	0,79

Threshold 0,80

Interface on modules score

Name	Score
Module M	4,52
Module I	4,01
Module R	3,82

Threshold 0,80

Modules on skin score

Name	Score
Module I	1,00
Module M	0,84
Module R	0,67

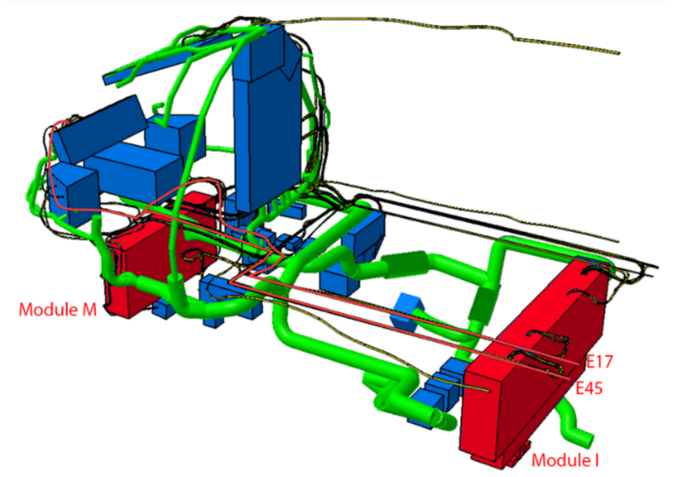
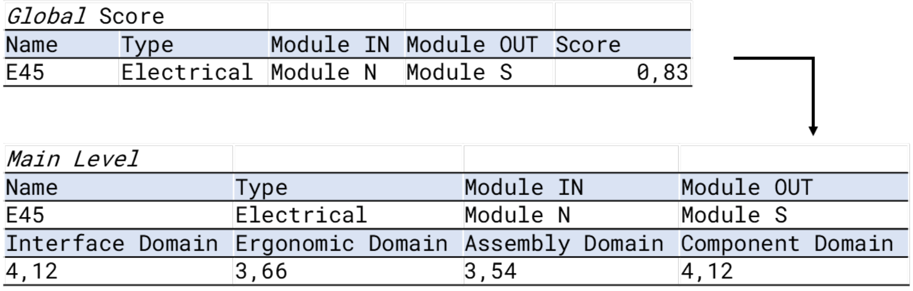


Figure 33 - Final Scores for modules and interfaces (including thresholds for redesign phase)

The obtained results provide an estimation of the complexity related to the assembly and installation of connections (interfaces) and modules inside the nose-fuselage. Final scores for the interfaces range from 0 to 1 due to the application of the TOPSIS method, while scores for modules range from 1 to 5 due to the use of the Mean operator. Interfaces and modules with the highest score represent the most complex to install. In this case, a threshold of 0.80 (in a range from 0 to 1) is used to filter out the most critical interfaces to install (i.e., E45 and E17), while a threshold of 4.00 (in a range from 1 to 5) is used to filter out the most critical modules to install (i.e., Module M and Module I). The red colour highlights the interfaces and modules with the highest scores (Figure 33). In the analysis of the interfaces' assessment, it is interesting to notice that electrical interfaces are the most critical for the installation process. The interface E45 has the highest score among all interfaces as reported in Figure 33. Interface E45 connects Module N to Module S and presents an overall score of 0,83. Moving from the final score to the main level, a better understanding of the E45 connection assembly complexity is obtained (Figure 34).



Global Score

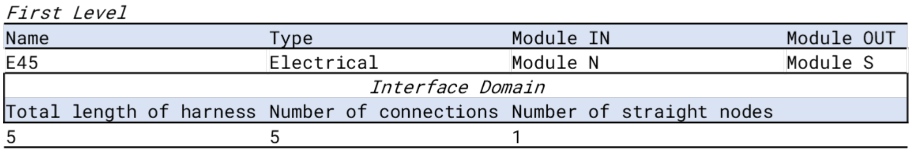
Name	Type	Module IN	Module OUT	Score
E45	Electrical	Module N	Module S	0,83

Main Level

Name	Type	Module IN	Module OUT
E45	Electrical	Module N	Module S
Interface Domain	Ergonomic Domain	Assembly Domain	Component Domain
4,12	3,66	3,54	4,12

Figure 34 - E45 from Global score to Main Level data

Among all, the highest score is associated with two domains: Interface Domain (4,12) and Component Domain (4,12). Analysing results for the Interface Domain, the attributes Total Length of Harness and Number of Connections present a score of 5 (Figure 35). Indeed, different modifications might be implemented to reduce the Interface Domain score: i) reduce the overall length of the connection by moving modules closer, ii) merge modules together to minimize the connection (i.e., build a new module that encompasses the two modules or assemble the two modules outside the aircraft and bring them inside as a single module) and, iii) make use of dedicated plate in order to install all connectors at the same time. In the Product Architecture Redesign phase, a structured way to assess which modification should be implemented first is presented.

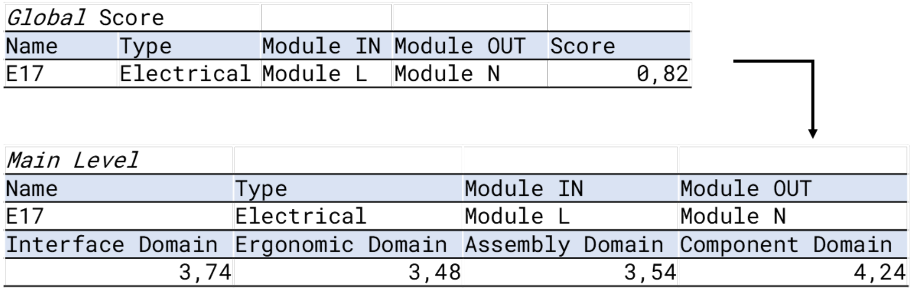


First Level

Name	Type	Module IN	Module OUT
E45	Electrical	Module N	Module S
	Interface Domain		
Total length of harness	Number of connections	Number of straight nodes	
5	5	1	

Figure 35 - E45 Interface Domain

The same study might be performed for the Component Domain. Moving to the second critical interface (E17) and repeating the analysis, it is possible to notice that the most critical domain is represented by the Component Domain, with a score of 4.24 (Figure 36).

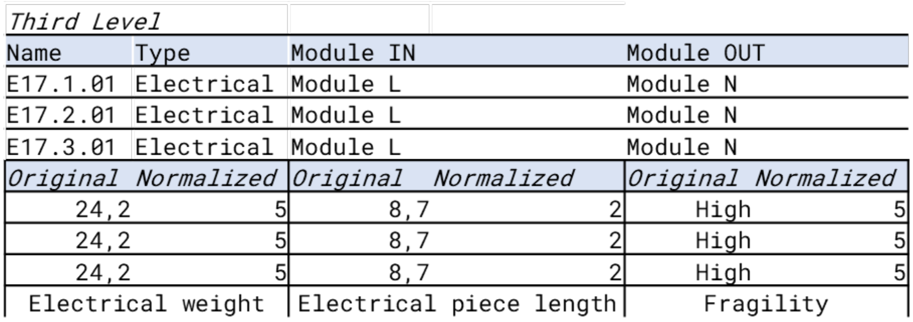


Global Score				
Name	Type	Module IN	Module OUT	Score
E17	Electrical	Module L	Module N	0,82

Main Level			
Name	Type	Module IN	Module OUT
E17	Electrical	Module L	Module N
Interface Domain	Ergonomic Domain	Assembly Domain	Component Domain
3,74	3,48	3,54	4,24

Figure 36 - Result analysis of E17 interface (from Global score to Main level data)

Electrical Weight and Fragility attributes, which belong to the Component Domain, have a score of 5. All electrical connections have the same score. In the current design, electrical connections are not split into parts like air interfaces; in fact, the same data is repeated for each interface sub-section (Figure 37). Electrical harnesses are heavy and difficult to manage. Furthermore, due to their lack of stiffness, harnesses are delicate and require special attention during installation. The Fragility score represents this characteristic (i.e., 5). To improve the installation aspects of E17 and all other electrical interfaces, electrical harnesses might be split into sub-harnesses and installed separately, or a special frame can be developed to increase the overall rigidity.



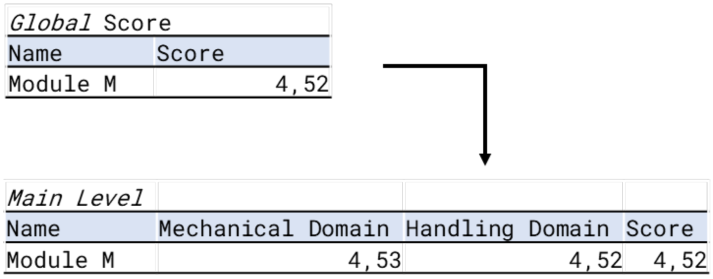
Third Level			
Name	Type	Module IN	Module OUT
E17.1.01	Electrical	Module L	Module N
E17.2.01	Electrical	Module L	Module N
E17.3.01	Electrical	Module L	Module N

Original	*Normalized*	*Original*	*Normalized*	*Original*	*Normalized*
24,2	5	8,7	2	High	5
24,2	5	8,7	2	High	5
24,2	5	8,7	2	High	5
Electrical weight		Electrical piece length		Fragility	

Figure 37 - extract of Weight and Fragility attributes for E17 interface

The same analysis was performed for the modules' assessment. From the global score, the most complex module to install is Module M, followed by Module I. Starting with Module M and moving from the global score to the

Main Level is possible to identify criticalities in both domains: Mechanical domain and Handling domain (Figure 38).

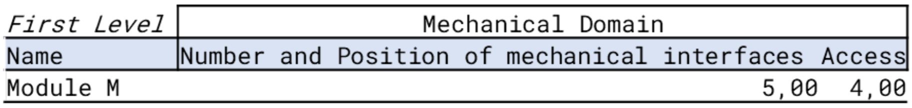


Global Score	
Name	Score
Module M	4,52

Main Level			
Name	Mechanical Domain	Handling Domain	Score
Module M	4,53	4,52	4,52

Figure 38 - Result analysis of Module M module (from Global score to Main level data)

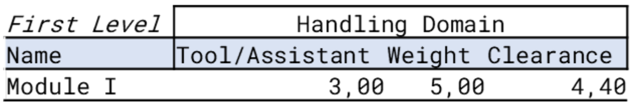
Analysing the Mechanical domain (Figure 39), it is possible to see that the most impacting attribute is Number and Position of Mechanical Interfaces, followed by Access. To change the number and position of mechanical interfaces, it is necessary to consider structural design requirements. By changing this attribute (i.e., reducing the number and changing the position of mechanical connection) a structural problem may arise. To avoid this issue, it could be necessary to reinforce the module, increasing the module weight. Before proceeding with the choice of which modification should be implemented, it is necessary to keep in mind some other aspects of the product analysed (i.e., the increment of weight). Indeed, most of the time, a weight increment cannot be tolerated for this kind of product. The same process can be repeated with the attribute Access, and with the Handling domain in order to identify design actions to reduce their scores.



First Level	Mechanical Domain	
Name	Number and Position of mechanical interfaces	Access
Module M	5,00	4,00

Figure 39 – Module M Mechanical Domain

Moving to the analysis of Module I, which is the second most critical module, from the Main Level it appears that the Handling domain has a higher score than the Mechanical domain. From the analysis of the Handling domain (Figure 40) is noticed that the most critical attribute is the Weight, followed by Clearance and finally Tool/Assistant. Several modifications might be performed to reduce their scores. An interesting solution might be to split Module I into two or more sub-modules. In this way, each submodule will be more manageable having a direct impact on the attribute Weight (i.e., the overall module weight will be reduced) and Clearance (i.e., available space between modules will increase). However, the split of a module might lead to a worsened interfaces score due to an increment in the overall number of interfaces.



First Level	Handling Domain		
Name	Tool/Assistant	Weight	Clearance
Module I	3,00	5,00	4,40

Figure 40 – Module I Handling Domain

The same analysis performed on the most critical modules and interfaces can be carried out to each interface and module for identifying criticalities during the installation phases. It is worth noting that the method does not consider the assembly sequence of the product (i.e., system dynamicity), and some interfaces/modules might be critical if assembled at the end of the installation process.

PHASE 4: PRODUCT ARCHITECTURE REDESIGN

The Product Architecture Redesign phase help designers to understand which modification should be implemented first in order to reduce the product architecture assembly complexity. Design guidelines were defined for the Nose-Fuselage and analysed according to the CDfA assessment results. Finally, the identified design guidelines were implemented and a new run of the CDfA method was performed.

Phase 4.1 Design Guidelines definition

Design guidelines were established using knowledge score matrices and brainstorming sessions with the participation of senior designers. The information inside knowledge scoring matrices was collected and structured according to the ontology defined. At the end of the process, 24 design guidelines were identified, 18 for interfaces and 6 for modules. In Table 12 an extract of the derived design guidelines for interfaces is presented. In Appendix A the design guidelines used for interfaces and modules are presented.

Table 12 – Extract of design guidelines for interfaces

#	Source	Related Domain	Related Attribute	Applicable
1	Scoring Matrix	Interface Domain	Total length of ducts	YES

Design guidelines

Reduce the length of air interface

Explanation

Reducing the length of the interface will reduce the overall assembly effort, since a shorter interface need to be installed.

#	Source	Related Domain	Related Attribute	Applicable
2	Scoring Matrix	Interface Domain	Branches	YES

Design guidelines

Reduce the number of branches

Explanation

Interfaces with fewer branches are easier to handle and install. Reducing the branches will lead to better interface management.

#	Source	Related Domain	Related Attribute	Applicable
3	Scoring Matrix	Interface Domain	Total length of harness	YES

Design guidelines

Reduce the length of electrical interface

Explanation

Reducing the length of the interface will reduce the overall assembly effort, since a shorter interface needs to be installed

Phase 4.2 Design Correlation Matrix derivation

Once Design Guidelines were obtained, two Design Correlation Matrices were derived, one for interfaces and one for modules. Each Design Guideline was read and analysed, and a score ranging from -2 to +2 was given to each attribute affected by the design guideline itself. Negative scores (-2 and -1) indicate that the application of the design guidelines will increase the assembly complexity of the specific parameter, hence a worsening of the product architecture will be produced. On the other hand, positive scores mean better assembly properties can be obtained for the specific attribute by applying the identified design guideline. An extract of the DCM for interfaces (Table 13) and modules (Table 14) are presented below.

Table 13 - Design Correlation Matrix for Interface modification

D.G.	*Tot. length of ducts*	*Branches*	*Tot. length of harness*	*# of connect.*	*# of straight nodes*
1	2	0	0	0	0
2	0	2	0	0	0
3	0	0	2	0	0
4	0	0	0	2	1
5	0	0	0	-1	2
6	0	0	0	0	0

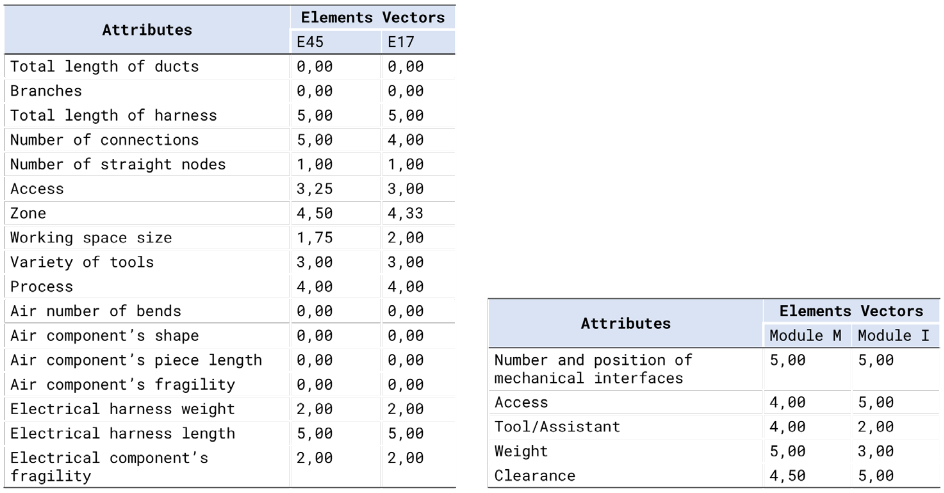
Table 14 - Design correlation matrix for module modification

D.G.	*# of Mech. Inter.*	*Pos. of Mech. Inter.*	*Access*	*Tool/Assistant*	*Weight*
1	2	1	0	0	0
2	0	2	0	1	0
3	0	0	2	1	0
4	0	0	0	2	0
5	0	0	0	1	2
6	0	0	-1	0	0

The generation of DCMs is a difficult task that necessitates the coordination of many experts. The correct population of DCMs is a must for the product architecture redesign to be efficient and error-free.

Phase 4.3 Elements' vector derivation

Results of the CDfA assessment were used to derive elements' vectors. By analysing the hierarchical structure of the cBoMs, it was possible to create the elements' vector. Normalized scores for each attribute were collected for modules and interfaces that were identified as critical (i.e., with a score higher than the 80% of the maximum available score). Module M and Module I are the modules that must be examined, while Interface E54 and Interface E17 are the interfaces that must be examined. In Figure 41 the EVs for Interface E54, Interface E17, Module M and Module I are shown.



