33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

EXTENSIBLE AIRCRAFT FUSELAGE MODEL GENERATION FOR A MULTIDISCIPLINARY, MULTI-FIDELITY CONTEXT

Jan-Niclas Walther[1], Christian Hesse[1], Jörn Biedermann[1] & Björn Nagel[1]

[1] Institute for System Architectures in Aeronautics, German Aerospace Center (DLR),
Hein-Saß-Weg 22, 21129 Hamburg, Germany

Abstract

In this paper, a knowledge-based approach to support analysis model generation is proposed, which aims to improve consistency among the models used by different contributors in collaborative multi-disciplinary design optimization, while also being open for adaptation to novel product architectures. This is accomplished by providing linked design rules in a centralized repository, outlining relationships between different data points in a central product data format such as CPACS (Common Parametric Aircraft Configuration Schema). Using graph-based methods, the rules can be combined and executed to provide new product data, which is required for increasingly detailed analysis model generation, based on previously available information in CPACS. In this way, the method is open for extension to account for novel product architectures or different design paradigms, simply by adding new rules to the rule repository.

The tool FUGA (Fuselage Geometry Assembler), which implements the approach for the fuselage design problem is also presented. Fuselage design includes aspects of preliminary aircraft design, structural design, cabin design, which are closely interlinked. Furthermore, the range of requirements w.r.t. the level of detail of the different analysis models involved is large, ranging from a simple outer mold line and mass points in preliminary design, to detailed 3D models required for industrialization. It is shown how FUGA is applied to provide consistent geometry models tailored to the specific requirements of the different disciplines.

The extensibility of the approach is demonstrated by including an additional set of rules for placing liquid hydrogen (LH2) tanks in the fuselage.

Keywords: Fuselage, KBE, LH2 integration, Multi-model geenration, CPACS

1 Introduction

In order to improve capabilities for aircraft assessment during early design phases, much of the recent research on multi-disciplinary design optimization is aimed towards extending the scope of the optimization to new disciplines, such as manufacturing [1–3] or certification [4, 5]. It has been shown how the complexity of the design problem can be managed in distributed collaborative processes by breaking it down into smaller disciplinary design tasks handled by disciplinary experts [6–8]. Whereas a common data format such as CPACS (Common Parametric Aircraft Configuration Schema) [9–11] is applied for product data exchange along the so-called digital thread in these processes [2, 3], the task of building the actual analysis model is usually left to the disciplinary tool provider [12–14].

Libraries exist to support the model generation by translating key sections of CPACS into CAD geometry [15–17]. However, depending on the respective fidelity levels of the current and the preceding tools in the process, the tool provider will have to augment any additional information they require to build the analysis model, which is not yet provided by the product data model. Taking the fuselage as an example, a structural designer might need to add frame and stringer distributions, while a cabin designer might add exit positions, both of which are mutually dependent. Currently, there is no automatic way to enforce consistency between the respective assumptions, instead requiring the experts to be aware of design choices made outside of their domain. As the number of disciplines increases, this becomes increasingly difficult to accomplish manually.

Another critical aspect is, that disciplinary analysis tool providers are responsible to adapt their initialization tools to changes in architectural requirements, e.g. newly added product components. This creates room for errors or inconsistencies and impairs the flexibility of the overall design process.

To overcome these limitations, a knowledge-based approach is introduced to automate the data initialization task and assert consistency with previously generated product data, which is outlined in section 2. The methodology relies on CPACS as a central product data format and a central repository of design rules to provide relationships between different data points. New product architectures can be taken into account by introducing additional rules. In addition to the augmented CPACS product data set, the method can also provide a CAD geometry model tailored to specific disciplinary requirements, further reducing ambiguities for analysis model generation.

As an example, the sets of design rules implemented in the fuselage design tool FUGA (Fuselage Geometry Assembler) are presented in section 3, which provide design and geometry model generation capabilities e.g. for the fuselage structure and the cabin.

Finally, the capability of the tool to produce consistent geometry models corresponding to the required levels of fidelity of different disciplinary analyses related to the fuselage, is demonstrated for a conventional single aisle configuration in section 4.1. In a second step, the rule set is updated to include liquid hydrogen (LH2) tanks in the fuselage, as proposed by Brewer [18], Silberhorn et al. [19], and Hartmann and Nagel [20]. The resulting design is compared to the conventional baseline in section 4.2.

2 Knowledge-based methodology

The methodology implemented in FUGA is inspired by knowledge-based engineering (KBE) as described e.g. by Chapman and Pinfold [21] and La Rocca [22]. The goal is not only to provide a model generator, which can provide geometry models at several levels of fidelity, but also to assert, that all additional model details are generated from a consistent parametric basis. This can be accomplished using a KBE system. La Rocca lists several fundamental elements of any such system, including

- A *work space* or *data repository*, which contains the product data,
- A *knowledge repository*, which contains design rules for the product,
- An *inference (or reasoning) engine*, which decides which rules to deploy to generate additional required data.

Implementations of these elements are provided in FUGA. To begin with, the available information on the product is stored in a central product data model, described in section 2.1, which takes the role of the data repository. As new information becomes available, the central product data model is updated accordingly.

The second element is the design rule or knowledge repository, introduced in section 2.2. Here, relationships between individual pieces of data are stored as executable software following a functional and declarative paradigm. Rules can be defined for both data from CPACS and non-CPACS data such as auxiliary variables or CAD geometry.

Finally, the design problem is set up based on the information provided by the central product data model and a query for specific pieces of product information made by a user. If the requested information is not available, graph-based techniques are applied to identify a sequence of rules for computing the requested values. This constitutes the inference engine, discussed in section 2.3.

In section 2.4, it is then explained, how the KBE methodology can be leveraged to produce multi-fidelity geometry models.

2.1 Central product data model: CPACS

The Common Parametric Aircraft Configuration Schema (CPACS)[1] has been introduced to provide a common data exchange format for different disciplines, scales and levels of fidelity in aircraft design [10]. Since its introduction, it has been developed continuously and applied in many projects at the German Aerospace Center (DLR) [11].

Many different authors have made contributions to develop CPACS in their respective fields. The earliest example has been published by Liersch and Hepperle [9], who not only introduce the overall structure of CPACS, but also provide a description of the outer mold line (OML) of aircraft bodies e.g. the wing and fuselage. Each body is defined using a sequence of predefined section profiles. Building upon this definition of the OML, Scherer and Kohlgrüber [23] introduce detailed descriptions of fuselage structure components to CPACS. Moreover, Walther et al. [24] present a cabin definition, which allows for the incorporation of detailed component models.

