DOI: ADD DOINUMBER HERE

Design automation strategies for aerospace components during conceptual design phases

Alejandro Pradas Gómez[*†], Massimo Panarotto[*] and Ola Isaksson[*]*
[*] *Chalmers University of Technology*
Hörsalsvägen 7A, floor 5
SE-412 96 Gothemburg, Sweden
alejandro.pradas@chalmers.se[†], massimo.panarotto@chalmers.se, ola.isaksson@chalmers.se
[†] Corresponding Author

Abstract

This paper explores the different design automation strategies used for the design of aerospace components. A literature review of the applicable strategies, together with the strategies used in the DEFAINE project are presented and compared. An opportunity to explore the combination of two strategies is presented (Enhanced Function-Mean and Knowledge Based Engineering), which has the potential to increase the discovery of novel design solutions while being able to assess their structural performance. The preliminary results of the combined strategy are presented, using a DEFAINE use case of a Turbine Rear Structure jet engine component.

1. Introduction

To keep competitive in the market, aerospace companies seek to reduce the efforts to design aircraft structures. Due to its complexity and variety of failure modes and scenarios to consider, it takes a long time to converge to a viable solution. The commercial and engineering relationship between Original Equipment Manufacturers (OEMs) and Tier One companies can delay further the process. The DEFAINE project [1] was launched with the aim to deliver and advanced design exploration framework able to: (1) Reduce recurring cost in design of aircraft systems by 10% and (2) Reduce the lead-time for design updates by 50%. By using a Front-Loading approach with the support of Knowledge Based Engineering (KBE) and Artificial Intelligence (AI) techniques, the design turnaround time can be reduced.

The purpose of this paper is not to focus solely on the automation techniques to make the process faster, but in the support needed to generate new configurations. Given the increased societal awareness on climate change, the aerospace companies are putting the focus in new systems and radical architectures that will ultimately impact the way structural components are designed. New boundary conditions, environmental loading and interfaces will change the functional requirements of these components.

In aerospace, the performance of the product (weight, stiffness, specific fuel consumption, etc.) is critical even at the conceptual stage. Therefore, on top of supporting designers with concept generation, there must be followed by a concept evaluation that satisfies at least, the basic criteria to take the design concept to the next maturity gate.

It is easier for designers to adopt a design strategy if it is based on existing tools and methods. Therefore, the following research questions (RQs) are addressed in this paper:

RQ1: What are the existing strategies for the design automation of the assessment of aerospace products?
RQ2: How can an automation technique support the conceptual generation and evaluation of physical products?

2. A selection of current approaches to design automation in industry

Two sources for gathering design automation strategies were used in this paper: Literature review and interviews with aerospace companies from the DEFAINE project.

2.1 Related automations in literature

Rolls-Royce

The design of turbomachinery is one of the most challenging problems for designing products, considering multi-disciplinary optimization. Rolls-Royce uses a modular approach to be able to define the different engine sections and their performance, called SOPHY [2]. The framework has a strong focus on the evaluation of the performance of physical components. For example, there is a predefined parametrization to define geometries and mesh them (PADRAM) [3], uses different in house solvers (HYDRA for CFD, Sc03 for mechanical) vs commercial solvers (FLUENT, ANSYS) as required. The SOPHY system has a clear focus for the geometrical multi-disciplinary optimization of turbomachinery geometries. Later demonstrations at recent conferences [4] show the application of this system for both early conceptual phases and problem solving at later phases.

A more conceptual approach to modelling the jet engine geometry system is proposed by di Mare and Kularni [5], [6]. Compared to the SOPHY approach, the modelling technique relies on an object oriented methodology (ceded in C++) with a focus on a product architecture and functional perspective. The objects cover for the geometrical and semi analytical models [7]. A particular characteristic of this particular modelling technique is the "negotiation" between neighboring components to agree on the appropriate interfaces given the needs of both objects [5]. This makes it particularly useful for a functional or architectural design space exploration. The application of his modelling techniques has been reported for the whole engine model, as well as for OGVs and the secondary air system.

Knowledge Based Engineering (KBE) Approaches

Since its inception in the 1980's, KBE has been difficult to define clearly defined "*It is like a butterfly - as soon as it has been called a 'caterpillar', it becomes a 'chrysalis'*"[8]. Most usually it is defined in line with an intersection of different disciplines: CAD, Knowledge Engineering, AI and software [9], [10]. In practice, the authors have found that each designer familiar with KBE defines it differently. And the latest reviews in the area still find it difficult to define what is and isn't KBE [11]. The most comprehensive definition of KBE found in the literature is made by La Rocca [10]: "*a technology based on the use of dedicated software tools called KBE systems, which are able to capture and systematically reuse product and process engineering knowledge, with the final goal of reducing time and costs of product development by means of the following: automation of repetitive and non-creative design tasks and support of multidisciplinary design optimization in all the phases of the design process*". For the purpose of this paper, two distinct KBE areas are made: KBE systems and KBE tools.

KBE Systems are defined in this paper as a purpose-built environment in which to code and execute knowledge. They are expected to be executed stand alone, or as part of a MDO workflow. Knowledge is stored as a software code built in a new programming language or on top of existing ones. Examples are ICAD [12], AML or more recently ParaPy. These languages allow KBE distinctive characteristics such as lazy evaluation, automating the workflow and results caching. Some use cases where KBE systems are used in the design of airframe structural components are [13], and for the jet engine components are [14], [15]

On the other hand, KBE *tools* are defined in this paper as add-ons to existing system whose purpose is to automate a specific task of the design process, but do not control the product architecture nor the parameter evaluation workflow. They may also live within other software, such as Knowledge Fusion in NX or KnowledgeWare in CATIA. In this category is where the lines start to blur between KBE and traditional Design Automation techniques. For the purpose of this paper, for a design automation to be considered KBE it should contain he intention to (1) automate geometric and non-geometric knowledge within a common object and (2) intend to use it at different projects and product architectures, so there should be some consideration for a modular combination of different objects.

Generative Design and Topology optimization

Topology optimization approaches allow for analytic and finite element methods to suggest shapes and geometries that minimize an objective (typically mass) given a set of external loading applied through predefined interfaces [16]. This automation approach has a focus on the physical definition of the component, and typically focus on small structural components such as beams, brackets, and other supporting devices. It is not used for the system or functional requirement. Topological optimization and additive manufacturing techniques complement each other as the organic shapes generated are attractive use cases that promise significant weight savings. In the search of design automation techniques for structural components, the benefits of this approach is a clear definition of surrounding interfaces and the goal of minimizing weight. However, this approach has a significant downside. During the optimization process, it looks for a limited set of objective (typically stress and global stiffness). Aerospace structures have a complex loading process like for example: Limit, Ultimate and Particular Risk Assessment loading, or fatigue and damage tolerance that require several analysis workflows and external tools to be used to consider the failure modes. Introducing all failure modes has the effect to decrease the iterations in the optimization, and potentially stopping the convergence of the results [17].