<table>
<tr><th rowspan="2">Attributes</th><th colspan="2">Elements Vectors</th></tr>
<tr><th>E45</th><th>E17</th></tr>
<tr><td>Total length of ducts</td><td>0,00</td><td>0,00</td></tr>
<tr><td>Branches</td><td>0,00</td><td>0,00</td></tr>
<tr><td>Total length of harness</td><td>5,00</td><td>5,00</td></tr>
<tr><td>Number of connections</td><td>5,00</td><td>4,00</td></tr>
<tr><td>Number of straight nodes</td><td>1,00</td><td>1,00</td></tr>
<tr><td>Access</td><td>3,25</td><td>3,00</td></tr>
<tr><td>Zone</td><td>4,50</td><td>4,33</td></tr>
<tr><td>Working space size</td><td>1,75</td><td>2,00</td></tr>
<tr><td>Variety of tools</td><td>3,00</td><td>3,00</td></tr>
<tr><td>Process</td><td>4,00</td><td>4,00</td></tr>
<tr><td>Air number of bends</td><td>0,00</td><td>0,00</td></tr>
<tr><td>Air component’s shape</td><td>0,00</td><td>0,00</td></tr>
<tr><td>Air component’s piece length</td><td>0,00</td><td>0,00</td></tr>
<tr><td>Air component’s fragility</td><td>0,00</td><td>0,00</td></tr>
<tr><td>Electrical harness weight</td><td>2,00</td><td>2,00</td></tr>
<tr><td>Electrical harness length</td><td>5,00</td><td>5,00</td></tr>
<tr><td>Electrical component’s fragility</td><td>2,00</td><td>2,00</td></tr>
</table>

a. Modules’ vector

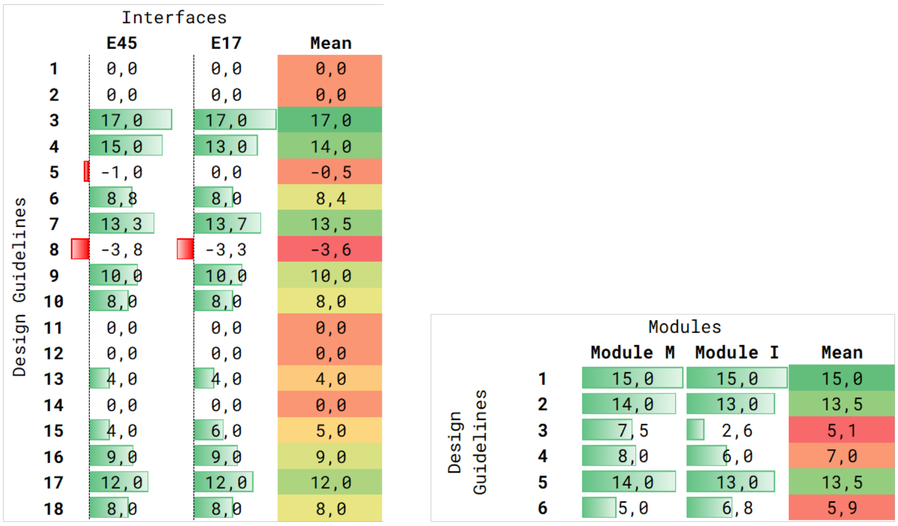
<table>
<tr><th rowspan="2">Attributes</th><th colspan="2">Elements Vectors</th></tr>
<tr><th>Module M</th><th>Module I</th></tr>
<tr><td>Number and position of mechanical interfaces</td><td>5,00</td><td>5,00</td></tr>
<tr><td>Access</td><td>4,00</td><td>5,00</td></tr>
<tr><td>Tool/Assistant</td><td>4,00</td><td>2,00</td></tr>
<tr><td>Weight</td><td>5,00</td><td>3,00</td></tr>
<tr><td>Clearance</td><td>4,50</td><td>5,00</td></tr>
</table>

b. Interfaces’ vector

Figure 41 – Elements’ vector. On the left side (a.) the EV of the modules is presented, while on the right side (b.) the EV of interfaces is presented.

Phase 4.4 Impact Vector computation

The multiplication between elements’ vector and DCMs was performed. The result of the computation was a [6x2] matrix (i.e., two impact vectors) for installation and [18x2] matrix (i.e., two impact vectors) in which each item represents the impact of the specific design guideline on the specific attribute. The mean of each design guideline impact vector (i.e., mean over each row) was computed. The mean allows having a global understanding of the impact that a given design guideline has on the overall modules. Indeed, assuming only a few designs guideline can be performed on a given product architecture, the ones with the highest mean values should be selected since they will provide the highest global positive impact on every element (Figure 42). In the specific case, only global (i.e., mean value) and element (i.e., value in each item) are presented. In fact, the identified critical elements belong to the same family.



Design Guidelines	Interfaces E45	Interfaces E17	Mean
1	0,0	0,0	0,0
2	0,0	0,0	0,0
3	17,0	17,0	17,0
4	15,0	13,0	14,0
5	-1,0	0,0	-0,5
6	8,8	8,0	8,4
7	13,3	13,7	13,5
8	-3,8	-3,3	-3,6
9	10,0	10,0	10,0
10	8,0	8,0	8,0
11	0,0	0,0	0,0
12	0,0	0,0	0,0
13	4,0	4,0	4,0
14	0,0	0,0	0,0
15	4,0	6,0	5,0
16	9,0	9,0	9,0
17	12,0	12,0	12,0
18	8,0	8,0	8,0

Design Guidelines	Modules Module M	Modules Module I	Mean
1	15,0	15,0	15,0
2	14,0	13,0	13,5
3	7,5	2,6	5,1
4	8,0	6,0	7,0
5	14,0	13,0	13,5
6	5,0	6,8	5,9

a. Interfaces' Impact Vectors

b. Modules' Impact Vectors

Figure 42 - Impact Vectors represented with colour bars. Green indicates improvements in terms of assembly product complexity. Red indicates worsened of the assembly product complexity.

Phase 4.5 Elements modification

Analysing results obtained in the last step, it was possible to select the design guideline to implement which generates the highest positive impact on the product architecture. Regarding the modules' modification, DGs #1, #2 and #5 are the best-in-class.

Design guideline #1 suggests to *reduce the number of mechanical interfaces*. The reduction of mechanical interfaces will improve the assembly process, since less interfaces will be required to be installed to fix the module at the structure. The elimination of mechanical contacts, on the other hand, may have an impact on force distributions. In fact, the remaining mechanical contacts must be modified to tolerate greater forces. Due to this reason, it is a modification that require a structural analysis and it will have an important amount of work to be implemented.

Design guideline #2 suggests to *change the position of mechanical interfaces*. According to the position of mechanical interfaces, operator needs to perform installation process in different ergonomic conditions (e.g., arms above the head, kneeling, etc.). Differently from DG #1 this modification is easier to implement and will increase the assembly performances allowing operators to reach mechanical interfaces easier.

Finally, design guideline #5 suggests to *reduce the module weight*. This DG can be achieved in a variety of ways. For example, it is possible to split modules into two sub-modules, reducing overall weight; alternatively, it is possible to modify the module design to reduce weight using different materials.

Regarding interfaces' modification, design guidelines #3, #7 and #4 were analysed, since they had the highest positive impact.

Design guideline #3 suggests to *reduce the length of the electrical interface*. The identified DG is feasible to implement. To do so, designers will be required to find another route that will decrease the overall interface length or move modules connected by interface E45 and E17 closer.

Design guideline #7 suggests to *change the zone of the working area* meaning it is necessary to move interfaces connections in zones easier to be reached by the operator. This modification will improve the ergonomic aspect (ergonomic domain) of the assembly process. This DG will necessitate a change in the interface path, allowing workers to undertake installation operations in a more ergonomically favourable environment (i.e., without the need to perform assembly operations overhead or crouched).

Design guideline #4 suggests to *reduce the number of connections.* It will be required to perform less assembly operations (i.e., positioning, fixation, etc.). The identified DG can be implemented in several ways, for instance, it is possible to change the connection fixation. Designing a special connector that clusters all interfaces into one-single operation will reduce the number of connections. Another possibility lies in adding a further interface between the module and the interface. The new interface can be collected in a plate that allows connecting all interfaces outside the nose-fuselage (i.e., on-ground-operation) and then, connecting the plate to the module. This will greatly reduce the number of connections to perform.

Among all DGs identified, it is interesting to notice that some DGs have a negative impact. For instance, DG #8 suggests to *move interface into a bigger working area*. The analysis shows that for Interfaces E45 and E17 this modification should not be implemented, because it will worsen the overall assembly process. In fact, changing the working area will probably lead to a change in the zone and access attributes, which, for the specific interfaces, are more important than the working area itself. However, the DG #8 must not be deleted since, for other elements, might lead to positive impacts.

For the nose-fuselage, design guideline #7 for interfaces (i.e., E45 and E17) and design guideline #5 for modules (i.e., M and I) were virtually implemented to test out the validity of the method. In fact, due to manufacturing problem it was not possible to physically perform modifications.

NEW CDFA ASSESSMENT

Following the implementation of design guideline #7 and #5, a second CDfA evaluation was conducted to evaluate the results. The zone of Interface E45 and E17 were changed. Table 15 presents the values pre and post modification. All values were set to "middle" since it represents the zone which has the lowest score (i.e., more efficient).

Table 15 - Interfaces modification

Interface	Pre modification Zone	Post Modification Zone
E45.1	Middle	Middle
E45.2	Upper	Middle
E45.3	Upper	Middle
E45.4	Lower	Middle
E17.1	Lower	Middle
E17.2	Upper	Middle
E17.3	Upper	Middle

The new CDfA assessment showed that an improvement of 8% and 9% were obtained respectively for interface E45 and E17 (Table 16).

Table 16 - CDfA new assessment Interfaces

Interface	Pre modification Score	Post modification Score	Improvement
E45	0,90	0,83	8%
E17	0,88	0,80	9%

Regarding modules, the DG #5 was implemented. Module M and Module I were divided into two sub-modules, respectively Module M1 and Module M2 and Module I1 and Module I2. All attributes' values of the sub-modules were inherited from the originating module. Exception was made for the attribute weight. In fact, with this configuration, Module M and Module I weight score moved from 5 to a score of 2 for each submodule (i.e., Module M1 and M2 weight score 2; Module I1 and I2 weight score 2). It is crucial to note, however, that this change may have unintended consequences that reduce assembly efficiency. Because two modules must now be installed, it will be necessary to perform interface installation twice (i.e., system interfaces, mechanical connections, and so on). Anyway, both modules' installation activities may be completed at the same time (i.e., operation parallelization). The modification will not worsen the installation process in this way. The final modules scores are presented in Table 17.

Table 17 - CDfA new assessment modules

Module	Pre modification Score	Post modification Score	Improvement
I	4,01	N/A	
I1	N/A	3,55	11%
I2	N/A	3,55	
M	4,23	N/A	
M1	N/A	3,71	12%
M2	N/A	3,71	

NOSE-FUSELAGE RESULTS DISCUSSION

The nose-fuselage is one of the most complex systems in an aircraft. It presents several interfaces and modules that need to be installed and assembled in a very confined and limited area. The Conceptual Design for Assembly method has shown great potentiality. After the initial effort made to obtain the functional scheme and the conceptual Bill of Material, the method allowed obtaining an estimation of the assembly performances of interfaces and modules inside the analysed system. Through the redesign phase, it was possible to obtain and prioritize design guidelines to help drive the product architecture redesign. Finally, once modifications are implemented, it is possible to re-perform the CDfA assessment to quantify the level of improvement reached.

Results showed that between the pre modification and post modification nose-fuselage architecture, an improvement of around 10% has been obtained for both modules and interfaces. These results have been reached by modifying the product architecture following suggestions retrieved from the redesign phase.

However, the application of the CDfA method at the nose-fuselage presented some challenges. For instance, the definition of the functional scheme required a great effort in terms of time. Several data were required to obtain a complete view of the nose-fuselage from its functional point of view. Moreover, the definition of the hierarchical structure and the mathematical model required a concurrent approach with a strong collaboration between manufacturing, assembly, and system departments. These led to management problems (i.e., arranging meetings, finding the right expertise, etc.) that further extended the complexity of the method.

Nevertheless, the CDfA approach applied to the nose-fuselage can assess product architectures in terms of assembly complexity, spot criticalities inside them, and identify design guidelines to advise designers through the redesign effort.

4.2 CABIN SYSTEM

The cabin is the aircraft system in which passengers and luggage are accommodated during the flight. It is composed of several elements (i.e., seats, toilets, panels, etc.) and connections (e.g., oxygen pipes, electrical cables, water pipes, etc.) that need to be assembled inside a pre-defined area (i.e., cabin airframe). The cabin architecture is mainly constrained by layout requirements. In fact, according to the size of the aircraft, different cabin layouts are used to accommodate the different number of passengers. Due to this requirement, the number of modules and interfaces cannot be changed. Moreover, interfaces are constrained by modules. Indeed, since modules need to follow a strict layout, interfaces installation must follow a specific path as well.

Due to these considerations, the analysis of the cabin system is performed only for modules, without considering interfaces installation themselves (Figure 43).

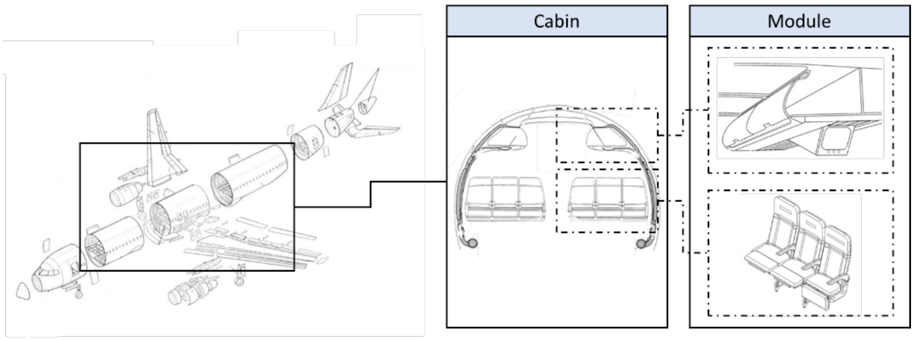


Figure 43 - Cabin modules

The goal of the presented case study is to numerically assess the cabin architecture with the aim to identify critical modules for the installation process, providing a list of the most critical items to install. Finally, design improvements are identified within the product redesign phase. In the following paragraphs, each phase of the CDfA method is presented and described with respect to the considered case study.

PHASE 1: FUNCTIONAL DECOMPOSITION

The functional scheme was obtained considering the cabin of an already existing civil aircraft. In fact, it is necessary to start from design actions already available to be able to derive a fully functional scheme. Functions were derived using several documents among which technical documents describing systems used in aircraft, technical drawing of the cabin, use and maintenance manual, accident, and malfunctions reports, etc. The analysis of these documents allowed obtaining information regarding systems inside the aircraft, how they interact, and how they are subdivided. The main function of the cabin is "*to accommodate passengers and hand luggage during the flight, to provide comfort for passengers (e.g., fresh air, toilets, water, etc.), while ensuring passenger safety (e.g., seats-belts, oxygen, etc.)*". As previously stated, the analysis focused on the installation of cabin modules; in fact, systems interfaces (i.e., electrical cables and harnesses, air ducts, etc.) are outside the scope of this work. Three hierarchical levels were obtained. Figure 44 shows an extract of the second level. The overall functional scheme is not provided due to confidentiality.

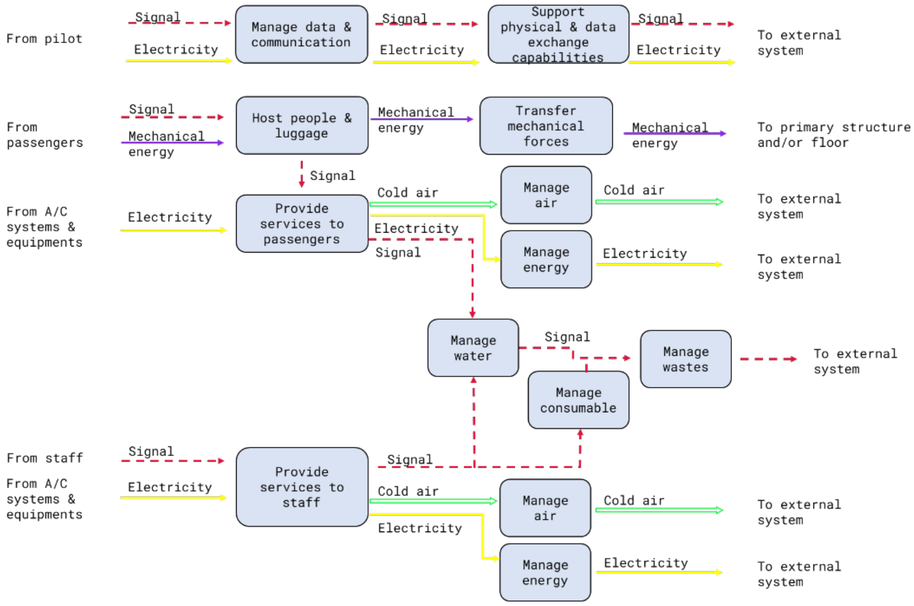


Figure 44 - 2nd level of the Cabin functional analysis

The modules heuristics method was chosen as a modularization technique because it provides a wide solution space. Indeed, it enables considering several constraints related to the aerospace sector (e.g., safety, maintenance, etc.). Modules identified with the three heuristics were compared with the ones already existing in the cabin and a one-to-one correspondence was found. A list of modules identified is shown in Table 18. These modules are difficult to merge and, therefore, to delete due to their specific functions, position allocation constraints, and aircraft building process. For example, the module Toilet is limited to its basic function because its location would not enable encompassing another cabin module function. In addition, it cannot be deleted because is required by the number of passengers and associated with the cabin layout.

Table 18 - Aircraft cabin modules derived by heuristics

Identified modules	Acronym
Toilet	T
Seats	S
Galley	G
Cabin Crew Rest Compartment	CCRC
Passengers Service Units	PSU
Hat Racks	HR
Doors Entrance	DE
Side Walls	SW

From an assembly point of view, it is interesting to understand the module ranking in relation to their complexity and therefore to their installation constraints driving the assembly performance, and then to assess different cabin architecture in relation to assembly performance.

PHASE 2: ARCHITECTURAL GEOMETRICAL DEFINITION

The sDMU is built by using the functional and the module derivation, indeed it aims at transforming that information into geometrical ones by translating functions and interfaces into simple geometries (e.g., boxes, cylinder, etc.). Modules are represented with their bounding box (parallelepiped) and interfaces are represented by cylinder (i.e., cables, pipes). However, since in the Cabin system no interfaces were analysed, the sDMU presents only modules.

The creation of the hierarchical structure to model the cabin modules installation process, called conceptual Bill of Material (cBoM). The cBoM is used to translate information from the sDMU into numerical data to be processed with mathematical algorithms. The modelling process consists of defining attributes, domains, and levels. The definition process was performed using a concurrent approach. For this task, five meetings were organized, involving cabin manufacturing engineering department and product architecture department with the aim to identify parameters characterizing the assembly process and their effects on the assembly lead-time and workload. Parameters considered critical from an assembly point of view, such as the module shape, attachment points location, etc. were defined and justified with the help of manufacturing engineers. These attributes were then clustered into domains considering their thematic (e.g., characterization of mechanical interfaces, of systems interfaces, of handling difficulty, etc.) and domains collected into levels according to the information granularity (i.e., at what time of the conceptual design phase the attribute information is available). At the end of the modelling process, the hierarchical structure was obtained: a total of 17 attributes were identified, clustered into 4 domains (Table 19). Differently from the nose-fuselage case study, for the cabin case study only modules were assessed. Thus, only one cBoM was derived.

