Conceptual Design for Assembly methodology formalization: systems installation analysis and manufacturing information integration in the design and development of aircraft architectures

Original
Conceptual Design for Assembly methodology formalization: systems installation analysis and manufacturing information integration in the design and development of aircraft architectures / Formentini, G.; Bouissiere, F.; Cuiller, C.; Dereux, P. -E.; Favi, C.. - In: JOURNAL OF INDUSTRIAL INFORMATION INTEGRATION. - ISSN 2452-414X. - 26:(2022), p. 100327.100327. [10.1016/j.jii.2022.100327]

Publisher:
Elsevier B.V.

Published
DOI:10.1016/j.jii.2022.100327

Authors

Giovanni Formentini[a], Francois Bouissiere[b], Claude Cuiller[b], Pierre-Eric Dereux[b], Claudio Favi[a]

[a] University of Parma, Parco Area delle Scienze 181/A, 43124, Parma, Italy

[b] Airbus S.A.S., 1 Rond-Point Maurice Bellonte, 31700, Blagnac, France

Corresponding author: Giovanni Formentini, giovanni.formentini@unipr.it, +39 0521 90 6344, Parco Area delle Scienze 181/A, 43124, Parma, Italy

Title

Conceptual Design for Assembly methodology formalization: systems installation analysis and manufacturing information integration in the design and development of aircraft architectures

Abstract

In recent years, the air transport market has experienced strong growth, increasing the demand for new civil aircraft, challenging the actual production rate of aerospace industries. The bottleneck of the production for the aviation industry lies in the capability of the manufacturing and assembly facilities to fulfill the module arrangement in the current design. The development of optimized product architecture requires the implementation of design for assembly principles at the conceptual design phase closing the gap between the design and the production departments. The study proposes a Conceptual Design for Assembly (CDfA) methodology which aims at the assessment of aircraft systems installation and assembly at the early phase of product development (conceptual design). The CDfA methodology allows comparing assembly performance of different aircraft architectures identifying critical modules and interfaces as well as assembly/installation issues. The methodology is based on a specific framework (hierarchical structure) which is characterized by levels, domains, and attributes. Levels enable the analysis of product architectures at different levels of granularity, splitting the global analysis into sub-problems (problem discretization). Domains and attributes are defined with a knowledge-based engineering approach considering available information at the conceptual design phase and production criteria. A complex system (the nose fuselage of a commercial aircraft) was chosen as a case study to test the robustness of the methodology in relation to the assembly performance observed within the manufacturing facilities. Results revealed the architectural elements (modules and interfaces) that contribute to inefficient assembly operations, as well as the rationales enabling to elaborate alternative architectures for an improved product industrial efficiency.

Keywords

product design and development; product architecture; aircraft; system installation; modularity; conceptual design; design for manufacturing and assembly; DFMA.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Summary

1. Introduction

The aviation industry, as well as the demand for commercial aircraft, is growing fast in the last decades, notwithstanding the recent COVID-19 pandemic. Aircraft manufacturers are being urged to enhance their industrial performance while keeping costs, safety requirements, and manufacturing lead times under control. To achieve this goal, product and manufacturing engineers are called to work closely together from the beginning of the design process to create aircraft architecture that meets both product and industrial performances. The conceptual design is the design stage when optimized product architectures are conceived with lower costs in terms of manufacturing and assembly. Design for Assembly (DFA) methods have been consolidated over years allowing to consider assembly concerns throughout the aircraft development process. DFA methods have been developed for late design phases (i.e., embodiment design and detail design) where project parameters and design information are available. On the other hand, DFA approaches developed for conceptual design phase are not mature enough, despite the significant impact that these methods can lead to the manufacturing and industrial production.

The purpose of this paper is to propose and formalize a Conceptual Design for Assembly (CDfA) methodology which aims at the assessment of assembly/installation performance during the development of product architectures in the specific context of complex products such as an aircraft. The CDfA methodology is developed to analyse aircraft system architectures in the early phases of product design (i.e., conceptual design). The CDfA methodology is a result of applied research in engineering design starting from aeronautical industry needs and requirements. Results of the CDfA methodology allow to identify critical modules/interfaces to install as well as the rationale behind these criticalities. Following the results of the CDfA assessment, new alternative architectures can be developed and compared with the original one to verify possible improvements. In addition, architectural design guidelines can be retrieved based on the CDfA result, inspiring the re-design phase.

Two novel concepts have been developed within the CDfA methodology. The first one recalls the possibility to translates product architecture data into a set of numerical values associated to fit for assembly assessment criteria (manufacturing-driven knowledge-based engineering). Assembly/manufacturing knowledge is turned from tacit implicit knowledge (unstructured information) to explicit knowledge (scoring matrices with numerical data). The second concept recalls the possibility to create a mathematical model (framework) enabling the analysis of the overall aircraft assembly problem by using sub-problems that are limited in terms of complexity (problem discretization). The model is characterized by: i) *levels* which represent the boundaries of a given problem/sub-problem, ii) *attributes* which are the identified fit for assembly assessment criteria, and iii) *domains* that represent a collection of attributes belonging to the same assembly/manufacturing aspect.

The paper is structured as follow: after the analysis of literature on this field (section 2), the CDfA methodology is described in detail (section 3). A case study investigates the CDfA analysis for a nose fuselage of a civil aircraft (section 4), and a conclusion section discusses limitations and future developments of the presented approach (section 5).

2. Literature background

The engineering design process is characterized by several phases in which several disciplines are collaborating in the development of products and industrial goods. Pahl et al., (Pahl et al. 2007) provided a classification of the engineering design process by identifying four distinct phases: i) planning and task clarification, ii) conceptual design, iii) embodiment design, and iv) detail design. A more challenging engineering environment has led to the creation of a concurrent engineering methods, where constant interactions between the design team and other departments (i.e., manufacturing and production) are required (Boothroyd et al. 2011) (Lyu and Chang 2010). Despite the important benefits of concurrent engineering methods, such as the shortening of the design phase and the reduction of product lead time (K. C. Tseng & Abdalla, 2006), several issues arose at the management level to control specific aspects of the engineering process (Jun et al. 2006). This gave rise to design for *X* (DfX) methods, in which the *X* stands for the optimization objective (e.g., assembly, manufacturing, cost, etc.) (Huang et al. 1999; Kuo et al. 2001; Holt and Barnes 2010). Due to the relevance of the subject in terms of time and cost (Favi et al. 2016), design for manufacturing and assembly (DfMA) has gotten particular attention among the DfX techniques (Coma et al. 2004). Assembly and installation are critical in large and complex products like an aircraft, where they account for more than 40% of the final cost (Bullen 1999; Paik et al. 2009; Hermansson et al. 2013).

Few attempts have been made to develop DfMA methods compliant with aerospace products. Some of them evaluated the assembly performance considering the aircraft assembly line (Butterfield et al. 2007; Mas et al. 2013, 2016; Gómez et al. 2016), while others were focusing on the identification of feasible improvements for the manual installation of the wiring system (Lockett et al. 2014). Despite the important contribution of these works, a few concerns were raised, such as the level of detail necessary to utilize and manage information that feeds DFMA techniques, which is mostly available in the late stages of design. The way to overcome this issue lies in the possibility to work at the conceptual design phase when only partial and high-level information is available (El-Nounu et al. 2018, Pokojski et al. 2018, Bouissiere et al. 2019). The need to use schemes (i.e., functional and modular representations) that incorporate a restricted amount of data with high granularity is one of the main challenges while working at the conceptual phase. Several methods and tools were developed to create schemes able to represent the gathered data, such as the black box model (Pahl et al. 2007) the function means tree (Malmqvist 1997) and the functional evolution process (Shimomura et al. 1998). All these tools describe how product functions are assigned to physical modules/components creating a product architecture (Ulrich 1995). Improved product architectures, as well as the development of the modular products, can have a favourable influence on the assembly phase (Jiao et al. 2007;AlGeddawy and ElMaraghy 2013;Stief et al. 2020). This outcome was demonstrated by the development of DFMA methods for the conceptual design phase (Stone et al. 2004;Favi and Germani 2012), even if the proposed methods were developed for small appliances or electric tools made of few components. On the other hand, moving on complex systems, the assessment of assembly advantages brought by modularity requires validation through mathematical models (Bonvoisin et al. 2016). The theory of modularity was first proposed in the aviation sector to address the problem of producing aircraft sub-parts in various geographical locations (Monnoyer and Zuliani 2007). With the development of electronic components, it became feasible to design individual sub-parts or modules such as wings, cockpits, and cabins, which could then be assembled at a later stage (Frigant and Talbot 2005). In relation to aircraft assembly, modular analyses were developed concerning the design of cabin interior or other reconfigurable systems (Jonas et al. 2009; Jung and Simpson 2017). Furthermore, product modularity allows for the development of aircraft product families with several benefits in terms of product reconfigurability, changeability (product evolution), serviceability, and survivability among others (Erens and Verhulst 1997; Miller et al. 2002; Fricke and Schulz 2005; Siddiqi and de Weck 2008). However, for aircraft and aircraft systems, the adoption of a modular approach may not be advantageous due to other requirements (e.g., weight reduction, fuel consumption, etc.) (Hölttä et al. 2005).

Following the outcomes of the literature analysis, conceptual design is the most critical phase to prevent installation and assembly issues. Engineering processes based on functional and modular decomposition seem to be the most promising to develop optimized product architectures. Since the assembly activities are the bottleneck of aircraft production due to the increasing product complexity, aerospace industries are experiencing a gap closure between product conceptualization and manufacturing (i.e., assembly and installation) in the early stages of design. Preliminary research tried to address assembly and manufacturing issues within the context of aircraft architecture development, (e.g., problem formalization, the definition of assembly parameters, etc.) but dedicated methodologies and tools are necessary to fill this gap.

