



Function model-based generation of CAD model variants

Downloaded from: <https://research.chalmers.se>, 2025-12-04 23:38 UTC

Citation for the original published paper (version of record):

Müller, J., Panarotto, M., Isaksson, O. (2021). Function model-based generation of CAD model variants. *Computer-Aided Design and Applications*, 18(5): 970-989.
<http://dx.doi.org/10.14733/cadaps.2021.970-989>

N.B. When citing this work, cite the original published paper.



Function Model-based Generation of CAD Model Variants

Jakob R. Müller¹ , Massimo Panarotto²  and Ola Isaksson³ 

¹Chalmers University of Technology, Göteborg, jakob.muller@chalmers.se

²Chalmers University of Technology, Göteborg, massimo.panarotto@chalmers.se

³Chalmers University of Technology, Göteborg, ola.isaksson@chalmers.se

Corresponding author: Jakob R. Müller, jakob.muller@chalmers.se

Abstract. A product is an artefact which fulfils a specific function. However, most design automation (DA) approaches which are used to generate multiple alternative design concepts focus on the generation of CAD models. These neglect to represent the functional aspects of the product, and are furthermore deemed too rigid for the introduction of novel solutions. Pure function modelling approaches on the other hand provides methods such as design rationale representation, introduction of novel solutions or instantiation of combinatorial alternative concepts, but the resulting models are insufficient for analysis. To alleviate this, a design space exploration (DSE) approach which couples function modelling and CAD is presented. The approach links the product's design rationale modelled in enhanced function-means (EF-M) to a DA approach via the here introduced object model for function and geometry (OMFG). The resulting method is able to automatically generate CAD models of alternative concepts based on combinations of alternative design solutions defined in the function model. The approach is presented through a case study of an aircraft engine component. Sixteen different concepts are generated based on four functions with alternative solutions. In an initial computation of the effort to generate all alternative concepts, the DA aspect of the approach's effort pays off as soon as five functions have two or more alternative solutions. Beyond the benefit of efficient instantiation of CAD models of alternative product concepts, the approach promises to provide the design rationale behind each concept, and thereby a more systematic way of exploring and evaluating alternative design concepts.

Keywords: Function Modelling, Design Space Exploration, Product Development, Design Automation, Method Development

DOI: <https://doi.org/10.14733/cadaps.2021.970-989>

1 INTRODUCTION

The goal of product development is to generate an artefact which fulfils a certain set of functions which generate value for its users and other stakeholders [29]. In this, to express what a product is *to do* – that is, its function – is the most important task of a designer [33]. This definition guides the entire subsequent design process, as it describes the design space – the set of all possible products which fulfil the functional requirements. This design space exploration (DSE) is one of the main tasks in the conceptual development phases of the design process. At the end of these conceptual phases, the designers are to have selected a concept for further development of which they are sure it fulfils all functions and meets the initial requirements.

This concept, the design of a product, is placed in the *physical* domain; which is to be distinguished from the abstract, *functional* domain, in which the expression of the function lies.

A majority of applied product development methods focus on the representation of the physical domain: models that represent the solutions, products and concepts in geometrical form (for example computer aided design (CAD)) or their behaviour (for example finite element method (FEM)) [9]. The use of models representing the function of a product is not common in industrial product development [34]. Although research has proven that the use of function models can be beneficial for DSE, a majority of these approaches remains at the academic level [1, 11, 23, 24]. One of the major challenges for function models, however, is that they lack a means for sufficient product behaviour analysis, which commonly requires a geometrical representation of the product [9]. This publication builds on the hypothesis that DSE can benefit from the use of function modelling (FM) in combination with geometric product models [24]. As a result, 'a primary question [...] is how to make function elements know to which geometric parts they are related to' [6].

Several efforts have been made to improve the generation of CAD models of design variants. Among the most prominent is knowledge-based engineering (KBE), which combines the efforts "to computerise as many individual operations involved in the design process as possible" [25] with the goal of generating knowledge about the design, eventually enabling fact-based decisions about the design process [39]. However, such approaches rely on complex master models [3] or specialised software routines [31] and are therefore hard-pressed to justify the initial setup with later trade-offs in resource and effort gains. Furthermore, many KBE approaches focus on design automation (DA), the generation of variant CAD models, without providing a means of representing the product's function. It has further been stated that 'functional reasoning is indispensable to the development of [...] CAD' [37].

This situation has left a gap for a DSE approach which provides a representation of the product's function, the ease of introducing novel solutions and the automated generation of the respective geometrical product models. Therefore, in this publication, a DSE approach based on an object model coupling the functional to the geometrical domain is presented. The approach is based on a FM framework optimised for the representation of alternative design solutions, enhanced function-means (EF-M) modelling. The functional domain model is linked to a method for the automatic generation of CAD models through an object model for function and geometry (OMFG).

This publication is structured as follows: Section 2 explains the aspects of CAD and function modelling relevant for the OMFG. Section 3 reports on how the approach was developed. In Section 4, the theoretical workings and underlying object model of the presented approach are illustrated, and then the approach is applied to a case product in Section 5. Lastly, the approach is discussed on its technical performance, relation to existing technologies and potential for further work in Section 6, before a summary of the paper in Section 7.

2 BACKGROUND

Ultimately, the aim of the presented approach is to improve DSE as it is performed in the early phases of the product development process, for example the conceptual product development phase as defined by [36]. As

such, the goal is to generate (synthesize) and analyse as many relevant design concepts as possible to be able to determine the ideal available concept based on facts. A *concept* is, in this context, defined as the design of a product, independent of the availability of models, which is distinguishably different from other concepts either by use of different dimensions or sub-solutions. However, the study presented in this publication only takes into account variation in solutions and sub-solutions; hence concepts with alternative dimensions are only implied – the approach only takes into account 'modular bandwidth' and ignores 'parametric bandwidth' as defined by [20].

The design space is the near infinite number of concepts that fulfil a specific set of functions, bordered by the product's requirements. For DSE, this space first needs to be defined; then the concepts residing inside this space need to be represented and then analysed for which concept fulfils the required functions best [18]. In engineering design, the representation of concepts is commonly done with different product models, those being geometry models in the form of CAD in the majority of cases [9]. The approach presented in this publication focuses on the population of the design space, that is the generation of representations of concept in the design space.

2.1 Product Geometry Models

The modelling approach presented in this publication is based on *feature-based CAD models*. A feature-based CAD-system can be distinguished from a direct-modelling CAD system in that it creates an editable record of each performed action, a so called feature, containing the resulting geometrical element. Features are containers for geometrical objects and operations [15]. They are automatically generated when creating new geometry objects and listed in the feature tree in chronological order, providing a 'design intent' [27]. However, this does not alleviate the lack of design rationale in CAD: even though the feature tree may show the 'design intent', it merely represents the design intent of the model and not of the product which is represented [15]. This is enhanced by the problem that CAD models are commonly not able to carry information beyond the pure geometrical information [14].

Beyond the references between different features, parameters can be used to govern dimensions or other feature properties [27]. In the case of Siemens NX® (NX), these parameters are called expressions, and both terms are used interchangeably throughout this publication.

Building on the concept of *features*, several CAD systems, among them NX, have introduced what are known as user defined features (UDFs). UDF enable the collection of several features and objects into one editable container [19]. The UDF can be configured using parameters (expressions) and be re-used via pre-defined interfaces. This enables the re-use of self-contained modular modelling units, which can be placed in different product concepts [7]. A UDF can contain any number of features, interfaces and expressions and therefore be of arbitrary levels of complexity. However, while King [19] claims UDF are created 'in a Design Intent fashion', it is again only the design intent of the CAD model, rather than the function-based design rationale. Design intent in this publication refers to the *logic of how a CAD model is set up*, while design rationale refers how a product *fulfils its function*.