Table 19 - cBoM structure (attributes and domains for cabin equipping)

Domain	Attributes
Mechanical Domain	Number of mechanical interfaces
	Standardization
	Design principle
	Rigging
	Gaps
Furnishing Domain	Number of furnishings
	Screwed
System Interface Domain	Number of system interfaces
	Number of plates
	Number of interfaces on plate
	Number of interfaces stand-alone
Handling Domain	Tool/assistant
	Clearance
	Access
	Weight

The model created for the cabin assessment is composed of only one level due to the characteristic of the system of interest (cabin) that does not require a further discretization in additional levels. Thus, all domains are considered at the same level. In Figure 45 the cabin hierarchical structure is presented. To make the mathematical explanation easier to follow, IDs were utilized to refer to attributes.

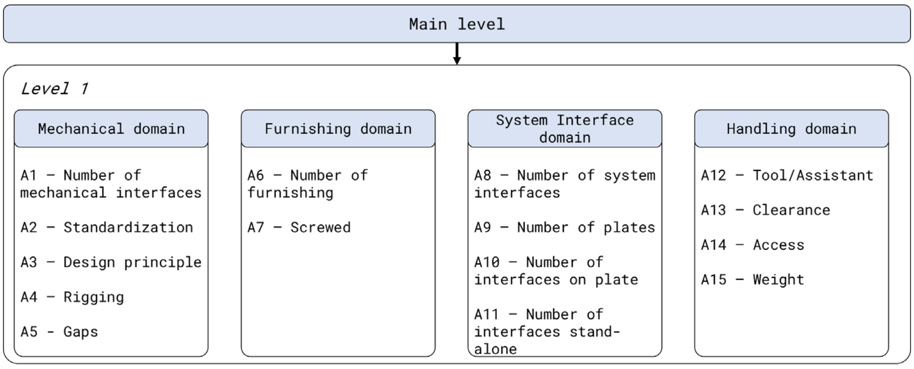


Figure 45 - Cabin hierarchical structure

The sensitivity analysis using One-Factor-At-Time (OFTA) method was performed on the CDfA model created. The analysis was repeated two times using different mathematical operators to cluster attributes inside domains. Specifically, for the first analysis, the mechanical, furnishing and handling domains were clustered with equation (VIII) (RMS), while the System Interface Domain with equation (IX). For the second analysis the mechanical, furnishing and handling domains were clustered with equation (IX) (Mean) while the System Interface Domain with equation (XI). Equations (X) and (XI) used for the System Interface Domain were developed according to the engineering knowledge to better model the CDfA cabin outcomes.

$$Mechanical/Furnishing/Handling\ Domain\ Score_{(I)} = \sqrt{\frac{\sum_{i=1}^{n}(A.i)^2}{n}}$$

(VIII)

$$Mechanical/Furnishing/Handling\ Domain\ Score_{(II)} = \sum_{i=1}^{n}\frac{A.i}{n}$$

(IX)

$$System\ Interface\ Domain\ score_{(I)} = \sqrt{\frac{\left(\left(1+4*\left(\frac{A.10}{A.9+A.10}\right)\right)+\left(\frac{A.8}{A.9}\right)\right)^2+(A.11)^2}{2}}$$

(X)

$$System\ Interface\ Domain\ score_{(II)} = \sqrt{\frac{\left(\left(1+4*\left(\frac{A.10}{A.9+A.10}\right)\right)+\left(\frac{A.8}{A.9}\right)\right)^2+(A.11)^2}{2}}$$

(XI)

Finally, the domain scores obtained were collected with a Mean operator for each module that is installed within the cabin. In Figure 46 results obtained for the mechanical domain and furnishing domain when RMS operator is applied are shown. Figure 47 presents results for the same domains when mean operator is used.

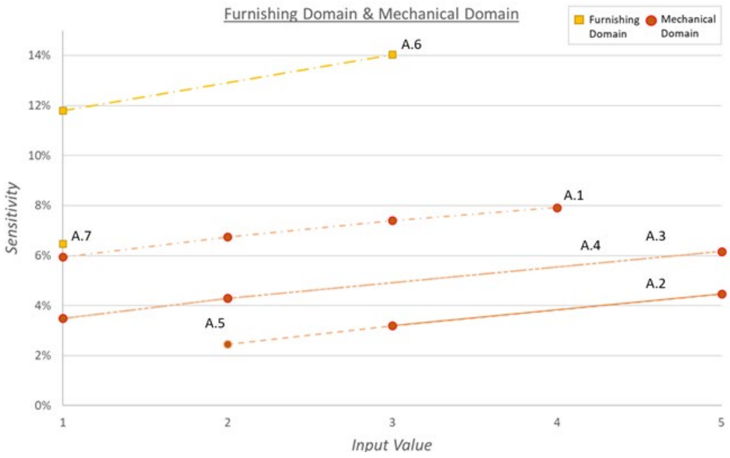


Figure 46 - SA results for RMS Mechanical and furnishing Domain

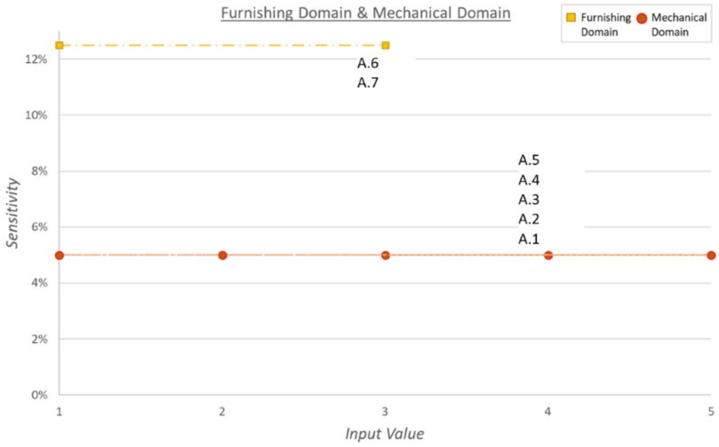


Figure 47 - SA results for MEAN mechanical and furnishing domain

Finally, Figure 48 presents results for the System Interface domain, where a mathematical equation derived though engineering judgment was used.

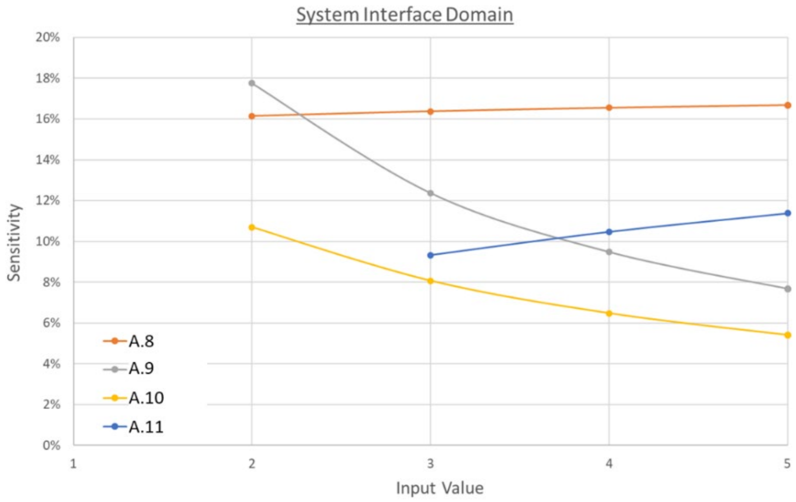
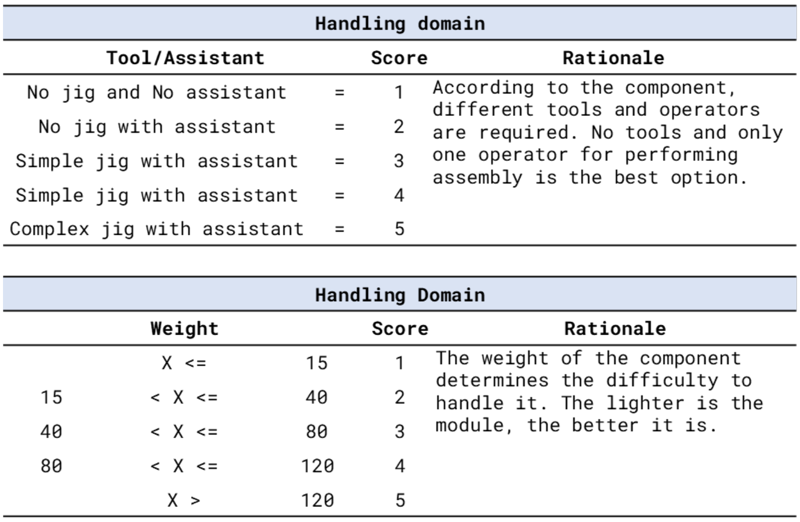


Figure 48 - System Interface Domain SA

The obtained results showed that: i) the use of the Mean operator to collect attributes inside a domain allows obtaining constant sensitivity for all attributes regardless of the baseline, ii) the use of RMS operator to collect attributes inside a domain gives different sensitivity results according to the baseline, and iii) the use of *ad-hoc* mathematical models might lead to inconsistent results, meaning the sensitivity of parameters involved in the computation does not follow a particular trend. Furthermore, by increasing the number of attributes collected within a domain, the sensitivity of each attribute decreases. Results of the SA allowed understanding the effect that domain creation had on the product architecture assessment. Moreover, even though the *ad-hoc* mathematical model presented an inconsistent result, it was not changed because it provided good results. The SA might be performed using other SA methods (i.e., global methods) to better understand how the variation of attributes might impact the final result.

After the definition of the cBoM structure, it was required to collect manufacturing knowledge using knowledge scoring matrices. kSM were built concurrently with assembly attributes identification. A scoring matrix was obtained for each attribute identified in the cBoM through brainstorming sessions involving manufacturing and architectural department. Figure 49 shows the scoring matrices for the attributes "Tool/Assistant" and "Weight" of the Handling domain.



Handling domain			
Tool/Assistant		Score	Rationale
No jig and No assistant	=	1	According to the component, different tools and operators are required. No tools and only one operator for performing assembly is the best option.
No jig with assistant	=	2	
Simple jig with assistant	=	3	
Simple jig with assistant	=	4	
Complex jig with assistant	=	5	

Handling Domain				
Weight			Score	Rationale
	X <=	15	1	The weight of the component determines the difficulty to handle it. The lighter is the module, the better it is.
15	< X <=	40	2	
40	< X <=	80	3	
80	< X <=	120	4	
	X >	120	5	

Figure 49 - Cabin scoring matrices

PHASE 3: CONCEPTUAL DESIGN FOR ASSEMBLY ASSESSMENT

The Conceptual Design for Assembly Assessment required the normalization of all attributes inside the cBoM, using the knowledge scoring matrices derived in the previous step. Then, the definition of a mathematical algorithm to collect the normalized data inside the cBoM to obtain the final score. Since the case study aims at assessing only modules installation, the sole module assembly final score was derived. Moreover, the cBoM presents only one level thus only two mathematical operators were required. The RMS was used to collect values inside each domain. The reason lies in the need to consider possible errors, in fact, initial data present different roots (i.e., some data are measured, others are derived by engineering knowledge) and they might present some evaluation errors. The use of RMS allowed obtaining a more conservative result with respect to other mathematical operators. The Mean operator was used to collect domains’ scores and compute the final score. In fact, after the normalization process and the domains’ collection, scores have all the same roots, and no further source of error needs to be considered.

Due to the nature of the case study (i.e., cabin), three different scores were obtained: (i) module score, (ii) family score, and (iii) global score (Figure 50).

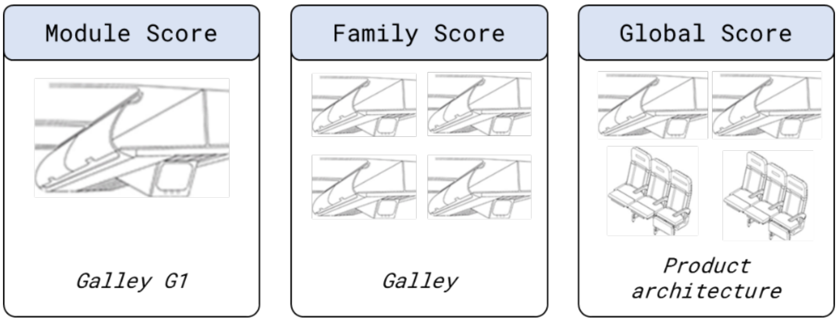
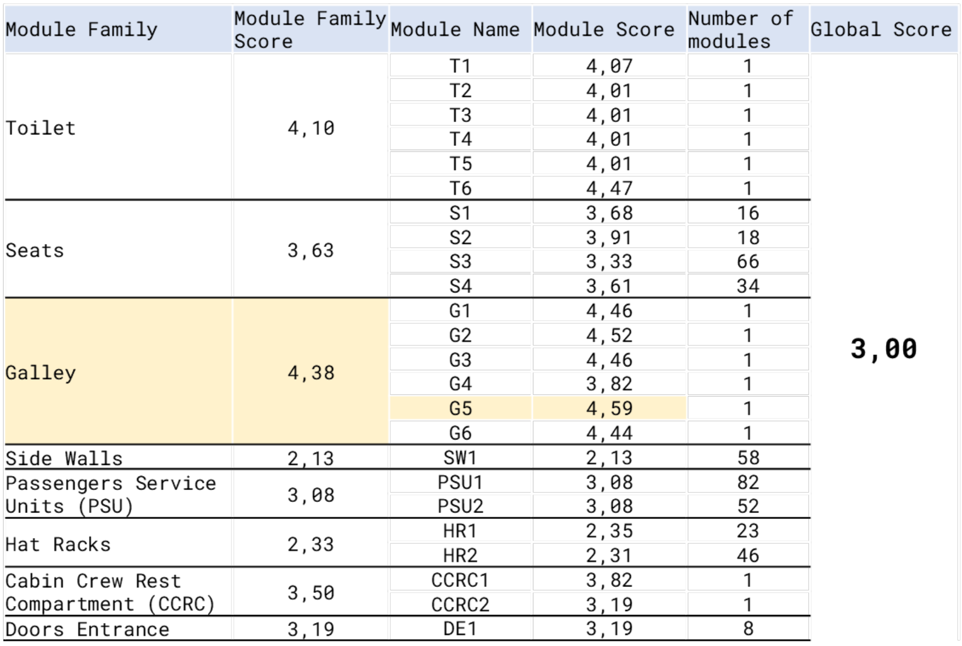


Figure 50 - Scores derived

The score called "Module Score" represents the assembly complexity of the specific module (e.g., Galley G5). It is the score obtained from the clustering of domains' scores with the mean operator. The score called "Family Score" was computed through the mean of all Modules' Scores belonging to the same type. For instance, in the analysed cabin, six different types of Galleys were identified (G1, G2, G3, G4, G5 and G6). They all belong to the same family which is "Galley". The aim of the "Family Score" is to provide an estimation of the most difficult type of modules to install. Finally, the score "Global Score" was computed considering the mean of Modules' Scores multiplied for the overall number of modules present in the analysed cabin. For example, the analysed cabin presents 16 Seats Business Lateral modules. Results obtained from the assessment are presented in Figure 51.

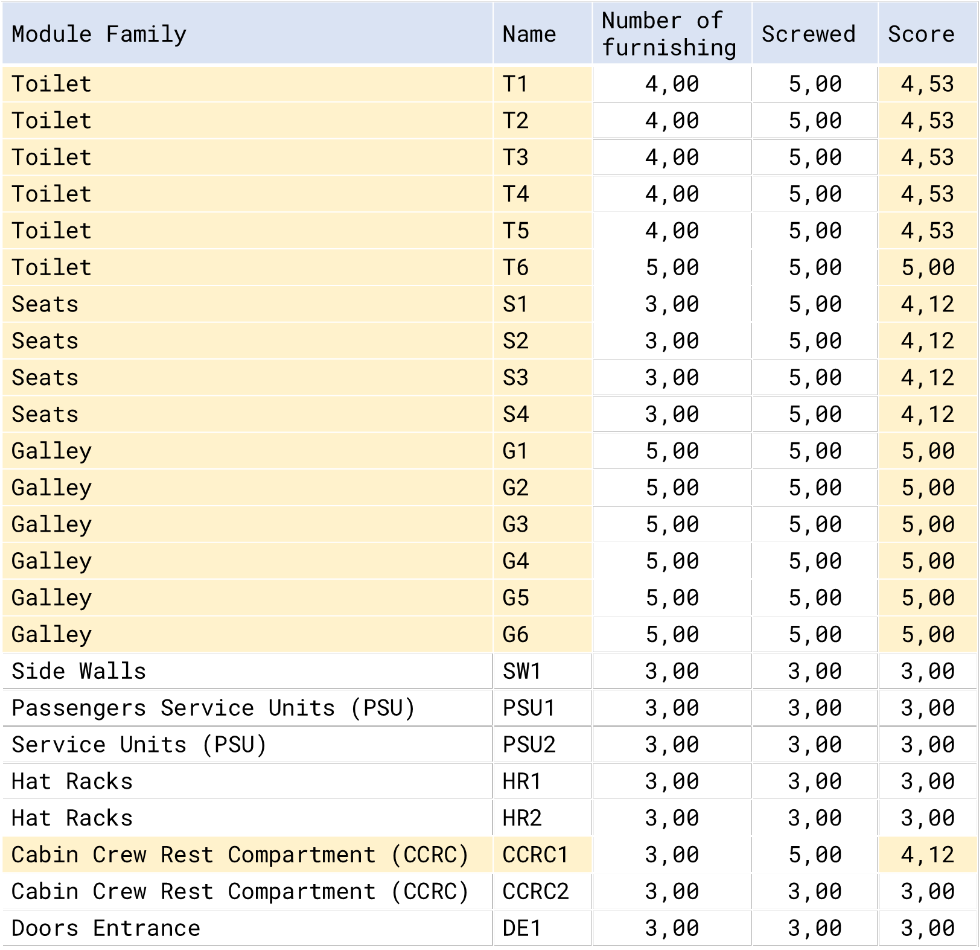


Module Family	Module Family Score	Module Name	Module Score	Number of modules	Global Score
Toilet	4,10	T1	4,07	1	3,00
		T2	4,01	1	
		T3	4,01	1	
		T4	4,01	1	
		T5	4,01	1	
		T6	4,47	1	
Seats	3,63	S1	3,68	16	
		S2	3,91	18	
		S3	3,33	66	
		S4	3,61	34	
Galley	4,38	G1	4,46	1	
		G2	4,52	1	
		G3	4,46	1	
		G4	3,82	1	
		G5	4,59	1	
		G6	4,44	1	
Side Walls	2,13	SW1	2,13	58	
Passengers Service Units (PSU)	3,08	PSU1	3,08	82	
		PSU2	3,08	52	
Hat Racks	2,33	HR1	2,35	23	
		HR2	2,31	46	
Cabin Crew Rest Compartment (CCRC)	3,50	CCRC1	3,82	1	
		CCRC2	3,19	1	
Doors Entrance	3,19	DE1	3,19	8	

Figure 51 - Results Cabin CDfA assessment

The obtained results provide an overview of the cabin's module. Module family scores show that the most critical module family is the Galley with a score of 4.38, followed by the module family Toilet with a score of 4.10 then the family Seats with a score of 3.63. A more detailed analysis of results is given by the single module score. Among all, the most critical module is the Galley G5 with a score of 4.59, while the second most critical module is the Galley G2 with a score of 4.52. In order to understand where design criticalities are, it is possible to proceed backward in the analysis of the cBoM.