FUGA manages data from CPACS as part of its data repository. However, the contents of the data repository are not limited to entries in CPACS. Instead, any data type in Python can potentially be stored. This includes CAD models via the Open Cascade Technology (OCCT) CAD library [25]. It is also possible to use container formats, such as lists or arrays, which enables efficient vectorized formulations of rules.

2.2 Design rule repository and rule set management

All design rules in FUGA are implemented and stored as Python classes, which inherit from a common base class. According to the functional paradigm, each rule can only provide the value of a single entry of the data repository, which can be accessed via its unique resource identifier (URI). The URI is stored in the `provides` attribute of the class. In contrast, any number of data repository entries can be requested as inputs. The corresponding URIs are stored in a set in the `requires` attribute. The `compute` method accepts a Python-dictionary as input, which contains all data repository entries specified in the `requires` set, and returns the new value of the data repository entry specified by the provides attribute. Beyond this, knowledge engineers are free to adapt the classes as they require.

Since only one return value can be provided per function, the design knowledge base for any given component usually consists of many different rules. Therefore, rules can be grouped into rule sets,

[1] https://cpacs.de

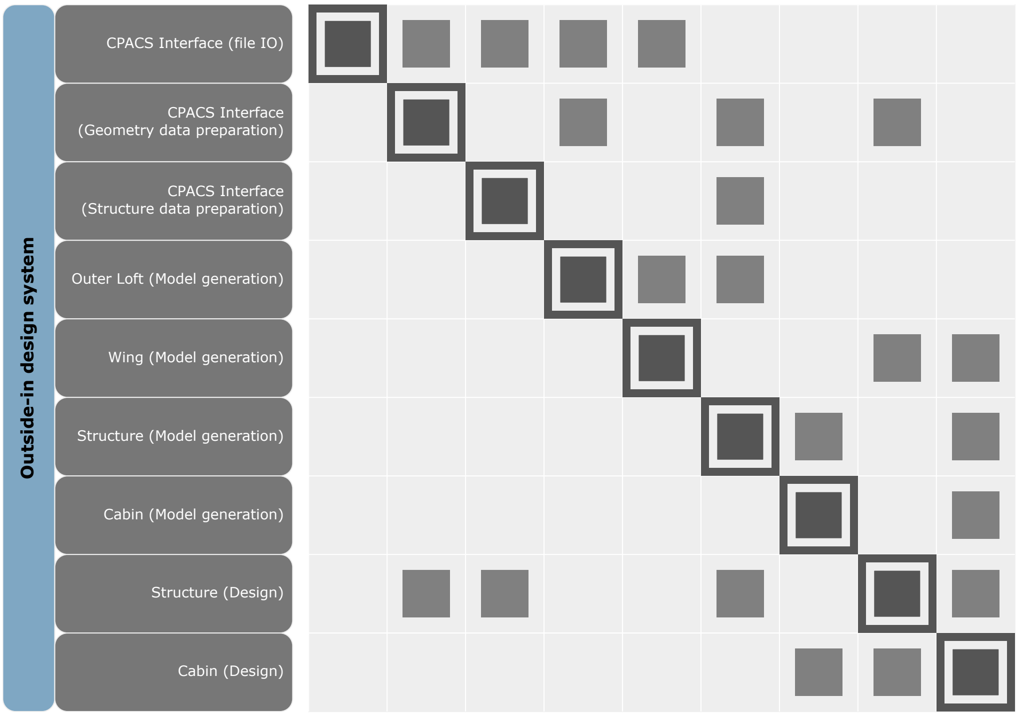


Figure 1 – N² chart showing the rules sets for a basic outside-in design.

to provide a more accessible structure. Integrated problems can be addressed by combining the capabilities of multiple rule sets. By default, FUGA provides design rule sets fort he structure and the cabin layout. Corresponding rule sets for the geometry model generation are also available. However, it is also possible to deploy additional rule sets, to reflect changes in the top level system architecture. Similarly to the method resolution in object-oriented programming, newly added rules will overrule existing ones if they share the same `provides` attribute.

Figure 1 provides an overview of the default rule sets in FUGA, corresponding to a basic outside-in cabin design, in an N² chart, built using the openMDAO framework [26]. Contrary to inside-out design, where the fuselage is designed around the cabin, the cabin is designed to fit a given outer fuselage geometry. Aside from rule sets providing interfaces to CPACS and geometric modeling capabilities for the individual components, the two aforementioned design rule sets are present in the system. They are further discussed in sections 3.2 and 3.3.

The N² chart also provides an understanding of the dependencies between rule sets, marked by the off-diagonal entries. Outgoing connections of the individual rule sets, placed on the main diagonal, are placed in the same row, whereas incoming connections are placed in the same column. It shows, that the inputs for the outer loft and wing model generation are provided by the CPACS file IO component, whereas the inputs for the structure and the cabin are provided by the design rule sets. Since the structure design rule set produces raw CPACS data, not all of its outputs go straight to the structural modeling rule set, but are passed through the data preparation sets first.

Moreover, it is worth noting, that the two design rule sets are mutually dependent, i.e. cabin data is required for the structural design and vice versa.