2.2 Product Function

A product's geometry – or structure, form –, is the result of a cognitive process by the designer, also called 'synthesis' in design theory (see process 2 in the function-behaviour-structure (FBS) framework in Figure 1)[13]. This structure is based on the *expected behaviour* as envisioned by the designer, which is formulated (process 1 in Figure 1) as *function*.

As a corollary, each geometric object can be associated with a design decision, or design solution and subsequently its function. However, the inverse is not possible – there are design solutions which do not (directly) impact geometry, such as material choices. Gero defines function as 'the intended behaviour' of the

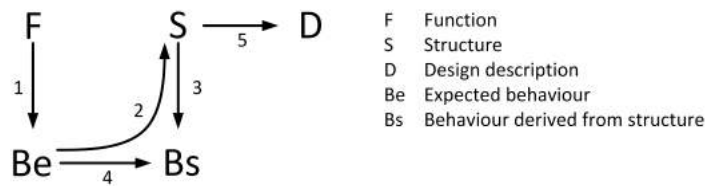


Figure 1: FBS framework, based on [13]. The different processes are as follows: 1 Formulation – 2 Synthesis – 3 Analysis – 4 Evaluation – 5 Documentation.

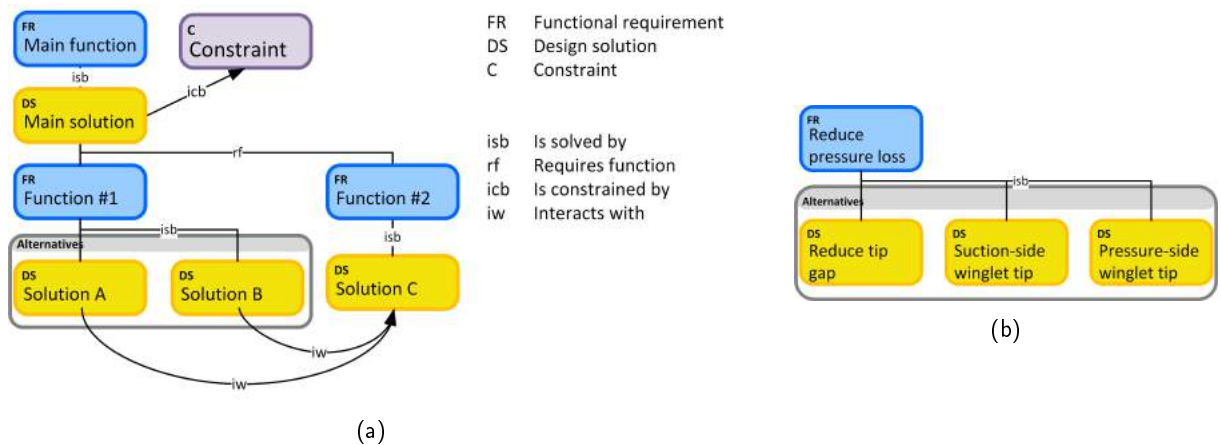


Figure 2: EF-M modelling. (a) basic elements, adapted in colour-scheme and reduced to the elements used in the presented approach from [30]; (b) example for alternative DS in the design of a turbine

product [13]. However, to be able to capture the rationale of all design solutions, the concept of *function* needs to cover product properties such as aesthetics, manufacturing considerations or compliance with regulations. The relations between function, behaviour and design are captured in *function models*. The structure, that is the product's physical elements, however, is generally represented in another modelling domain than the function – commonly the CAD model (see above, Section 2.1).

2.3 Enhanced Function-means Modelling

The use of function modelling supports DSE by enabling the capture of many new solutions and sub-solutions for identified functions on different levels of product abstraction, generating variants in both a modular as well as parametric bandwidth [20] and providing an initial analysis of the concept [11]. Therefore, while providing developers with an understanding of a product's design rationale as well as effortless means to model new design solutions, function modelling lacks a means of analysing (Process 4 in Figure 1) whether a design performs as desired. Since the product's behaviour is derived from its structure, product geometry models are necessary. However, a majority of function modelling theories treat the geometric models as peripheral results of the design process, as is exemplary in the FBS framework in Figure 1. As a result, the connection of the two modelling domains is a goal of engineering design research [6, 34, 37].

One function modelling approach which models this connection between functions and solutions is EF-M

modelling. The basic modelling elements of EF-M based on [30] are presented in Figure 2a. Building on Hubka's law [16], where a product's functions and solutions are sorted hierarchically, an EF-M model alternates between functional requirement (FR), the design solution (DS) for this requirement and the therefrom resulting sub-FR.

A DS represents a *solution principle* which fulfils the respective function. Such a solution can be a structural element such as a bolt, a combustion engine or a chamfer but it is not constrained to geometrical elements. Other examples are software, physical working principles or materials, the re-use of existing elements or the *reduction* of structure (that is the removal of other elements).

An example of a geometrical DS without a body is the tip-clearance between the rotor and stator in a turbine. The width of the gap, a result of a design decision, has a direct impact on the performance of the turbine [38]. This gap – or more precisely the width of it – can be modelled as a DS in the EF-M model for the FR 'reduce pressure loss in fan'. Alternative DS can represent different gap widths or further, different wing-tip geometries [41]. In this example, the DS using different wing tip geometries directly relate to geometric entities, whereas an alteration in the wing tip gap width relates to one (or several) parameter(s). The alternative solutions modelled in EF-M are presented in Figure 2b.

Through the capture of alternative DS for a FR, different variants of a concept can be captured in model. The combinatorial combinations of different sub-solutions are then *instantiated* into variant concepts. Each concept fulfils the same top-level functions, but uses different solution principles on different hierarchy levels.

Another core mechanic of the EF-M framework is the use of constraints (Cs). These are used for example to delimit the design space when exploring novel alternative solutions. Constraints can act either upon the design parameter (DP) of a design rationale, or behaviour parameters of a DS. In this function, they enable a screening of potential solutions for feasibility or requirements conformity. This makes the EF-M framework a suitable candidate for a DSE modelling approach. However, in this presentation of the approach, C are omitted since they do not contribute to the direct connection between geometry and function.

3 RESEARCH METHOD

This publication presents an approach to generate multiple variants of a product concept, based on a legacy design. The approach was developed in the progress of a research project in collaboration with an industrial partner in the aerospace field. Therefore, the approach is a result of multiple previous studies, where practitioners were observed, interviewed and their behaviour analysed in workshops. Based on these results, the approach was developed as an object model based on existing research on EF-M, then implemented in a software prototype tool. The tool was realised using Python as a scripting language, the Python package Django for database management and the web-interface, and the Siemens NXOpen application programming interface (API) for Python *Journaling*.

The approach was tested with a simplified product based on real development cases at the industrial project partner. The product is a turbine rear structure (TRS), a static component of aircraft engines. All product data such as functions or dimensions are similar to real development cases. However, to protect the companies intellectual property they were slightly altered.

4 PROPOSED APPROACH: OBJECT MODEL FOR GEOMETRY AND FUNCTION

As stated above, the geometry of a product is the result of a cognitive process to generate an artefact exerting a specific function. Therefore, each geometric element can be associated with its respective function.

The approach builds on the identified functionality resulting from a functional decomposition, as described in [4]. This functional decomposition presents the design rationale, and a way to capture alternative solutions (in the form of DS) for specific functions (as FR). On the geometrical side, the approach makes use of customisable geometrical modules in the form of UDF as provided by the CAD software Siemens NX.