Lately, the term Generative Design has also been used for the generation of topologically optimized structures. It has been marketed by some vendors and included in their software such as PTC Creo, Autodesk Fusion 360, Altair Inspire Ansys discovery or Workbench. In general, the vendors claim the support of AI techniques to be able to suggest better designs. Often, the term generative design is used interchangeably with topology optimization, but the authors acknowledge that the term Generative Design has also other contexts [18]. For example, some of the previous KBE references also include this term to highlight that KBE Objects can be combined by an algorithm to generate new designs. Or the use of deep learning techniques to generate new designs [19]. In this paper, for a design strategy to be considered Generative Design must include the following components [18]: (1) A design schema, (2) a means of creating variations and (3) a means of selecting desirable outcomes. Note that how the variations and selections are made may be either manually by the designer or performed by an algorithm.

Commercial Off the Shelf Software+ Design Automation Techniques (COTS+DA)

This section covers a wide section of sub-techniques. In general, the design process of an aerospace structural component is divided into different specialist areas, like geometrical definition, structural analysis, thermodynamics or aerodynamics. Each discipline has their own models and specific software to develop it, typically a Commercial Of The Shelf (COTS) software such as NX or Ansys. This section covers the design automations techniques to (1) connect the different discipline models, (2) execute the models and (3) optionally optimize the results for a given objective and constraints. Existing tools exist to manage the workflow of different discipline models, called Process Integration and Design Optimization (PIDO) tools. An example of this methodology is [20] using iSight software, but other ones exist such as ModeFrontier [21], OpiSLang [22], Optimus [23] or RCE [24]

These PIDO techniques can help when:

- The product architecture is relatively mature and there are just a limited number of parameters to change.
- It is a linear workflow process (vs a convergence needed, see [25] otherwise)
- Disciplines have distinct models that can be executed independently
- The execution of the different sub-steps are performed by different workstations or servers, and not in the designer user machine.
- Users do not have programmatic experience, since these tools can be configured via an user interface

The PIDO can also be used in combination with other design automation techniques described above like KBE [26].

In addition to PIDO tool, the “COTS+DA” techniques also include automation of different discipline tools by means of scripting. If the appropriate standards exist within a company for the input/outputs of each discipline, then the tools can be executed from a batch script and the output of each sub-model or tools is parsed and transformed to the next sub-model.

Enhanced Function-Means

The previous approaches have been focused on the automation modelling of the physical product and its performance. The Enhance Function-Means (EF-M) approach [27] looks at the product and asks what should the product do. It originates from the Function-Means trees, see [28] for more information. It follows Hubka's law: “The primary functions of a machine system are supported by a hierarchy of subordinate functions, which are determined by the chosen means” [29]. So in practice it is a hierarchical decomposition of a product into the different subfunctions.

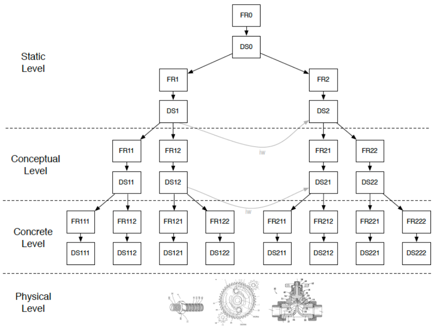

Figure 1: *Schematic of a product architecture using EF-M, from [30]*

The methodology's benefits and challenges has been discussed in [31], from which the following considerations are extracted. It is a good framework to perform architectural alternatives and studies. However it suffers from a precise geometrical embodiment. The lack of 3D CAD representation was later explored in [32] by automating the connection between the Design Solutions in the EF-M tree to User Defined Features (UDFs) in Siemens NX.

The following is an example to see the process rational of the function-means thinking. Consider the rib component inside the main wing. For simplification, it will fulfill only two functional requirements: Maintain wing profile shape and prevent fuel slosh. If an electrical/hydrogen configuration is selected, there is no fuel in the wing and no need for the rib to act as a barrier, so that requirement is removed and new possible design solutions appear, such as a truss like rib, that could be manufactured using AM techniques like Laser Wire deposition.

Architecture Design Space Graph

An alternative to model the function-means behavior of EF-M is the Architecture Design Space Graph (ADSG) [33], [34]. While EF-M is based on the generic product development field, ADSG is related to Systems Engineering. It attempts to control the different configuration alternatives and generate performance assessments of the design space, with the ability to connect it to MDO tools.

2.2 Approaches at DEFAINE companies

Even though all companies are within the aerospace business, each one of them has a different scenario: OEM vs Tier One, and each Tier One deals with a different product type: wire harnesses, airframe or engine structural elements. Therefore, even though they all apply design automation techniques to reduce the lead-time for designs, the implementation are different. Also each company is at different maturity stages of the implementation of the automation techniques. This section gathers those needs, the chosen strategy used and its motivation. It is important to notice that these design approaches are only the subset of strategies applied during the DEFAINE project. Companies apply additional strategies outside the scope described in this paper.

OEM - Saab

The need of the OEM is to design an aircraft system, at the design phase is kept at a system level. The starting point is a list of Top Level Aircraft Requirements, such as payload, range, altitude and speed to be able to meet the operational requirements. Through an evaluation of a range of promising configurations and technology embodiments, the final aircraft configuration and performance is estimated. The output is a configuration based on basic parameters, such as wing loading, wing and tail configuration, engine type, system architecture, movable types number etc. A physical CAD is not expected at this stage.

In this design activity, different models for evaluating the different performance metrics are used. More than one model may be available depending on the resolution and number of inputs available. Additionally, the model variables can be both an input and an output, and is the designer the one selecting what variable performs as an input or an output. It is a challenge to develop a workflow that contains the relevant models and connected appropriately, and it is also a challenge to update the variables, models and workflow with each design iteration.

Tools like Pacelab are specifically built for the conceptual aircraft design. The main advantage of this tool is that the design space exploration process is orchestrated from the software itself, having a central user interface to define the execution of the models: connect models, select inputs and outputs and run models. The software also acts as an aircraft configurator with a library of design alternatives. The configurations and alternatives are connected with different viewpoints. For example, the operational scenario (mission) uses the architectural configuration. In addition, in-house specific models can be used instead of the generic models provided by default. On the other hand, one of the challenges that this design approach has is the difficulty to represent different architectures and technologies within the same model. For example, having to artificially set some masses to 0 kg if a given technology is not used in that configuration.

The definition of a conceptual architecture is an iterative process that requires the interaction between the Aircraft OEM and the Tier One supplier. Once the architecture and component interfaces are sufficiently mature, the Tier One takes over the detail design in the aircraft subsystem or component. At the component level a new conceptual design exercise starts that triggers a negotiation between OEMs and Tier one's to get to the desired performance levels while maintaining weight and cost. This negotiating scenario is precisely the operational environment of the DEFAINE project and the context in which the lead-time and cost reductions are targeted. The following sections within this chapter describe the automation strategies for Tier One Companies.

GKN Aero Engine Systems (GKN AES)

The company designs and manufactures components for jet engines OEMs such as General Electric, Pratt and Whitney or Rolls Royce. The components designed are mostly static (non-rotating) such as a Fan Frames, Fan Outer Guide Vanes or Turbine Rear Structures. The inputs are typically surrounding component interfaces and interface loading. The objective of the conceptual study is to generate a geometry for a given manufacturing and assembly process that fulfils a set of performance requirements (such as structural stiffness) or aerodynamic performance (such as total pressure loss). The manufacturing risk, manufacturing schedule and cost predictions are sometimes added to performance metrics, depending on the product type and needs. The outcome of the design study are 3D solid CAD models and its performance metrics.