3. Conceptual Design for Assembly methodology

In this section, the CDfA (Conceptual Design for Assembly) methodology is presented. The methodology is based on four main phases as presented in Fig. I (CDfA methodology flowchart). For each phase of the methodology, tasks and dedicated tools used to perform the analysis are presented.

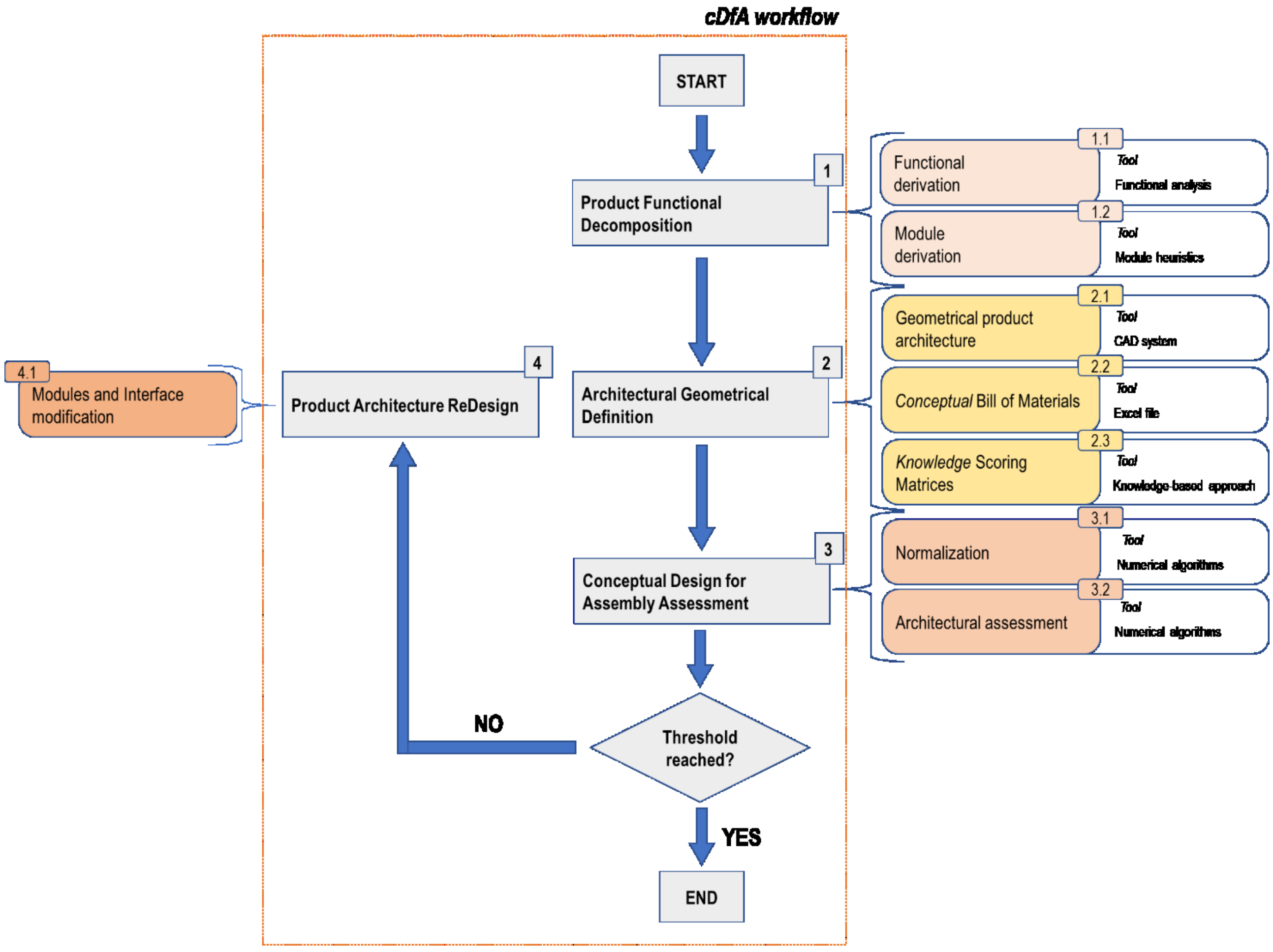


Fig. I - Conceptual Design for Assembly methodology (flowchart)

Here below, each phase of the CDfA methodology is described in detail.

3.1. Phase 1: product functional decomposition

The product functional decomposition is the starting point for the investigation. It allows for the identification of functional modules and their functional interconnections, which are subsequently used to describe physical modules and their physical links. The functional decomposition of a product encompasses two tasks: (i) functional derivation and (ii) module derivation. It is worth noting that these two tasks are not a novel step considering the available literature in this field; however, few changes have been introduced to proceed on to the next phase of the process.

Functional derivation

The functional analysis is used to obtain the functional scheme of the product under investigation (i.e., an aircraft or a complex system belonging to it). The functional analysis consists of defining primary and auxiliary functions, as well as basic fluxes used to connect them. The black-box model is used to describe functions, while material, energy, and signal are the basic fluxes used to describe flows going in and out of functions. A modified version of the functional analysis presented by (Pahl et al. 2007) is used in the proposed method to describe the given product (aircraft). Different types of fluxes are calculated and linked with a specific colour within the same functional flow(see Table 1). For example, if two general functions are coupled by a flux of material, the flux of material can be further specified by indicating the type (e.g., gas), the sub-type (e.g., air) and the colour (e.g., green). The modified approach allows the following improvements: i) to make easier the readability of the functional representation for large and complex systems, ii) to increase the level of detail of the functional analysis without requiring data from a lower design phase, iii) provide a better understanding of the implications of each requirement, and iv) to make the transition from fluxes (functional representation) to physical interconnections (interfaces) in an easier way.

After all the basic fluxes have been identified, the system interfaces among modules are obtained. Starting from the defined fluxes, four interfaces are derived: i) electrical, ii) mechanical, iii) fluid, and iv) air. There is a relation between the interface type and the basic flux that originated the interface, even if the same interface can be associated to different basic fluxes sub-types (e.g., interface fluid is associated to fuel, oil, liquid waste, and water fluxes sub-types). Table 1 shows all basic fluxes detected in an aircraft (Hirtz et al. 2002) and the relationship that exists with the four identified interfaces. There are no interactions connected with human and solid fluxes due to their inherent nature.

TABLE 1 - BASIC FLUXES AND INTERFACES FOR AIRCRAFT

Aircraft Basic Fluxes							Aircraft Interfaces		
Basic Flow	***Symbol***	***Type***	***Explanation***	***Example***	***Sub-Type***	***Colour***	***Type***	***Interface***	***Colour***
Material	⇧	Human	All or part of a person who physically interacts with the system	Cabin crew entering the cabin	*N.A.*	Black ⬛	Human	*N.A.*	*N.A.*
		Solid	Object with mass and shape which physically interacts with the system	Luggage entering the hat rack	Waste	Green 🟩	Solid	*N.A.*	*N.A.*
					Objects/Parts	Grey ■			
		Liquid	Fluid which physically interacts with the system	Fuel that flows in pipes	Liquid waste	Purple 🟪	Liquid	Fluid	Blue 🟦
					Fuel	Brown 🟫			
					Oil	Pink ■			
					Water	Blue 🟦			
		Gas	Gas that physically interacts with the system	Air entering the fans and ducts	Air	Green 🟩	Gas	Air	Green 🟩
					Gas mixture	Orange 🟧			
					O2	White ⬜			
Energy	↑	Human	Work performed by a person on the system	Energy generated by the pilot to move the cloche	*N.A.*	Black ⬛	Human	*N.A.*	*N.A.*
		Acoustic	Work performed to produce and trasmitt sound	Energy generated by turboprop motion converted into noise	*N.A.*	Orange 🟧	Acoustic	Electrical	Yellow 🟨
		Chemical	Work resulting from chemical reactions	Energy produced by aircraft batteries	*N.A.*	Green 🟩	Chemical		
		Electrical	Work resulting from motion of electrons	Energy transmitted by aircraft harnesses	*N.A.*	Yellow 🟨	Electrical		
		Hydraulic	Work performed by moving fluids	Energy used to actuate the landing gear	*N.A.*	Black ⬛	Hydraulic		
		Thermal	Work resulting from a themal system	Energy exchanged in the aircraft cooling system	*N.A.*	Red 🟥	Thermal		
		Pneumatic	Work resulting from the motion of gas	Energy used by the pnematic aircraft system	*N.A.*	Brown 🟫	Pneumatic		
		Mechanical	Work performed by a mechanical system	Energy generated by a mechanical connection to fix the seat on the cabin	*N.A.*	Purple 🟪	Mechanical	Mechanical	Purple 🟪
Signal	⇡	Control	Command sent to an apparatus to regulate it	Pilot that regulates air in the cabin	Sound	Orange 🟧	Control	Electrical	Yellow 🟨
					Tactile	Blue 🟦			
					Visual	Red 🟥			
		Feedback	Information about the state of the system	Indicator of the fuel level in the aircraft	Visual	Green 🟩	Feedback		
					Sound	Yellow 🟨			
					Tactile	Purple 🟪			

Module derivation

After the functional scheme is obtained, aircraft modules are derived. Module definition is a key task for the methodology development, and it must be coupled with demanding engineering specifications (e.g., the presence of redundant elements placed in different areas for safety reasons). The module heuristics (Stone et al. 2004) are used in this task due to their ability to consider engineering requirements when a module is developed. Module heuristics gives a consistent module breakdown in respect to the product functions even though necessitates engineering judgment to be defined. After the modules have been identified, the physical architecture of the product is developed by creating the physical arrangement (layout) of modules and system interfaces (derived in the previous phase).

3.2. Phase 2: architecture geometrical definition

The architecture geometrical definition phase consists of translating information of the product architecture into numerical data. This phase is characterized by three tasks: i) Geometrical product architecture, ii) *Conceptual* Bill of Materials (cBoM), and iii) *knowledge* Scoring Matrices (kSM).