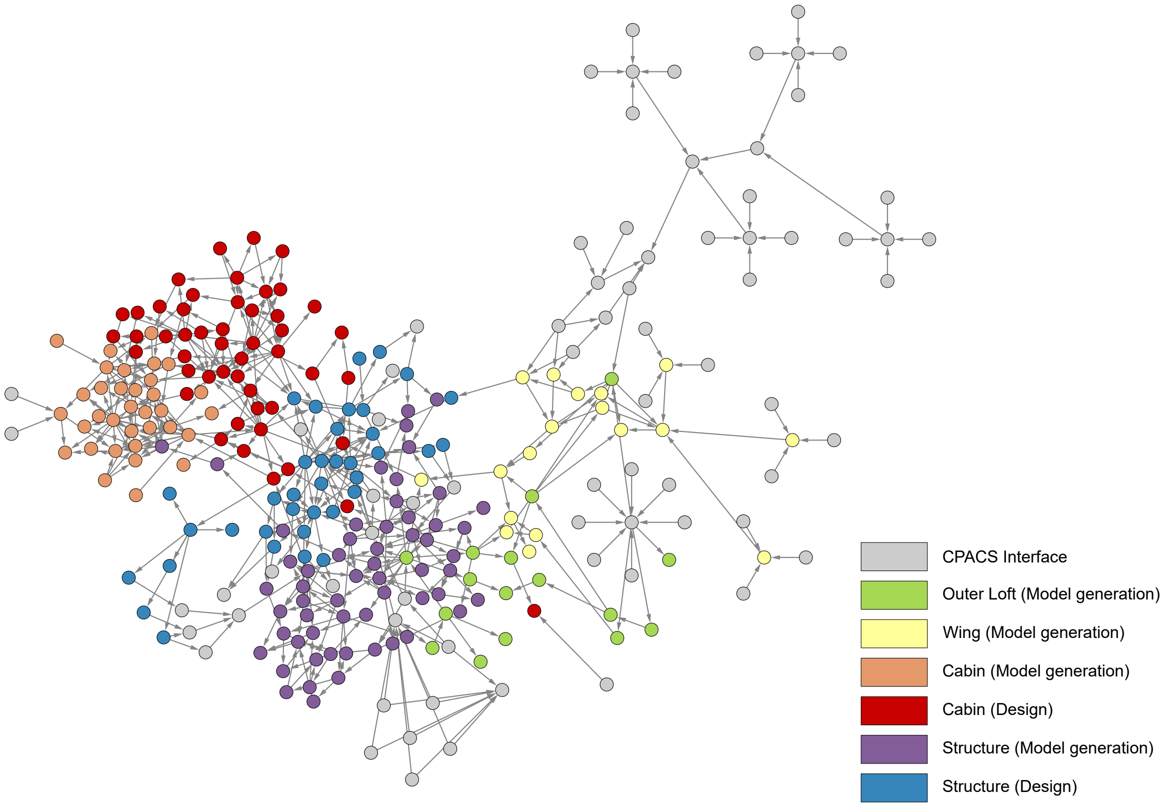


Figure 2 – Maximal connectivity graph of the design and modeling rules by rule sets.

2.3 Graph-based inference engine

The approach to rule representation introduced in 2.2 follows a declarative programming paradigm, where logical relationships are stated without a predefined control flow. It is left to the framework to determine the appropriate execution sequence at runtime. Graph-based techniques have been applied to resolve the dependencies in multi-disciplinary design analysis and optimization workflows as shown e.g. by Pate et al. [27] and Gent [28]. Very similar techniques can be applied for the inference engine in KBE problems. The approach implemented in FUGA is based on the NetworkX Python package [29] and has been described in detail by Walther et al. [17].

The system of rules constituted by the rule sets can be modeled as a directed graph as shown in figure 2, where the nodes represent a rule (or a data repository entry, since only one is provided per rule) and the edges illustrate the data flow between the rules. This graph is the maximal connectivity graph (MCG), which contains all known relationships in the design system.

During system initialization, the data repository is filled with the available information. This can be accomplished e.g. by importing data from a CPACS data set. To generate additional data the design engineer makes a query, for missing entries. The KBE problem is then formulated based on which data is available and which data is requested. To determine the solution, the full problem graph (FPG) is be extracted from the MCG. The root nodes of the FPG, i.e. nodes with no incoming connections, contain the data available in the data repository, whereas the leaf nodes, i.e. the nodes with no outgoing connections are rules, which provide the data requested.

Based on the FPG, an execution sequence for the active rules can be determined, which is referred to as the problem solution graph (PSG). If no cycles are found, i.e. the FPG is a directed acyclic graph, Kahn's topological sorting algorithm [30] will yield a feasible execution sequence. The approach works irrespectively of the available rules, as long as a path can be traced from the known to the requested nodes. As such, extensions of the rule repository via additional rule sets can be taken into account without additional effort.

Another advantage of the graph-based approach is that the information on the descendants of a data repository entry, i.e. to which rules it is a direct or indirect input, is always available. This is valuable, if an entry in the data repository is changed unexpectedly, e.g. through user intervention. In this case, the system will automatically invalidate all of the entry's descendants to avoid inconsistencies in the data.

2.4 Multi-fidelity geometry

The above building blocks, can be utilized to reason on the available information from CPACS and generate additional product data consistently. In this way, design capabilities can be implemented.

In addition, the same methodology can also be used to derive product geometry models from available parametric data in CPACS. This task is commonly performed using the TiGL Geometry Library for CPACS [16, 31], which can not only generate CAD representations of the outer geometry, but also some additional details such as wing structures and movables. It is implemented using the OCCT CAD library. Furthermore, TiGL provides the TiGL Viewer, a popular tool for visualizing CPACS files. However, TiGL does not provide modeling capabilities for the fuselage interior descriptions in CPACS. Furthermore, only one geometric representation at a specific level of detail for a given component in CPACS is available. Thus, if more or fewer details are required to build a given disciplinary analysis model, the geometry must be manipulated by the responsible engineer, which increases the risk of inconsistencies in distributed processes.

The concept, that different analyses require different abstractions of the product geometry is referred to as multi-fidelity [32, 33]. Several authors have proposed so-called Multi-Model Generators (MMG) to cater to multi-fidelity requirements [22, 34].

To support MMG capabilities, the geometric model generation in FUGA based on CPACS is formulated in a way similar to a feature tree known from CAD applications, on top of the OCCT library. A feature tree links geometric shapes and operations, such as Boolean operations, extrusions, etc., to create new components and details based on previously known shapes. This allows for the creation of more detailed geometry, such as a wing with wingbox and movables, based on less detailed geometry, e.g. the wing outer shape, and additional information, e.g. wingbox and movable definition volumes.

With this approach, geometry can be requested from any point along the feature tree. The inference engine in FUGA will detect and compute the required input geometries, and omit unnecessary details, which have not been requested. This allows for very efficient geometry model generation. Another benefit of this formulation is the possibility to connect design and model generation capabilities. This enables the ad-hoc generation of additional design details, which are missing from CPACS as they are required for the model generation.