By looking at the overall score for each domain, the highest score for the module Galley G5 is the Furnishing Domain with a score of 5, followed by the Mechanical Domain with a score of 4.76. It is possible to evaluate which attributes are relevant by moving inside the Furnishing Domain. For Galley G5 both attributes (number of furnishing and screwed) are critical, indeed they present a score of 5. Looking at the whole Furnishing Domain scores, 70% of scores are above 4 (Figure 52).



Module Family	Name	Number of furnishing	Screwed	Score
Toilet	T1	4,00	5,00	4,53
Toilet	T2	4,00	5,00	4,53
Toilet	T3	4,00	5,00	4,53
Toilet	T4	4,00	5,00	4,53
Toilet	T5	4,00	5,00	4,53
Toilet	T6	5,00	5,00	5,00
Seats	S1	3,00	5,00	4,12
Seats	S2	3,00	5,00	4,12
Seats	S3	3,00	5,00	4,12
Seats	S4	3,00	5,00	4,12
Galley	G1	5,00	5,00	5,00
Galley	G2	5,00	5,00	5,00
Galley	G3	5,00	5,00	5,00
Galley	G4	5,00	5,00	5,00
Galley	G5	5,00	5,00	5,00
Galley	G6	5,00	5,00	5,00
Side Walls	SW1	3,00	3,00	3,00
Passengers Service Units (PSU)	PSU1	3,00	3,00	3,00
Service Units (PSU)	PSU2	3,00	3,00	3,00
Hat Racks	HR1	3,00	3,00	3,00
Hat Racks	HR2	3,00	3,00	3,00
Cabin Crew Rest Compartment (CCRC)	CCRC1	3,00	5,00	4,12
Cabin Crew Rest Compartment (CCRC)	CCRC2	3,00	3,00	3,00
Doors Entrance	DE1	3,00	3,00	3,00

Figure 52 - Furnishing Domain scores

This result suggests that a modification of the furnishing technology would result in a more efficient assembly process by improving the score of all modules. For instance, from the scoring matrix associated with the attribute "Screwed", it is possible to understand that changing the attachment principle by removing screws and making use of hook and loop closure would bring a score of 3. The same process can be repeated for other domains and other modules, to identify possible improvements to the current cabin architecture.

In order to obtain the best design guidelines to apply for redesigning the cabin architecture, it is possible to proceed with the final phase.

PHASE 4: PRODUCT ARCHITECTURE REDESIGN

The Product Architecture Redesign phase helps to guide designers through the product architecture modification. Design guidelines were defined for the Cabin and analysed according to the CDfA assessment results. Finally, the identified design guidelines were implemented and a new run of the CDfA method performed.

Phase 4.1 Design Guidelines definition

First, a list of design guidelines (DG) was obtained. The list is composed of 24 design guidelines of which 9 derived from scoring matrices and 15 from engineering knowledge. Design guidelines were obtained following a concurrent approach, in which experts from cabin manufacturing and cabin architecture design were involved. Design guidelines were derived and listed according to the ontology defined. An extract of the derived guidelines is presented in Table 20. It is worth nothing that even though some design guidelines might not be applicable, they must be retained. The complete list of design guidelines for modules installation is provided in Appendix A.

Table 20 - Extract of design guidelines for modules

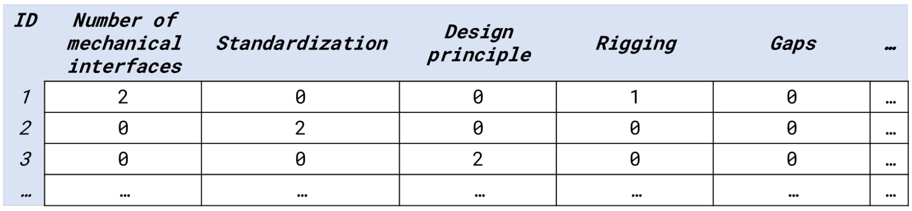
#	Source	Related Domain	Related Attribute	Applicable
1	Scoring Matrix	Mechanical Domain	Mechanical Connections	NO
	Design guidelines			
	Reduce the number of mechanical connections			
	Explanation			
	Reducing the number of mechanical connections will reduce the assembly complexity. However, stresses must be considered.			

#	Source	Related Domain	Related Attribute	Applicable
2	Scoring Matrix	Mechanical Domain	Part standardization	YES
	Design guidelines			
	Increase number of standard parts			
	Explanation			
	The use of standard parts will reduce the assembly complexity, since less tools will be needed.			

#	Source	Related Domain	Related Attribute	Applicable
3	Scoring Matrix	Mechanical Domain	Change design principles	YES
	Design guidelines			
	Obtain one design principle per set of attachment			
	Explanation			
	Using one design principles per set of attachment increase the assembly performances reducing the need of different tools.			

Phase 4.2 Design Correlation Matrix derivation

Then, the Design Correlation Matrix (DCM) was obtained indicating the impact of each design action on each attribute. An extract of the DCM for the first three design guidelines is reported in Figure 53. The example reports the first five attributes which are referred to the mechanical domain.



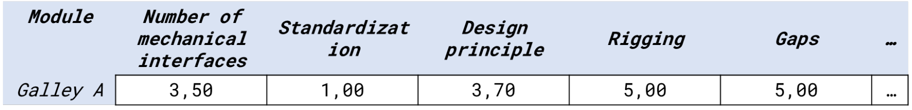
ID	*Number of mechanical interfaces*	*Standardization*	*Design principle*	*Rigging*	*Gaps*	*…*
1	2	0	0	1	0	…
2	0	2	0	0	0	…
3	0	0	2	0	0	…
…	…	…	…	…	…	…

Figure 53 - Extract of the DCM. Impact of the first three DGs.

Attributes belonging to the mechanical domain address the number of mechanical interfaces, the standardization, the design principles of mechanical connections, aesthetic features and process issues. The impact derivation was performed concurrently with architecture engineers' experts. An interesting result was obtained: impacts associated with design options derived from scoring matrices had only positive values (0, 1, 2) while the ones derived from engineering expertise had both positive and negative values. Reasons lie in the fact that scoring matrices are built to improve the product architecture assemblability, and they do not consider other attributes. In other words, scoring matrices reflects the best solution that can be reached for a given attribute, without considering others. Design guidelines based on engineering expertise, on the other hand, assess the impact of a modification on multiple attributes. To make the analysis easier only design guidelines derived from scoring matrices have been considered (9 design guidelines) for the computation of the impact vector.

Phase 4.3 Elements' vector derivation

The elements vectors were calculated. For the specific case study, elements vectors of the most critical modules were obtained. Module Galley G5, G4, and G2; Toilet T6, T1 and T5; Hat Racks HR1, HR2 and Cabin Crew Rest CCRC1, CCRC2 were analysed, and associated vectors derived. An extract of the obtained element vector for the Galley G5 is presented in Figure 54.

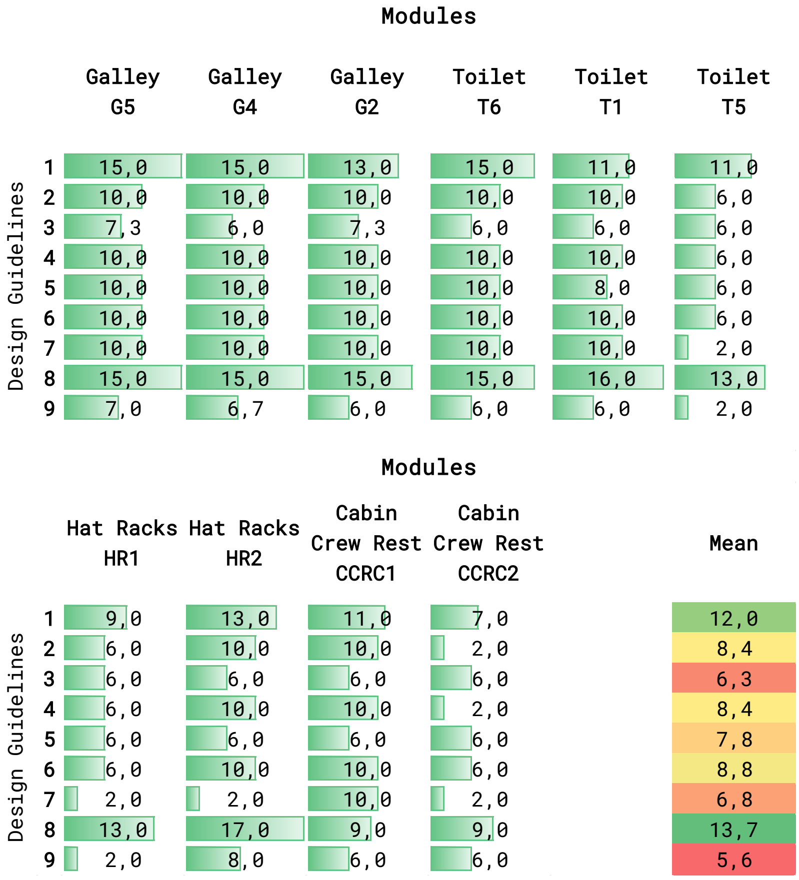


Module	Number of mechanical interfaces	Standardization	Design principle	Rigging	Gaps	…
Galley A	3,50	1,00	3,70	5,00	5,00	…

Figure 54 - Extract of the element vector of the Galley G5

Phase 4.4 Impact Vector computation

Finally, the Impact vector for the examined modules was calculated by multiplying the elements vectors by the Design Correlation Matrix. Results of the computation is a matrix [8x10] (i.e., 8 impact vectors) in which each item represent the impact of the specific design guideline on each module. To understand the overall impact of each design guidelines on all analysed modules the mean was computed. The matrix obtained composed of impact vectors is reported in Figure 55.

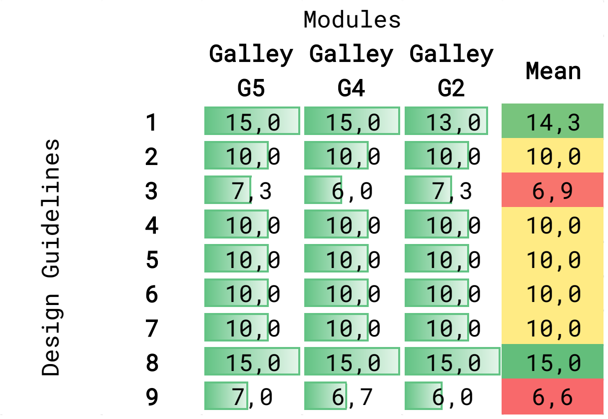


Design Guidelines	Galley G5	Galley G4	Galley G2	Toilet T6	Toilet T1	Toilet T5
1	15,0	15,0	13,0	15,0	11,0	11,0
2	10,0	10,0	10,0	10,0	10,0	6,0
3	7,3	6,0	7,3	6,0	6,0	6,0
4	10,0	10,0	10,0	10,0	10,0	6,0
5	10,0	10,0	10,0	10,0	8,0	6,0
6	10,0	10,0	10,0	10,0	10,0	6,0
7	10,0	10,0	10,0	10,0	10,0	2,0
8	15,0	15,0	15,0	15,0	16,0	13,0
9	7,0	6,7	6,0	6,0	6,0	2,0

Design Guidelines	Hat Racks HR1	Hat Racks HR2	Cabin Crew Rest CCRC1	Cabin Crew Rest CCRC2	Mean
1	9,0	13,0	11,0	7,0	12,0
2	6,0	10,0	10,0	2,0	8,4
3	6,0	6,0	6,0	6,0	6,3
4	6,0	10,0	10,0	2,0	8,4
5	6,0	6,0	6,0	6,0	7,8
6	6,0	10,0	10,0	6,0	8,8
7	2,0	2,0	10,0	2,0	6,8
8	13,0	17,0	9,0	9,0	13,7
9	2,0	8,0	6,0	6,0	5,6

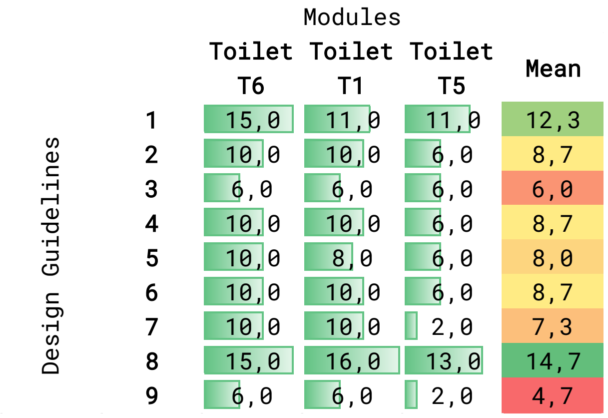
Figure 55 - Impact Vectors represented with green bars.
The longer is the bar, the higher is the impact of each design guideline on the specific module.

The results were clustered by module family and the mean was computed to determine the overall influence of each design guideline on each family. The results for the families Galley and Toilet are presented in Figure 56 and Figure 57, respectively.



Design Guidelines	Modules Galley G5	 Galley G4	 Galley G2	Mean
1	15,0	15,0	13,0	14,3
2	10,0	10,0	10,0	10,0
3	7,3	6,0	7,3	6,9
4	10,0	10,0	10,0	10,0
5	10,0	10,0	10,0	10,0
6	10,0	10,0	10,0	10,0
7	10,0	10,0	10,0	10,0
8	15,0	15,0	15,0	15,0
9	7,0	6,7	6,0	6,6

Figure 56 - Family Galley Impact Vector results



Design Guidelines	Modules Toilet T6	 Toilet T1	 Toilet T5	Mean
1	15,0	11,0	11,0	12,3
2	10,0	10,0	6,0	8,7
3	6,0	6,0	6,0	6,0
4	10,0	10,0	6,0	8,7
5	10,0	8,0	6,0	8,0
6	10,0	10,0	6,0	8,7
7	10,0	10,0	2,0	7,3
8	15,0	16,0	13,0	14,7
9	6,0	6,0	2,0	4,7

Figure 57 - Family Toilet Impact Vector

Results show that, for the family Galley, the most impactful design guideline is #8 which suggests to "*change gaps to avoid visible gaps*" followed by DG #1 which suggests to "*reduce the number of mechanical interfaces*". Then, design guidelines #2, #4, #5, #6 and #7 have all the same impact, thus they can be implemented without any preference. The same results are shown for the family Toilet. However, differently from family Galley, the third design guideline to be implemented can be chosen freely from DG #2, DG #4 and DG #6. Analysing results for the most critical module (Figure 58), which is Galley G5, it is possible to see that the design guidelines suggested are the same for the family galley. Indeed, galley G5 belongs to the galley family.

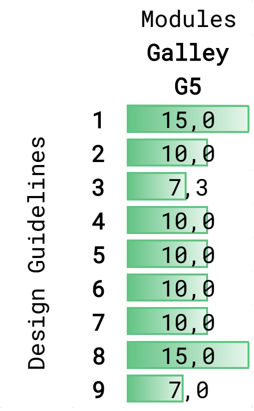


Figure 58 - Galley G5 Impact Vector

Phase 4.5 Elements modification

The impact vectors allowed identifying which design guideline shall be implemented to optimize the product architecture from a global, family and specific module point of view.

The first design guideline to implement is DG #8 which suggests to "*change gaps to avoid visible gaps*". Gaps are linked to aesthetic requirements. In fact, when modules are installed on the cabin structure, according to the attachment design they might present gaps that will need to be covered subsequently. Changing the module attachment, it is possible to avoid the creation of gaps.

Another design guideline that should be implemented is the DG #1 which suggests to *"reduce the number of mechanical connections"*. However, this modification cannot be applied with current manufacturing technologies, and it will require a deep study of forces associated with a reduction of the number of mechanical connections.

Looking at the results, design guidelines DG #2, DG #4, DG #6, will provide an improvement for both toilets and galley family, while DG #5, DG #7 will have a greater benefit only for the family galley. Thus, DG #2, DG #6, and DG #7 were applied to improve the assemblability of both galley and toilets families.

DG #2 suggests to *"change the part standardization to reach all standard parts"*. Part standardization refers to the ability to use same parts to accomplish different functions (i.e., same screws to fix different parts of the galley). Using all standard parts will reduce the need for different tools and it will increase the assembly performance since blue collars will not need to change tools from one part to another.

DG #4 suggests to *"modify the rigging process to avoid rigging"*. The act of rigging is a final operation that needs to be performed on modules to fix them on the structure and on near modules. Modifying the rigging process in order to remove it for the assembly operation will improve the overall assemblability since fewer operations will need to be performed to accomplish the final assembly.

DG #6 suggests to *"change the screwing process to avoid screwing"*. The process of screwing refers to the need of fixing modules on the structure. By changing the fixing procedure, it is possible to attach a module to the structure without the need for screwing (e.g., snap-fit, etc.), improving the overall assembly process.

Finally, the DG #5 was implemented only on the family Toilet since the impact vector showed that it will produce a great benefit. DG #5 suggests to *"modify the number of furnishings to avoid furnishing"*. Reducing the number of furnishings will improve the module assemblability since fewer operations will be required to accomplish the module installation. The reduction of the number of furnishings to zero means avoiding furnishing operations.

For Galley and Toilets, the design guidelines #8, #2, #4, and #6 were implemented, while the design guideline #5 was adopted only for the family Toilet.

Finally, the CDfA Assessment was rerun to see how much the product architecture had improved.

NEW CDFA ASSESSMENT

A new run of the CDfA Assessment was performed once the identified design guideline was implemented.

On the attribute Gaps, the DG #8 was applied by changing the initial value "*Existing and Visible Gaps*" to the final value "*No Visible Gaps*." The DG #2 was then applied, which changed the Part Standardization property from "*Less than 50% standard parts*" to "*100% standard parts*". The DG #4 was implemented by changing the value of the Rigging attribute from "*Rigging in FAL*" to "*No Rigging*". In the case of DG #6, the attribute Type of Fixation was modified from "*Screwed fixation*" to "*No screwed fixation*". Finally, the DG #5 was implemented, which changed the Number of Furnishing attribute from "*Above 21*" to "*No Furnishing*".