Geometrical product architecture

The possibility to evaluate fit for assembly metrics of aircraft systems architecture requires converting conceptual features (from the functional/modular representations) into parameters that can be represented graphically and quantified. The data gathered in the previous phase (i.e., functional modules and schemes) is used to build a virtual representation of the product under investigation. The use of a virtual tool (e.g., CAD software) allows representing, with simple geometries, the information available at the conceptual design phase. The simplified Digital Mock-Up (sDMU) is a geometrical representation of the product architecture, which is characterized by 3D geometrical objects (i.e., boxes, cylinders), and describing how modules and interfaces are built and interconnected.

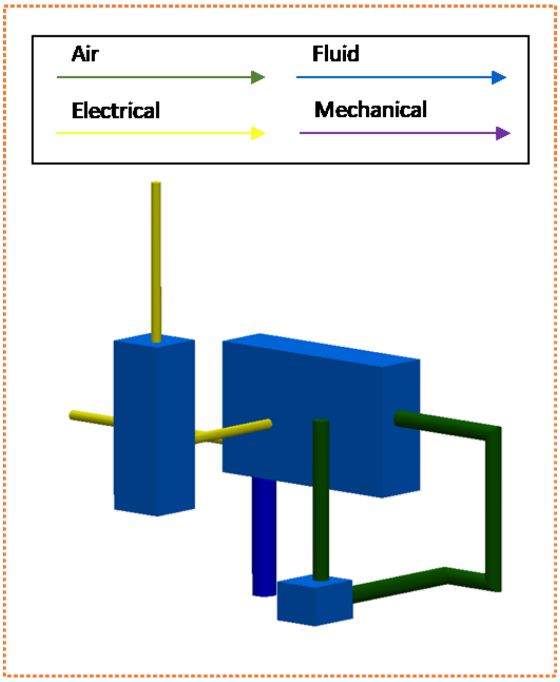


FIG. II - *SIMPLIFIED* DIGITAL MOCK-UP

The sDMU provides the minimum set of information required to complete the fit for assembly evaluation. In particular, geometrical and graphical information for modules (Table 2) and interfaces (Table 3)are represented by the use of the sDMU.

TABLE 2 - INFORMATION MODULES

	Information collected in sDMU (Modules)
Modules	Position (x, y, z) based on a given reference
	Shape (i.e., rectangular box, cylinder, other)
	Size (bounding box)
	Colour (i.e., blue)

TABLE 3 - INFORMATION INTERFACES

	Information collected in sDMU (Interfaces)
Interfaces	Position (x, y, z) based on a given reference
	Path (x, y, z) based on a given reference
	Overall length
	Shape (i.e., cylinder)
	Size (i.e., diameter)
	Colour (based on interface type)

Conceptual Bill of Materials

The conceptual Bill of Materials (cBoM) is a document that translates the functional and geometrical data, obtained from the previous phases, into numerical data aiming at assessing the fit for assembly score. The cBoM is organized in a table format, where a row represents an element (i.e., module or interface), and a column represents a feature (attribute) linked to the assembly complexity for the analysed element. The cBoM is designed by following a hierarchical structure, and the overall arrangement of information is based on levels (layers), domains, and attributes. The hierarchical structure of the cBoM allows to analyse the product architecture considering the overall product breakdown into sub-problems. Table 4 and Table 5 list, respectively, the information for interfaces and modules that must be included in the *fixed information* section according to the analysed element.

TABLE 4 – INTERFACE FIXED INFORMATION OF THE CBOM STRUCTURE

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 5 – Module Fixed Information of the cBoM structure

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

Other information can be added within the established information framework if necessary.

Levels (layers) definition in the cBoM structure

A level is a collection of data that represents the most important assembly aspects within a given context (or sub-problem). Several levels can be identified based on the overall product assembly and the specificity of the system of interest. The way to pass from one level to another is based on the definition of the *product invariant*. A product invariant is a design feature or a project constraint that does not change and cannot be modified in the system architecture. Data collected at a given level represents the key assembly features to consider for the problem representation. The definition of product invariant that connects two neighbouring levels also requires the description of the relationship that exists between two adjacent levels. For instance, if the space distribution (i.e., compartments) inside a product is fixed and cannot be modified, then the "space distribution" might be considered an invariant. Invariants enable the global analysis to be divided into sub-problems with a lower level of complexity (problem discretization).

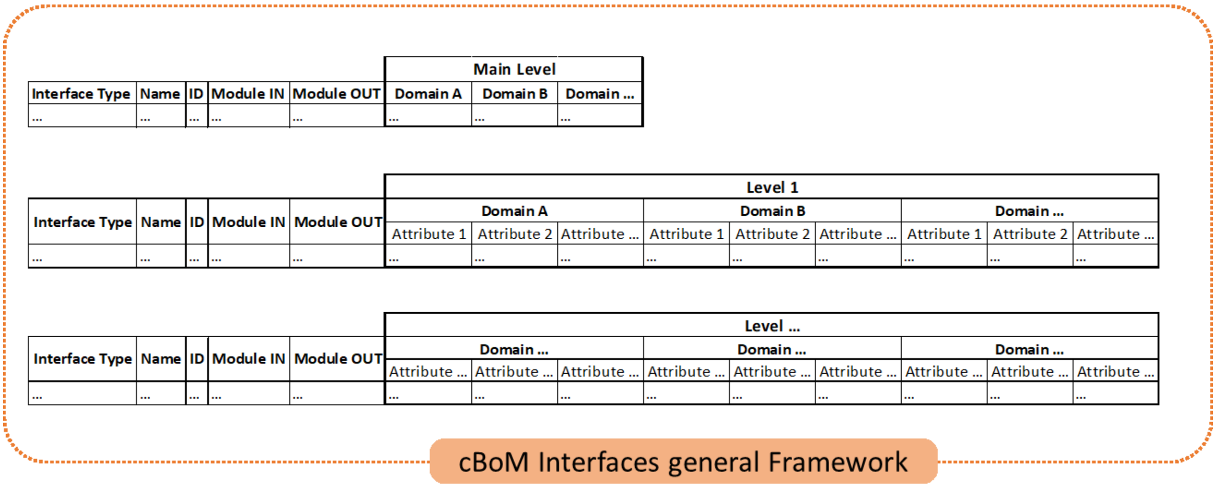
Attributes Class definition

An attribute (*a*) is defined as a key feature that impacts the assembly processes. The attribute *a* [1 to x; x N>0] is a parameter that characterize a specific aspect of the assembly process. To define an attribute, it is required to provide the following information: i) the attribute's name, (ii) the dimension (i.e., the unit measure or amount), and (iii) the level at which the attribute is available. Usually, design information is used to generate a list of important features representing assembly difficulties. Attributes might be of interest or not, according to the level in which they are placed.

Domain Class definition

A domain (*d*) is a cluster of one or more attributes (*a*) that belong to the same context (i.e., mechanical, ergonomic, operation, etc.). Thus, a domain represents a cluster of attributes that must be clearly understandable for every designer/engineer. Given a generic domain D [1 to t; t x; t N>0], the domain D is a vector with n attributes [n 1; n N>0]. Each domain does not have the same number of attributes.

The general architectures of the cBoM for interfaces (Fig. III) and modules (Figure IV) with fixed information are presented below.



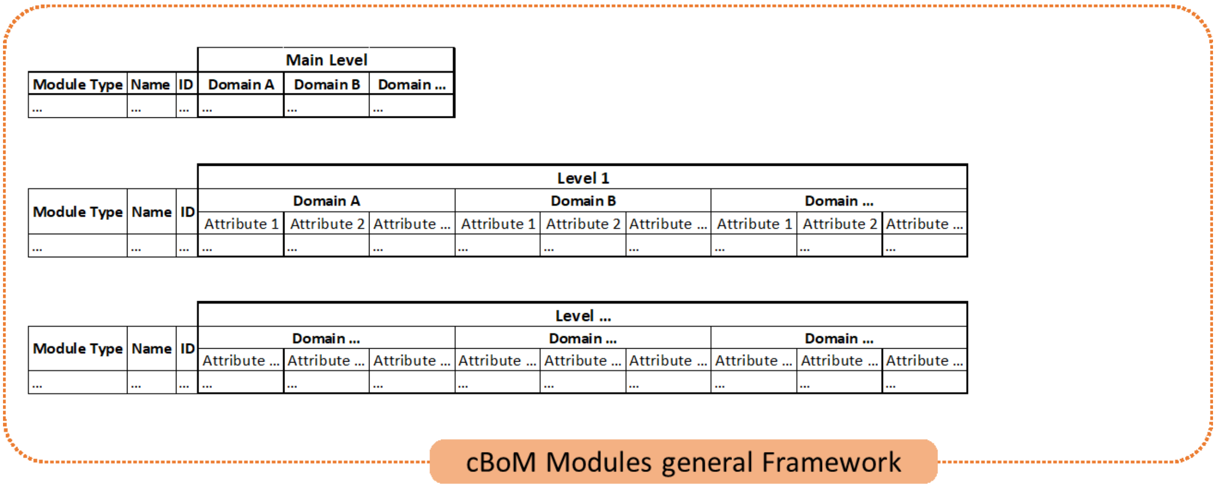
					Main Level		
Interface Type	**Name**	**ID**	**Module IN**	**Module OUT**	**Domain A**	**Domain B**	**Domain ...**
...	...	...	...	...	...	...	...

					Level 1								
					Domain A			**Domain B**			**Domain ...**		
Interface Type	**Name**	**ID**	**Module IN**	**Module OUT**	Attribute 1	Attribute 2	Attribute ...	Attribute 1	Attribute 2	Attribute ...	Attribute 1	Attribute 2	Attribute ...
...	...	...	...	...	...	...	...	...	...	...	...	...	...