The performance metrics are multidisciplinary and therefore different models are required, requiring most of them a detail 3D model definition. GKN considered a multidisciplinary KBE approach to manage the creation of such models in the past [14]. Two decades ago, a strategic decision was made to use COTS and in house tools instead to develop models in a KBE system. Some of the reasons behind this rationale are:

- The cost of KBE systems at that time was high, several times the cost of a CAD license.
- The cost of maintaining such system.
- The specialist (software oriented) designer capabilities of the existing workforce. For example, there are plenty of engineers at the company (and outside) able to create CAD models using NX or FEM using Ansys. There are only a handful at GKN AES that can even develop or even run KBE applications. That is a risk in case any of the “KBE” engineers is not available for a new project.
- The ability to transition CAD models from the conceptual to the preliminary design teams, since the tools used by both teams were the same. The design practices and PLM architecture is also the same.
- A KBE system is seen as being in control of the geometrical definition. And for geometrical definition it is preferred to have a CAD for the complex aerodynamic shape definitions required. Then specific disciplines, such as the aerodynamic performance, can be responsible to define *only* the aerodynamic profiles and leave the rest of the parameter for the CAD software to control.

It is recognized at GKN AES that having a KBE system could have advantages, such as being able to “rule-base” any parameter and connect it to any other parameter. Nevertheless, the strategy selected for conceptual component space exploration – Engineering Work Bench (EWB) – is based on automation scripts to connect different in house tools and COTS tools (NX, Ansys). In addition of solving these perceived KBE disadvantages, this approach is valued for the flexibility to be able to integrate independent functionality developed by different designers and tools. Furthermore, the effort to maintain, train and develop the engineer capabilities falls within the general disciplines strategies and there is no need to have an internal provision for those activities.

GKN Fokker Elmo (GKN FE)

The company designs and manufactures the Electrical Wiring Interconnection Systems (EWIS) that provide power distribution and signal transmission among the different subsystems distributed around the aircraft, such as control units, sensors or actuator subsystems. On the initial phases, the input to the design is a list of subsystems, their location on the aircraft, the connection requirements to other subsystems. Together with a skeleton 3D model of the AC and the main paths, a first EWIS design is generated that includes a 3D route for each connection, the harnesses lengths and weights.

GKN Fokker is developing an in house tool based on a KBE System (ParaPy) that allows to generate in minutes what otherwise would take weeks to do manually, while increasing the quality of the result. The combinatorial amount of possibilities, together with the complex rules needed to satisfy the requirements suggest that a code based logic and a software development perspective to be the best approach to assess the product performance.

GKN Fokker Aerostructures (GKN FAE)

The company designs and manufactures lightweight structures in metal and composite materials, like for example movables, HTP or fuselage sectors. For the use case of DEFAINE, they develop a movable structure where the input is the Outer Mold Line (OML) or aerodynamic shape and the hinge interface points. The Outcome of the design process is a product configuration (spars, ribs, covers, stringers in surface models, Fittings in 3D models) together with the analysis of its performance (Structural and cost for the DEFAINE example). The results can be a one-shot analysis, DoE or optimization activity. The company uses an in house KBE System called Multi-Disciplinary Modeler (MDM) [35] based on ParaPy, which is developed and maintained by a dedicated team.

The main benefits are the ability to generate multiple configurations very quickly. Thanks to its programmatic nature, the detail phase analysis tools can be implemented at the conceptual phases. The combination of both allows the exploration of the design space to be front-loaded and to be easily customized for a particular customer requirement. This approach has been used for both traditional aerospace OEMs as well as for new players in the market looking for non-conventional configurations such as electrical, hydrogen or Urban Aerial Mobility (UAM) platforms.

2.3 Characterization of the design automation strategies

Despite being different design automation approaches used for different purposes, there are dimensions in which they can be compared. The following section explore some of those dimensions. Given that a quantitative and even a qualitative comparison is difficult, a linear scale was defined, and the authors subjectively placed the approaches on the scale with the intention of having a comparative measure of each dimension. Their placement was then shared and discussed by other DEFAINE partners prior to publication.

- **Domain to assess**
  Some automations are intended to define what the product should do (functional domain) while others focus on how the product should perform, for which a physical definition is often required (physical domain).

- **Storage of Knowledge**
  Some automation techniques allow for the design rationale to be included in the artifacts (KBE), the user interaction with the automation is through writing and importing code. Others require the experienced designer to build it using a GUI (EWB, Pacelab).

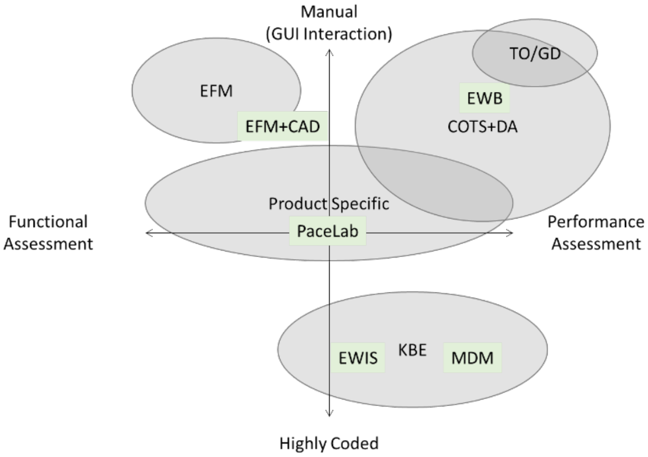

Figure 2*: Positioning the design automation techniques in terms of the domain to assess (horizontal) and storage and generation of knowledge (vertical)*

- **Design Phase**
  The design automation activities can be used at different phases of the design. The phases from [36] are used to position the different approaches: Planning and clarifying, Conceptual Design, Embodiment Design, Detailed Design. For this dimension, the company role is considered. For example, the OEM may consider that a product may be in the embodiment phase, while for a Tier one may be in the conceptual phase. The categorization is made from the point of view of the design team who uses the design automation strategy.

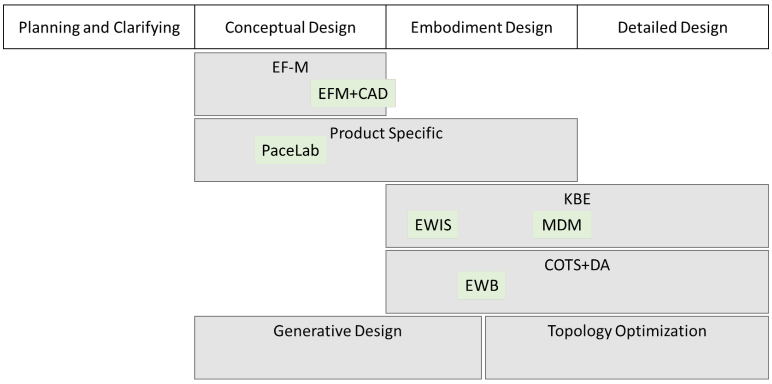

Figure 3: *Different Design Automation Strategies used at different design phases*

2.4 Outcomes of the design automation approach interviews

The main outcome of the interview process is that the selection of a design automation strategy is multi-faceted and comparing them is complex as they have different purposes and needs.