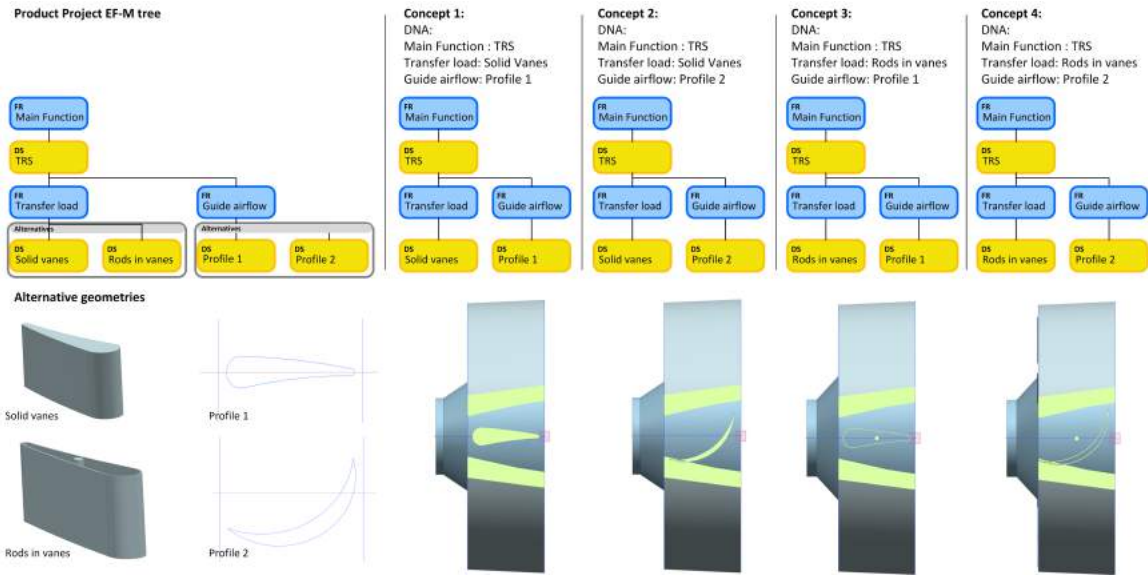


Figure 4: Instantiation of a product project with alternative DS in EF-M and the respective DNA of each concept, with respective embodiments – UDF for each DS, as well as fully assembled concepts.

alternative DS, which leads to a total of $2^2 = 4$ concepts. Each concept with its respective DNA is illustrated to the right. As a result, each concept can be identified and generated by using only the DNA.

This setup allows for the modelling of a holistic EF-M tree for the entire product, which contains all possible alternatives on different levels of the tree. The project model can then be instantiated into individual concepts, each based on the same original modelling data but individual in its configuration of DS and FR.

In the functional decomposition of the legacy object, the FM is generated in a bottom-up fashion by identifying the function of geometrical features, and describing them in DS. Their function is captured in the same process, and the DS can be linked to the respective FR object.

The product's top-level of functions is derived from the product's requirements and stakeholder needs. The first identified lowest-level FR and DS are then collected in overarching functions and linked to the top-level. For a more detailed description of functional decomposition of a legacy product, see [4]. New DS can now be added on any level of the tree structure. Those new DS are linked to the parent-FR via a CC object, which later on handles the instantiation via DNA. In case of a green-field design process, only the top-level FR are derived from the product's requirements and needs, and all subsequent DS and FR are developed through design methods.

4.2 Object Model of Design Automation Approach

The object model of the geometrical side of the OMFG is largely based on the definition of the UDF as found in NX. UDF are a container for geometrical objects of all kinds. For this approach, they are generated via the wizard function in NX. Figure 5 shows the summary of a UDF generation process in NX. A UDF is a collection of CAD objects and rules, captured in a special CAD part file. The UDF used as an example in this screenshot contains two geometrical elements, a datum plane and a sketch (in the second highlight box).

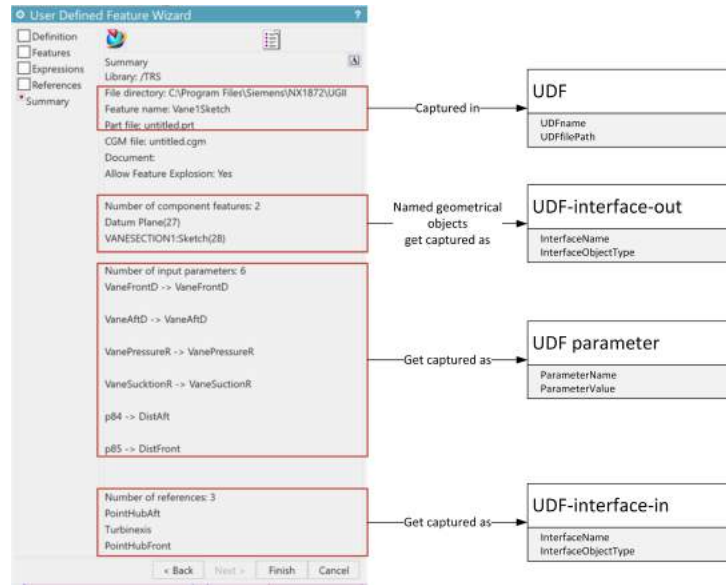


Figure 5: Siemens NX UDF 'wizard' summary interface and how certain elements are captured in the OMFG

Other UDF used in the example in Section 5 contain a wider variety of geometrical objects, such as sketches, extrusions, patterns or other boolean operations. Furthermore, it has six input parameters (third box) and three geometrical references (bottom box). The references are needed for the placement of a UDF in a part – each reference needs to be matched with an existing geometrical object of the same type.

In the OMFG, the UDF name and part-file (top box in Figure 5) are captured in a *UDF* object. The 'input parameters' are captured as *UDF parameter* objects and associated with their respective UDF parent object. For each geometrical reference, a *UDF interface in* object is generated and associated with the UDF object. The geometrical features are not individually captured in the OMFG, but they are scanned for named geometrical objects for which a *UDF interface out* object is created and associated with the parent UDF object.

To be able to place a UDF in a part, that is to assemble the concept in the instantiation phase from all the specific DS's UDF, the assembly algorithm needs to match each of the UDF's reference, that is *UDFinterfaceIn* object, with a fitting geometrical object. To enable this, the user has to match each *UDFinterfaceIn* object with a fitting *UDFinterfaceOut* object. This process has to be performed only for one concept, for example placing the UDF into the legacy concept. As a result of this, a *InterfaceMatch* object is created. It provides an object name, for which the the assembly algorithm finds the matching geometrical object, and feeds it to the NX UDF as a reference. This means that for all *UDFinterfaceIn* objects respective named elements in other UDF need to be created to have the respective *UDFinterfaceOut* objects available in the OMFG.

Since the relation *UDFinterfaceIn* to *InterfaceMatch* is a 1 : 1 relation, the assembly would not work in all concepts: if the respective *UDFinterfaceOut* is not in this concept, the assembly would fail. To alleviate this, the UDF of the alternative DS can provide an *AlternativeMatch*. If the UDF related to the *InterfaceMatch* object is not in the assembly, the assembly falls back on the interface provided in the *AlternativeMatch*.

This demand driven approach enables an efficient replacing of UDF objects: a UDF of a novel, alternative DS only needs to be placed into the OMFG and its interfaces connected once.

A more detailed example of the interface matching is presented in Section 5.2.