					Level ...								
					Domain ...			**Domain ...**			**Domain ...**		
Interface Type	**Name**	**ID**	**Module IN**	**Module OUT**	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...
...	...	...	...	...	...	...	...	...	...	...	...	...	...

cBoM Interfaces general Framework

Fig. III - cBoM general framework for interfaces



			Main Level		
Module Type	**Name**	**ID**	**Domain A**	**Domain B**	**Domain ...**
...	...	...	...	...	...

			Level 1								
			Domain A			**Domain B**			**Domain ...**		
Module Type	**Name**	**ID**	Attribute 1	Attribute 2	Attribute ...	Attribute 1	Attribute 2	Attribute ...	Attribute 1	Attribute 2	Attribute ...
...	...	...	...	...	...	...	...	...	...	...	...

			Level ...								
			Domain ...			**Domain ...**			**Domain ...**		
Module Type	**Name**	**ID**	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...	Attribute ...
...	...	...	...	...	...	...	...	...	...	...	...

cBoM Modules general Framework

Figure IV - cBoM General Framework for Modules

Knowledge Scoring Matrices (knowledge-based approach)

Knowledge Engineering is a research field dealing with the formalization of knowledge, which is a key challenge in the industrial context (Staab et al. 2001; Ahmed 2005; Reed et al. 2011). Engineering knowledge is usually unstructured, spread in several forms (e.g., technical drawings, spreadsheets, etc.), and different departments. The approach used to collect and formalize engineering knowledge focuses on two aspects: i) knowledge collection, and ii) ontology definition (Guarino 1995). In the current method, relevant information in the context of aircraft assembly is obtained following a concurrent knowledge-based approach (Favi et al. 2020). Three principles are defining the structure and the syntax (i.e., ontology) of knowledge: i) role-limiting, ii) knowledge typing, and iii) reusability (Musen and Schreiber 1995). The means used to translate engineering knowledge into numerical data within the CDfA methodology is the *knowledge* Scoring Matrix (kSM). The kSM is a table with a defined number of rows, each representing a different score (i.e., from 1 to 5). The kSM is needed to transform the collected data into dimensionless values and to perform mathematical calculations. The kSM consists of three parts: i) the attribute name, ii) the numeric range or string, and iii) the score for each attribute in the cBoM. It is worth mentioning that each kSM is based on the same range of scores (i.e., a score of integer numbers ranging from 1 to 5). 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 is 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.

3.3. Phase 3: conceptual Design for Assembly assessment

The Conceptual Design for Assembly assessment is the computational phase of the method and concerns the assessment of assemblability score based on the previous defined cBoM documents and kSMs. This phase is characterized by two tasks: (i) normalization, and (ii) architectural assessment.

Normalization

The normalization task is necessary to switch from heterogeneous data (i.e., text, integer, etc.) collected for each attribute into dimensionless values (i.e., dimensionless scores). For this task, the previously defined kSMs are used allowing to convert information presented in numerical or string form into a score. For example, when the characteristic "length" is implemented as an attribute (i.e., 3,5 [m]), the normalization process is able to convert the value into a dimensionless score (i.e., 1) using the specific kSM defined for the length attribute. 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:

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

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:

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

with $a_1, \dots, a_0$ represent qualitative attributes that require normalization (i.e., the o_i elements in the matrix (I)), while $\mathrm{a}_{\mathrm{Q+1}}, \dots, \mathrm{a}_{\mathrm{Q+P}}$ 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 $\underline{\underline{A}}(e^*,d)$ with $e^* \in [1, \mathrm{E}^*]$ is defined as:

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

(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., $e_{s1}, \dots, e_{sm}$).

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

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

which is composed of a_{k} elements deriving from quantitative kSMs (I), and $\mathrm{a}_{\mathrm{k+w}}$ 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 $\underline{\underline{A}}_{\mathrm{norm}}(e^*,d)$

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

(IV)

Once the normalization process is completed for all attributes, all data in the cBoM is normalized and dimensionless, enabling to proceed further with the architecture assessment.

Architectural assessment

The architectural assessment task consists of several mathematical steps which allows to obtain, 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 $\mathrm{H}(\cdot) = H(\mathrm{A}_{\mathrm{norm}},\mathrm{l},\mathrm{t},\mathrm{d})$ as $\mathrm{H}: \mathbb{R}^{(Ax(P+Q))} \to \mathbb{R}^{(Ax1)}$ with A = dim(A,1) which transforms the normalized attributes matrix (IV) into the domain vector $\underline{d}(e^{*}, d)$:

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

(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 $\mathrm{G}: \mathbb{R}^{(Ax1)} \to \mathbb{R}$ 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)*:

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

(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:

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

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 $\underline{C}$:

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

where:

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

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

- $\mathrm{m_i}$ 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 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 for 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. 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.

3.4. Phase 4: product architecture re-design

The redesign of the product architecture is composed by only one task, which is the modification of elements identified as critical. However, this phase is out of the scope for this research work since aims at providing a method to assess the installation performance of aircraft systems architectures.

Modules and interface modification

Critical elements can be identified by setting a threshold on the final score. Threshold may be set according to the goal of the analysis. For instance, assuming the goal is "*to identify and modify the 20% of the most complex elements to install*", then the threshold should be set to highlight these elements. Once threshold is set and elements identified, it is possible to proceed with the elements' modification. To modify critical elements, it is required to make use of the cBoM framework. By investigating the hierarchical structure, it is possible to spot domains and attributes which lead to a higher element assembly score. The type of modification that can be implemented is identifiable making use of the kSM. In fact, once the hierarchical structure is investigated and attributes with a critical score are identified, it is possible to analyse their kSM to understand what design action will lower their score (i.e., improve their assembly complexity). Once modifications are implemented, the CDfA analysis should be performed again, to check if the score of critical elements has been lowered.

4. Case study

The CDfA methodology previously presented has been applied to assess the fit for assembly performances of the nose fuselage (Fig. V), which is one the most challenging system of a civil aircraft. Inside the nose fuselage several elements, such as modules and interfaces, are present and they interact with each other. The nose fuselage architecture is constrained by several aspects, for instance the need to install elements in a confined area (i.e., compartments), the impossibility to change the structural framework (i.e., aircraft skeleton), and the need to have elements redundancy for safety reasons.

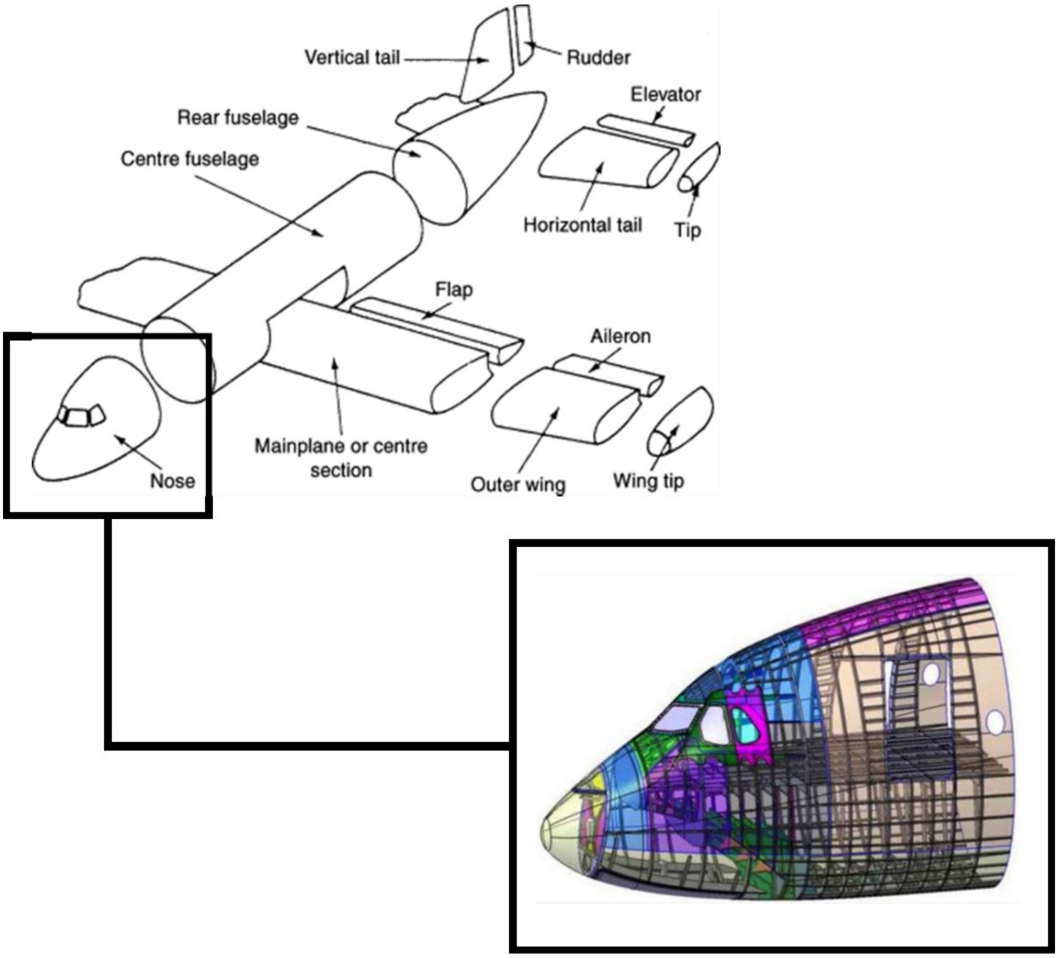


FIG. V - AIRCRAFT NOSE FUSELAGE

The goal of this exercise is to numerically assess the nose fuselage architecture, identifying critical elements (both modules and interfaces) from an assemblability/installation point of view. The ranking of elements is used to drive the re-design phase by highlighting hotspots and offering rationales for assembly activities. This information will then be exploited to enhance the design; however, this step is beyond the scope of this research work. In the following sections, each step of the CDfA methodology is described and applied to the nose-fuselage system.