3 Fuselage and cabin design and model generation rule sets

The design rules are a crucial element of any KBE system. In order to manage the complexity, the design rules in FUGA are grouped into rule sets roughly corresponding to design tasks i.e. preliminary design, structural design, cabin layout and hydrogen tank integration. An introduction of the

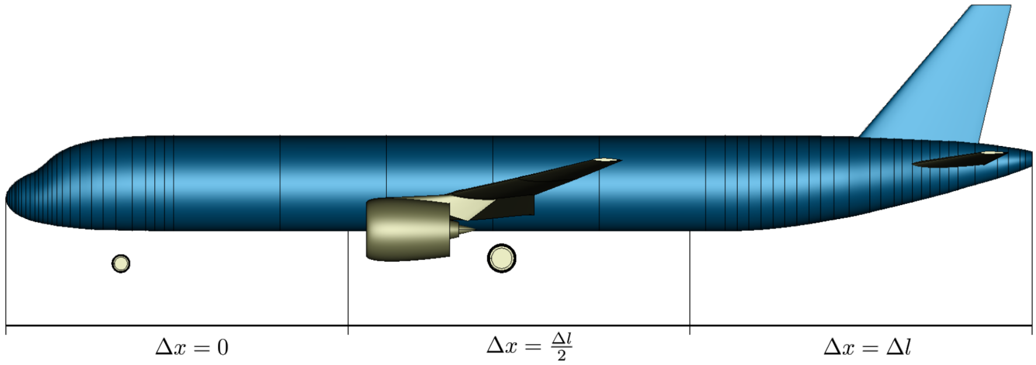
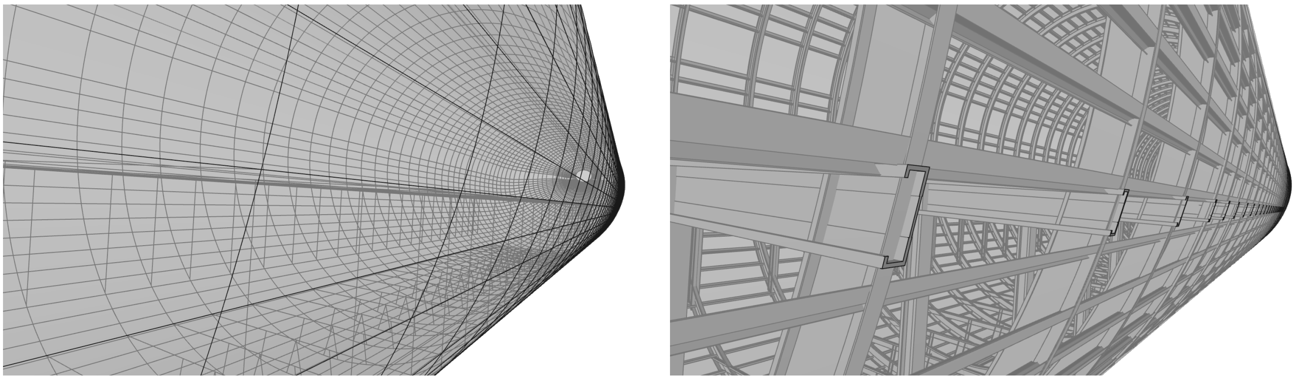


Figure 3 – Distribution of profile and component offsets Δx.

knowledge covered by the different rule sets is provided in the following. Both design relationships, which provide the means to compute unknown product data based on the known data, and geometric modeling rules, which are required to assemble CAD geometry from the parametric data stored in CPACS are included.

3.1 Fuselage outer mold line manipulation rule set

Different approaches to describe the outer geometry of a passenger aircraft fuselage have been described. Depending on the specific design task, either a clean sheet design or a modification of an existing design, e.g. for family concepts, which share the same cross-section, can be performed.
Many authors show clean sheet geometric design approaches, leveraging CAD modeling [16, 35, 36]. A common trait of all these approaches is, that the fuselage surfaces are built from a series of cross-section profiles combined with several guide curves, usually the top and the bottom and the side curve of the fuselage. A clean sheet design offers the freedom, to manipulate both the cross section e.g. according the number of seats abreast in the cabin, and the lengths of the different segments.
Even so, in many cases, some aspects of the design may already be prescribed, e.g. the cross-section for family concepts. The fuselage geometry design task may therefore be reduced to the task of finding the necessary fuselage length. This can be accomplished by modifying only the length of the cylindrical section for an existing design provided by a preliminary design synthesizer such as openAD [37]. Due to its simplicity and the easy implementation in CPACS, the approach has been selected for the tank integration task in this study.
To apply the desired length difference Δl, an offset Δx must be applied to each fuselage profile in CPACS. As illustrated by figure 3, this is accomplished by dividing the fuselage into three sections. While the forward section is not modified, the rear section, including the empennage, is offset by the length difference Δl. Applying constant offsets across the sections has the advantage, that the shape of the non-cylindrical sections remains unchanged, which is in line with a length increase of the cylindrical section. In the middle section, which also contains the wing, the engines and the main landing gear, an offset of $\frac{\Delta l}{2}$ is applied, to retain a credible wing position.
This distribution is only an estimate and the final result should be validated by re-running the preliminary design synthesizer with the updated fuselage assumptions. However, it is considered sufficiently accurate to illustrate the subsequent integration problems within the scope of this study.



(a) Curve-based (b) Volumetric

Figure 4 – Multi-fidelity representations of profile-based structural elements.

3.2 Structural layout and geometry generation rule set

Based on the OML, several elements of the fuselage structure must be generated, due to their significance to the cabin layout generation. Key structural elements include the frames, the floor structure and the pressure bulkheads. A set of rules, which can be applied to create a structural layout for CPACS, has been described by Walther and Ciampa [38] and is included in FUGA.