The selection of the strategy is based mainly on the following considerations:

1. The type of product that the company manufactures: each product has different input and output requirements; it is difficult to define a superior strategy even for the same type of product.
2. The organization and management: A strategic management style contributes towards a design automation strategy. The preference may be implicit, by for example the limited budgeting for developing, maintaining and training for in-house applications may lead to use COTS solutions.
3. The experience of the engineers: Even if there was a superior design automation strategy, the experience of the engineer – especially the team lead – has a major impact on the design automation chosen.
4. The tools available: Engineers joining a project may have experience on more suitable design approaches, but the licenses may not be available and therefore the approach is discarded. Lead times to get the licenses are of the same order as some of the conceptual design projects.
5. The legacy of the team or project: Teams and projects have a history on how to do things, that may be even coded in the Quality Management System or agreed by contract with the customer. Changing the design automation approach is compared to “swimming against the tide”.

3 Proposed approach: Connecting EF-M Design Solutions to KBE Primitives

In order to support designers in generating both new design solutions in the functional domain and physical evaluation of those, a new link between different design strategies is proposed. On one hand the EF-M approach can support the functional domain, generation of new potential design solutions and support different product variants within the same tree. This is something needed for the new aerospace engine architectures. However, this technique does not allow for concept evaluation at the physical level. On the other hand, KBE techniques and libraries of primitives allow for the quick generation and evaluation of different products, but do not have explicit support to look at functional level on its own.

As discussed in the previous sections, an previous attempt exist to connect the EF-M to the physical (CAD) domain [32] but there are no automatic means of evaluating the concept, for which considerable effort and expertise is needed. Based on this, it seems promising to connect EF-M to a KBE approach to generate both the CAD and the evaluation models. Both strategies use an object-oriented approach: EF-M uses Design Solutions and KBE uses primitives.

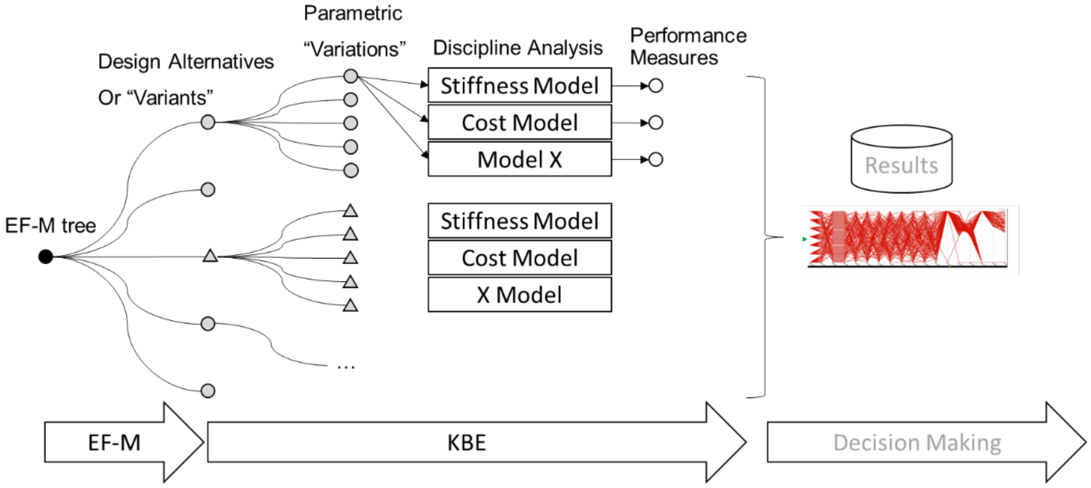

Figure 4: Schematic of the design process. A EF-M tree contains all the information, that get expanded into different variants (think of different design solution combinations) and each variant can have different values for each variable. Then a multidisciplinary set of models is used to assess each variation. Every variant and variation configuration, together with the multidisciplinary performance value is stored in a results database to assess the design space.

The design process proposed has the following steps:

1. **Define EF-M tree**. This is done as if it was an independent step, see [37].
2. **Define and configure the study**. Select the parameters to vary, or the Design Solutions to consider. Select the performance metrics desired as an output of the evaluations. Select the analysis approach: Design of experiments, Multidisciplinary optimization, one shot, etc.
3. **Link EF-M Design Solutions (DS) to KBE Primitive instances.** Use the existing library at the company, or develop new primitives if a new means to solve the functional requirement has been created.
4. **Enrich the EF-M tree with additional information.** Provide as much information as the primitive requires into the EF-M tree. This can be parametric information or constraints (or link to other files) that the KBE Primitives will need to be instantiated
5. **Create a product architecture.** Similar to a Module Interface Graph [38], Primitives are physically located next to each other and connected
6. **Create/Update KBE Application**. An existing skeleton is expected to exist at the company for similar products, but may not consider new Design Solutions.
7. **Execute Application**. Execute KBE application to obtain the desired outputs.
8. **Review results.** Evaluate performance metrics and decide if a new loop is required or a final result can be selected.

The key of this new strategy lays on steps 3, 4 and 5 for which further clarification is provided below. These steps deal with how to transfer the information from a functional domain to a physical domain.

Step 3 is required to identify what Feature needs to be instantiated in the KBE application when the variant includes a given Design Solution. Alternative approaches were considered to avoid this space, such as a particular naming convention, but were discarded in order to give the maximum design freedom while generating EF-M trees.

Step 4 solves the problem of passing down the required inputs for a KBE application to be instantiated. KBE Primitives can have default arguments, but some other are mandatory. Another consideration to include all the information in the EF-M tree is to avoid having the inputs dispersed in different places. If all the information needed is on the tree, there is a single source of truth. Note that the inputs could be parameters or design variables, for which ranges or list should be specified instead.

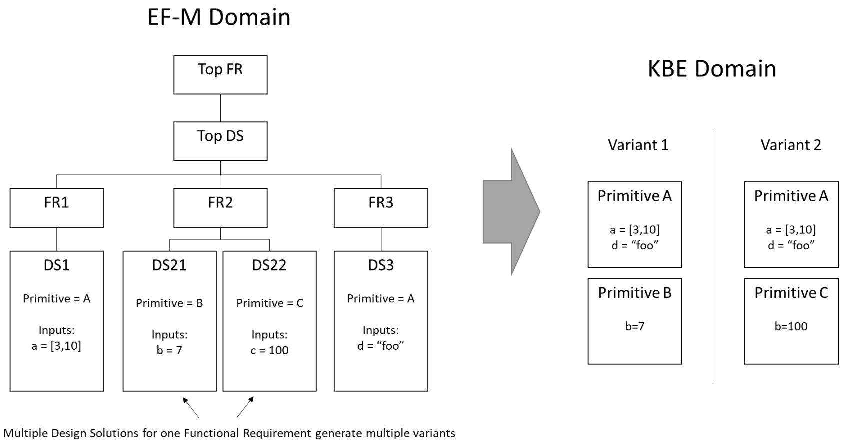

Figure 5: *Step 3 and 4 visualizations. Note how generating more than one Design Solution under FR2 creates 2 possible alternatives with two different primitives. Also note how to design solutions (DS1 and DS3) relate to the same primitive A and complement their input.*

Step 5 deals with the problem of the KBE structured approach of having child and parent architecture. EF-M trees have a functional hierarchy, not architectural. While step 4 passed information about inputs to primitives, this step provides information about physical connectivity. This is required because there may be more than one Primitive of the class A in a product, with different inputs, related to different DS. While step 4 was a many-to-one relationship, this step is a one-to-one relationship.