4.1. Phase 1: nose fuselage functional decomposition

Following the analysis presented in Fig. I, using the functional derivation the nose fuselage functional decomposition is obtained. The functional analysis is used to determine all the functions and fluxes in the nose fuselage. Only electrical and air interfaces were examined in this study to limit the scope of work of the analysis. Indeed, electrical and air interfaces are considered the most challenging interfaces in the overall assembly process. Once the functional scheme was acquired, module heuristics (Stone et al. 2004) were used to derive modules. Fig. VI shows an extract of the identified modules starting from the functional scheme. Representation is based on the colour map and taxonomy provided in Table 1.

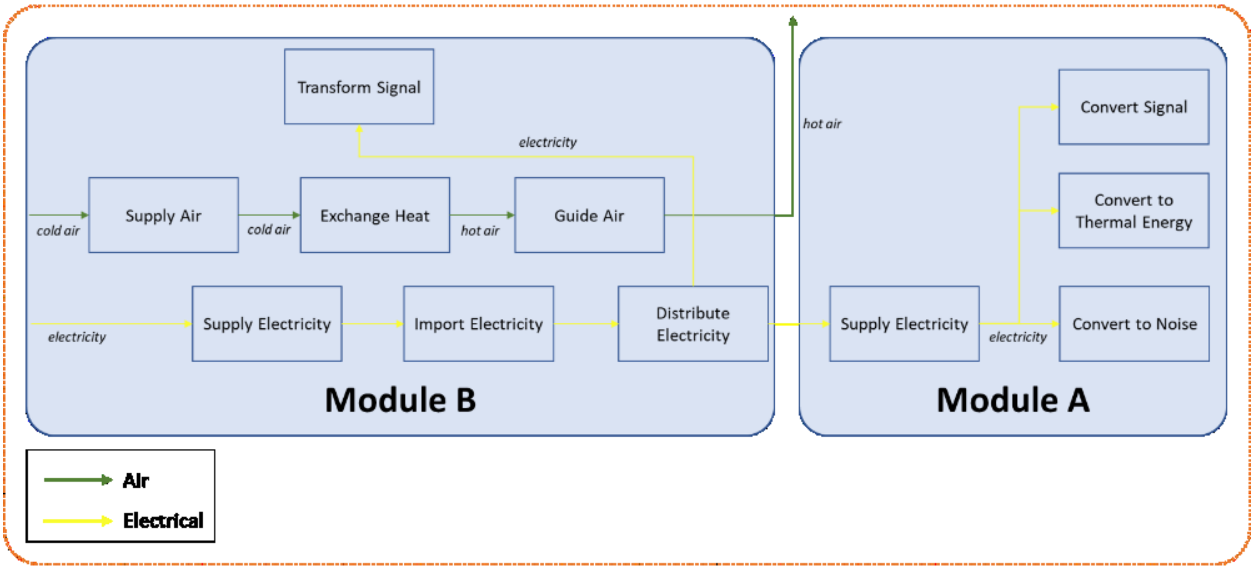


FIG. VI – EXTRACT OF THE APPLICATION OF THE MODULE HEURISTICS TO FUNCTIONAL ANALYSIS

4.2. Phase 2: nose fuselage architecture geometrical definition

The sDMU is created to graphically represent the nose fuselage architecture based on the information derived from the previous step. The module arrangement and the spatial layout were defined with the help of existing nose fuselage information: (i) the modules' bounding box, (ii) the length of interfaces (i.e., connections) among modules, (iii) the modules and connections' location, and (iv) the connections diameters. Modules (blue colour) were represented by rectangular boxes, whereas interfaces were represented by cylinders (yellow and green colours respectively for electrical and air interfaces) as presented in Fig. VII.

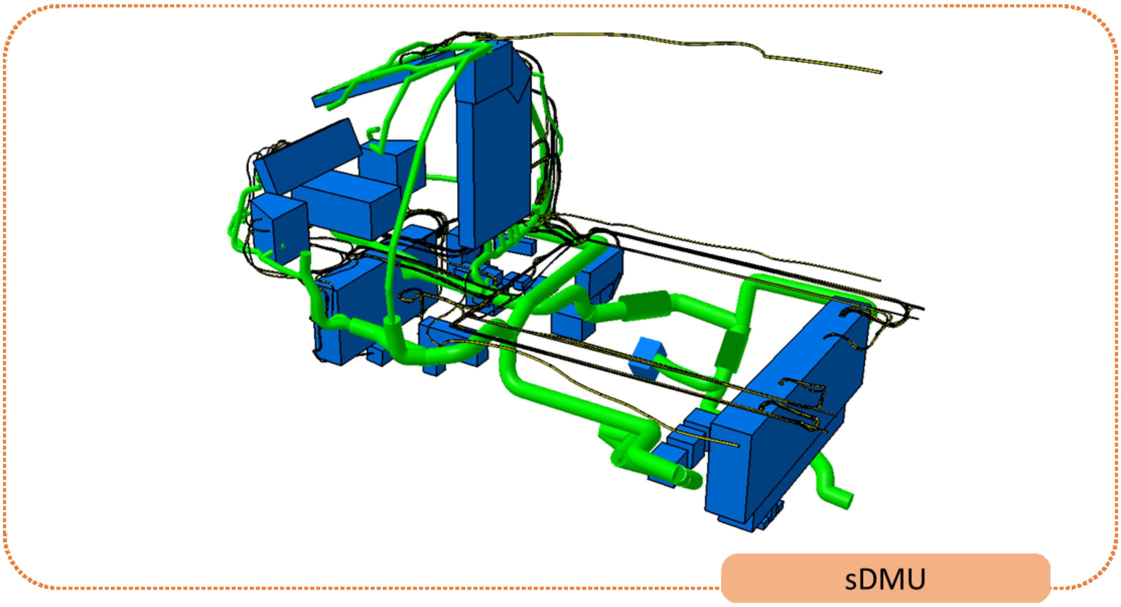
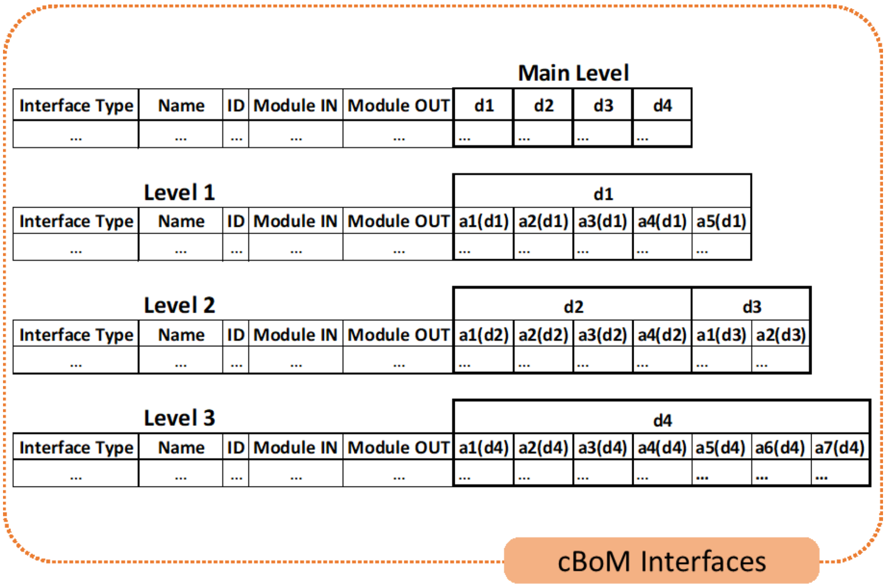


FIG. VII - SDMU OF THE NOSE FUSELAGE

The assessment of both, modules and interfaces, was performed considering two cBoMs. The cBoM used to assess nose fuselage interfaces is composed of three levels of data: Level 1, Level 2, and Level 3. Only one domain, named "Interface Domain" is present in Level 1, and it contains attributes that refer to the geometrical features of the interfaces (i.e., length, number of connections, etc.). The invariant *working area* was used to move from Level 1 to Level 2. Indeed, working areas are pre-defined compartments within the nose fuselage that cannot be changed due to structural constraints (e.g., design of beams, skin, and floor). The “Ergonomic Domain” and the “Assembly Domain” are the two domains present in Level 2. The former has four attributes referring to ergonomic elements of the installation procedure (e.g., space of the working areas, access, etc.), while the latter has two attributes that indicate the installation process's complexity (e.g., tool used and process). The invariant *connection* was used to move from the Level 2 to the Level 3. Indeed, a connection represents the physical element associated to an interface and cannot be changed in the final product design since it is defined starting from the functional analysis. At Level 3, only one domain called "Component Domain" is present and it contains features related to the physical elements (e.g., shape, weight, number of bends, etc.). The cBoM framework for the nose fuselage created to assess interfaces is presented in Fig. VIII.



Main Level

Interface Type	Name	ID	Module IN	Module OUT	d1	d2	d3	d4
...	...	...	...	...	...	...	...	...

Level 1

					d1				
Interface Type	Name	ID	Module IN	Module OUT	a1(d1)	a2(d1)	a3(d1)	a4(d1)	a5(d1)
...	...	...	...	...	...	...	...	...	...

Level 2

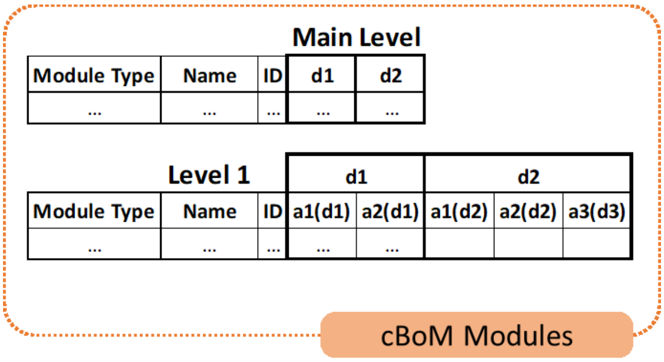
					d2				d3	
Interface Type	Name	ID	Module IN	Module OUT	a1(d2)	a2(d2)	a3(d2)	a4(d2)	a1(d3)	a2(d3)
...	...	...	...	...	...	...	...	...	...	...