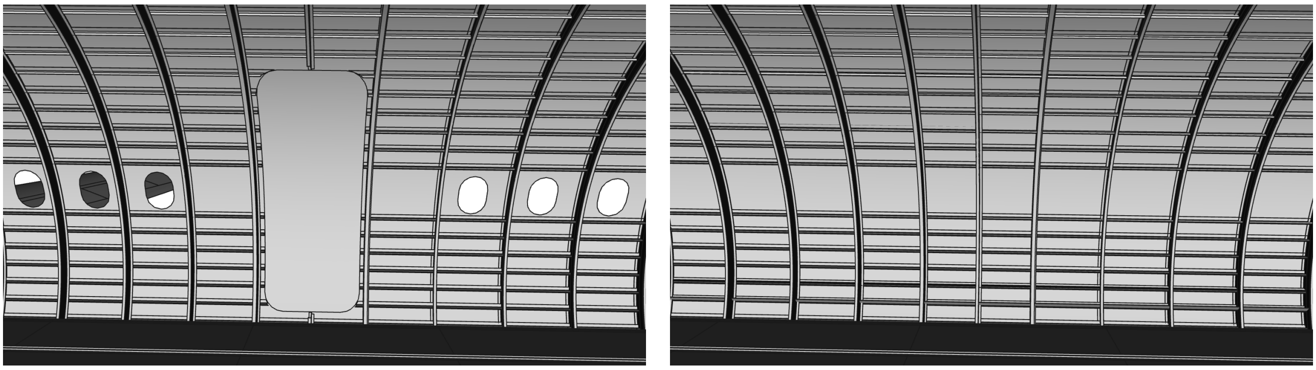
To begin with, the positions of the pressure bulkheads along the longitudinal axis of the fuselage are determined. A common passenger transport aircraft has pressure bulkheads at the front and at the rear. The positions are described in terms of offsets from the fuselage end points. If structural information for a vertical tail plane (VTP) is available, the rear bulkhead can also be placed at the VTP front spar.

The bulkhead positions are part of a set of main frame positions, which also include wing attachments and door frames. Regular frames are distributed between the main frames based on a nominal frame pitch. The default rule is to place frames at equal distance between two main frames. The number of frames is selected so that the actual frame pitch is less than or equal to the nominal frame pitch.

The floor is determined by a given height for the cabin floor, specified by a z-coordinate. The floor structure consists of crossbeams, longitudinal beams, support struts and floor panels. A crossbeam is placed at each frame position, at an offset prescribed by the longitudinal beam profile height. Support struts may be added at each crossbeam at a given lateral position based on the designated cargo container size.

Aside from the design, the model generation is also implemented in FUGA. Using the knowledge-based approach, the level of detail of the geometry can be tailored to requirements due to different disciplines or levels of fidelity. An example of this is the representation of extruded structural stiffeners such as frames or stringers, which are defined in CPACS as an extrusion of a cross-section profile along a curve as described by Scherer and Kohlgrüber [23]

In global finite element models, used for structural analysis and sizing, the stiffeners are modeled as beam elements. In this case, the profile is not modeled geometrically, but taken into account via beam properties instead. As a result, it is not necessary to compute the extrusion of the profile along the curve explicitly. The corresponding geometry model is shown in figure 4a. For higher fidelity analysis or detailed visualization, it may however be desirable to consider the cross-section in more detail using a volumetric representation of the components. In this case, the extrusion can not be avoided, as displayed in figure 4b. For intermediate cases the extrusion of the profiles is also



(a) Cutouts on (b) Cutouts off

Figure 5 – Multi-fidelity representations of structural cutouts.

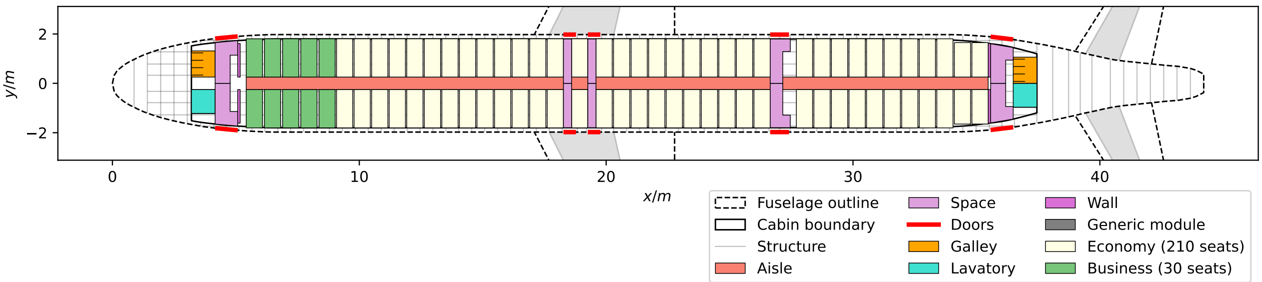


Figure 6 – Example LOPA for a single aisle configuration.

possible without explicitly modeled wall thickness, which is useful e.g. for shell representations in detailed FEM models.

Explicitly modeled structural cutouts, as displayed in figure 5a, are another example for optional details. They are defined using Boolean operations, which can become expensive to compute. Similarly to the extruded profiles, explicitly modeled cutouts are not required e.g. for global finite element models. Therefore, the computation of the cutouts can also be omitted as shown in figure 5b.

3.3 Cabin layout generation rule set

Some structural details, such as the placement of the door cutouts, and consequently the frame distribution, can not be determined reliably without knowledge of the cabin layout. Furthermore, accurate knowledge of the cabin is useful to determine the required length of the fuselage. Walther et al. [39] introduce an approach to generate cabin details based on external component models, which has been implemented in FUGA.

A primary result of the cabin generation is the layout of passenger accommodation (LOPA) as given in figure 6. It contains the distribution of the seats and the monuments, such as galleys and lavatories. In addition, the spaces allocated for the aisle and the passageways and assist spaces at the exits are shown. The description of the cabin arrangement as shown is fully supported by CPACS [24].

The number of passengers, the seat pitch, the types of seats and number of seats abreast for each class are among the necessary inputs for the cabin generation. Furthermore, the number and widths of the aisles must be provided. The exit types can either be prescribed or computed automatically based on the number of passengers using an optimization algorithm [40]. The longitudinal positioning can be defined using either passenger percentages or explicit seat block sizes. The corresponding

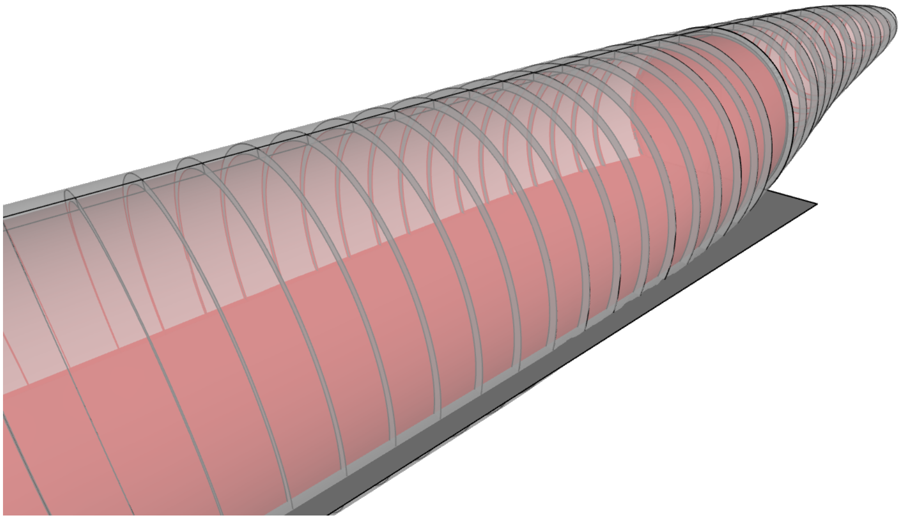


Figure 7 – Cabin space determination based on frame heights.

passageways and assist spaces are designed according to certification requirements [41]. Monuments are placed at predefined slots relative to the exit passageways.