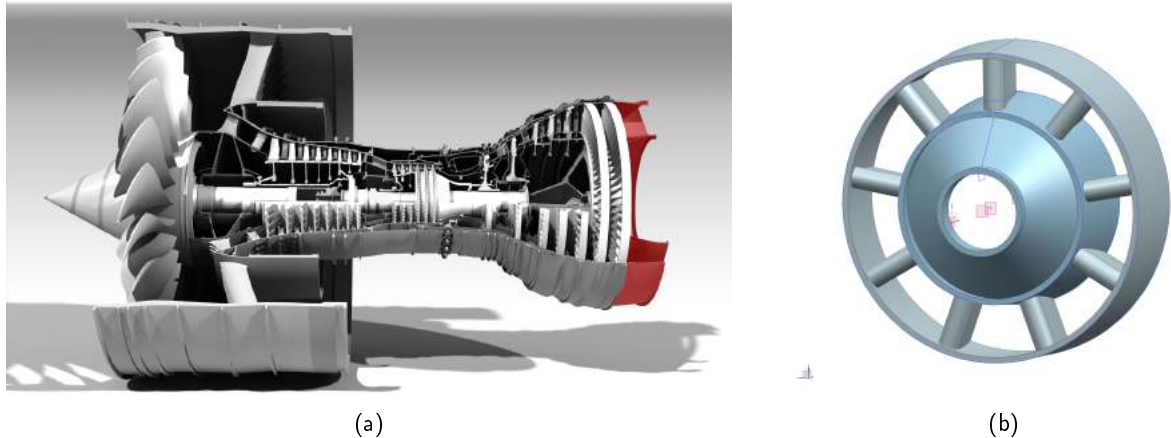


Figure 6: (a) CAD model (Modelled by Chris Shakal for grabcad.com) of a Trent 9000 turbofan engine with the TRS highlighted in red. (b) CAD model of the TRS legacy design as used in this study.

5 DEMONSTRATION OF APPROACH WITH AN ENGINEERING DESIGN CASE

To demonstrate the workings of the OMFG presented above, an exploration of alternative design options was conducted. The demonstration was performed using a simplified TRS, a static element positioned in the aft section of a civil aircraft engine, as shown in Figure 6a. The legacy design, presented in Figure 6b, as well as the functional decomposition of it, was based on case studies performed in collaboration with an industry partner, but simplified in geometry and functionality to better serve the presentation of the approach. The novel alternative DS are based on actual innovative ideas from workshops with practitioners in the case company.

5.1 Decomposition and Modularisation

In the first step, the legacy model of the TRS was decomposed into a function model. The EF-M model, shown in Figure 7, covers the main functions to 'Provide structural connection for turbine on pylon', due to the TRS providing the rear mounting points for the entire turbine, and 'Deswirl the airflow', for the TRS being the last aerodynamic component in the turbine's gaspath. Furthermore, a manufacturing relevant function, 'Assemble TRS' is present. The different DS are equipped with DP, representing relevant dimensions in the CAD model.

The geometrical modularisation is presented here with the example of the UDF 'Vane1Sketch.prt', as shown in Figure 9. The UDF is associated to the DS 'Aerodynamic optimised vanes' of the legacy design. In the legacy CAD model, the sketch of the vane outline determines the vane's shape and therefore fulfills the solution to the FR 'Deswirl airflow'. As can be seen in Figure 9, the UDF requires the references 'hubFrontPoint', 'hubAftPoint' and 'turbineAxis' to be placed. As a result, the respective UDFinterfaceIn objects (red) were created, and associated with the UDF. Based on the parameterisation of the UDF, a UDFparameter object was created for all seven parameters and associated with the UDF. Lastly, the named geometrical elements in the part were scanned and captured as UDFinterfaceOut (green) objects. This UDF now represents all the geometry elements needed to realise the DS 'aerodynamic optimised vanes'. It can be used in the assembly of the full TRS, wherever a vane geometry needs to be constructed – as is the idea of UDF [19]. However, beyond the geometrical reusability, it is now also linked to an EF-M object representing its function. Furthermore, since the interface objects are captured in the OMFG, it can be automatically reused or replaced as is illustrated

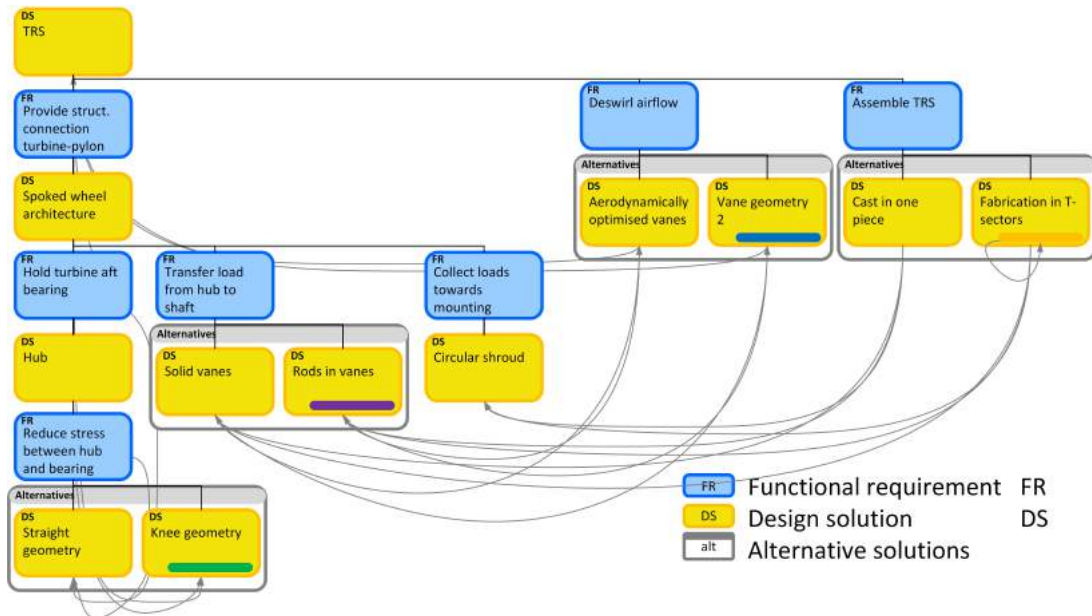


Figure 7: EF-M model of TRS including alternative DS for most FR, and showing iw based on geometrical interfaces. New DS in the grey alternative boxes are marked with an a bar in the color the respective UDF have in Figures 11b and 12.

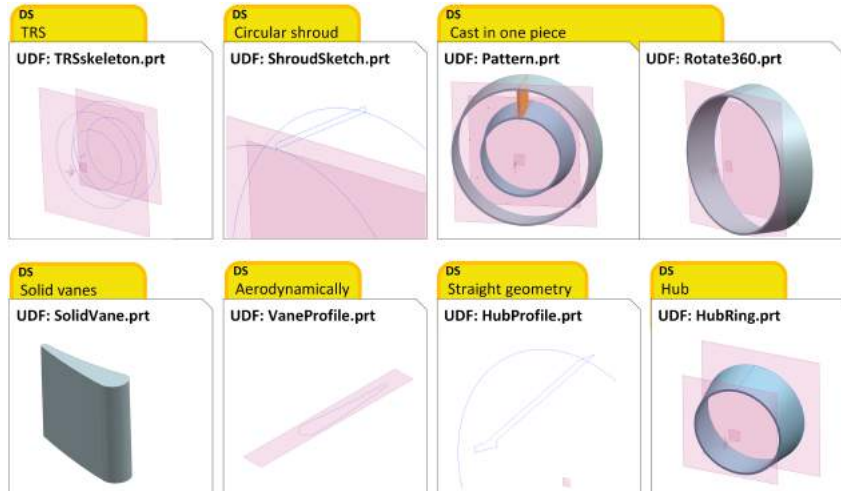


Figure 8: Geometrical decomposition of the legacy design. All UDF and the respective DS. However, since the UDF in some cases dynamically use features of other UDF, the final geometry in the concept may look different.