The modification led to an improvement of 19% for family Toilet and 30% for family Galley (Figure 59).

Module Family	Pre modification	Post modification	Improvement	Module name	Pre modification	Post modification	Improvement
Toilet	4,10	3,30	19%	T1	4,07	3,26	20%
				T2	4,01	3,20	20%
				T3	4,01	3,20	20%
				T4	4,01	3,20	20%
				T5	4,01	3,20	20%
				T6	4,47	3,76	16%
Seats	3,63	3,63	0%	S1	3,68	3,68	0%
				S2	3,91	3,91	0%
				S3	3,33	3,33	0%
				S4	3,61	3,61	0%
Galley	4,38	3,07	30%	G1	4,46	3,06	31%
				G2	4,52	3,21	29%
				G3	4,46	3,15	29%
				G4	3,82	2,51	34%
				G5	4,59	3,31	28%
				G6	4,44	3,14	29%
Side Walls	2,13	2,13	0%	SW1	2,13	2,13	0%
Passengers Service Units (PSU)	3,08	3,08	0%	PSU1	3,08	3,08	0%
				PSU2	3,08	3,08	0%
Hat Racks	2,33	2,33	0%	HR1	2,35	2,35	0%
				HR2	2,31	2,31	0%
Cabin Crew Rest Compartment (CCRC)	3,50	3,50	0%	CCRC1	3,82	3,82	0%
				CCRC2	3,19	3,19	0%
Doors Entrance	3,19	3,19	0%	DE1	3,19	3,19	0%

Product Architecture	Pre modification	Post modification	Improvement
Cabin	3,00	2,97	1%

🟨	Most complex module Post modification
🟧	Most complex module Pre modification

Figure 59 – Cabin Final results

Moreover, Galley G5 and Galley G2 which were the most complex module to install before the modification have reduced their score. With the new product architecture, the most complex modules to install are Cabin Crew Rest Compartment CCRC1 and Seat S2.

However, the overall global score improved by only 1%. The reason lies in the fact that only a limited amount of Galley and Toilets are present in the cabin architecture, with respect to other modules. In fact, for the analysed aircraft cabin concept, there are 6 toilets and six 6 galleys while the number of seats is 134. As a result, improving the seat family will result in a global score reduction that is bigger than the reduction produced by changing the toilets and galleys.

CABIN RESULTS' DISCUSSION

The CDfA methodology allowed assessing and improving the Cabin architecture of a civil aircraft. Cabin systems present several constraints such as the need to have a specific number of modules (i.e., number of toilets, seats, etc.) according to the size of the cabin. This resulted in interface constraints; they must follow a specified path and cannot be altered in any way. Because of these limitations, the methodology was only used on modules (i.e., no interfaces were analysed). The CDfA assessment phase provided three different scores: i) Module Score, ii) Family Score and iii) Global Score. Each score provided a different understanding of the cabin architecture. From the performed analysis, module Galley G5 resulted to be the most challenging to install with a score of 4,59, followed by the module Galley G2 with a score of 4,52. Next, the Product Redesign Phase provided suggestions on design guidelines that had to be implemented to reduce module and module family scores, meaning improving their assemblability. The implemented modification led to a new product architecture. For the new product architecture, the most complex modules to install are the Seat S2 and the Cabin Crew Rest Compartment CCRC 1 with a score of, respectively, 3,91 and 3,82. The identified design guidelines enabled improving the Galley G5 of 28% and Galley G2 of 29%, while Family Galley and Family Toilets improved by 30% and 19%, respectively. Finally, the overall product architecture was improved by 1%, switching from a Global Score of 3,00 to a Global Score of 2,97.

The Global Score improved the least of all the results. This is because the number of galleys and toilets in the cabin architecture under consideration is lower than the number of other modules (Table 21). As a result, a change to the Toilets and Galley will have a less impact on the overall assembly complexity of the Cabin architecture than a change to the Seats or Passenger Service Units, because fewer units will be involved.

Table 21 - Number of modules inside the Cabin

Module family	Overall number
Toilet	6
Seats	134
Galley	6
Side Wall	58
Passenger Service Units (PSU)	134
Hat Racks	69
Cabin Crew Rest Compartment	2
Doors Entrance	8

Few challenges have been identified during the application of the CDfA methodology on the Cabin case study. First, due to the impossibility to modify cabin layout and modules, the functional decomposition presented little added value. In fact, modules were obtained following a bottom-up approach to retrieve already available modules.

Another challenge was the derivation of the Design Correlation Matrix during the Redesign phase. The association of impacts to each design guideline required great effort. Much expertise (i.e., architects, cabin experts, manufacturing department) and several iterations were needed to derive the DCM in compliance with the given system of interest.

The CDfA method appeared to be a great way to assess and provide modification of complex product architecture at the conceptual design phase when the granularity level is high and few data available. However, it requires an initial effort to set up the mathematical model that better fits the system analysed. Moreover, several iterations are required to fine-tune the model before proceeding to assess brand-new complex product architectures.

5 RESULTS

The Conceptual Design for Assembly method proposed in this research work aims at assessing and improving architectures of complex products at early design phases. In literature, different methods are present to tackle the assembly phase of products. In the following paragraphs, a comparison between the CDfA method and other methods to perform assembly optimization is proposed. The approach used to perform the comparison is introduced, then the B&D, the Lucas method, and the Large Design for Installation method are analysed according to the comparison framework proposed. Finally, the comparison and discussion of DFMA methods are proposed.

5.1 COMPARISON FRAMEWORK

The comparison among various methods requires the need of applying them to the same product and record different parameters. Due to the nature of the method proposed in this research, it is not possible to directly compare the CDfA approach with other DFMA methods. The CDfA method has been specifically developed to assess and redesign large products with a long-lead time, while DFMA methods are mainly focused on small products, easy to handle with a short lead time. Moreover, the CDfA approach, even if can handle data of different granularity, has been developed to be mainly used with conceptual design information. In order to provide a meaningful and scientific comparison, a framework collecting several parameters which can be identified in all analysed DFMA methods is provided. The framework is based on the work of Owensby (Owensby, et al., 2011) which performed a comparison between two DFMA methods. The framework provided in this research work consists of twelve fields:

- **Reference**
  it collects works from which method information is collected.
- **Method**
  it provides an overview of the method used. Methods vary according to the work analysed.
- **Design phase**

it describes the design phase at which the method can be applied. Three choices are available:

  - Conceptual
  - Detail
  - Embodiment
- **Analysis time**
  it provides an estimation of the time required to perform the analysis. This field accepts strings. Three choices are available:
  - Short – within 1 day
  - Moderate – within 1 week
  - Long – more than 1 week
- **Predicted assembly time**
  When available, it indicates the goodness of the estimated assembly time with respect to the benchmark. Three choices are available:
  - Good – same as the benchmark
  - Fair – within 10% of the benchmark
  - Poor – more than 10% of the benchmark
- **Amount of information**
  It indicates the amount of information required to perform the analysis. Three choices are available:
  - High – more than 20 parameters required
  - Moderate – within 11 to 20 parameters required
  - Low – less than 10
- **Information type**
  It indicates the type of information required. It is used to compute the method repeatability. It can be:
  - Subjective – information depends on the operator
  - Objective – information does not depend on the operator
- **Repeatability**
  It indicates the ability to obtain the same result if the analysis is repeated by other users. It is the ratio between the number of objective information over the total information required.
- **Redesign guidelines**
  It indicates if the method can provide redesign guidelines.
- **Possibility to add cost evaluation**
  It indicates if the method can provide a cost estimation.

- **Index for evaluating design goodness**
  It indicates if the method provides one or more indices to evaluate the goodness of the design in terms of assemblability performances

Using the proposed framework, it was possible to analyse and compare B&D DFMA method, the Lucas method, and the Large DFI method with the Conceptual Design for Assembly approach overcoming the difficulty of applying the three methods to the same case studies.

5.2 DFMA METHODS ANALYSIS

Four DFMA methods are analysed to perform the comparison. The four methods are: i) B&D DFMA, ii) The Lucas Methods, iii) Large Design for Installation, and iv) Conceptual Design for Assembly.

B&D DFMA

The B&D Design for Manufacturing and Assembly method is well-known and widely applied in both academia and industry field. The method required the user to define parameters such as Theoretical Minimum Number of Parts or Handling Time, answering a specific set of questions. According to the type of question answered, parameters were classified as objective or subjective. Table 22 presents the full list of parameters required and the type of information.

Table 22 - B&D Subjective/Objective information

B&D required information (Gupta & Kumar, 2019)	**Subjective**	**Objective**
Part name		X
Part number		X
Number of times operations required		X
Handling code	X	
Handling time *(s)*		X
Insertion code	X	
Insertion time *(s)*		X
Theoretical minimum number of parts	X	
Total operation time *(s)*		X

The repeatability index is computed considering the ratio of objective parameters (6) over the total number of them (9). This makes the repeatability of the B&D method of 67%.

In literature, there is an extreme number of works presenting the application of the B&D DFMA on different fields and case studies. Among all, the work of Owensby (Owensby, et al., 2011) applied the B&D DFMA approach to three products made of a few parts (i.e., One Touch Copper, Black & Decker Cordless Drill, and RIVAL Can Opener). The paper provides information regarding the analysis time. It states that the application of the B&D method required less than a day to be performed and that the predicted assembly time was in line with the available baseline. Finally, the analysis identified 11 types of issues on which design efforts had to be focused to improve the product, and, together with the Design Efficiency Index, helped the user to improve the product in terms of manufacturing and assembly. However, no redesign guidelines were provided. Finally, to perform the B&D analysis, it was necessary to have real products available and to be able to disassemble them in order to retrieve the required information. Information collected in the comparison framework is presented in Table 23.

Table 23 – Information in the comparison framework for the B&D DFMA

Reference	(Owensby, et al., 2011) (Gupta & Kumar, 2019)
Method	B&D
Design phase	Detail
Analysis time	Short
Predicted assembly time	Good
Amount of information	Low
Type of information	Objective & Subjective
Repeatability	67%
Feature to Redesign	Yes
Redesign guidelines	No
Possibility to add cost evaluation	Yes
Index for Design goodness	Yes

The Lucas Method

The Lucas Method is based on rating factors. The method consists of performing four different analyses: i) Functional analysis, ii) Feeding analysis, iii) Fitting analysis, and iv) Manufacturing analysis.

Each analysis requires estimating parameters with the help of scoring tables. Parameters required to perform the analysis are collected in Table 24, where their type is specifically expressed (i.e., subjective or objective).

Table 24 - The Lucas Method subjective/objective information

Lucas Method required information (Dochibhatla, et al., 2017)	**Subjective**	**Objective**
Part name		X
Number of essential functions	X	
Number of non-essential functions	X	
A feeding value		X
B feeding value	X	
C feeding value	X	
D feeding value		X
A fitting value		X
B fitting value	X	
C fitting value		X
D fitting value		X
E fitting value	X	
F fitting value	X	
Relative cost		X
Complexity factor		X
Material factor		X
Minimum section		X
Tolerance factor		X
Processing cost		X
Waste coefficient		X
Material cost		X
Volume		X

The Lucas Method, similarly to the B&D, provides repeatability of 68% since 14 parameters out of 22 are subjective. Dochibhatla (Dochibhatla, et al., 2017) provided a comparison between the B&D and Lucas Method, assessing that the two methods are, indeed, very similar in the result they provide. The comparison framework (Table 25) shows that the Lucas Method requires a greater amount of information with respect to the B&D method.

Table 25 - Information in the comparison framework for the Lucas Method

Reference	(Dochibhatla, et al., 2017)
Method	Lucas method
Design phase	Detail
Analysis time	Medium
Predicted assembly time	Fair
Amount of information	High
Type of information	Objective & Subjective
Repeatability	68%
Feature to Redesign	No
Redesign guidelines	No
Possibility to add cost evaluation	Yes
Index for Design goodness	Yes

The Lucas Method cannot identify features that need to be redesigned as well as it cannot suggest modifications to engineers and designers. Finally, the predicted assembly time is fair. Indeed, the method is not time-based, as the B&D method.

Large Design for Installation (DFI)

The method called Large Design for Installation (LDFI) for long life and large size product aims at optimizing their design in terms of assembly performances (Retolaza, et al., 2021) (Mora, et al., 2020). The method uses Design Structure Matrix together with standard DFMA methods. It starts evaluating the product design according to the B&D criteria, then it evaluates the design using indices based on the Lucas Method. However, new criteria and penalty indices are proposed to consider the analysis of large products. For instance, the Index “Fine Tuning” was introduced to consider the type of tool used in the installation process (i.e., Hand Made or Special Tools). The method is based on both B&D DFMA and The Lucas Method, therefore the number of information required is the sum of the two methods. In Table 26, information and information type are collected.

Table 26 - LDFI Objective/Subjective Information

Large Design for Installation (Retolaza, et al., 2021)	Subjective	Objective
Part name		X
Part number		X
Critical part number	X	
Number of times operations required		X
A handling value		X
B handling value	X	
C handling value	X	
D handling value		X
A fitting value		X
B fitting value	X	
C fitting value		X
D fitting value		X
E fitting value	X	
F fitting value	X	
Fine tuning		X

The LDFI method has a repeatability of 60%. In fact, 9 parameters are objective while 6 are subjective. The analysis time is medium since it requires more than one day to collect and perform the analysis. The predicted time is fair since the method is based both on B&D and Lucas Method and it is not a pure time-based method. Finally, it does not provide information regarding what modifications should be implemented to improve the product assemblability (i.e., Redesign guidelines). However, it identifies features to redesign, it presents an index for evaluating the design goodness and, it can be used also to perform cost analysis. Table 27 presents the comparison framework for the LDFI method.

Table 27 - Information in the Comparison Framework for the LDFI method

Reference	(Retolaza, et al., 2021) (Mora, et al., 2020)
Method	Large Design for Installation
Design phase	Detail
Analysis time	Medium
Predicted assembly time	Fair
Amount of information	Moderate
Type of information	Objective & Subjective
Repeatability	60%

Reference	(Retolaza, et al., 2021) (Mora, et al., 2020)
Feature to Redesign	Yes
Redesign guidelines	No
Possibility to add cost evaluation	Yes
Index for Design goodness	Yes

The DFI method was applied to study the installation performances of a Solar Tracker (Remirez, et al., 2019) and an Elevator (Retolaza, et al., 2021). Analysing the proposed case studies, it was shown that the LDFI method can be applied on Large Products with a moderate amount of information to spot assembly difficulty of elements composing the product itself.

Conceptual Design for Assembly (CDfA)

The Conceptual Design for Assembly approach aims at assessing and improving complex product architectures with conceptual information. The method is composed of several steps to guide the user through the assessing and redesign phases. The core of the methodology is the creation of the hierarchical structure, called conceptual Bill of Material, to collect information of low granularity. The hierarchical structure changes according to the system that needs to be assessed. Due to this characteristic, it is not possible to clearly identify information which is subjective and objectives. In Table 28, subjective and objective information for two different systems, cabin modules, and nose-fuselage interfaces are presented.

Table 28 - Subjective and Objective information for the cabin and the nose-fuselage systems

Conceptual Design for Assembly	Subjective	Objective
CABIN		
Module		X
Type of system interfaces		X
Number of system interfaces		X
System interface position		X
Number of plates		X
Number of interfaces on plate		X
Number of interfaces stand-alone		X
Number of furnishings		X
Screwed		X
Number of mechanical interfaces		X

Conceptual Design for Assembly	Subjective	Objective
CABIN		
Mechanical interfaces position		X
Standardization		X
Design principle		X
Rigging		X
Gaps		X
Tool/assistant		X
Clearance		X
Weight		X
NOSE-FUSELAGE		
Interface type		X
Name		X
ID		X
Module in		X
Module out		X
Total length of ducts		X
Branches		X
Number of connections		X
Number of straight nodes		X
Access		X
Zone	X	
Working space size	X	
Variety of tools		X
Process	X	
Air number of bends		X
Air component's shape		X
Air component's weight		X
Air component's piece length		X
Air component's fragility		X
Electrical harness weight		X
Electrical harness length		X
Electrical component's fragility		X

The nose-fuselage model presents repeatability of 87% (20 objective parameters with respect to 23 total parameters), while the cabin of 100% (no subjective parameters). The CDfA approach presents an analysis time which is long. This is justified by the need to collect information for several parameters, in fact, the amount of information required is elevated. Moreover, differently from other methods, the CDfA methodology does not provide information about the assembly time, but only assembly complexity.

Indeed, it is not possible to estimate the goodness of the predicted assembly time. Finally, it can identify elements which need to be redesigned (i.e., complex elements), suggest design actions (redesign guidelines) and provide several indices for evaluating the design goodness. However, it does not allow adding a cost evaluation. In Table 29 the comparison framework for the CDfA method is presented.

Table 29 - Information in the comparison framework for the CDfA method

Reference	(Formentini, et al., 2021.a) (Formentini, et al., 2022.a)
Method	Conceptual Design for Assembly
Design phase	Conceptual
Analysis time	Long
Predicted assembly time	*N/A*
Amount of information	Elevate
Type of information	*Depends on the mathematical model*
Repeatability	*Depends on the mathematical model*
Feature to Redesign	Yes
Redesign guidelines	Yes
Possibility to add cost evaluation	No
Index for Design goodness	Yes

The CDfA method was applied to two aeronautical case studies (Bouissiere, et al., 2019) (Formentini, et al., 2021.a). Results showed the ability of the method to assess assembly complexity of aeronautical products, identifying critical elements, and providing redesign suggestions.

5.3 CDFA AND DFMA COMPARISON

The proposed comparison highlighted several differences among typical DFMA methods (i.e., B&D, Lucas Method), updated DFMA methods (i.e., LDFI), and the Conceptual Design for Assembly method developed in this research work.

The first one is the detail of the design required. While all DFMA methods analysed required a detailed design of the product, the CDfA method does not require it. This feature enables the application of the method on complex products, which are generally difficult to design and require a great amount of time to provide a detailed design. The LDFI, which is the only DFMA method that can study complex products, present the same limitation- In fact, LDFI required time is greater than standard DFMA methods.

Regarding the analysis time, the majority of DFMA methods analysed require a few days to be applied, the CDfA method requires a longer time. The reason lies in the nature of the method itself. In fact, CDfA method requires the involvement of several experts in order to be used, meaning from the functional decomposition until the guideline definitions. Moreover, differently from others DFMA methods, the CDfA approach requires the creation of the hierarchical structure and the mathematical model, which change according to the system analysed, extending the analysis time. However, once models are created and data collected, the analysis can be performed in a systematic way.