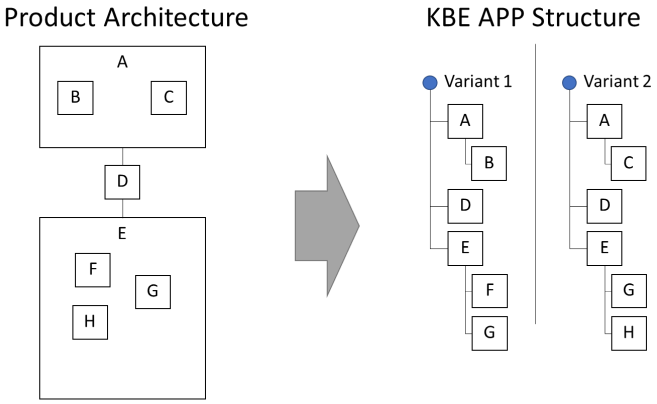

Figure 6: *Visualization of Step 5 purpose: Defining an architecture of primitives (all possible variants included) and translating it to a hierarchical class structure for each variant KBE app instantiation*

Demonstration on how it is applied at the DEFAINE use case.

The object to be designed is a jet engine Turbine Rear Structure (TRS), also called Turbine Exhaust Case (TEC) or Tail Bearing Housing (TBH). In a conventional commercial turbofan it sits after the Low pressure turbine.

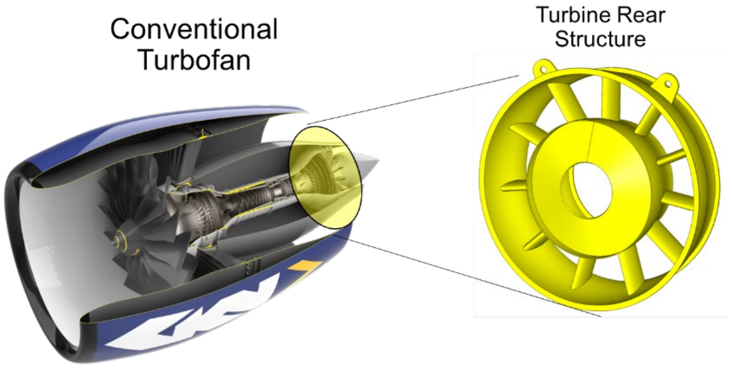

Figure 7: *Turbine Rear Structure (TRS) in the context of a conventional turbofan*

The Functional Requirements considered for this product are:

- Guide airflow from turbine to nozzle
- De-swirl airflow
- Provide containment of turbine blade
- Transfer loading through structure

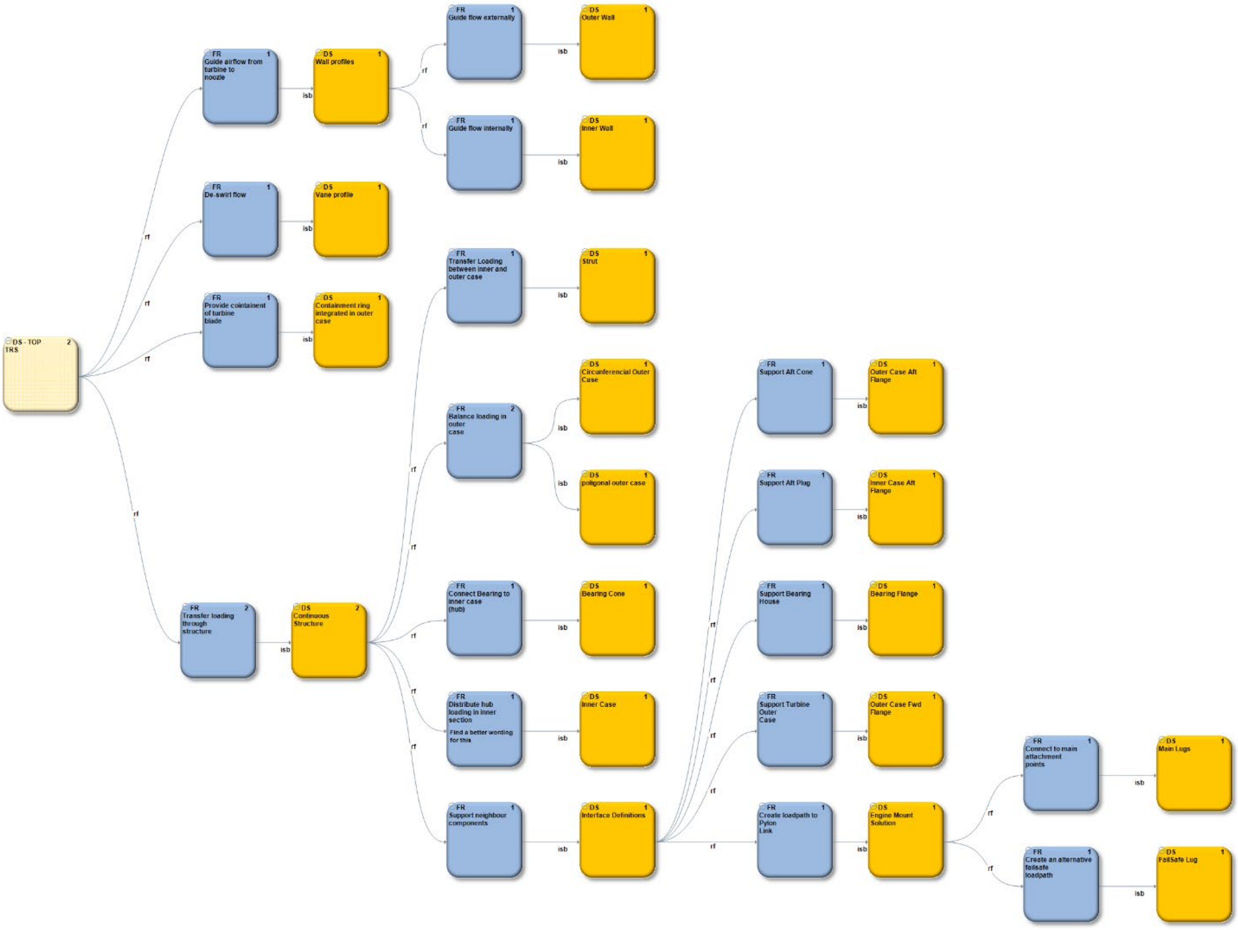

Figure 8: EF-M tree of a TRS. Note how there is only one Functional Requirements that contain two possible design solutions, making 2 the number of variants in this study.

For each Design Solution, a Library of Primitives has been created, with the ability to generate CAD and Finite Element Meshes. The following picture provides an example of the features and the product architecture generated for Step 5.