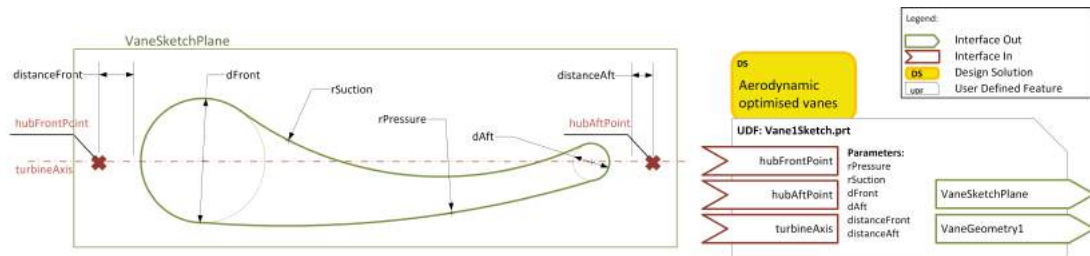


Figure 9: The UDF 'Vane1Sketch.prt', which is associated to the DS 'Aerodynamic optimised vanes in airflow'. The left-hand side shows the sketch which comprises the majority of the UDF, while the right-hand side shows an illustration of the created UDF object, parameters and interfaces. Input interfaces are drawn in red in the sketch and object-illustration, output-interfaces in green.

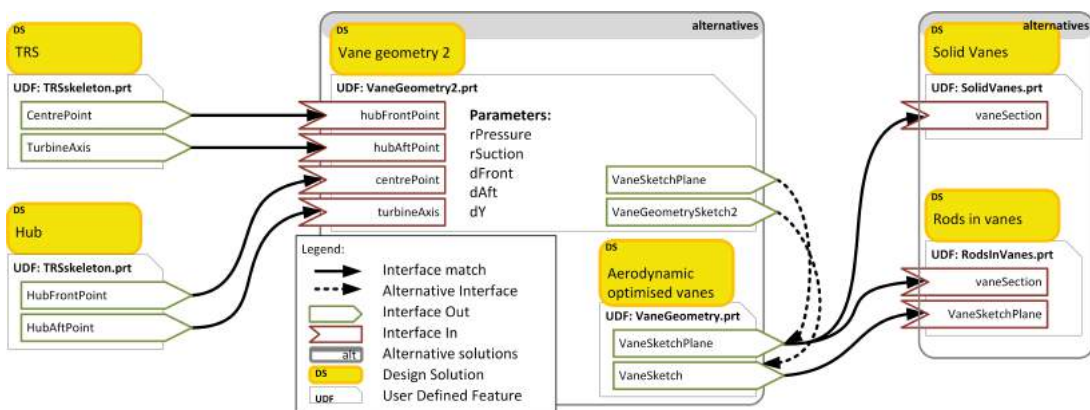


Figure 10: Illustration of the objects around the DS 'Vane geometry 2'. The left-hand side shows UDF and the owning DS objects which provide input interfaces for the UDF 'VaneGeometry2.prt', while the right-hand side shows the UDF and respective DS which make use of the output interfaces. The arrows show the interface matches, while the dotted arrow shows the link for 'alternative interface'. The grey areas capture DS which are alternative solutions for one FR.

below in Section 5.3.

Screenshots of all UDF of the legacy design as well as the respective DS are shown in Figure 8

After the OMFG automatically generated the UDF and interface objects from the UDF part, the interfaces had to be connected manually. All UDFInterfaceIn of the UDF needed to be connected to UDFInterfaceOut of other UDF. This was done via a web-interface which is used to maintain the OMFG.

This process was performed for all DS and geometric elements.

5.2 Redesign, Embodiment and Interface Matching

Based on the functions in the EF-M model in the form of FR, alternative DS are developed. The alternative solutions, for example Vane geometry 2, are then modelled as new DS associated to the respective FR.

In total, 4 different new DS for 4 FR are modelled in this example, leaving 4 CC with 2 alternatives each. The alternative DS are illustrated in Figure 7 in grey boxes. All new DS are captured in the OMFG in the

same way as in the decomposition stage. In the redesign phase, new DS for some of the FR are devised. As can be seen in the grey 'alternative' boxes in figure 7, these are

- *Knee geometry*: a bent hub cross-section to reduce the stress in thermal load cases
- *Rods in vanes*: instead of using the entire vane body as a structural element, the load is carried by a single rod
- *Vane geometry 2*: a different geometry for the vanes (exaggerated in the geometry, as can be seen in Figure 4
- *Fabrication in T-sectors*: each vane and the respective shroud segment are cast individually, and then welded together

In this redesign phase, the UDFinterfaceOut need to be linked in a special way to be able for the rest of the model to be able to be assembled. In the case of an alternative DS, the new UDFs need to be able to provide the same interfaces as the legacy UDFs. Therefore, UDFinterfaceOut of the new UDF are linked as *alternativeMatch* towards the legacy interfaces in the same CC. This is shown in Figure 10, where the interfaceOut of the UDF of the new DS 'Vane geometry 2' are linked as *alternativeMatch* (dotted line) to the UDFinterfaceOut of the legacy DS 'Aerodynamic optimised vanes'. The new UDFinterfaceOut objects need to have the same object-type, or an object type which can be converted to the same, as the original interfaces, that is for an interface of type section the alternative interface has to be either a section, too or a sketch object.

When generating the concept geometry, when the UDF 'SolidVanes.prt' of the DS needs to be placed, the assembly algorithm searches via the interfaceMatch object for the UDFinterfaceOut 'VaneSketch' in the UDF 'VaneGeometry.prt'. If this UDF cannot be found, the assembly algorithm searches for an *alternativeMatch* object that is linked to a UDF in the same concept.

As a benefit of this approach, a new UDF only needs to be linked into the existing structure once, in two steps: first to be placed itself via its UDFinterfaceIn, and second with *alternative* interfaces towards existing InterfaceMatch objects.

5.3 Instantiation and Efficiency

As described above, four FR have each two alternative DS, resulting in $4^2 = 16$ alternative concepts. Each concept is generated from the respective UDF as one part file. The differences in the geometry in the concepts are illustrated in Figure 11, each alternative geometry with its own colour. Figure 4 as well shows how different UDF were combined into one concept.. The fully assembled CAD models of all the different instances are illustrated in Figure 12, starting with the legacy design top-left. The modelling approach using UDF and dedicated interfaces proved to be robust enough to generate all 16 concepts automatically. The alternative DS are highlighted in the figure in colour.

To illustrate the efficiency of the presented approach in terms of generating alternative design concepts compared to a manual generation, a computation of the gains in effort was performed. The modelling effort for more complex DSE projects was computed by interpolating from the presented case study. The efficiency is measured in *effort* or *interactions*, where one interaction roughly equals a selection in a user interface or the entering of a value into a form. Albeit not a precise metric, it was chosen since it represents the smallest denominator of interactions of a user of both systems. All respective effort values are listed in Table 1.

The number of concepts, n in equation 1, is computed using the number of alternative DS per FR, D_{each} , to the power of F , the number FR with alternative DS. The number of possible concepts grows exponentially with each new FR that has alternative DS. Equation 1 assumes for simplicity's sake that all FR have the same amount of alternative DS, and two scenarios were simulated, one with two and one with three alternative DS per FR. The manual generation and implementation of one alternative solution (e_{change}) into an existing

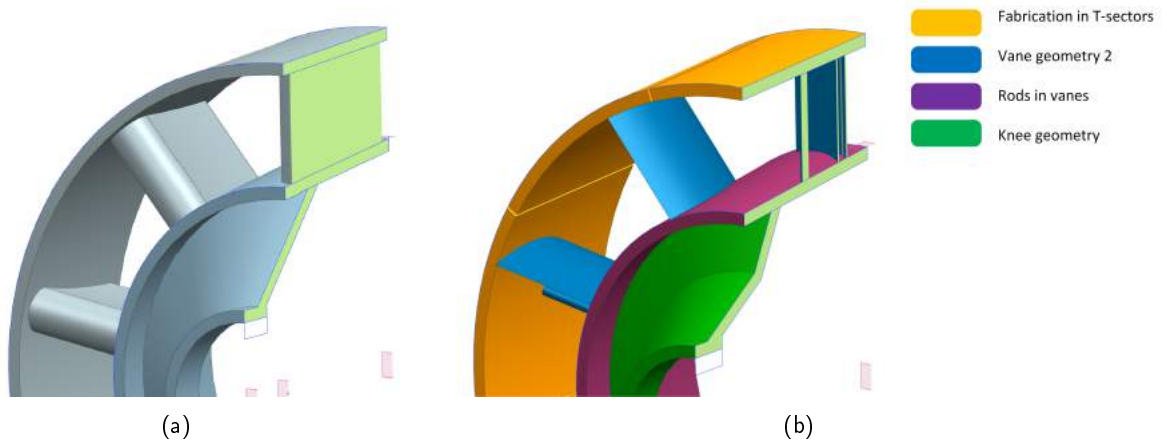


Figure 11: Cross sections through (a) the legacy design, and (b) concept 16 which is composed of only alternative solutions. The alternative solutions are colour coded to be recognised throughout the figures. However, in case of the DS 'Rods in vanes', the hub is coloured purple, since the vanes are already colored respective to the DS 'Vane geometry 2' in blue.