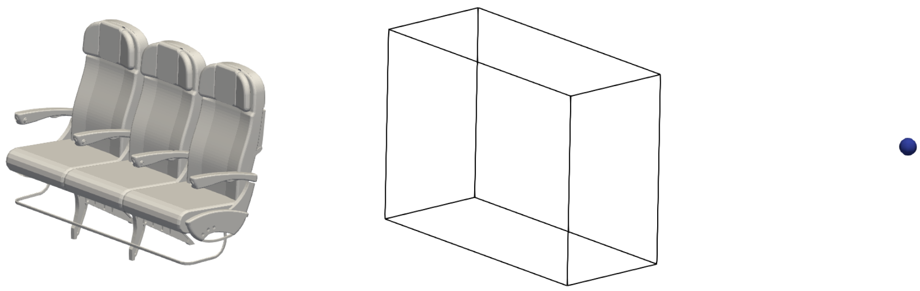
Aside from the elements of the LOPA, rules for the placement of the secondary structure, including sidewall panels, overhead storage bins, and ceiling panels are also provided. The placement and sizing of these components is performed based on the frame positions and the available cabin space. The space is constructed from the volume circumscribed by the tips of the frames, as illustrated by figure 7, which is intersected with a box representing the space limits due to the floor height, the cockpit rear wall and the rear pressure bulkhead.

Similarly to the structure, a variety of geometry models can be derived from the cabin data, depending on disciplinary analysis requirements. The CPACS definition allows for triangulated models of cabin components, like the one shown in figure 8a, as inputs, which can provide an arbitrarily high level of detail. For many applications, such as space allocation analysis, the high amount of detail is, however, not needed and unnecessarily reduces performance. Instead, a simplified bounding box representation can be derived from the model, as displayed in figure 8b.

For applications, such as dynamic structural analysis, only the mass and position of the component is relevant, which can be provided in a point mass representation as illustrated by figure 8c. To this end, cabin components of the groups operator items, payload and furnishings are evaluated based on empiric and geometric relations. Combined, the three weight chapters contribute approximately the same mass as the empty mass of the aircraft. For the mass estimation, the geometric placement information available from the cabin design process can be reused and does not need to be recalculated. These capabilities are deployed as an additional rule set for FUGA an can therefore be integrated seamlessly with the other design and modeling capabilities.

3.4 Liquid hydrogen tank geometry rule set

To further illustrate the extensibility of the knowledge-based approach, a new rule set to model the integration of LH2 tanks into the fuselage is introduced. For the purpose of this integration study, a simplified parametric representation of the tanks is selected, based on simple geometric input parameters. The estimation of more accurate parameter values has been discussed in separate studies e.g. by Silberhorn et al. [19] and Burschyk et al. [42].



(a) Original seat mesh (b) Bounding box from seat mesh (c) Point mass from bounding box

Figure 8 – Multi-fidelity cabin components.

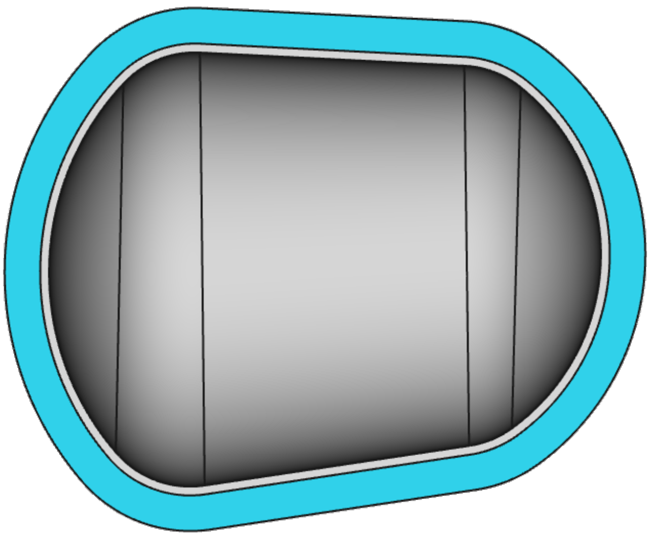


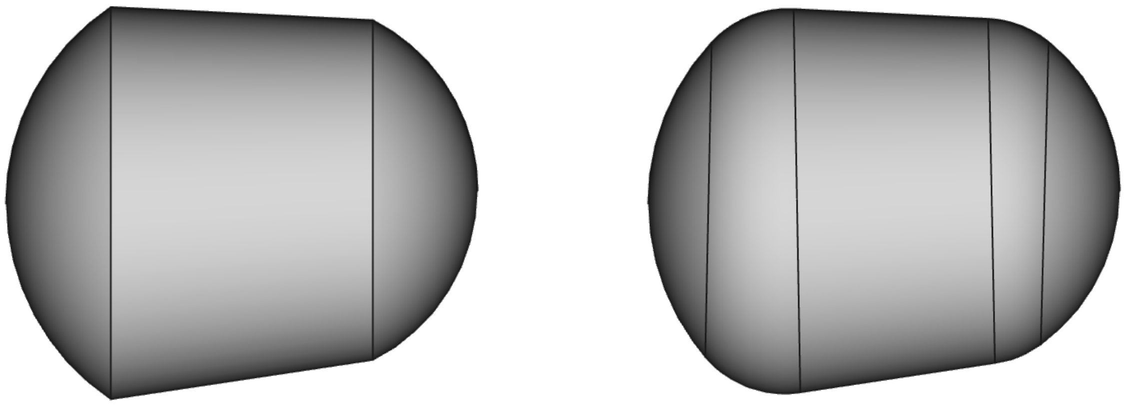
Figure 9 – Section view of the liquid hydrogen tank with insulation (blue) and wall (grey).