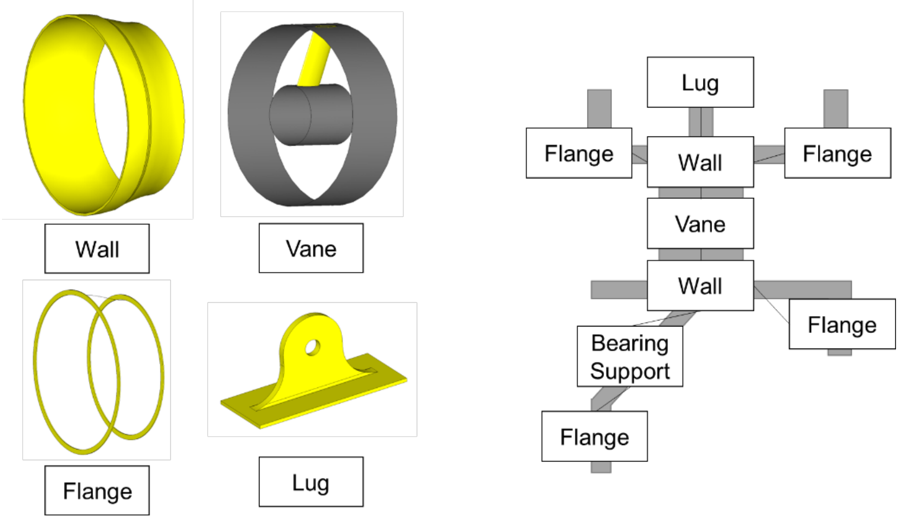

Figure 9: *On the left side there is a CAD representation of some of the primitives, and on the right the product architecture is superposed to a 2D cross section of the TRS, showing the location where each primitive is. For visualization purposes, the type of primitive is presented, not the actual instance (for example "Flange" instead of "Forward Outer Flange")*

KBE application is able to generate two CAD architectures for the two turbine blade containment solutions: integrated or independent. For each architectural solution, there are approximately 30 parameters that can be configured individually.

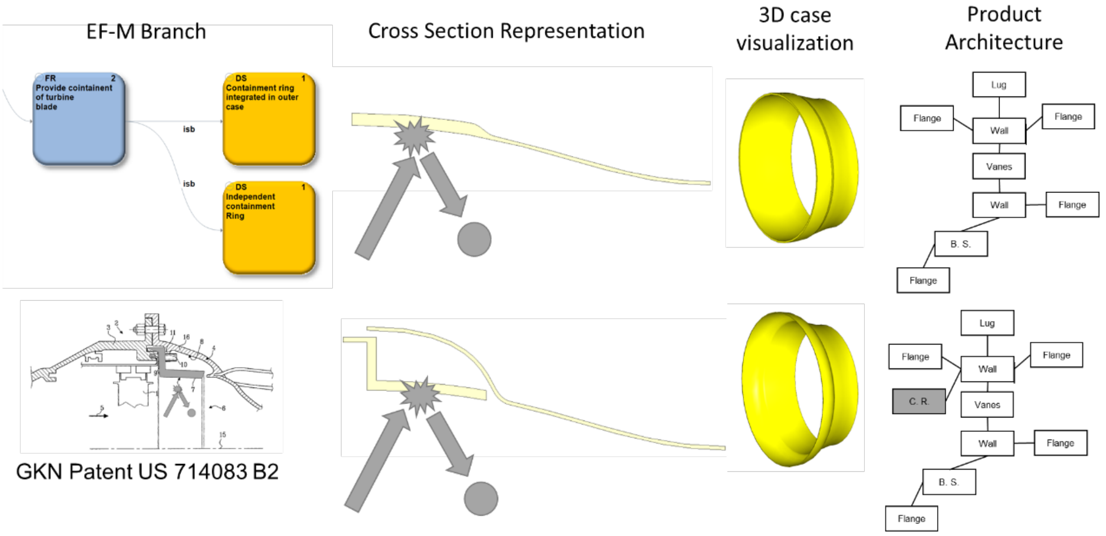

Figure 10: *EF-M detail of the Functional Requirement (FR) that has two possible Design Solutions (DS) to contain the turbine blade.*

The application is also able to generate a mesh, that contains the appropriate mesh details (element types, mesh densities, boundary conditions and loading application) that are needed to run in a finite element solver, in this case, Ansys mechanical.

4 Preliminary Results

It is possible to generate a product from an EF-M tree and automatically evaluate it. The models generated are able to represent the CAD geometry and detail required to perform a stiffness analysis. However, the goal of performing it quickly the using a KBE system can be seen from two angles:

- **Time to develop**: the author was inexperienced in KBE Systems, it took approximately 100 hours to learn the System and develop the primitives to be able to support the use case. It is hard to distinguish between

learning and developing, but the estimation is that these two task were equal in duration (50 hours + 50 hours)
- **Time to generate models**: The time to generate the models varies significantly with the KBE experience. Initial attempts took 10 minutes to generate a CAD model plus 12 minutes to generate a mesh. However with the support from KBE experts, this time was reduced to 13 seconds to generate the CAD model and 95 seconds to generate a mesh. Running and extracting the FEM results using a dedicated server took only 5 seconds. With these timescales, it is affordable to perform design of experiments and optimization loops.

The development time of models, particularly in cases where knowledge is stored in libraries, is further explored. Differentiating between recurring and non-recurring development efforts is often a challenge. Although the initial creation of a primitive requires time, its subsequent reuse in future studies does not require additional effort. As primitives are repurposed in new projects, it's anticipated that a reduction in their development time will occur. For this purpose, two new use cases were studied where some new primitives were needed. Figure 9 shows the architectural changes.