Figure 12: All 16 instances of the alternative solutions for the TRS. The top-left design is the legacy design. All alternative solutions are colour-coded as in Figure 11

model is assumed to average at 50 interactions, and therefore also for the generation of one concept (that is one change per new concept), since the alterations grow combinatorially. This results in a growth of effort linearly to the number of alternative concepts – that is exponentially to the number of FR and DS. The effort to create one UDF geometry is to be assumed the same as a change to an existing model. The implementation of a new UDF into the OMFG, however, is computed with a penalty $e_{interface} = 15$ interactions for the integration into the OMFG, that is the manual setting of interfaces.

Based on these assumptions, the effort for the creation of all possible concepts can be computed using the equations below, both for the manual (e_{manual} , equation 2) and the OMFG approach (e_{omfg} , equation 3):

$$n = D_{each}^F \quad (1)$$

$$e_{manual} = n * e_{change} \quad (2)$$

$$e_{omfg} = e_{setup} + n * (e_{change} + e_{interface}) \quad (3)$$

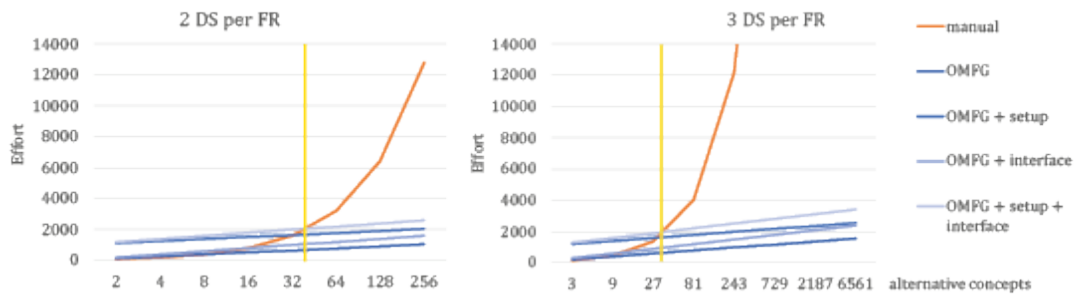


Figure 13: Effort to generate all possible combinatorial concepts, either manually (orange) or using the OMFG approach. Left hand side shows for $D_{each} = 2$, right hand side for $D_{each} = 3$, and break even for $n > 32$ or $n > 27$ concepts respectively with a yellow bar.

where

n = number of combinatorial concepts

D_{each} = number of alternative DS per FR

F = number of FR that have alternative DS

e_{change} = the effort to introduce or create a new DS geometry

$e_{interface}$ = the effort to link the interfaces of a UDF

e_{setup} = the effort of setting up the OMFG

case	e_{change}	$e_{interface}$	e_{setup}
manual	50	0	0
OMFG	50	15	0
OMFG with setup penalty	50	15	1000
OMFG with interface penalty	50	65	0
OMFG with setup and interface penalty	50	65	1000

Table 1: Values for the computation of effort for the generation of alternative concepts using different approaches and penalties.

To account for inaccuracies in the assumptions of effort, the effort for the OMFG has been computed using a penalty for the initial functional decomposition (e_{setup}), and a higher effort for integration as well as a combination of both. The penalty values are listed in detail in Table 1. All efforts are computed relative to the number of concept variants in Figure 13.

The effort to generate all concepts breaks even, under assumption of the highest penalties, after five FR with alternative FR, that is for $2^5 = 32$ different concepts. Using the original assumptions, the OMFG approach already breaks even for more than three FR with alternative DS.

The right hand side of Figure 13 shows the computation for 3 alternative DS per FR. This leads to a higher exponential growth in alternative concepts and therefore a break-even for the additional effort of the OMFG at $3^4 = 81$ different concepts, assuming the highest penalties.

6 DISCUSSION

The presented OMFG approach for the generation of alternative product concepts based on alternative sub-solutions has shown to deliver CAD models of different product concepts based on alternative design solutions from a function model. As such, it contributes to the closure of the gap between models of different abstraction levels, as described by [6, 34, 37] and provides one more DSE method.

The presented approach builds on the introduction of novel sub-solutions into an existing product model. This follows the argumentation of Wyatt et al. [40] that most of applied product development is an iterative process building on top of already defined products. Nonetheless could the described approach also be used in a greenfield design case, starting from the top-level functions and iteratively building up the DS and respective geometries.

By coupling a function modelling framework to a design automating process, the approach follows the recommendations of Lund et al. [21] 'to make datasets that link the parametric models to the parameter data stored in database tables'. EF-M is used as the backbone for function modelling part of OMFG. This choice proved to be beneficial, since EF-M already provides mechanisms for the capture of alternative solutions as well as the instantiation of concepts based on combinatorial sub-solutions. While the use of EF-M for DSE, or other FM frameworks inspired by axiomatic design [33], such as [17], is not novel, the actual object-oriented implementation of the OMFG is a novel contribution. While other publications about EF-M, for example [5], have discussed the potential coupling between functional and geometrical domain, this appears to be the first time such an approach has been implemented in practice.

It could be argued that the generation of a similar bandwidth of alternative concepts can be done using only a parameterised meta-model. While such approaches may be able to generate similar amounts of concepts with potentially less effort, the introduction of new sub-solutions into these meta-models are confronted with the same problems of rigidity as stated in the introduction [15]. Therefore, the use of a function model to map the design space does not only provide a more structured and flexible approach, it also presents opportunities for improvement and development to designers [4]. Furthermore, if the OMFG is extended to use the full ability of EF-M including constraints, it can provide an even more powerful tool for assessment of the individual concepts' feasibility – thereby providing not only DA functionality, but also KBE features such as knowledge capture, storage and representation. The motivation of the presented approach, to provide a holistic product model coupling a concepts form, function and behaviour, is shared with other frameworks such as the Core Product Model (CPM) [12]. However, while the CPM attempts a more generalised product model, for example by capturing material properties, it appears to focus on the *capture* and *storage* of product information, similar to as prescribed by MOKA [32]. The OMFG, however, aims beyond that, to *generate* product information by enabling the capture of new design solutions, and their automated integration into existing product geometry – with the final goal of supporting DSE.

Only based on this assumption that each geometrical element in a product has an identifiable function is the functional decomposition, a core step in the presented approach, actually possible. The idea that a product is created to fulfil a function, and that the design process is guided by it, is a core idea of design theory [26, 33]. Roozenburg and Eekels state further that 'designing is to reason from function to form' [29]. Therefore, while not all design decisions may lead to forms, all form is the result of function-based synthesis. The decomposition aims to revert this process, to uncover the relation between function and form. The possibility of this has been shown in previous studies such as [1, 11]. However, other studies as well as the underlying industrial cooperation for this publication have shown that practitioners have problems applying the abstract concept of *function* [4, 8, 34]. The most issues are too high abstraction, no added value and not being practical. The presented approach attempts to relieve these challenges: The coupling of the abstract concept of function to tangible geometry, while difficult in its creation, can deliver a better understanding of the product's form *and* function. Furthermore, the use in a DSE approach delivers added value, while also a practical use.