The tank wall is assumed to consist of two layers as shown in figure 9. On the one hand, there is the inner tank wall made of aluminum or steel, which must sustain the pressure loads from within the tank. On the other hand, an insulation layer is required to avoid excessive heating of the fuel, whose temperature is significantly lower than the ambient temperature.

To facilitate integration into the fuselage, the tank is designed inward from the outermost layer. To this end, a surface representation of the outer shape is constructed first. As illustrated by figure 10a, the tank consists of a truncated eccentric cone, which is defined by two circular sections with the centers $\mathbf{c}_1$ and $\mathbf{c}_2$ and the radii r_1 and r_2, and two spherical closing caps with the lengths l_1 and l_2. To achieve a smooth transition, a fillet with the radius r_{fillet} is applied at the common edges, as shown in figure 10b.

Based on the outer shell, offset-surfaces by the thicknesses of the insulation $t_{insulation}$ and the wall t_{wall} are computed. The innermost offset-surface circumscribes the available fuel storage volume, which can be computed via OCCT using Gauss integration.

To integrate the tank, the aforementioned design parameters must be determined based on the design data available for the fuselage. It is assumed, that the tank is placed between the rear bulkhead and the front spar of the horizontal tail plane (HTP), which yields the available length. The number of tanks installed back-to-back n_{tanks} is another design parameter, that can be specified. For redundancy



(a) Base primitives (b) Fillets applied

Figure 10 – Tank outer shape modeling steps.

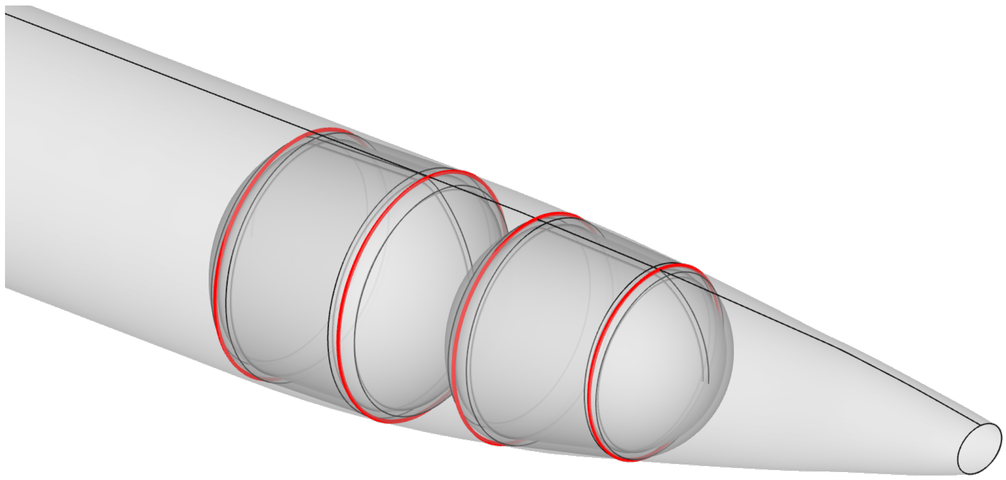


Figure 11 – Determination of sections from cabin space surface.

reasons, at least two tanks should be installed. If multiple tanks are present, it is assumed, that the available length is divided evenly between the tanks.
To determine the positions $\mathbf{c}_i$ and radii r_i of the circular sections, the surface shown in figure 7, which has been constructed from the frame tips to determine the cabin space, can be reused. Assuming a fixed closing cap length $l_{cc} = l_1 = l_2$, the longitudinal positions of the sections can be determined from the bounds. The circular section parameters are then given by the inner circle of the slice of the fuselage surfaces at these positions, as illustrated by figure 11.

4 Application examples

The design capabilities of FUGA are demonstrated using a stretched single aisle configuration for 240 passengers, similar to the Airbus A321neo. In a first example in section 4.1, the multi-fidelity

Use case	Extruded structure	Cutouts	Cabin components	Consider secondary	Target tool
GFEM geometry	false	false	point	true	Gmsh
TiGL export	false	true	box	false	TiGL Viewer
Immersive visualization	true	true	mesh	true	Blender

Table 1 – Assignment of fidelity settings to the three use cases.

geometry capabilities are highlighted by deriving models at different levels of fidelity from a single source file, according to the requirements of different use cases. A second example provided in section 4.2, where a hypothetical liquid hydrogen tank is added to the baseline configuration, serves to illustrate the extensibility of the knowledge-based approach.

4.1 Multi-fidelity geometry for the baseline single aisle configuration

Table 1 lists the three use cases considered in this section along with the respective requirements for the level of fidelity of the models. As discussed in section 3, the level of detail of the extruded reinforcements and the cutouts can be manipulated on the structural side, whereas simplified representations may be used for the cabin components. It is also possible to omit certain component groups, e.g. the secondary structure.

GFEM geometry with cabin masses From a purely geometrical perspective, the global finite element model (GFEM) geometry shown in figure 12 is the simplest use case. The goal is to assemble a geometry model, which can be fed to a tool like Gmsh [43], to generate a suitable finite element mesh. As mentioned previously, structural details are usually expressed via element properties, rather than detailed geometric representation.

Although the contribution of the cabin is rarely considered in typical GFEM analyses e.g. for structural sizing, cabin mass points are also provided, which can be added to the model as point mass elements. The figure shows the final mesh and the point masses plotted in the scientific data visualization tool Paraview[2].

On the other hand, some model details commonly found in the state of the art for CPACS-based GFEM (s. [12, 13]), have not yet been implemented, such as detailed wing-fuselage intersection regions.

CAD export for TiGL As mentioned previously, the TiGL Viewer is a visualization tool, which is widely used in the CPACS community. Therefore, an adequate interface is desirable, in order to make the additionally supported component geometries from FUGA accessible to people familiar with TiGL. Ideally, the model should convey information similar to the LOPA shown in figure 6.