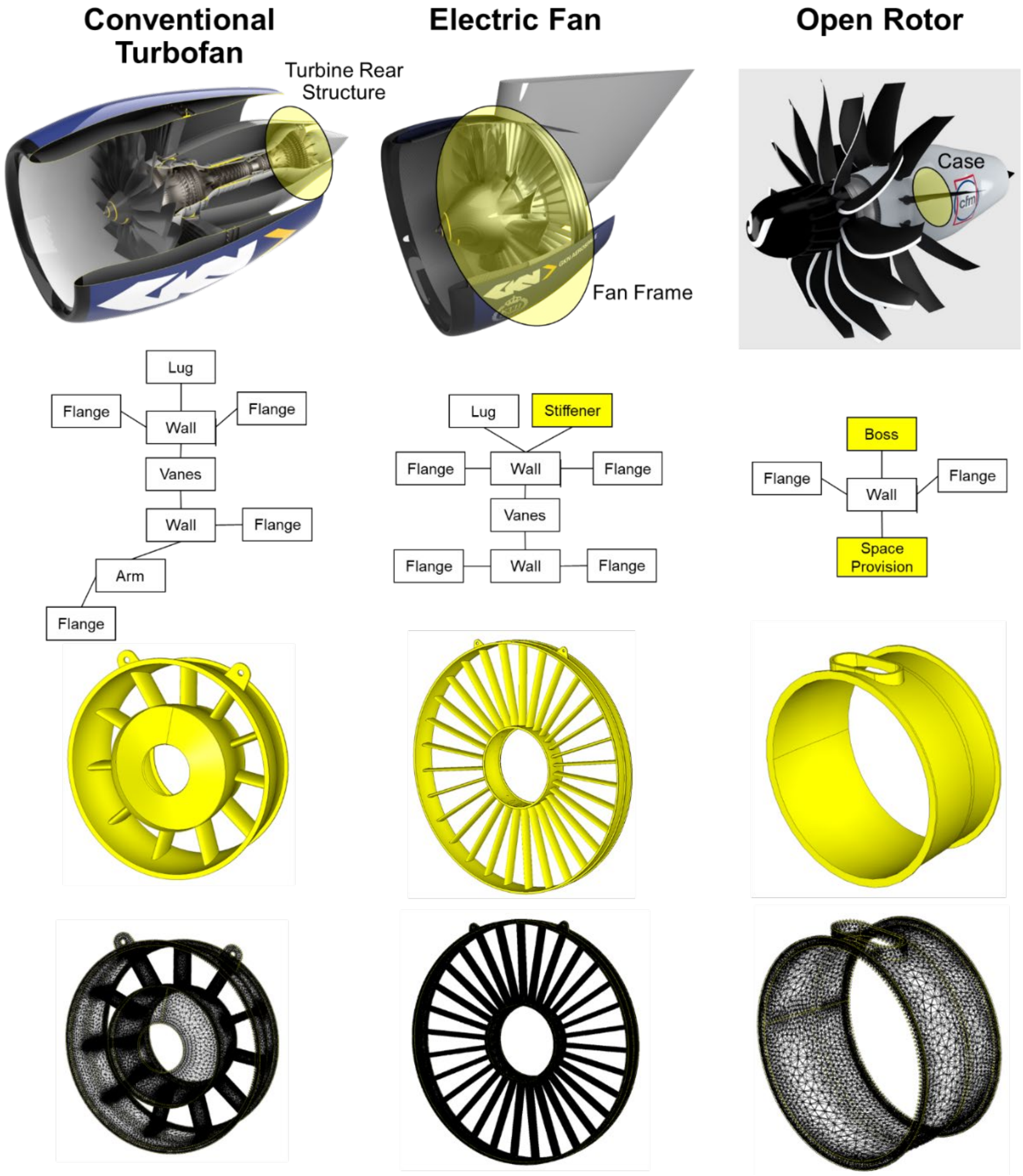

Figure 11: *Comparison of the different architectures used to test the impact of new primitives development. To the left there is the TRS use case described (100 hours effort). In the center, a new stiffener primitive was required for the fan frame which took only one hour to develop. To the right, the case with a space provision was required and it took 5 hours to develop.*

5 Discussion

The current approach is work in progress and it is expected to improve its capability as the DEFAINE project evolves. There are some missing automation steps, namely executing the model to run and extract results, and generating other types of performance measures. This has been proven possible in other cases by GKN FAE [35] so it is not the focus of this discussion.

An initial expectation of this approach was to allow for non-KBE experts to define their product architecture and variability and let the process combine and generate KBE applications. In practice, this will be challenging as it is likely that the new products to study will need to create or update primitives. So the support of a KBE developer will be needed. While EF-M allows for the rapid generation of many design alternatives and configure them all in a single location, the creation of a KBE primitive that enables the intended behavior is the bottleneck for its analysis.

Preliminary results indicate that an investment in a KBE library has long-term benefits. However, this strategy is only feasible if the products share a common architecture and if the primitives can be generalized. Consequently, this approach may not be suitable for companies that do not have a Product Family strategy. It is also discouraged for companies who are not willing to invest in developing and maintaining KBE libraries.

The advantage of using EF-M is that it gives a framework for the designer to come up with innovative solutions and manages all the different design variants in one single tree. If there is no need to consider innovative solutions, an alternative approach to model the architectural design space may be [33], [34].

6 Conclusion and future outlook

Answering the research question 1, a description of the different design automation strategies for aerospace products has presented from literature and from the DEFAINE use cases. The strategies are compared in some dimensions, and it was found that there a need to systematically create new design solutions and evaluate them. To answer the research question 2, a new automation design strategy is proposed which combines two existing and complementary approaches: EF-M for conceptual generation and KBE for design evaluation. A DEFAINE use case using a structural jet engine component (TRS) is used to develop the feasibility of this approach.

This work is part of the ongoing DEFAINE project, the results are preliminary and it is expected that the approach will be refined in the upcoming months, adding more automations on the process and increasing the performance assessments of the use case. In particular:

- An approach to automatically generate a KBE parent object without any manual intervention
- The ability to interact with the inputs from a web interface, without any service installed locally.
- Execution and retrieval of Finite Element Model results

Acknowledgement

The research presented in this paper has partially been performed in the framework of the DEFAINE (Design Exploration Framework based on AI for froNt-loaded Engineering) project and has received funding from ITEA 3 programme.

References

[1] Max BAAN *et al.*, “DEFAINE - Design Exploration Framework based on AI for froNt-loaded Engineering,” presented at the Aerospace Europe Conference 2023 – 10TH EUCASS – 9TH CEAS, Lausanne, Switzerland, 2023.

[1] Shahpar S. SOPHY: An Integrated CFD BASED Automatic Design Optimisation System. 2005.

[3] S. Shahpar and L. Lapworth, “PADRAM: Parametric Design and Rapid Meshing System for Turbomachinery Optimisation,” in *Volume 6: Turbo Expo 2003, Parts A and B*, Atlanta, Georgia, USA: ASMEDC, Jan. 2003, pp. 579–590. doi: 10.1115/GT2003-38698.

[4] S. Shahpar, “Role of MDO in Turbomachinery Design,” IHES - Paris, Bures-sur-Yvette, Sep. 20, 2022.

[5] D. Kulkarni, G. Lu, F. Wang, and L. di Mare, “Virtual Gas Turbines Part I: A Top-Down Geometry Modeling Environment for Turbomachinery Application,” *Journal of Engineering for Gas Turbines and Power*, vol. 144, no. 3, p. 031010, Mar. 2022, doi: 10.1115/1.4052560.

[6] L. Di Mare, D. Y. Kulkarni, F. Wang, A. Romanov, P. R. Ramar, and Z. I. Zachariadis, “Virtual Gas Turbines: Geometry and Conceptual Description,” in *Volume 1: Aircraft Engine; Ceramics; Coal, Biomass and Alternative Fuels; Wind Turbine Technology*, Vancouver, British Columbia, Canada: ASMEDC, Jan. 2011, pp. 347–358. doi: 10.1115/GT2011-46437.

[7] L. di Mare, M. Carnevale, M. Rife, D. Kulkarni, and R. Northall, *Modelling and Design of LPC Components With Semi-Analytical Models: Part II — Design of Optimal OGV Cascades*. 2019. doi: 10.1115/GT2019-90481.

[8] E. M. Stokes, “Managing Engineering Knowledge, MOKA,” p. 292.

[9] W. J. C. Verhagen, P. Bermell-Garcia, R. E. C. van Dijk, and R. Curran, “A critical review of Knowledge-Based Engineering: An identification of research challenges,” *Advanced Engineering Informatics*, vol. 26, no. 1, pp. 5–15, Jan. 2012, doi: 10.1016/j.aei.2011.06.004.