The core of the DA partition of the OMFG is the management of interfaces between UDF, which enables the exchange of alternative DS and their geometry. This is handled via the naming of CAD elements in each UDF to designate them as *UDFInterfaceOut*. However, during the test-runs of the prototype tool, this has shown several potential challenges. For one, the CAD software NX occasionally, in a non-reproducible fashion, renamed surfaces with a trailing '_0', which lead to failing interfaces in the assembly process. Second, the naming by users leads to challenges with double-naming and other human-induced errors, as has also been observed by [35]. Further work will lead investigations into alternative ways of identifying and labelling interfaces between geometric modules.

The ease to generate a vast number of design concepts based on alternative sub-solutions is one of the main arguments for the presented approach. However, it can be questioned whether the availability of such a large amount (reaching 6561 as shown in Figure 13) of CAD models actually improves the product development process. While being less resource intense than manual creation, the approach still consumes computational resources in the creation of the concepts' CAD models. Furthermore, the subsequent analysis and evaluation of the concepts is more laborious.

Previous research has shown that a function or architecture model can be used to already analyse and down-select the available concepts before the embodiment stage [11, 24]. Since the presented approach makes use of actual, geometrical data already connected to the function model, such a reduction of the search space based on the FM data has shown feasible in previous DSE studies. Borgue et al. [4] used the C object of the EF-M framework as a delimitation to the design space. The integration of such methods is a goal of further development of the method.

Along with the growing number of alternative solutions, the growing complexity of the OMFG in terms of UDF interfaces and other relations can lead to some concepts failing to assemble. In an optimal case, the failed concepts are technically non-feasible – for example a concept using a combustion engine with a battery as energy source – and the failing assembly algorithm provides a reliable filter for those concepts. However, it is also likely that the mapping of interfaces, or changing elements in the UDF cause by intersections, can lead to failing assemblies even though the concept would be theoretically possible. In the presented sample case, this was avoided through carefully setting up the different interfaces, relations and alternative interface matches. However, this level of fine-grained model tuning is not possible once the model is scaled up in complexity. While the object of the above case study, the TRS, is already a product of a certain complexity [28] (aerodynamic as well as structural requirements, manufacturing constraints and highly integrated design), the approach's application on more complex products with more parts and requirements is to be the subject of further studies.

7 CONCLUSIONS

This publication presents an approach to generate variants of CAD models, based on alternative design solutions captured in an EF-M model. The goal of the approach is to support product developers in the early product development phases to explore a wider portion of the design space. It achieves this by not only taking into account the geometric domain, as common DA approaches do, but additionally extending the modelling into the functional domain.

The approach takes into account that a majority of product development cases are not greenfield development cases, but evolve an existing design. To achieve this, the approach provides a method for the capture of novel design solutions on a design rationale basis, enhanced function-means.

The connection between the different modelling domains is achieved through an object model, the *OMFG*. Using the instantiation of alternative product concepts inherited from the use of EF-M, the approach generates the combinatorial of all available sub-solutions. While the original instantiation algorithm only creates concept models as function models, the novel object model of the approach together with an assembly algorithm

enables the automated generation of CAD models for each product concept.

Therefore, the major contribution of this publication is the object model *OMFG*, which connects the abstract function model of the product with the concrete geometrical product representation in CAD, which is needed for subsequent product analysis. By this, they fulfil a gap that has been presented in literature [2, 6, 10, 22, 29, 34, 37] as hindering both creativity and DSE.

Assuming values in "effort" for the generation of CAD models, variants or UDF, a computation of expected gains in efficiency in the generation of alternative CAD models has been presented. The gains show that as soon as there are more than four functions for which there is one alternative solution, the approach is more efficient in generating all possible variants.

Further work is performed into validating the approach in an actual industrial product development project to establish whether the approach does contribute to applied DSE, and to investigate the scalability of the OMFG. Lastly, the approach is constantly being refined and expanded in functionality, usability and compatibility.

Acknowledgements

This work was carried out at the Wingquist Laboratory within the Area of Advance - Production at the Chalmers University of Technology, Sweden. The research is supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA) through the NFFP7 program in the MEPHISTO project.

ORCID

Jakob R. Müller, <http://orcid.org/0000-0002-2270-6253>

Massimo Panarotto, <http://orcid.org/0000-0001-5216-0944>

Ola Isaksson, <http://orcid.org/0000-0003-0373-3720>

REFERENCES

- [1] Albers, A.; Matthiesen, S.; Thau, S.; Alink, T.: Support of Design Engineering Activity Through C & CM - Temporal Decomposition of Design Problems. In Tools and Methods for Competitive Engineering Conference, vol. Dubrovnik, 295–306, 2008. ISBN 9789051550443.
- [2] Albers, A.; Ohmer, M.; Eckert, C.M.: Engineering design in a different way: cognitive perspective on the contact and channel model approach. Visual and Spatial Reasoning in Design, 22(23), 2004.
- [3] Amadori, K.; Tarkian, M.; Ölvander, J.; Krus, P.: Flexible and robust CAD models for design automation. Advanced Engineering Informatics, 26(2), 180–195, 2012. ISSN 14740346. <http://doi.org/10.1016/j.aei.2012.01.004>.
- [4] Borgue, O.; Müller, J.R.; Panarotto, M.; Isaksson, O.: Function modelling and constraints replacement for additive manufacturing in satellite component design, 2018.
- [5] Claesson, A.: A configurable component framework supporting platform-based product development. Ph.D. thesis, Chalmers University of Technology, Gothenburg, 2006.
- [6] Cohrs, M.; Klimke, S.; Zachmann, G.: Streamlining Function-oriented Development by Consistent Integration of Automotive Function Architectures with CAD Models. Computer-Aided Design and Applications, 11(4), 399–410, 2014. ISSN 16864360. <http://doi.org/10.1080/16864360.2014.881182>.
- [7] Danjou, S.; Lupa, N.; Koehler, P.: Approach for automated product modeling using knowledge-based design features. Computer-Aided Design and Applications, 5(5), 622–629, 2008. ISSN 16864360. <http://doi.org/10.3722/cadaps.2008.622-629>.