The need of creating different models for different systems impacts the repeatability of the method. If only objective information is used in the mathematical model, then the repeatability is 100%, otherwise, the repeatability decreases. For instance, the Cabin system presents only objective information (e.g., Gaps, Weight, Clearance, etc.) while the nose-fuselage system has subjective information such as the attribute Process, which can assume the value of “Very Easy”, “Easy”, “Normal”, “Complex” and “Very Complex”. This value may change according to the user who performs the analysis.

Among all the DFMA methods analysed, the CDfA approach is the only one that provides redesign guidelines to the user, guiding him/her through the redesign process. This feature leads to two positive impacts i) it allows modification of the product architecture by junior designers and engineers, and ii) it allows collecting and passing industrial knowledge inside the firm in a structured manner since guidelines are derived with the help of senior engineers.

As previously noted, the CDfA approach necessitates adaptation based on the system being studied. In fact, it cannot be generalized, unlike other DFMA approaches. The most significant disadvantage of the required customization is the amount of time it takes to define the method framework, which includes hierarchical structure, mathematical model, and so on. CDfA method, on the other hand, allows for greater accuracy in the analysis, allowing for the identification of aspects that cannot be studied using other methods. As a result, compared to other DFMA approaches, the Conceptual Design for Assembly method is more powerful, at the cost of a higher initial effort to build the overall framework. It can also be observed that the CDfA approach is only applicable at the conceptual design phase. This advantage comes with some drawbacks.

The first one is the analysis time. Among all methods analysed the CDfA requires a great amount of time to be performed. The main reason is the need of creating the hierarchical structure and the mathematical model, as stated above. However, another reason is the need to gather information characterized by low granularity, making the collection process complex and time-consuming. This difficulty is not present in DFMA methods which are used in later design phases, such as the detail phase. For these methods, almost all required information can be derived directly from several sources such as components drawings, 3D models, or on the product itself.

Another drawback of working with conceptual information is the elevated number of information required. Conceptual design information is characterized by great uncertainties. To reduce the effect of information uncertainties, a lot of information must be collected.

DFMA and CDfA methods analysed can provide indices to evaluate the goodness of the product in terms of assembly performances. However, the CDfA approach is the only one that cannot provide a cost estimation. In other words, it does not allow adding a model cost to the analysis. The reason lies in the nature of the method. The analysed DFMA can estimate the assembly time and assembly complexity with a set of parameters that can easily be linked to cost analysis. For instance, the B&D method can estimate the product assembly time. By adding the man-labour cost, the overall product assembly cost can be obtained. The CDfA method does not assess assembly complexity with high granularity information, and it is not time-based. Indeed, no parameters can be associated with a cost model. Table 30 summarizes the comparison between Design for Assembly methods and the CDfA approach.

Table 30 - DFMA methods and CDfA method comparison summary

Method	B&D	Lucas Method	LDFI	CDfA
Design Phase	Detail	Detail	Detail	Conceptual
Analysis Time	Short	Medium	Medium	Long
Predicted Assembly Time	Good	Fair	Fair	*N/A*
Amount of Information	Low	High	Moderate	Elevate
Types of Information	Objective & Subjective	Objective & Subjective	Objective & Subjective	*Depends on mathematical model*
Repeatability	67%	68%	60%	*Depends on mathematical model*
Feature to Redesign	Yes	No	Yes	Yes
Redesign Guidelines	No	No	No	Yes
Possibility to Add Cost Evaluation	Yes	Yes	Yes	No
Index for Design Goodness	Yes	Yes	Yes	Yes

The CDfA approach is the only method that can be used during the conceptual design phase and can suggest redesign actions to improve the product design in terms of assembly complexity. However, it requires to be personalised on the system analysed and a great amount of information to be used. Finally, differently from other DFMA methods, it does not support the implementation of a cost model. However, this drawback is not in the critical path since the literature shows that the optimization of the assembly phase at the conceptual design stage leads to great benefits in terms of product costs.

6 SOFTWARE PROTOTYPE

In recent years, the increase of computational power allowed developing software tools able to handle a huge quantity of information. To date, virtually any engineering activity is supported by software tools to help users perform the required analysis. The product development process is supported by several software tools such as the 3D CAD modeler, which allows easily creating the representation of the final product. Following this trend, a software prototype to support the Conceptual Design for Assembly methodology is presented. In the following paragraphs, a brief introduction to the development of software is introduced. Then, the software architecture of the CDfA software is presented in detail. Finally, the created minimum viable product of the software is described together with a discussion of current limitations and future developments.

6.1 SOFTWARE DEVELOPMENT

Methodologies require the support of tools enabling users to perform analysis in a lean manner. Successful software is the one that allow not educated users to perform analysis effortlessly. Moreover, the use of software can reduce the required time, saving further costs (Domeshek, et al., 1994). Before proceeding with the development of software, it is necessary to provide a software architecture. The software architecture is the core of the software itself. In literature, there are several definitions of software architecture, among all the ones proposed by Engineers C.E. (Engineers, 2000), which defines architecture as

> *"[…] fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution."* (Engineers, 2000)

The creation of a software architecture requires the observation of three principles (Bass, et al., 2003) (Martin, et al., 2018):

- **Structured organization**
  The software must be divided into small parts. Each part must be tested independently.

- **Object-Oriented programming**
  The software must be organized around data rather than functions and logic. It allows having control over each part of the code.
- **Functional programming**
  The software must be developed on unchanging variables, meaning it is necessary to use variables which do not change during the process.

Software is composed of several parts, here called components, which are released separately. Decisions regarding the software architecture must be contained in the high-level components. Components are defined high-level when they are not related to the use-case but contain general rules that need to be followed by the software. It is possible to classify the component level using the instability - abstractness graph (Figure 60) (Martin, et al., 2018).

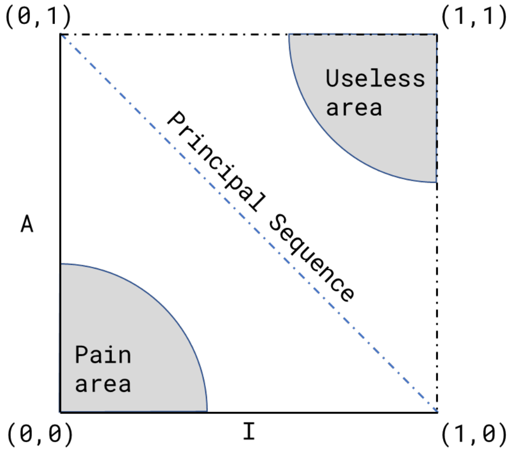


Figure 60 - Instability - Abstractness graph

Components in the pain area are stable and concrete. They are hard to handle due to the strict correlation with data. For example, databases are components in the pain area. On the other hand, components that are abstract and unstable lie in the useless area. They are subjected to frequent changes. Correct software architecture is obtained by components lying in the principal sequence.

When software is developed with the Object-Oriented programming paradigm, the concept of class is key. A class is defined as a data structure which contains both data and lines of code which are acting on the structure itself (Bass, et al., 2003).

When developing software, it is necessary to understand how to allocate classes to each component. There are three rules to follow (Bass, et al., 2003):

- **Reuse Equivalence principle**
  It must be allowed the use of components of different releases.
- **Common Closure principle**
  It is necessary to collect classes in the same components when they change due to the same cause.
- **Common Reuse principle**
  It is necessary to collect classes that are used at the same time.

Specifically, for software architecture there are five more rules to follow (Martin, et al., 2018):

- **Single Responsibility principle**
  It is necessary to separate the code affecting two different entities. For instance, the code affects two different functions.
- **Open-Close principle**
  It is necessary to create an internal hierarchy, to separate high-level functions and low-level functions.
- **Liskov Substitution principle**
  Objects working for a given type of data must work for all sub-set of that type of data.
- **Interface Segregation principle**
  Reduce dependencies among different modules, making the modification of modules easier.
- **Dependency Inversion principle**
  Avoid dependencies among modules of low levels (i.e., concrete but unstable).

Following these rules, it is possible to obtain a software architecture made of shells, where high-level components are positioned at the core, while low-level ones are positioned on outer shells. These rules lead to the creation of a well-design software architecture, which is crucial to reduce the time and cost of further modifications or updates, along with the possibility to easily implement new features.

6.2 SOFTWARE ARCHITECTURE

The software developed to support the application of the Conceptual Design for Assembly methodology is composed of four components:

- *Simplified* Digital Mock-Up generator
- Data extractor and reader
- Computational engine
- Graphic User Interface

The four components enable the creation of a shell software architecture. The obtained architecture is shown in Figure 61.

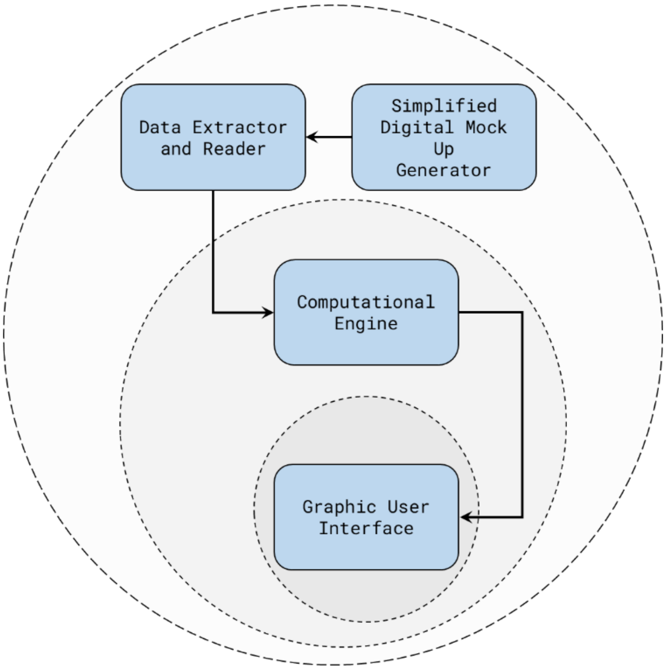


Figure 61 - Software Architecture of the Conceptual Design for Assembly tool

The architecture presents the Graphic User Interface as the core. In fact, it requires information from the computational engine to be operated. The computation engine is placed in the middle since it needs to operate with both inner and outer layers. Finally, the *simplified* Digital Mock Up Generator and the Data Extractor and Reader components are located in the outer layer, since they need to communicate with external systems to gather the required information. In the following paragraph, each component will be described and analysed.

Simplified Digital Mock-Up generator

The *simplified* Digital Mock-Up Generator component consists of a tool to create 3D representations. The 3D representations need to be composed of simple forms such as cubes, parallelepiped, cylinders, etc. Currently, several tools are available in the market to generate these representations. Any of these tools, such as CATIA V6, SolidWorks, etc. can be used by users to generate sDMU, without the need to create a program *ex-novo*. The use of already existing tools allows obtaining a better implementation inside companies, reducing the overall software development costs.

Data Extractor and Reader

It is the component that needs to extract data from the sDMU, collecting them according to the conceptual Bill of Material structure (i.e., hierarchical structure) provided by the method. Since the cBoM depends on the system analysed, it is required to make the component able to handle different data for different systems. The Data Extractor and Reader requires to be linked with the *simplified* Digital Mock-Up Generator component. This aspect is crucial: to avoid compatibility problems and make the overall coding process easier, it is necessary to choose a coding language which is supported by the software used to generate the *simplified* Digital Mock-Up. For instance, CATIA v6 is compatible with Visual Basic.NET, making the overall coding process leaner and easy to perform. The Data Extractor and Reader is divided into high-level and low-level classes. The former refers to classes which do not interact with components outside the component. In other words, high-level classes are abstract. Low-level classes, on the other hand, need to reach components outside the component and there are more likely to be changed. For instance, these classes might need to interact with external files or the user interface in order to fully operate.

The Data Extractor and Reader is composed of three classes (Figure 62):

- Import
- Structure
- Export

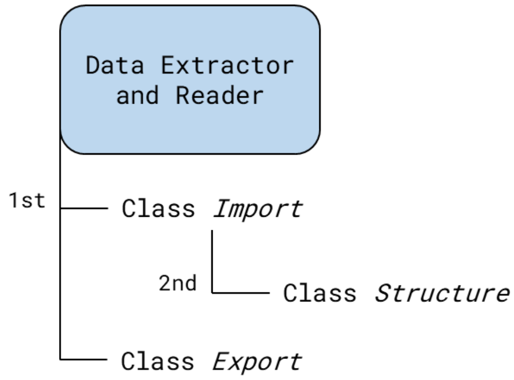


Figure 62 - Classes inside Data Extractor and Reader component

The class *Import* consists of extracting all information from the sDMU and importing it into the computer memory. At this stage, data are unstructured and do not resemble the hierarchical framework required by the method. To support this phase, a feature recognition algorithm must be used. Information such as interfaces length, modules' bounding box, number of modules in each working area, etc. can be gathered with this class. The class is placed at the low-level since it requires interaction with components outside itself.

The class *Structure* aims at providing the CDfA hierarchical framework associated with the system analysed. The imported data are clustered following the Levels, Domains, and Attributes structure defined for the product. The output of the class is the *conceptual* Bill of Material filled with information retrieved from the *simplified* Digital Mock-Up. Since the class interacts only with the Data Extractor and Reader component itself, it is placed in the high-level.

The class *Export* performs two tasks. First, it saves the hierarchical framework and data associated with it in the computer hard disk. This operation is optional, and it is required to avoid errors during computations. In fact, it allows checking if data were retrieved and collected rightfully from the Import and Structure classes. Then, it exports the data previously structured to the next component. It is a low-level class since it enables the communication between Data Extractor and Reader and Computational Engine components.

Computational Engine

The Computational Engine component is the core of the software. It collects all required information to perform mathematical steps (i.e., normalization process, collecting scores, etc.) and to obtain assembly indices. It needs to be developed in order to handle different types of information that may arise from the Data Extractor and Reader component.

Following the same concept of the Data Extractor and Reader components, the component is divided into high-level and low-level classes (Figure 63).

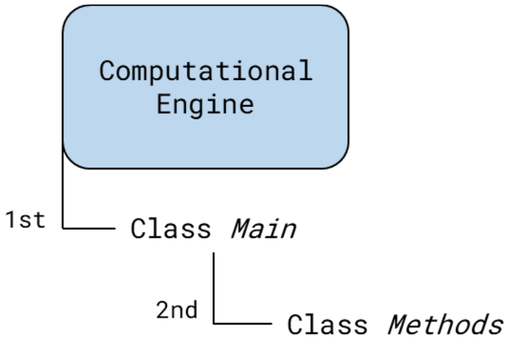


Figure 63 - Classes composing the Computational Engine

According to this decomposition, two classes are identified:

- Main
- Methods

The class *Main*, which is low-level, is used to interact with components outside of the computational engine. Information to initiating variables (e.g., lists, dataset, tables, etc.) is collected in this class. Variables are needed to correctly perform computations.

The class *Methods* belongs to the high-level. It collects functions, mathematical operators, algorithms to handle lists and tables, etc. used to perform the required mathematical steps. Every time a new mathematical operator is added to the methodology, it can be directly added to this class, without the need to rewrite the overall software from scratch.

Graphic User Interface

It is the inner component of the software architecture. In fact, it does not need to receive information outside the software itself. The aim of this component is to make the visualization of results and the navigation of the hierarchical structure user-friendly. The research area which focuses on studying user friendly GUI is called Human Interaction Design. It consists of developing GUI having in mind the final user during the whole design process. In other words, features such as bottoms' position and dimensions, labels position, etc. need to be carefully designed to facilitate the final user during the use phase. Several rules are available in the literature to design an optimized GUI. To cite a few, the Fitt's Law (Jagacinski, et al., 1980), formulated by the psychologist Fitt, suggests increasing the size of main and frequently used bottoms and placing them in easy-to-reach positions. The Poka-Yoke principle (Misiurek, 2016), derived from the lean manufacturing area, aims at reducing user errors by implementing verifications on the hard typed data, forcing users to correct themselves if the wrong information is inserted.

The GUI presents four classes, divided into high-level and low-level (Figure 64).

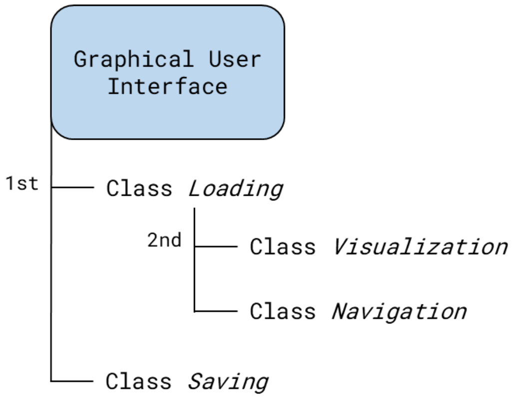


Figure 64 - Classes composing GUI component

The class *Loading* is necessary to retrieve and update all data necessary to navigate inside the cBoM framework. Data updated are: i) initial cBoM data, ii) normalized cBoM data, iii) final assembly indices, iv) sDMU visualization, and v) design guidelines suggested. Since the class *Loading* needs to communicate with components outside itself, the class is placed in the first level.

The class *Visualization* is used to initiate all variables and visualize them. It communicates with classes which are internal to the Graphic User Interface component, thus it is placed in the second level. The class *Visualization* is necessary to provide the visual structure (e.g., bottom position, dimensions, labels, etc.) of the GUI.

The class *Navigation* is required to allow users to navigate the GUI. Following the same idea of the class *Visualization*, it is placed in the second level. The class *Navigation* collects all methods required to move inside the GUI. For instance, it aggregates methods to search, filter, select, etc. data. Each method is called-out by bottoms, which were previously initialized with the class *Visualization*.

Finally, the class *Saving* is an auxiliary class. It allows the component GUI to save the current visualization, interacting with the operating system. In other words, it allows users to export desired data outside the software. Without this class, it would not be possible for the GUI component to export any information.

6.3 SOFTWARE MOCK-UP

The software architecture provided allows the creation of components which can be handled separately. To make an analogy with product architectures, the software architecture developed is *modular*. By creating a modular product architecture, updates and future developments of the software can be implemented reducing errors. The subdivision of the tool into components and classes allows approaching the software development from a conceptual point of view. The conceptual approach enables understanding and accurately planning the development of the software itself.

The provided software architecture can be further detailed by decomposing each class in order to reach a more detailed design phase. By following the architecture provided, a mock-up of the software was created. The mock-up represents the minimum viable product of the software. It presents basic functions, and it was developed for studying *only* the nose-fuselage case study. In Figure 65 the initial form of the mock-up is presented.