[10] G. La Rocca, “Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design,” *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 159–179, Apr. 2012, doi: 10.1016/j.aei.2012.02.002.

[11] P. Kügler, F. Dworschak, B. Schleich, and S. Wartzack, “The evolution of knowledge-based engineering from a design research perspective: Literature review 2012–2021,” *Advanced Engineering Informatics*, vol. 55, p. 101892, Jan. 2023, doi: 10.1016/j.aei.2023.101892.

[12] P. N. Payne, “STRUCTURAL DESIGN USING INTELLIGENT OBJECTS.” 1991.

[13] Tobie van den Berg, Ton van der Laan, Bas van Manen, Darpan Bansal, Ioana Ciobotia, and Jente Sonneveld, “A multidisciplinary modelling system for aircraft structural components,” presented at the Aerospace Europe Conference 2023 EUCASS CEAS, 2023.

[14] O. Isaksson, “A GENERATIVE MODELING APPROACH TO ENGINEERING DESIGN,” 2003.

[15] M. Sandberg, I. Tyapin, M. Kokkolaras, O. Isakasson, J.-O. Aidanpää, and T. Larsson, “A Knowledge-based Master-model Approach with Application to Rotating Machinery Design,” *Concurrent Engineering*, vol. 19, no. 4, pp. 295–305, Dec. 2011, doi: 10.1177/1063293X11424511.

[16] G. I. N. Rozvany, “A critical review of established methods of structural topology optimization,” *Struct Multidisc Optim*, vol. 37, no. 3, pp. 217–237, Jan. 2009, doi: 10.1007/s00158-007-0217-0.

[17] J. Alanko and M. Wallin, “Evaluation of AI-driven Generative Design and Redesign of a MINI-LINK Mounting Kit,” Chalmers, 2023.

[18] S. Krish, “A practical generative design method,” *Computer-Aided Design*, vol. 43, no. 1, pp. 88–100, Jan. 2011, doi: 10.1016/j.cad.2010.09.009.

[19] S. Oh, Y. Jung, S. Kim, I. Lee, and N. Kang, “Deep Generative Design: Integration of Topology Optimization and Generative Models,” *Journal of Mechanical Design*, vol. 141, no. 11, Sep. 2019, doi: 10.1115/1.4044229.

[20] P. Hecker, D. Delimar, H. Brandl, and M. Lo¨tzerich, “Process Integration and Automated Numerical Design Optimization of an Eigenfrequency Analysis of a Compressor Blade,” in *Volume 6: Structures and Dynamics, Parts A and B*, Vancouver, British Columbia, Canada: ASMEDC, Jan. 2011, pp. 791–800. doi: 10.1115/GT2011-45489.

[21] L. Knöös Franzén, I. Staack, P. Krus, and K. Amadori, “Optimization Framework for Early Conceptual Design of Helicopters,” *Aerospace*, vol. 9, no. 10, Art. no. 10, Oct. 2022, doi: 10.3390/aerospace9100598.

[22] A. Al-Qarni, P. Kumar, S. Koushan, and A. El-Refaie, “Comparative Analysis Between Various High Specific Power Permanent Magnet Motor Topologies for Aerospace Applications,” in *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*, Oct. 2022, pp. 1–8. doi: 10.1109/ECCE50734.2022.9947953.

[23] R. d’Ippolito, S. Donders, N. Tzannetakis, H. V. der Auweraer, and D. Vandepitte, “Integration of Probabilistic Methodology in the Aerospace Design Optimization Process,” in *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2005-2137.

[24] B. Boden *et al.*, “RCE: An Integration Environment for Engineering and Science,” *SoftwareX*, vol. 15, p. 100759, Jul. 2021, doi: 10.1016/j.softx.2021.100759.

[25] J. R. R. A. Martins and A. B. Lambe, “Multidisciplinary Design Optimization: A Survey of Architectures,” *AIAA Journal*, vol. 51, no. 9, pp. 2049–2075, Sep. 2013, doi: 10.2514/1.J051895.

[26] A. Raju Kulkarni, G. La Rocca, T. Berg, and R. Dijk, *A knowledge based engineering tool to support front-loading and multidisciplinary design optimization of the fin-rudder interface*. 2017.

[27] P. Schachinger and H. L. Johannesson, “Computer modelling of design specifications,” *Journal of Engineering Design*, vol. 11, no. 4, pp. 317–329, Dec. 2000, doi: 10.1080/0954482001000935.

[28] J. R. Müller *et al.*, “Enhanced function-means modeling supporting design space exploration,” *AI EDAM*, vol. 33, no. 4, pp. 502–516, Nov. 2019, doi: 10.1017/S0890060419000271.

[29] V. Hubka and W. E. Eder, *Theory of Technical Systems: A Total Concept Theory for Engineering Design*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988. doi: 10.1007/978-3-642-52121-8.

[30] C. Levandowski, D. Raudberget, and H. Johannesson, “Set-Based Concurrent Engineering for Early Phases in Platform Development,” *Moving Integrated Product Development to Service Clouds in the Global Economy*, pp. 564–576, 2014, doi: 10.3233/978-1-61499-440-4-564.
[31] Jakob Müller, Maria Daniela Irene Siiskonen, and Johan Malmqvist, “Lessons learned from the application of enhanced Function-Means modelling,” vol. 1, pp. 1325–1334, May 2020, doi: 10.1017/dsd.2020.87.
[32] J. Muller, M. Panarotto, and O. Isaksson, “Function Model-based Generation of CAD Model Variants,” *CADandA*, vol. 18, no. 5, pp. 970–989, Jan. 2021, doi: 10.14733/cadaps.2021.970-989.
[33] J. H. Bussemaker, P. D. Ciampa, and B. Nagel, “System Architecture Design Space Exploration: An Approach to Modeling and Optimization,” in *AIAA AVIATION 2020 FORUM*, VIRTUAL EVENT: American Institute of Aeronautics and Astronautics, Jun. 2020. doi: 10.2514/6.2020-3172.
[34] J. Bussemaker and L. Boggero, “From System Architecting to System Design and Optimization: A Link Between MBSE and MDAO”.
[35] T. Van Den Berg and A. H. Van Der Laan, “A multidisciplinary modeling system for structural design applied to aircraft moveables,” presented at the AIAA Aviation and Aeronautics Forum and Exposition, AIAA AVIATION Forum 2021, 2021. doi: 10.2514/6.2021-3079.
[36] G. Pahl and W. Beitz, *Engineering design: a systematic approach*, 3rd ed. London: Springer, 2007.
[37] J. R. Müller, M. D. I. Siiskonen, and J. Malmqvist, “LESSONS LEARNED FROM THE APPLICATION OF ENHANCED FUNCTION-MEANS MODELLING,” *Proc. Des. Soc.: Des. Conf.*, vol. 1, pp. 1325–1334, May 2020, doi: 10.1017/dsd.2020.87.
[38] D. Krause, S. Eilmus, and H. Jonas, “Developing Modular Product Families with Perspectives for the Product Program,” in *Smart Product Engineering*, M. Abramovici and R. Stark, Eds., in Lecture Notes in Production Engineering. Berlin, Heidelberg: Springer, 2013, pp. 543–552. doi: 10.1007/978-3-642-30817-8_53.