- [8] Eckert, C.M.: That which is not form: The practical challenges in using functional concepts in design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 27(03), 217–231, 2013. ISSN 0890-0604. <http://doi.org/10.1017/S089006041300022X>.
- [9] Eckert, C.M.; Albers, A.; Bursac, N.; Xin Chen, H.L.; Clarkson, P.J.; Gladysz, B.; Maier, J.F.; Rachenkova, G.; Shapiro, D.; Wynn, D.C.: Integrated Product and Process Models : Towards an Integrated Framework and Review. *Proceedings of the 20th International Conference on Engineering Design (ICED 15)*, 1(July), 1–10, 2015.
- [10] Eckert, C.M.; Alink, T.; Albers, A.: Issue driven analysis of an existing product at different levels of abstraction. *International Design Conference - Design 2010*, 673–682, 2010.
- [11] Eisenbart, B.; Gericke, K.; Blessing, L.; McAloone, T.C.: A DSM-based framework for integrated function modelling: concept, application and evaluation. *Research in Engineering Design*, 28(1), 25–51, 2017. ISSN 14356066. <http://doi.org/10.1007/s00163-016-0228-1>.
- [12] Fennes, S.J.; Foufou, S.; Bock, C.; Sriram, R.D.: CPM2: A core model for product data. *Journal of Computing and Information Science in Engineering*, 8(1), 0145011–0145016, 2008. ISSN 15309827. <http://doi.org/10.1115/1.2830842>.
- [13] Gero, J.S.; Kannengiesser, U.: The situated function-behaviour-structure framework. *Design Studies*, 25(4), 373–391, 2004. ISSN 0142694X. <http://doi.org/10.1016/j.destud.2003.10.010>.
- [14] Heikkinen, T.; Johansson, J.; Elgh, F.: Review of CAD-model capabilities and restrictions for multidisciplinary use. *Computer-Aided Design and Applications*, 4360(January), 1–11, 2018. ISSN 1686-4360. <http://doi.org/10.1080/16864360.2017.1419639>.
- [15] Hoffmann, C.M.: Constraint-Based Computer-Aided Design. *Journal of Computing and Information Science in Engineering*, 5(3), 182, 2005. ISSN 15309827. <http://doi.org/10.1115/1.1979508>.
- [16] Hubka, V.; Eder, W.E.: *Theory of Technical Systems*. Springer Verlag, 1988. ISBN 978-3-642-52123-2. <http://doi.org/10.1007/978-3-642-52121-8>.
- [17] Jin, Y.; Li, W.: Design Concept Generation: A Hierarchical Coevolutionary Approach. *Journal of Mechanical Design*, 129(10), 1012, 2007. ISSN 10500472. <http://doi.org/10.1115/1.2757190>.
- [18] Kang, E.; Jackson, E.K.; Schulte, W.: An Approach for Effective Design Space Exploration. *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems: 16th Monterey Workshop 2010, Redmond, WA, USA, March 31- April 2, 2010, Revised Selected Papers*, 33–54, 2011. http://doi.org/10.1007/978-3-642-21292-5_{ }3.
- [19] King, L.: Benefits of using User-defined Features to Generate Preliminary Geometry. *Tech. Rep.* October, Rolls Royce plc, 2017. https://www.plm-europe.org/admin/presentations/2017/2006_PLMEurope_24.10.17-17-00_Liam-King_Rolls-Royce_creation_of_turbine_prelim_geometry.pdf.
- [20] Levandowski, C.E.; Jiao, J.R.; Johannesson, H.: A two-stage model of adaptable product platform for engineering-to-order configuration design. *Journal of Engineering Design*, 26(7-9), 220–235, 2015. ISSN 0954-4828. <http://doi.org/10.1080/09544828.2015.1021305>.
- [21] Lund, J.G.; Fife, N.L.; Jensen, C.G.: PLM-based parametrics for design automation and optimization. *Computer-Aided Design and Applications*, 2(1-4), 37–45, 2005. ISSN 16864360. <http://doi.org/10.1080/16864360.2005.10738351>.
- [22] McNeill, T.; Gero, J.S.; Warren, J.: Understanding Conceptual Electronic Design Using Protocol Analysis. *Research in Engineering Design*, 10, 129–140, 1998.
- [23] Mokhtarian, H.; Coatanéa, E.; Paris, H.: Function modeling combined with physics-based reasoning for assessing design options and supporting innovative ideation. *Artificial Intelligence for Engineering Design*,

- Analysis and Manufacturing (AIEDAM), 31(4 (Function Modeling: Benchmark Models, Problems, and Approaches)), 2017. ISSN 14691760. <http://doi.org/10.1017/S0890060417000403>.
- [24] Müller, J.R.; Isaksson, O.; Landahl, J.; Raja, V.; Panarotto, M.; Levandowski, C.E.; Raudberget, D.: Enhanced function-means modeling supporting design space exploration. *Artificial Intelligence for Engineering Design and Manufacturing*, 1(15), 2019. <http://doi.org/https://doi.org/10.1017/S0890060419000271>.
- [25] Ohsuga, S.: Toward intelligent CAD systems. *Computer-Aided Design*, 21(5), 315–337, 1989. ISSN 00104485. [http://doi.org/10.1016/0010-4485\(89\)90039-0](http://doi.org/10.1016/0010-4485(89)90039-0).
- [26] Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.H.: *Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung*. Springer-Verlag, 2003. ISBN 3540003193. <http://doi.org/10.1007/978-1-84628-319-2>.
- [27] Pratt, M.J.; Anderson, B.D.; Ranger, T.: Towards the standardized exchange of parameterized feature-based CAD models. *CAD Computer Aided Design*, 37(12), 1251–1265, 2005. ISSN 00104485. <http://doi.org/10.1016/j.cad.2004.12.005>.
- [28] Raja, V.; Kokkolaras, M.; Isaksson, O.: A simulation-assisted complexity metric for design optimization of integrated architecture aero-engine structures. *Structural and Multidisciplinary Optimization*, 2019.
- [29] Roozenburg, N.F.M.; Eekels, J.: *Product Design: Fundamentals and Methods.*, vol. 2. Wiley Chichester, 1995.
- [30] Schachinger, P.; Johannesson, H.: Computer modelling of design specifications. *Journal of engineering design*, 11(4), 317–329, 2000. ISSN 0954-4828. <http://doi.org/10.1080/0954482001000935>.
- [31] Shea, K.; Aish, R.; Gourtovaia, M.: Towards integrated performance-driven generative design tools. *Automation in Construction*, 14(2 SPEC. ISS.), 253–264, 2005. ISSN 09265805. <http://doi.org/10.1016/j.autcon.2004.07.002>.
- [32] Stokes, M.: *Managing engineering knowledge-MOKA: methodology for knowledge based engineering applications*. Book, Whole. Professional Engineering Publ, London, 2001. ISBN 1860582958;9781860582950;.
- [33] Suh, N.P.: *The principles of design*. Oxford University Press New York, 1990. ISBN 0195043456.
- [34] Tomiyama, T.; Van Beek, T.J.; Cabrera, A.A.A.; Komoto, H.; D'Amelio, V.: Making function modeling practically usable. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 27(3), 301–309, 2013. ISSN 08900604. <http://doi.org/10.1017/S0890060413000309>.
- [35] Ulliana, F.; Léon, J.C.; Palombi, O.; Rousset, M.C.; Faure, F.: Combining 3D Models and Functions through Ontologies to Describe Man-made Products and Virtual Humans: Toward a Common Framework. *Computer-Aided Design and Applications*, 12(2), 166–180, 2015. ISSN 15548740. <http://doi.org/10.1080/16864360.2014.962429>.
- [36] Ulrich, K.T.; Eppinger, S.D.: *Product Design and Development*. Boston, MA: McGraw-Hill/Irwin, 0–27, 2012. <http://doi.org/10.1016/B978-0-7506-8985-4.00002-4>.
- [37] Umeda, Y.; Tomiyama, T.: Functional reasoning in design. *IEEE Expert-Intelligent Systems and their Applications*, 12(2), 42–48, 1997. ISSN 08859000. <http://doi.org/10.1109/64.585103>.
- [38] Venter, S.J.; Kröger, D.G.: The effect of tip clearance on the performance of an axial flow fan. *Energy Conversion and Management*, 33(2), 89–97, 1992. ISSN 01968904. [http://doi.org/10.1016/0196-8904\(92\)90094-D](http://doi.org/10.1016/0196-8904(92)90094-D).
- [39] Verhagen, W.J.C.; Bermell-Garcia, P.; van Dijk, R.E.C.; Curran, R.: A critical review of Knowledge-Based Engineering: An identification of research challenges. *Advanced Engineering Informatics*, 26(1), 5–15, 2012.

- [40] Wyatt, D.F.; Eckert, C.M.; Clarkson, P.J.: Design of Product Architectures in Incrementally Developed Complex Products. In International Conference on Engineering Design ICED 09, 167–178, 2009.
- [41] Zhong, J.; Han, S.; Lu, H.; Kan, X.: Effect of tip geometry and tip clearance on aerodynamic performance of a linear compressor cascade. Chinese Journal of Aeronautics, 26(3), 583–593, 2013. ISSN 10009361. <http://doi.org/10.1016/j.cja.2013.04.020>.