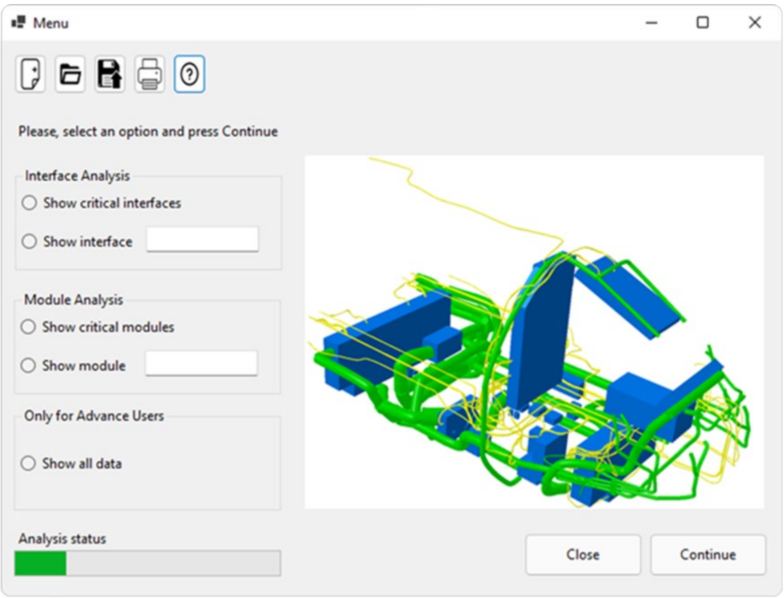
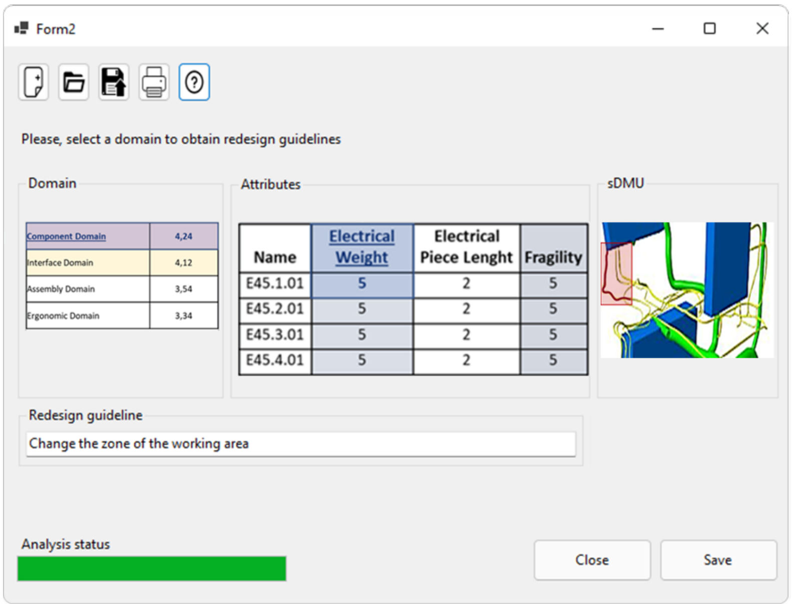


Figure 65 - GUI of the Software Mock-Up. The initial form of the software provides user with the available choices.

The GUI developed allowed to navigate the hierarchical structure using bottoms. Different options are provided for the user, which can be an expert user (i.e., educated on the CDfA method) or not. According to the level of the user, he/she can decide if: i) visualize all data collected in the *conceptual* Bill of Material, ii) show only most critical interfaces or modules, and iii) search a specific interface or module. The analysis then proceeds to show the required information, providing information regarding the complexity of the interface or module selected, and redesign suggestions. An example is provided in Figure 66 for Interface E45.



Please, select a domain to obtain redesign guidelines

Domain

Component Domain	4,24
Interface Domain	4,12
Assembly Domain	3,54
Ergonomic Domain	3,34

Attributes

Name	Electrical Weight	Electrical Piece Lenght	Fragility
E45.1.01	5	2	5
E45.2.01	5	2	5
E45.3.01	5	2	5
E45.4.01	5	2	5

Redesign guideline

Change the zone of the working area

Figure 66 - Analysis of the critical interface E45 using the Software Mock-Up

The software mock-up presents some limitations. Despite the software architecture allows connecting the tool with CAD tool, due to the nature of the mockup, this feature was not developed in this work and information needs to be retrieved manually from the sDMU. Moreover, the main challenge for the development of a fully working software is the ability to handle different cBoM according to the product analysed. In fact, the method provided requires having a different hierarchical structure and mathematical models every time a new product is analysed. To date, the creation of cBoM is not automated. The cBoM is fulfilled in an external file, which is read by the tool and creates the hierarchical structure accordingly.

Future developments will require the automation of this process through the implementation of a database collecting several cBoM structures and mathematical models for different aerospace systems. Moreover, it will be required to increase the processing performance of the software, from a computational point of view. This will be reached by optimizing the software from the coding side. Finally, the proposed software architecture can be modified accordingly to future needs that may arise during the software development phase, by adding/deleting components and/or classes.

7 CONCLUSION

The assembly phase is one of the most important stages of every product. It is well acknowledged that this can affect up to 40% of the product's cost. The recent rise in product demand and personalisation has resulted in a significant increase in the number of product units that need to be manufactured and assembled. This is especially true for products that are complex and have a significant lead time, such as airplanes. These products necessitate a significant amount of time to be assembled. For this type of product, optimizing the assembly step will result in significant cost reductions. There are currently no approaches or tools in the literature that can meet this demand.

The proposed study tries to overcome this problem, aiming at improving the assembly phase of complex products working at the conceptual design phase. This will allow avoiding the redesign of the product.

This research work developed a method called Conceptual Design for Assembly for evaluating and redesigning complex systems product architectures. The approach was developed having the aerospace industry in mind. The technique is composed of several steps which guide users through the application of the whole method. The CDfA approach is divided into two main phases which are the Conceptual Design for Assembly Assessment and the Conceptual Design for Assembly Redesign. The first one aims at assessing product architectures in terms of assemblability performances to spot components criticalities, while the second one consists of deriving design guidelines to optimize the product architecture, guiding users through components redesign. Furthermore, a mock-up of a software tool was presented to improve the whole approach’s usability.

The CDfA approach was tested on two aircraft systems, which are the nose fuselage and the cabin systems of a civil airplane. In the first case study, the approach was used to investigate the overall system installation. In fact, the installation of modules, interfaces on modules, and structural interfaces (i.e., aircraft airframe) were evaluated and challenged. Through the definition of the hierarchical structure, all elements inside the nose fuselage aircraft were mapped and critical ones were identified. Then, throughout the redesign process, the optimal design actions for minimizing product architecture assembly complexity were identified. A comparison of the modified nose fuselage design (i.e., post modification) to the original architecture (i.e., pre modification) revealed an improvement of 11% and 12% for interfaces and modules, respectively. Since both interfaces and modules were tackled and modified, the nose fuselage system provided an opportunity to demonstrate the power of the CDfA method, showing the possibility to operate with complex systems with information of low granularity.

The cabin system of a civil airplane was the second case study. Differently from the nose fuselage case study, the CDfA method was only used to study cabin modules. In fact, due to the nature of the cabin, other assembly and installation aspects could not be studied. The approach allowed for the identification of critical modules and their modification. which led to an improvement of 19 % for the family toilets and 30% for the module family galley, while the overall cabin architecture gained a 1% improvement. At the conceptual level, the cabin case study demonstrated the capacity of the CDfA method to identify critical elements and the optimal redesign activities to improve them. Moreover, it showed the ability to assess product architecture both from a global and more specific point of view (i.e., global score and element score).

A comparison of the CDfA method with other DFMA methods available in the literature was performed, in order to understand its benefits and drawbacks. The CDfA is the only method that enables conceptual assessment and, indeed, does not require a detailed design of the product. However, few limitations were identified. The CDfA method required a significant amount of time and experience to be implemented even though a concurrent approach is used to model the system of interest. In fact, the creation of the hierarchical structure, the mathematical model, the knowledge scoring matrices, and the design guidelines lists are activities that need to be performed with a concurrent approach. Furthermore, while a concurrent method improves the overall validity of the approach by allowing diverse points of view to be integrated with the same model, it also necessitates more managerial work. In other words, it is necessary to involve and coordinate different departments in the application of the CDfA. This might lead to a further increase of time. However, this is true only for the first application of the method.

During the first run of the CDfA approach, the system under study must be modelled according to the CDfA model. This process is extensively time-consuming, and it has been shown to be the main blocking point. Indeed, the modelling phase cannot be performed by non-expert users. However, once the system is modelled, it is possible to perform virtually an infinite amount of analysis within a very short period of time (i.e., hours). This possibility balances the initial modelling complexity and time effort. Moreover, the CDfA approach is system oriented. In fact, the same model cannot be used for analysing different systems. If on one hand, this characteristic leads to the drawbacks already stated above, on the other hand, it guarantees a high level of fidelity in the analysis that could not be reached with other general methods, such as the B&D or Lucas Method. In other words, the CDfA model create for each system is highly personalised.

A possible solution to speed up the use of the CDfA approach, it is to model only the main systems of interest using the CDfA hierarchical structure and build over time a repository of previous knowledge to fine-tune the CDfA model. For example, for an aircraft, only systems such as the cabin, the nose-fuselage, the wind, the tail, and the engine might be modelled using the CDfA approach. Then, if assembly technology advancements change the way some systems are installed, all that is required is to adjust the CDfA hierarchical structure of the system under investigation by adding/deleting attributes and domains. For instance, if the nose-fuselage system installation procedure is to be converted from manual to automatic, it will be necessary to eliminate the domain Ergonomic from the CDfA model without having to reconstruct the entire model.

In addition, the CDfA technique can propose design changes to improve the product architecture of the system being studied in terms of assembly complexity. It is possible to build up a repository of design guidelines and good practices for each system modelled over time in order to improve the accuracy of the CDfA redesign suggestions. However, the process of generating redesign guidelines is time-consuming and heavily reliant on expert knowledge.

Finally, it's worth noting that the CDfA method's inherent complexity, as well as the obligation to include design and manufacturing experts in the modelling process, result in a secondary effect: an increase in internal communication. From the standpoint of product design, the CDfA technique enables experts to share their knowledge across several departments (e.g., manufacturing, R&D, etc.), raising engineers' and designers' awareness of other issues that may occur during the design process. In this regard, the methodology provided in this research work expects to improve departmental collaboration and information sharing, requiring a different model for each system investigated. Approaches that attempt to offer a one-size-fits-all solution do not produce the same results. In fact, these approaches aim at collecting knowledge of different experts and using that for assessing all systems in the same way, without personalization.

The Conceptual Design for Assembly method enables the assessment and redesign of complex system architecture; however, it requires to be further developed and studied. Future developments will focus on the drawbacks highlighted above. The method needs to be expanded to include manufacturing aspects other than assembly. Indeed, the two aspects are closely linked. Moreover, the CDfA approach requires to be tested on other case studies, including the aircraft airframe system. Another interesting potential development is the extension of the CDfA approach to evaluating different figures of merits over the course of the system life cycle. Aspects like management, environmental impacts, and so on, could be added and considered within the CDfA framework. Furthermore, while the redesign process proved to be successful in identifying the best design guideline to reduce product architectural assembly complexity, the generation of design guidelines must be improved in order to limit human involvement in the process and reduce biases. Finally, the Conceptual Design for Assembly method will need to be extended to other complex systems outside of the aerospace industry and tested.

BIBLIOGRAPHY

Ahmed, S., 2005. Encouraging reuse of design knowledge: a method to index knowledge. *Design Studies,* Volume 26, p. 565–592.

Albin, S. L. & Crefeld III, P. J., 1994. Getting started: Concurrent engineering for a medium-sized manufacturer. *Journal of Manufacturing Systems,* Volume 13, p. 48–58.

Alkan, B., Vera, D., Ahmad, B. & Harrison, R., 2017. A method to assess assembly complexity of industrial products in early design phase. *IEEE Access,* Volume 6, p. 989–999.

Altfeld, H.-H., 2016. *Commercial aircraft projects: Managing the development of highly complex products.* s.l.:Routledge.

Azevedo, J. G. d., Arantes, A. C. & Costa, L. E. V. L. d., 2015. Fins module conception of the microsatellite launch vehicle based on design for manufacture and assembly method. *Journal of Aerospace Technology and Management,* Volume 7, p. 93–100.

Barbosa, G. F. & Carvalho, J., 2014. Guideline tool based on design for manufacturing and assembly (DFMA) methodology for application on design and manufacturing of aircrafts. *Journal of the Brazilian Society of Mechanical Sciences and Engineering,* Volume 36, p. 605–614.

Bass, L., Clements, P. & Kazman, R., 2003. *Software architecture in practice.* s.l.:Addison-Wesley Professional.

Benabdellah, A. C., Bouhaddou, I., Benghabrit, A. & Benghabrit, O., 2019. A systematic review of design for X techniques from 1980 to 2018: concepts, applications, and perspectives. *The International Journal of Advanced Manufacturing Technology,* Volume 102, p. 3473–3502.

Bonvoisin, J., Halstenberg, F., Buchert, T. & Stark, R., 2016. A systematic literature review on modular product design. *Journal of Engineering Design,* Volume 27, p. 488–514.

Boothroyd, G., 1987. Design for assembly—the key to design for manufacture. *The International Journal of Advanced Manufacturing Technology,* Volume 2, p. 3–11.

Boothroyd, G., Dewhurst, P. & Knight, W. A., 2010. *Product design for manufacture and assembly.* s.l.:CRC press.

Borjesson, F. R. E. D. R. I. K., 2010. *A systematic qualitative comparison of five approaches to modularity.* s.l., s.n.

Bouissiere, F. et al., 2019. *Conceptual Design for Assembly in aerospace industry: a method to assess manufacturing and assembly aspects of product architectures.* s.l., s.n., p. 2961–2970.

Bracewell, R., 2002. Synthesis based on function-means trees: schemebuilder. In: *Engineering design synthesis.* s.l.:Springer, p. 199–212.

Browning, T. R., 2015. Design structure matrix extensions and innovations: a survey and new opportunities. *IEEE Transactions on Engineering Management,* Volume 63, p. 27–52.

Brusoni, S. & Prencipe, A., 2001. Unpacking the black box of modularity: technologies, products and organizations. *Industrial and Corporate Change,* Volume 10, p. 179–205.

Bullen, G. N., 1999. *Assembly Automation and Implementation Issues,* s.l.: s.n.

Butterfield, J. et al., 2007. Optimization of aircraft fuselage assembly process using digital manufacturing.

Dahmus, J. B., Gonzalez-Zugasti, J. P. & Otto, K. N., 2000. *Modular product architecture.* s.l., s.n., p. 225–235.

Dochibhatla, S. V. S., Bhattacharya, M. & Morkos, B., 2017. *Evaluating assembly design efficiency: a comparison between Lucas and Boothroyd-Dewhurst methods.* s.l., s.n., p. V004T05A012.

Domeshek, E. A., Herndon, M. F., Bennett, A. W. & Kolodner, J. L., 1994. *A case-based design aid for conceptual design of aircraft subsystems.* s.l., s.n., p. 63–69.

Eakin, D., 2010. Aircraft Design for Manufacture and Assembly. *Encyclopedia of Aerospace Engineering.*

Emmatty, F. J. & Sarmah, S. P., 2012. Modular product development through platform-based design and DFMA. *Journal of Engineering Design,* Volume 23, p. 696–714.

Engel, A., Browning, T. R. & Reich, Y., 2017. Designing products for adaptability: insights from four industrial cases. *Decision Sciences,* Volume 48, p. 875–917.

Engineers, C. E., 2000. *Recommended Practice for Architectural Description of Software-Intensive Systems.* s.l.:Std.

Eppinger, S. & Ulrich, K., 2015. *Product design and development.* s.l.:McGraw-Hill Higher Education.

Erens, F. & Verhulst, K., 1997. Architectures for product families. *Computers in industry,* Volume 33, p. 165–178.

Ericsson, A. & Erixon, G., 1999. *Controlling design variants: modular product platforms.* s.l.:Society of Manufacturing Engineers.

Farid, A. M., 2008. Facilitating ease of system reconfiguration through measures of manufacturing modularity. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* Volume 222, p. 1275–1288.

Favi, C. et al., 2019. *Design for Assembly in the Conceptual Development of Aircraft Systems.* s.l., s.n., p. 268–278.

Favi, C. & Germani, M., 2012. A method to optimize assemblability of industrial product in early design phase: from product architecture to assembly sequence. *International Journal on Interactive Design and Manufacturing (IJIDeM),* Volume 6, p. 155–169.

Favi, C., Germani, M. & Mandolini, M., 2016. Design for manufacturing and assembly vs. design to cost: toward a multi-objective approach for decision-making strategies during conceptual design of complex products. *Procedia CIRP,* Volume 50, p. 275–280.

Formentini, G., Boix Rodríguez, N. & Favi, C., 2022.b. *Design for Manufacturing and Assembly methods in the product development process of mechanical products: a systematic literature review.* s.l.:submitted to International Journal of Advanced Manufacturing Technology.

Formentini, G. et al., 2022.a. CDFA method: a way to assess assembly and installation performance of aircraft system architectures at the conceptual design. *Research in Engineering Design,* January.

Formentini, G. et al., 2021.a. Impact assessment of design guidelines in the conceptual development of aircraft product architectures. *Procedia CIRP,* Volume 100, p. 223–228.

Formentini, G. et al., 2021.b. Conceptual design for assembly in aerospace industry: sensitivity analysis of mathematical framework and design parameters. *Proceedings of the Design Society,* Volume 1, p. 731–740.

Frigant, V. & Talbot, D., 2005. Technological determinism and modularity: lessons from a comparison between aircraft and auto industries in Europe. *Industry and Innovation,* Volume 12, p. 337–355.

Gao, F., Xiao, G. & Simpson, T. W., 2010. Identifying functional modules using generalized directed graphs: Definition and application. *Computers in Industry,* Volume 61, p. 260–269.

Gavel, H., Ölvander, J., Johansson, B. & Krus, P., 2007. *Aircraft fuel system synthesis aided by interactive morphology and optimization.* s.l., s.n., p. 653.

Gomez, A., Rios, J., Mas, F. & Vizan, A., 2016. Method and software application to assist in the conceptual design of aircraft final assembly lines. *Journal of Manufacturing Systems,* Volume 40, p. 37–53.

Guarino, N., 1995. Formal ontology, conceptual analysis and knowledge representation. *International journal of human-computer studies,* Volume 43, p. 625–640.

Gunasekaran, A., 1999. Agile manufacturing: a framework for research and development. *International journal of production economics,* Volume 62, p. 87–105.

Gupta, M. & Kumar, S., 2019. *Design Efficiency Analysis Towards Product Improvement Using DFMA.* s.l., s.n., p. 1–6.