Level 3

					d4						
Interface Type	Name	ID	Module IN	Module OUT	a1(d4)	a2(d4)	a3(d4)	a4(d4)	a5(d4)	a6(d4)	a7(d4)
...	...	...	...	...	...	...	...	...	...	...	...

Fig. VIII - cBoM framework for interfaces assessment

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 Fig. IX.



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(d3)
...	...	...	...	...			

Fig. IX - cBoM framework for modules assessment

The overall list of attributes defined for each cBoM framework is presented in Table 6 (assessment of interfaces and modules).

TABLE 6 – 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)	Overall air interface length
			Branches	a2(d1)	Number of times air interface branches out
			Total Length of Harness	a3(d1)	Overall electrical interface length
			Number of Connections	a4(d1)	Number of connections electrical interface collects
			Number of Straight Nodes	a5(d1)	Number of times electrical interface branches out
	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
			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
Module assessment	Mechanical Domain	d1	Number and Position of Mechanical Interfaces	a1(d1)	The overall number of module anchors' points and their 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 required to perform the assembly
			Weight	a2(d2)	Weight of the module
			Clearance	a3(d2)	Space available around the module to perform assembly operations

The creation of the cBoM framework was supported by sensitivity analysis (SA) methods. SA allows to understand 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). The method consists of changing the value of one parameter, keeping others fixed, to understand how each parameter affects the overall result. For the sake of brevity, only an extract of the SA performed on the modules assessment framework (cBoM) is shown in Fig. X. Results exhibit that 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). 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).

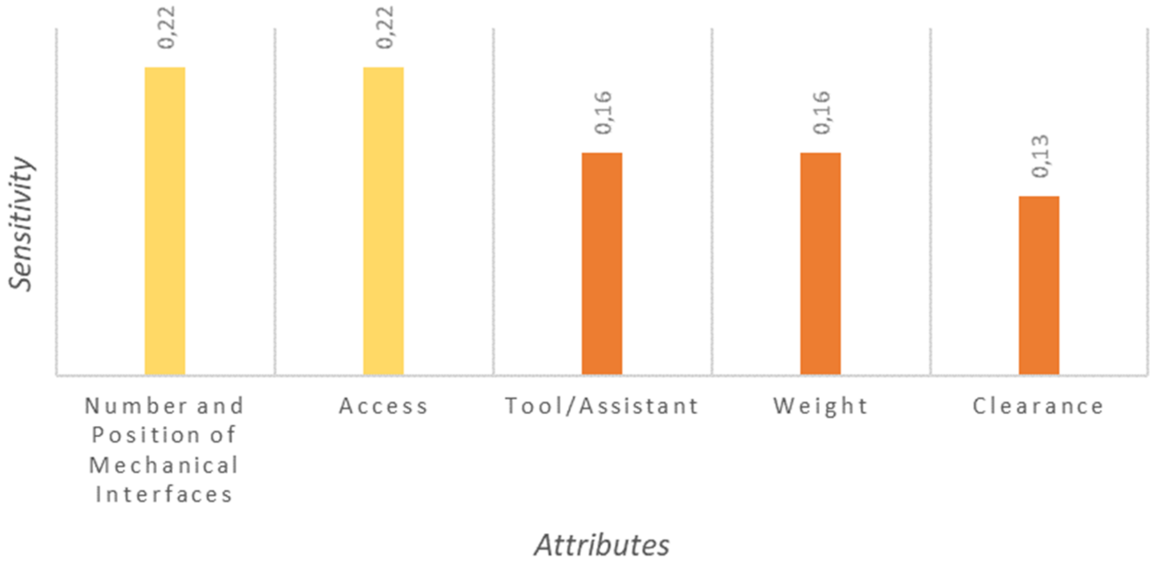
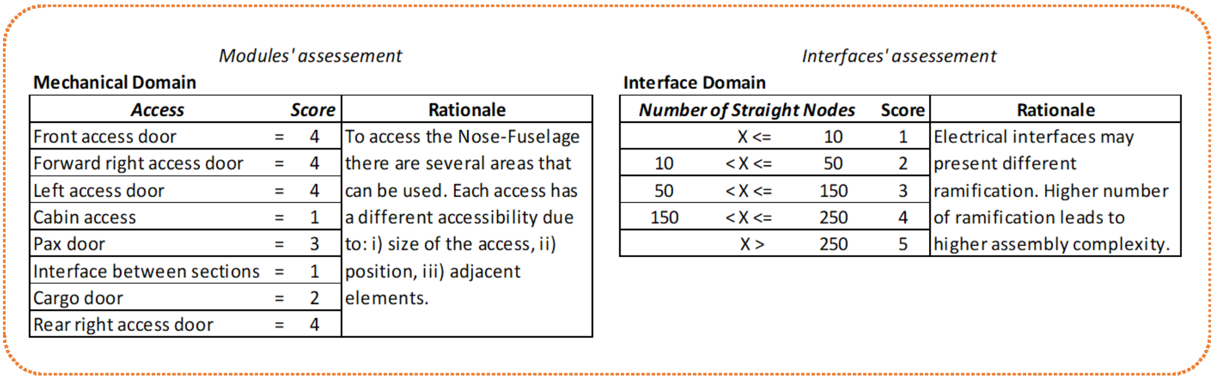


FIG. X - SENSITIVITY ANALYSIS FOR ATTRIBUTES INSIDE A DOMAIN

After the definition of the cBoM structure, the kSM for each attribute is defined following the ontology proposed in the methodology section. To collect and formalize the required engineering knowledge, several meetings with practitioners of aircraft development (from both design, and manufacturing/assembly departments) were organized. During the initial meeting (first meeting), the methodology was presented to clearly express the scope of work. Then, a *web* meeting (second meeting) was arranged, and a survey was submitted to the manufacturing department in order to collect expertise from assembly operations and related tasks. Afterwards, survey's results were analysed to obtain the first draft of the kSMs. Another *web* meeting (third meeting) was organized with the Product Architecture & Design department to show the kSMs obtained. In this case, the goal was to check the consistency between the obtained results and the expertise of designers, manufacturing, and system engineers. Few modifications were suggested and implemented for the final meeting, where the kSMs were finalized. An extract of the kSMs obtained is presented in Fig. XI.



Modules' assessement

Mechanical Domain

Access		*Score*	Rationale
Front access door	=	4	To access the Nose-Fuselage there are several areas that can be used. Each access has a different accessibility due to: i) size of the access, ii) position, iii) adjacent elements.
Forward right access door	=	4	
Left access door	=	4	
Cabin access	=	1	
Pax door	=	3	
Interface between sections	=	1	
Cargo door	=	2	
Rear right access door	=	4	

Interfaces' assessement

Interface Domain

Number of Straight Nodes			Score	Rationale
	X <=	10	1	Electrical interfaces may present different ramification. Higher number of ramification leads to higher assembly complexity.
10	< X <=	50	2	
50	< X <=	150	3	
150	< X <=	250	4	
	X >	250	5	

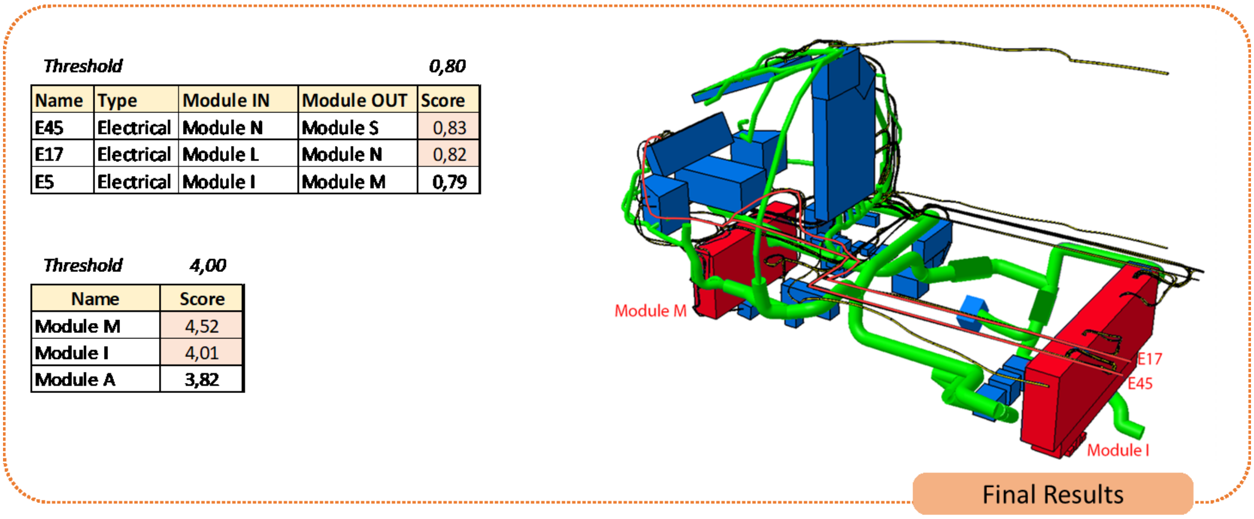
FIG. XI – EXAMPLE OF TWO KSMS (ON THE LEFT FOR THE MODULES' ASSESSMENT AND ON THE RIGHT FOR THE INTERFACES' ASSESSMENT)

4.3. Phase 3: nose fuselage *conceptual* Design for Assembly assessment

This phase allows to obtain a final global score for each element. In this case, the mathematical operator used to collect the attributes scores for each domain is the Root Mean Square (RMS). The choice of the RMS was done based on the type of data collected inside the cBoM which is coming from different sources (heterogeneous parameters with a possible source of error). The overall domain score is more conservative (i.e., higher) than the one obtained, for example, with the adoption of the Mean operator. On the other hand, to gather level scores (i.e., from Level 3 to Level 2, from Level 2 to Level 1, and from Level 1 to Main Level) the Mean operator was preferred. In fact, domain scores collected with the RMS are already normalized, and any possible source of error was already being accounted.

Once the global score for each interface inside the nose fuselage was obtained and the Main Level was reached, the TOPSIS method was applied to collect all scores into a single one. The choice of TOPSIS lies in the possibility to apply weights on each domain to tune the overall assessment. In fact, during the model definition some attributes and domains might be considered less or more critical in relation to the product assembly complexity. This may lead to some shortfalls that can be recovered afterwards making use of weights. Since the obtained results are in line with engineering expertise and perception of the problem for the nose section, no weights were added in this case.

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 score obtained is shown in Fig. XII.



Threshold *0,80*

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

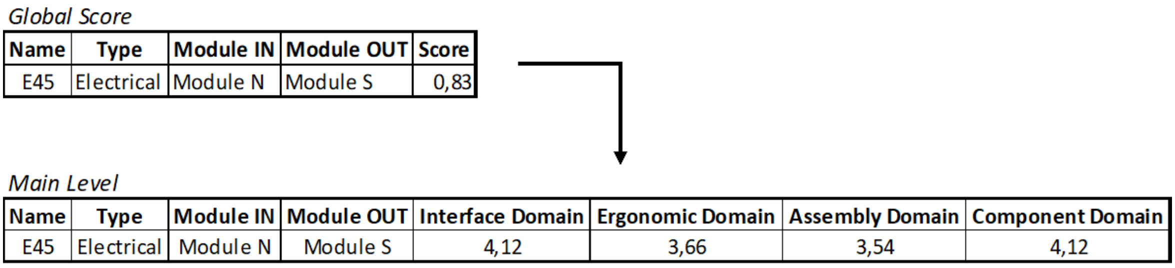
Threshold *4,00*

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

FIG. XII - FINAL SCORES FOR MODULES AND INTERFACES (INCLUDING THRESHOLDS FOR REDESIGN PHASE)

4.4. Phase 4: nose fuselage architecture re-design

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 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). Red colour highlights the interfaces and modules with the highest scores (Fig. XII). Results analysis is only the initial task of the re-design phase since the re-design phase falls outside the scope of this work. In the analysis of the interfaces' assessment, it is interesting to notice that electrical interfaces are the most critical for the installation process. The hierarchical structure of the CDfA methodology enables to spot criticalities for each interface. The interface E45 has the highest score among all interfaces as reported in Fig. XII. 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 (Fig. XIII).



Global Score

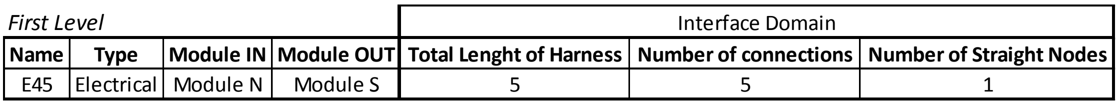
Name	Type	Module IN	Module OUT	Score
E45	Electrical	Module N	Module S	0,83

Main Level

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

FIG. XIII - E45 FROM GLOBAL SCORE TO MAIN LEVEL DATA

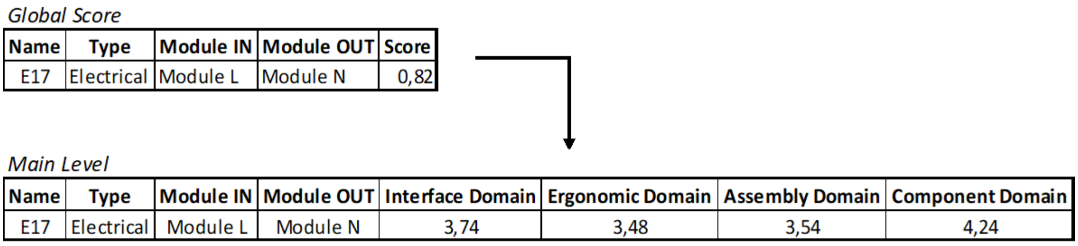
Among all, the highest score is associated to two domains: Interface Domain (4,12) and Component Domain (4,12). Analysing results for the Interface Domain, the attributes *Total Length of Harness* and *Number of Connections* present a score of 5 (Fig. XIV). Indeed, different modifications might be implemented 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 which 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.



First Level				Interface Domain		
Name	**Type**	**Module IN**	**Module OUT**	**Total Lenght of Harness**	**Number of connections**	**Number of Straight Nodes**
E45	Electrical	Module N	Module S	5	5	1

FIG. XIV - 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.



Global Score

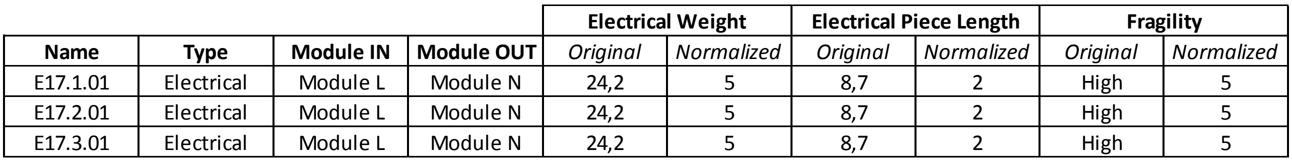
Name	**Type**	**Module IN**	**Module OUT**	**Score**
E17	Electrical	Module L	Module N	0,82

Main Level

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

FIGURE XV – RESULT ANALYSIS OF E17 INTERFACE (FROM GLOBAL SCORE TO MAIN LEVEL DATA)

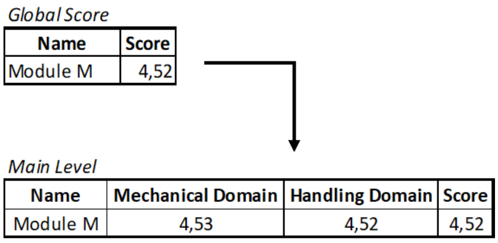
Electrical Weight and *Fragility* attributes, which belong to the Component Domain, have a score of 5. All electrical connections have the same score. In the current design, electrical connections are not split into parts like air interfaces; in fact, the same data is repeated for each interface sub-section (Fig. XVI). 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.



				Electrical Weight		**Electrical Piece Length**		**Fragility**	
Name	**Type**	**Module IN**	**Module OUT**	*Original*	*Normalized*	*Original*	*Normalized*	*Original*	*Normalized*
E17.1.01	Electrical	Module L	Module N	24,2	5	8,7	2	High	5
E17.2.01	Electrical	Module L	Module N	24,2	5	8,7	2	High	5
E17.3.01	Electrical	Module L	Module N	24,2	5	8,7	2	High	5

FIG. XVI – 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 the Module_M, followed by Module_I. Starting with the 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 (Fig. XVII).



Global Score

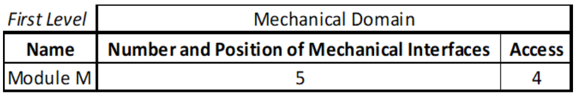
Name	Score
Module M	4,52

Main Level

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

FIG. XVII - RESULT ANALYSIS OF MODULE_M MODULE (FROM GLOBAL SCORE TO MAIN LEVEL DATA)

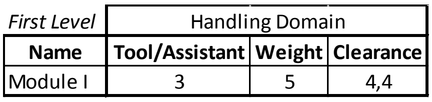
Analysing the Mechanical domain (Fig. XVIII), it is possible to see that the most impacting attribute is *Number and Position of Mechanical Interfaces*, followed by *Access*. To change the number and position of mechanical interfaces, it is necessary to consider structural design requirements. By changing 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	4

FIG. XVIII – 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 (Fig. XIX) 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 the Module_I into two or more sub-modules. In this way, each sub-module 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	5	4,4

FIG. XIX – 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. The assessment phase can be repeated in an iterative way after a modification is implemented to check the benefits of the proposed re-design action and to spot additional issues.

5. Conclusion

Design for assembly methods allow to reduce the product assembly complexity, however if applied during the late design phases their advantages might be limited. The Conceptual Design for Assembly(CDfA) methodology provided in this study is used to evaluate fit for assembly performance in the early design phases of aircraft product development. The approach is grounded on the functional/modular derivation and uses high-level design information which is properly stored within a given framework characterized by levels, attributes, and domains. For each module and interface analysed in this framework, a single score is calculated using mathematical operations. The highest score indicates the most difficult element to assemble. The CDfA methodology was used to evaluate the fit for assembly performance of modules and interfaces (i.e., electrical and air) of the nose fuselage of a civil aircraft. The method shows important novelties in the field of aircraft design and development targeting assembly/installation aspects: i) it collects and analyse data by moving from tacit implicit knowledge to explicit knowledge, ii) it develops a mathematical model and a framework capable of considering the global context of an aircraft assembly by using sub-problems with limited complexity (problem discretization), and iii) it provides rationales to engineers and designers to understand aircraft assemblability complexities.

The CDfA applied research work can be considered the initial step toward a complete design framework for aircraft product development, closing the gap between design and manufacturing departments. A few important aspects required to be included within this framework are: i) a fast analysis of the design guidelines that might be implemented to reduce the score, allowing for a reduction of the solution space, and ii) the possibility to consider the assembly sequence. In particular, the last aspect is critical and challenging for the analysis of the assembly/installation phase, since the product evolution during the installation process can change the score associated with elements' attributes. For instance, the attribute *Clearance* for a module might vary during the installation process from *big* to *small* due to the number of elements already assembled in the surrounding area.