Helmer, R., Yassine, A. & Meier, C., 2010. Systematic module and interface definition using component design structure matrix. *Journal of Engineering Design,* Volume 21, p. 647–675.

Hermansson, T., Bohlin, R., Carlson, J. S. & Söderberg, R., 2013. Automatic assembly path planning for wiring harness installations. *Journal of manufacturing systems,* Volume 32, p. 417–422.

Herrera, A., 1997. The effectiveness of Design for Manufacturing and Assembly as applied to the design of the AH64D helicopter. *SAE transactions,* p. 346–352.

Herrera, A., 1998. *Design for manufacture and assembly, makes a difference among the boeing lean design initiatives,* s.l.: s.n.

Hirtz, J. et al., 2002. A functional basis for engineering design: reconciling and evolving previous efforts. *Research in engineering Design,* Volume 13, p. 65–82.

Hölttä, K., Suh, E. S., de Weck, O. & others, 2005. *Tradeoff between modularity and performance for engineered systems and products.* s.l., s.n., p. 449–450.

Hölttä-Otto, K., 2005. *Modular product platform design.* s.l.:Helsinki University of Technology.

Huang, G. Q., Lee, S. W. & Mak, K. L., 1999. Web-based product and process data modelling in concurrent "design for X". *Robotics and Computer-Integrated Manufacturing,* Volume 15, p. 53–63.

Jagacinski, R. J. et al., 1980. Fitts' law and the microstructure of rapid discrete movements.. *Journal of Experimental Psychology: Human Perception and Performance,* Volume 6, p. 309.

Jose, A. & Tollenaere, M., 2005. Modular and platform methods for product family design: literature analysis. *Journal of Intelligent manufacturing,* Volume 16, p. 371–390.

Judt, D. M. & Lawson, C., 2016. Development of an automated aircraft subsystem architecture generation and analysis tool. *Engineering Computations.*

Jung, S. & Simpson, T. W., 2017. New modularity indices for modularity assessment and clustering of product architecture. *Journal of Engineering Design,* Volume 28, p. 1–22.

Jun, H.-B., Park, J.-Y. & Suh, H.-W., 2006. Lead time estimation method for complex product development process. *Concurrent Engineering,* Volume 14, p. 313–328.

Kirschman, C. F. & Fadel, G. M., 1998. Classifying functions for mechanical design.

Kroll, E., 2013. Design theory and conceptual design: contrasting functional decomposition and morphology with parameter analysis. *Research in Engineering Design,* Volume 24, p. 165–183.

Kuo, T.-C., Huang, S. H. & Zhang, H.-C., 2001. Design for manufacture and design for 'X': concepts, applications, and perspectives. *Computers & industrial engineering,* Volume 41, p. 241–260.

Kusar, J., Duhovnik, J., Grum, J. & Starbek, M., 2004. How to reduce new product development time. *Robotics and Computer-Integrated Manufacturing,* Volume 20, p. 1–15.

Lai, K. & Wilson, W. R. D., 1989. FDL—a language for function description and rationalization in mechanical design.

Leenders, R. T. A. J., Van Engelen, J. M. L. & Kratzer, J., 2007. Systematic design methods and the creative performance of new product teams: Do they contradict or complement each other?. *Journal of Product Innovation Management,* Volume 24, p. 166–179.

Li, T., Lockett, H. & Lawson, C., 2020. Using requirement-functional-logical-physical models to support early assembly process planning for complex aircraft systems integration. *Journal of Manufacturing Systems,* Volume 54, p. 242–257.

Little, A. D., Wood, K. L. & McAdams, D. A., 1997. *Functional analysis: A fundamental empirical study for reverse engineering, benchmarking and redesign.* s.l., s.n., p. V003T30A015.

Li, Z., Cheng, Z., Feng, Y. & Yang, J., 2013. An integrated method for flexible platform modular architecture design. *Journal of Engineering Design,* January, Volume 24, p. 25–44.

Lockett, H., Fletcher, S. & Luquet, N., 2014. Applying design for assembly principles in computer aided design to make small changes that improve the efficiency of manual aircraft systems installations. *SAE International Journal of Aerospace,* Volume 7, p. 284–291.

Ltd, L. E. S., 1993. *Design For Manufacture and Assembly Practitioners Manual, Version 10.* s.l.:University of Hull.

Lyu, J. & Chang, L.-Y., 2010. A reference model for collaborative design in mould industry. *Production Planning and Control,* Volume 21, p. 428–436.

Martin, R. C., Grenning, J. & Brown, S., 2018. *Clean architecture: a craftsman's guide to software structure and design.* s.l.:Prentice Hall.

Mas, F., Ríos, J., Gómez, A. & Hernández, J. C., 2016. Knowledge-based application to define aircraft final assembly lines at the industrialisation conceptual design phase. *International Journal of Computer Integrated Manufacturing,* Volume 29, p. 677–691.

Mas, F., Rios, J., Menendez, J. L. & Gomez, A., 2013. A process-oriented approach to modeling the conceptual design of aircraft assembly lines. *The International Journal of Advanced Manufacturing Technology,* Volume 67, p. 771–784.

McDermott, C. M. & O'connor, G. C., 2002. Managing radical innovation: an overview of emergent strategy issues. *Journal of Product Innovation Management: an international publication of the product development & management association,* Volume 19, p. 424–438.

Milner, T., Volas, M. & Sanders, A., 2013. Systems engineering methodology for linking requirements to design complexity and manufacturing trade space constraints. *Procedia Computer Science,* Volume 16, p. 947–956.

Misiurek, B., 2016. *Standardized work with TWI: eliminating human errors in production and service processes.* s.l.:CRC Press.

Monnoyer, M.-C. & Zuliani, J.-M., 2007. The decentralisation of airbus production and services. *The Service Industries Journal,* Volume 27, p. 251–262.

Mora, B. et al., 2020. *DEVELOPMENT OF A NEW DESIGN METHODOLOGY FOR LARGE SIZE PRODUCTS BASED ON DSM AND DFMA.* s.l., s.n., p. 2315–2324.

Musen, M. A. & Schreiber, A. T., 1995. *Architectures for intelligent systems based on reusable components.* s.l.:Elsevier.

Nuñez, J. R., Anderton, C. R. & Renslow, R. S., 2018. Optimizing colormaps with consideration for color vision deficiency to enable accurate interpretation of scientific data. *PloS one,* Volume 13, p. e0199239.

Ölvander, J., Lundén, B. & Gavel, H., 2009. A computerized optimization framework for the morphological matrix applied to aircraft conceptual design. *Computer-Aided Design,* Volume 41, p. 187–196.

Owensby, E. et al., 2011. *Evaluation and comparison of two design for assembly methods: subjectivity of information inputs.* s.l., s.n., p. 721–731.

Pahl, G. & Beitz, W., 2013. *Engineering Design: A Systematic Approach.* s.l.:Springer Science & Business Media.

Paik, J. K. & Thayamballi, A. K., 2007. *Ship-shaped offshore installations: design, building, and operation.* s.l.:Cambridge University Press.

Rampersad, H. K., 1996. Integrated and assembly oriented product design. *Integrated Manufacturing Systems.*

Reed, N., Scanlan, J., Wills, G. & Halliday, S. T., 2011. Knowledge use in an advanced manufacturing environment. *Design Studies,* Volume 32, p. 292–312.

Remirez, A. et al., 2019. New design for assembly methodology adapted to large size products: Application on a solar tracker design. *Procedia CIRP,* Volume 84, p. 468–473.

Retolaza, I. et al., 2021. New design for installation (DFI) methodology for large size and long life cycle products: application to an elevator. *Proceedings of the Design Society,* Volume 1, p. 2237–2246.

Sadraey, M. H., 2012. *Aircraft design: A systems engineering approach.* s.l.:John Wiley & Sons.

Sage, A. P., 1996. *Manufacturing systems engineering and management.* s.l., s.n., p. 1–9.

Saltelli, A. et al., 2006. Sensitivity analysis practices: Strategies for model-based inference. *Reliability Engineering & System Safety,* Volume 91, p. 1109–1125.

Samadhi, T. A., Ma'ruf, A., Toha, I. S. & others, 2018. *A Framework for the Development of Automatic DFA Method to Minimize the Number of Components and Assembly Reorientations.* s.l., s.n., p. 012083.

Samy, S. N. & ElMaraghy, H., 2010. A model for measuring products assembly complexity. *International Journal of Computer Integrated Manufacturing,* Volume 23, p. 1015–1027.

Sanders, D., Tan, Y. C., Rogers, I. & Tewkesbury, G. E., 2009. An expert system for automatic design-for-assembly. *Assembly Automation.*

Shupe, J. A., Mistree, F. & Sobieszanski-Sobieski, J., 1987. Compromise: An effective approach for the hierarchical design of structural systems. *Computers & structures,* Volume 26, p. 1027–1037.

Sohlenius, G., 1992. Concurrent engineering. *CIRP annals,* Volume 41, p. 645–655.

Srinivasan, R. S., Wood, K. L. & McAdams, D., 1995. *Functional tolerancing: A design for manufacturing methodology.* s.l., s.n., p. 465–482.

Staab, S., Studer, R., Schnurr, H.-P. & Sure, Y., 2001. Knowledge processes and ontologies. *IEEE Intelligent systems,* Volume 16, p. 26–34.

Stone, R. B., McAdams, D. A. & Kayyalethekkel, V. J., 2004. A product architecture-based conceptual DFA technique. *Design studies,* Volume 25, p. 301–325.

Stone, R. B., Wood, K. L. & Crawford, R. H., 2000. A heuristic method for identifying modules for product architectures. *Design studies,* Volume 21, p. 5–31.

Sudin, M. N., Chin, N. S., Shamsudin, S. A. & Yusuff, M. A., 2016. Design efficiency analysis towards product improvement for eco-friendly using DFMA method. *The Open Mechanical Engineering Journal,* Volume 10.

Swift, K. G. & Brown, N. J., 2003. Implementation strategies for design for manufacture methodologies. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* Volume 217, p. 827–833.

Thompson, M. K., Jespersen, I. K. J. & Kjærgaard, T., 2018. Design for manufacturing and assembly key performance indicators to support high-speed product development. *Procedia CIRP,* Volume 70, p. 114–119.

Tseng, K. C. & Abdalla, H., 2006. A novel approach to collaborative product design and development environment. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture,* Volume 220, p. 1997–2020.

van Beek, T. J., Erden, M. S. & Tomiyama, T., 2010. Modular design of mechatronic systems with function modeling. *Mechatronics,* December, Volume 20, p. 850–863.

Wymore, A. W., 2018. *Model-based systems engineering.* s.l.:CRC press.

Yuan, Z., Sun, C. & Wang, Y., 2018. Design for Manufacture and Assembly-oriented parametric design of prefabricated buildings. *Automation in Construction,* Volume 88, p. 13–22.

APPENDIX A: DESIGN GUIDELINES TABLES

Table 31 - Design Guidelines for Nose-Fuselage Interfaces' Installation

#	**Source**	**Related Domain**	**Related Attribute**	**Design Guidelines**	**Explanation**	**Applicable**
D.G. number	*Type of Source*	*Domain of Interest*	*Main attribute related to the D.G.*	*verb + object*	*goal of the design guidelines, suggestion to reach the best design*	*applicability of the D.A.*
1	Scoring Matrix	Interface Domain	Total length of ducts	Reduce the length of air interface	Reducing the length of the interface will reduce the overall assembly effort, since a shorter interface need to be installed.	YES
2	Scoring Matrix	Interface Domain	Branches	Reduce the number of branches	Interfaces with fewer branches are easier to handle and install. Reducing the branches will lead to better interface management.	YES
3	Scoring Matrix	Interface Domain	Total length of harness	Reduce the length of electrical interface	Reducing the length of the interface will reduce the overall assembly effort, since a shorter interface needs to be installed	YES
4	Scoring Matrix	Interface Domain	Number of connections	Reduce the number of connections	Interfaces with fewer connections are easier to handle and install. Reducing the number of connections will lead to better interface management.	YES

#	Source	Related Domain	Related Attribute	Design Guidelines	Explanation	Applicable
5	Scoring Matrix	Interface Domain	Number of straight nodes	Reduce the number of straight nodes	Reducing the number of straight nodes will increase the handleability of electrical interfaces.	YES
6	Scoring Matrix	Ergonomic Domain	Access	Change the access to the working area	Changing the access inside the working area can improve the ergonomics of the assembly process	YES
7	Scoring Matrix	Ergonomic Domain	Zone	Change the zone of the working area	Moving the interface in a more ergonomic-friendly zone will improve ergonomics aspects, reducing assembly complexity	YES
8	Scoring Matrix	Ergonomic Domain	Working Space Size	Move interface in a bigger working area	Moving the interface and, indeed, the installation in a bigger area will increase the available space to perform assembly operations	YES
9	Scoring Matrix	Assembly Domain	Variety of Tools	Reduce the variety of tools needed	Reducing the variety of tools will increase the assembly performances, enabling the installation of more items with the same tool	YES

#	Source	Related Domain	Related Attribute	Design Guidelines	Explanation	Applicable
10	Scoring Matrix	Assembly Domain	Process	Reduce assembly process complex	The reduction of the assembly process complexity will lead to an overall improvement of the product complexity	YES
11	Scoring Matrix	Component Domain	Air number of Bends	Decrease the number of bends in air component	Decreasing the amount of bends in air components will lead to a better assembly process, requiring fewer operations to orientate the component	YES
12	Scoring Matrix	Component Domain	Air component's shape	Simplify the shape of air component	Making air components simple in shape (i.e., straight components) will reduce the operations required to orientate the component	YES
13	Scoring Matrix	Component Domain	Air component's weight	Reduce the weight of air component	Reducing the weight of air components will increase the handleability, making the overall assembly process easier	YES
14	Scoring Matrix	Component Domain	Air component's piece length	Reduce the length of air component	Reducing the length of air components will increase the handleability, making the overall assembly process easier	YES

#	Source	Related Domain	Related Attribute	Design Guidelines	Explanation	Applicable
15	Scoring Matrix	Component Domain	Air component's fragility	Make air component less fragile	Making air components less fragile will improve assemblability performances	YES
16	Scoring Matrix	Component Domain	Electrical harness weight	Reduce the weight of electrical harness	Reducing the weight of electrical components will increase the handleability, making the overall assembly process easier	YES
17	Scoring Matrix	Component Domain	Electrical harness length	Reduce the length of electrical harness	Reducing the length of electrical components will increase the handleability, making the overall assembly process easier	YES
18	Scoring Matrix	Component Domain	Electrical component's fragility	Make electrical component less fragile	Making electrical components less fragile will improve assemblability performances	YES

Table 32 - Design Guidelines for Nose-Fuselage Modules' Installation

#	Source	Related Domain	Related Attribute	Design Guidelines	Explanation	Applicable
D.G. number	*Type of Source*	*Domain of Interest*	*Main attribute related to the D.G.*	*verb + object*	*goal of the design guidelines, suggestion to reach the best design*	*applicability of the D.A.*
1	Scoring Matrix	Mechanical Domain	Number of Mechanical Interfaces	Reduce the number of mechanical interfaces	The reduction of mechanical interfaces will require the change of the module design to withstand forces. Usually, it is a complex modification to perform.	YES
2	Scoring Matrix	Mechanical Domain	Position of Mechanical Interfaces	Change the position of mechanical interfaces	Change the position of mechanical interfaces to make the assembly process easier.	YES
3	Scoring Matrix	Mechanical Domain	Access	Change the module access	Change the access to simply the entering of the module inside the working area.	YES
4	Scoring Matrix	Handling Domain	Tool/Assistant	Reduce the number of operators required	Modifying the number of assistants, adopting a different tool, will make the overall module assembly process easier.	YES

#	Source	Related Domain	Related Attribute	Design Guidelines	Explanation	Applicable
5	Scoring Matrix	Handling Domain	Weight	Reduce the module weight	Reducing the module weight will make the overall assembly process easier. It can be done in several ways.	YES
6	Scoring Matrix	Handling Domain	Clearance	Modify the module clearance	Increasing the clearance will allow to have bigger space around the module to perform assembly operations. The clearance is an assembly-sequence depending on feature. Not implementable for the time-being.	NO

Table 33 - Design Guidelines for Cabin Modules' Installation

#	Source	Related Domain	Related Attribute	Design Guidelines	Explanation	Applicable
D.G. number	*Type of Source*	*Domain of Interest*	*Main attribute related to the D.G.*	*verb + object*	*goal of the design guidelines, suggestion to reach the best design*	*applicability of the D.A.*
1	Scoring Matrix	Mechanical Domain	Number of mechanical interfaces, weight	Reduce the number of mechanical connections	The minimum number of mechanical connections is related to the ratio **Weight/Number of Interfaces.** Try to reach the value of 28.	YES
2	Scoring Matrix	Mechanical Domain	Standardization	Change part standardization to reach *all standard parts*	Modify the standardization of the part to obtain the *all standard parts* value.	YES
3	Scoring Matrix	Mechanical Domain	Design Principle	Change the attachment design principle to reach *1 design per set of attachments for all modules*	Modify the principle of the attachment design to obtain *1 design per set of attachments (Upper, Middle, Lower) for all modules.*	YES

#	Source	Related Domain	Related Attribute	Design Guidelines	Explanation	Applicable
4	Scoring Matrix	Mechanical Domain	Rigging	Modify the rigging process to avoid rigging	Modify the rigging process to obtain *no rigging*. The modification of rigging will increase the number of furnishing since it is not necessary to rig. Moreover, it will create gaps.	YES
5	Scoring Matrix	Furnishing Domain	Number of furnishings	Modify the number of furnishings to avoid furnishing	Modify the number of furnishings. Try to eliminate the need of furnishing (*No Furnishing*)	YES
6	Scoring Matrix	Furnishing Domain	Screwed	Change the screwing process to avoid screwing	Modify the screwing process. Try to eliminate the need of screwing process (N*o Screwed).*	YES
7	Scoring Matrix	System Domain	Number of interfaces on plate	Change interfaces on plates to collect more interfaces on a plate (i.e., move stand-alone interfaces on plates)	Change the number of interfaces on each plate. Try to obtain one plate for type of interface. It will increase the number of plates.	YES

#	Source	Related Domain	Related Attribute	Design Guidelines	Explanation	Applicable
8	Scoring Matrix	Mechanical Domain	Gaps	Change gaps to avoid visible gaps	Avoid visible gaps (i.e., reach *no visible gaps*).	YES
9	Scoring Matrix	System Domain	Interfaces Position	Change the position of system interfaces	Move the position from Upper/Lower to Central position	YES