Future work will be focused on overcoming these two shortfalls. Firstly, the assembly evolution will be considered in the approach to better assess installation complexities including the possibility to retrieve optimized assembly sequences. Secondly, the CDfA methodology will be coupled with a novel approach able to identify the most impacting and applicable design guidelines. This approach, which is called “loop-back approach” aims at the reduction of the design solution space allowing to optimize global product architecture scoring. Finally, the methodology will be implemented in a software solution coupled with commercial CAD systems to make more user-friendly the data acquisition (automated data entry from CAD systems) as well as the score assessment.

References

Ahmed S (2005) Encouraging reuse of design knowledge: A method to index knowledge. Design Studies 26:565–592. https://doi.org/10.1016/j.destud.2005.02.005

AlGeddawy T, ElMaraghy H (2013) Reactive design methodology for product family platforms, modularity and parts integration. CIRP Journal of Manufacturing Science and Technology 6:34–43. https://doi.org/10.1016/j.cirpj.2012.08.001

Bonvoisin J, Halstenberg F, Buchert T, Stark R (2016) A systematic literature review on modular product design. Journal of Engineering Design 27:488–514. https://doi.org/10.1080/09544828.2016.1166482

Boothroyd G (Geoffrey), Dewhurst Peter, Knight WA (Winston A (2011) Product design for manufacture and assembly. CRC Press

Bouissiere, F., Cuiller, C., Dereux, P. E., Malchair, C., Favi, C., & Formentini, G. (2019, July). Conceptual Design for Assembly in aerospace industry: a method to assess manufacturing and assembly aspects of product architectures. In Proceedings of the Design Society: International Conference on Engineering Design (Vol. 1, No. 1, pp. 2961-2970). Cambridge University Press.

Bullen GN (1999) Assembly automation and implementation issues. In: SAE Technical Papers. SAE International

Butterfield J, Crosby S, Curran R, et al (2007) Optimization of aircraft fuselage assembly process using digital manufacturing. Journal of Computing and Information Science in Engineering 7:269–275. https://doi.org/10.1115/1.2753879

Coma O, Mascle C, Balazinski M (2004) Application of a fuzzy decision support system in a Design for Assembly methodology. International Journal of Computer Integrated Manufacturing 17:83–94. https://doi.org/10.1080/0951192031000078185

El-Nounu AR, Popov A, Ratchev S (2018) Redesign methodology for mechanical assembly. Research in Engineering Design 29:107–122. https://doi.org/10.1007/s00163-017-0255-6

Erens F, Verhulst K (1997) Architectures for product families. Computers in Industry 33:165–178. https://doi.org/10.1016/s0166-3615(97)00022-5

Favi, C., Germani, M., & Mandolini, M. (2016). A multi-objective design approach to include material, manufacturing and assembly costs in the early design phase. *Procedia CIRP*, *52*, 251-256.

Favi C, Formentini G, Bouissiere F, et al (2020) Design for Assembly in the Conceptual Development of Aircraft Systems. In: Lecture Notes in Mechanical Engineering. Springer, pp 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 6:155–169. https://doi.org/10.1007/s12008-012-0147-y

Fricke E, Schulz AP (2005) Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. Systems Engineering 8:no. https://doi.org/10.1002/sys.20039

Frigant V, Talbot D (2005) Technological determinism and modularity: Lessons from a comparison between aircraft and auto industries in Europe. Industry and Innovation 12:337–355. https://doi.org/10.1080/13662710500195934

Formentini, G., Cuiller, C., Dereux, P. E., Bouissiere, F., & Favi, C. (2021). Impact assessment of design guidelines in the conceptual development of aircraft product architectures. Procedia CIRP, 100, 223-228.

Gómez A, Ríos J, Mas F, Vizán A (2016) Method and software application to assist in the conceptual design of aircraft final assembly lines. Journal of Manufacturing Systems 40:37–53. https://doi.org/10.1016/j.jmsy.2016.04.002

Guarino N (1995) Formal ontology, conceptual analysis and knowledge representation. International Journal of Human - Computer Studies 43:625–640. https://doi.org/10.1006/ijhc.1995.1066

Hermansson T, Bohlin R, Carlson JS, Söderberg R (2013) Automatic assembly path planning for wiring harness installations. Journal of Manufacturing Systems 32:417–422. https://doi.org/10.1016/j.jmsy.2013.04.006

Hirtz J, Stone RB, McAdams DA, et al (2002) A functional basis for engineering design: Reconciling and evolving previous efforts. Research in Engineering Design - Theory, Applications, and Concurrent Engineering 13:65–82. https://doi.org/10.1007/s00163-001-0008-3

Holt R, Barnes C (2010) Towards an integrated approach to “design for X”: An agenda for decision-based DFX research. Research in Engineering Design 21:123–136. https://doi.org/10.1007/s00163-009-0081-6

Hölttä K, Suh ES, de Weck O (2005) TRADEOFF BETWEEN MODULARITY AND PERFORMANCE FOR ENGINEERED SYSTEMS AND PRODUCTS

Huang GQ, Lee SW, Mak KL (1999) Web-based product and process data modelling in concurrent “design for X.” Robotics and Computer-Integrated Manufacturing 15:53–63. https://doi.org/10.1016/S0736-5845(98)00028-3

Jiao J, Simpson TW, Siddique Z (2007) Product family design and platform-based product development: A state-of-the-art review. Journal of Intelligent Manufacturing 18:5–29. https://doi.org/10.1007/s10845-007-0003-2

Jonas H, Gumpinger T, Krause D (2009) FLEXGALLEY – INNOVATIVE APPROACH FOR A MODULAR DESIGN OF AN AIRCRAFT-GALLEY

Jun HB, Park JY, Suh HW (2006) Lead time estimation method for complex product development process. Concurrent Engineering Research and Applications 14:313–328. https://doi.org/10.1177/1063293X06073302

Jung S, Simpson TW (2017) New modularity indices for modularity assessment and clustering of product architecture. Journal of Engineering Design 28:1–22. https://doi.org/10.1080/09544828.2016.1252835

Kuo TC, Huang SH, Zhang HC (2001) Design for manufacture and design for “X”: Concepts, applications, and perspectives. Computers and Industrial Engineering 41:241–260. https://doi.org/10.1016/S0360-8352(01)00045-6

Lockett H;, Fletcher S, Luquet N (2014a) 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 7:284–291. https://doi.org/10.4271/2014-01-2266

Lyu J, Chang LY (2010) A reference model for collaborative design in mould industry. Production Planning and Control 21:428–436. https://doi.org/10.1080/09537280903363217

Malmqvist J (1997) Improved function-means trees by inclusion of design history information. Journal of Engineering Design 8:107–117

Mas F, Ríos J, Gómez A, Hernández JC (2016) Knowledge-based application to define aircraft final assembly lines at the industrialisation conceptual design phase. International Journal of Computer Integrated Manufacturing 29:677–691. https://doi.org/10.1080/0951192X.2015.1068453

Mas F, Ríos J, Menéndez JL, Gómez A (2013) A process-oriented approach to modeling the conceptual design of aircraft assembly lines. International Journal of Advanced Manufacturing Technology 67:771–784. https://doi.org/10.1007/s00170-012-4521-5

Miller J, Guerrero J, Goldstein D, Robinson T (2002) Spaceframe: Modular Spacecraft Building Blocks for Plug And Play Spacecraft. Small Satellite Conference

Monnoyer MC, Zuliani JM (2007) The decentralisation of airbus pruduction and services. Service Industries Journal 27:251–262. https://doi.org/10.1080/02642060701207056

Musen MA, Schreiber AT (1995) Architectures for intelligent systems based on reusable components. Artificial Intelligence In Medicine 7:189–199

Pahl G, Beitz W, Feldhusen J, Grote KH (2007) Engineering design: A systematic approach. Springer London

Paik JK, Thayamballi AK, Paik JK, Thayamballi AK (2009) Design Principles, Criteria, and Regulations. In: Ship-Shaped Offshore Installations. Cambridge University Press, pp 55–81

Pokojski J, Oleksinski K, Pruszynski J (2018) Knowledge Based Processes in the Context of Conceptual Design. Journal of Industrial Information Integration

Reed N, Scanlan J, Wills G, Halliday ST (2011) Knowledge use in an advanced manufacturing environment. Design Studies 32:292–312. https://doi.org/10.1016/j.destud.2010.10.002

Saltelli, A., Ratto, M., Tarantola, S., Campolongo, F., & Commission, E. (2006). Sensitivity analysis practices: Strategies for model-based inference. *Reliability Engineering & System Safety*, *91*(10-11), 1109-1125.

Shimomura Y, Yoshioka M, Takeda H, et al (1998) Representation of design object based on the functional evolution process model. Journal of Mechanical Design, Transactions of the ASME 120:221–229. https://doi.org/10.1115/1.2826962

Siddiqi A, de Weck OL (2008) Modeling methods and conceptual design principles for reconfigurable systems. Journal of Mechanical Design, Transactions of the ASME 130:1011021–10110215. https://doi.org/10.1115/1.2965598

Staab S, Studer R, Schnurr HP, Sure Y (2001) Knowledge processes and ontologies. IEEE Intelligent Systems and Their Applications 16:26–34. https://doi.org/10.1109/5254.912382

Stief P, Dantan JY, Etienne A, et al (2020) Product design improvement by a new similarity-index-based approach in the context of reconfigurable assembly processes. Journal of Engineering Design 31:349–377. https://doi.org/10.1080/09544828.2020.1748181

Stone RB, McAdams DA, Kayyalethekkel VJ (2004) A product architecture-based conceptual DFA technique. Design Studies 25:301–325. https://doi.org/10.1016/j.destud.2003.09.001

Ulrich K (1995) The role of product architecture in the manufacturing firm. Research Policy 24:419–440. https://doi.org/10.1016/0048-7333(94)00775-3