Since both TiGL and the FUGA geometry modeling rule sets are built upon the OCCT library, an interface can be implemented using the native *BRep* format. *BRep* is designed for CAD geometry, and is therefore not well-suited for exchange of triangulated mesh data, which is used to represent the cabin component models. Consequently, the bounding box representations are stored instead. Furthermore, the structural cutouts are included to show the door positions. The frames and floor structures are again represented using curves, to manage file size and display performance in the

[2]https://www.paraview.org/

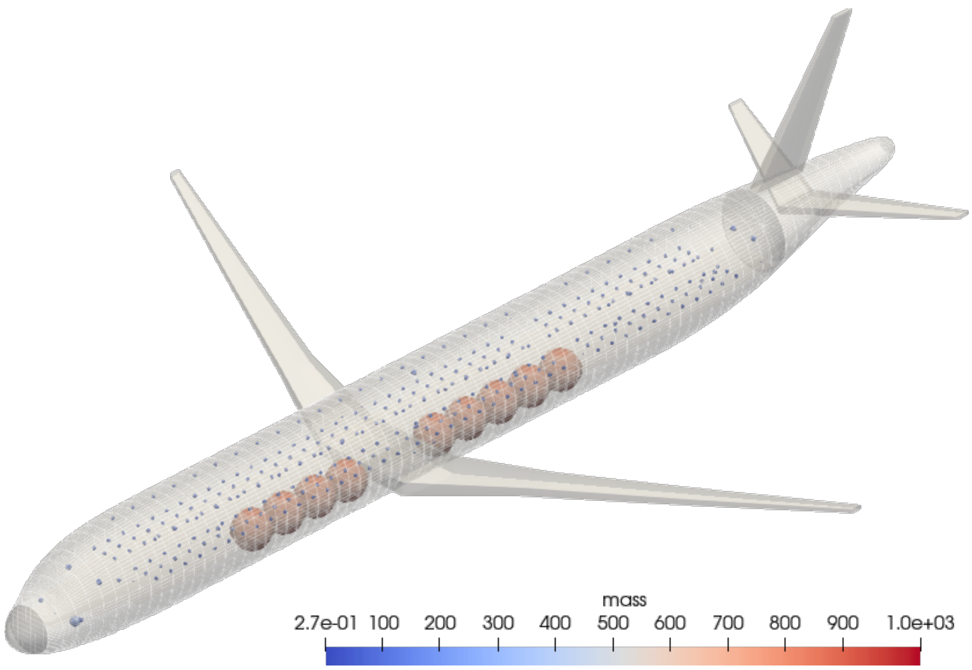


Figure 12 – Geometry model for GFEM generation and cabin point masses.

TiGL Viewer. The stringers are omitted to reduce clutter. The resulting model as displayed by the TiGL Viewer is given in figure 13.

Immersive visualization mesh The immersive visualization model shown in figure 14 is the most complex of the models shown. It is intended as an alternative to physical cabin mock-ups, e.g. for evaluation of human factors, providing a digital counterpart by leveraging e.g. virtual reality (VR) technology. The success largely depends on the level of immersion, which can be augmented by providing a high level of detail. Consequently, all the options in table 1 are set to their maximum values. A particularly important point is that, unlike in the previous example, the secondary structures i.e. panels and overhead storage bins are not omitted, but represented by detailed meshes, since they occupy a large section of the field of vision in the cabin. The resulting mesh is exported to the general-purpose 3D modeling environment Blender[3], where it can be further prepared for application in VR-compatible video game engines [44].

4.2 Fuselage-integrated LH2 tank

In this section, a liquid hydrogen tank is integrated into the fuselage from the previous example to showcase the extensibility of FUGA. To accommodate the tank, the length of the fuselage is increased, while the number of passengers remains unchanged w.r.t. the baseline design. Accordingly, the outside-in design system shown in figure 1 must be extended as shown in figure 15. Both the fuselage extension rule set described in section 3.1 and the tank design rule set described in section 3.4, which are highlighted in orange, have been added.
The fuselage extension rule set only requires the baseline geometry data from CPACS as input, along with a nominal frame pitch from the structural design rule set to provide an option to express the added length in terms of number of frames. The updated raw geometry data is then either passed through data preparation again or accessed directly by the outer loft model generation.

[3] https://www.blender.org/

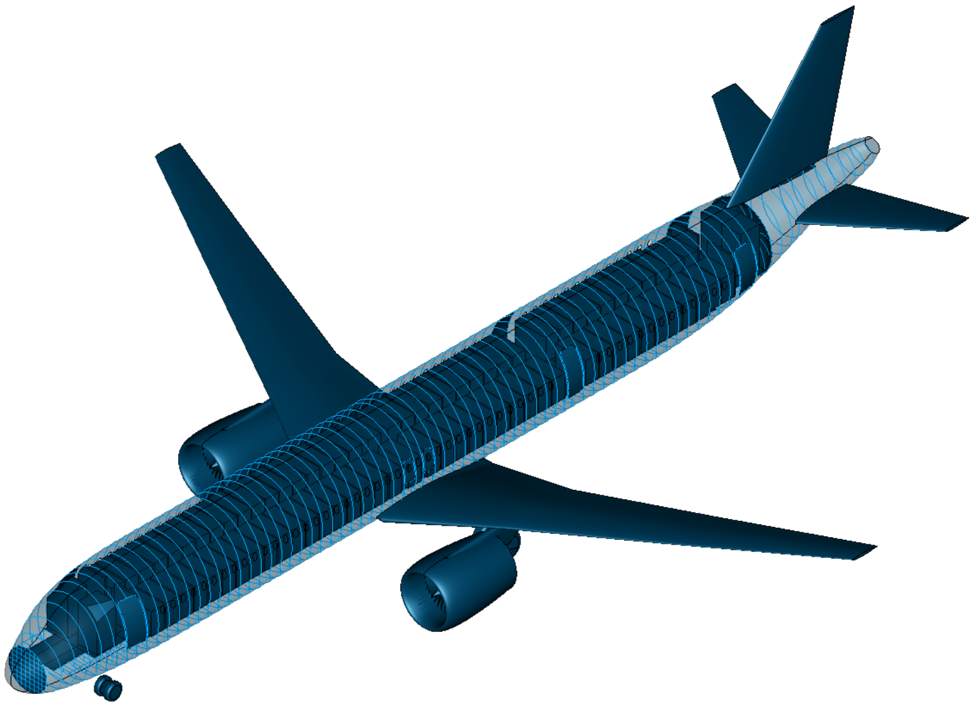


Figure 13 – Fuselage details shown in TiGL viewer.

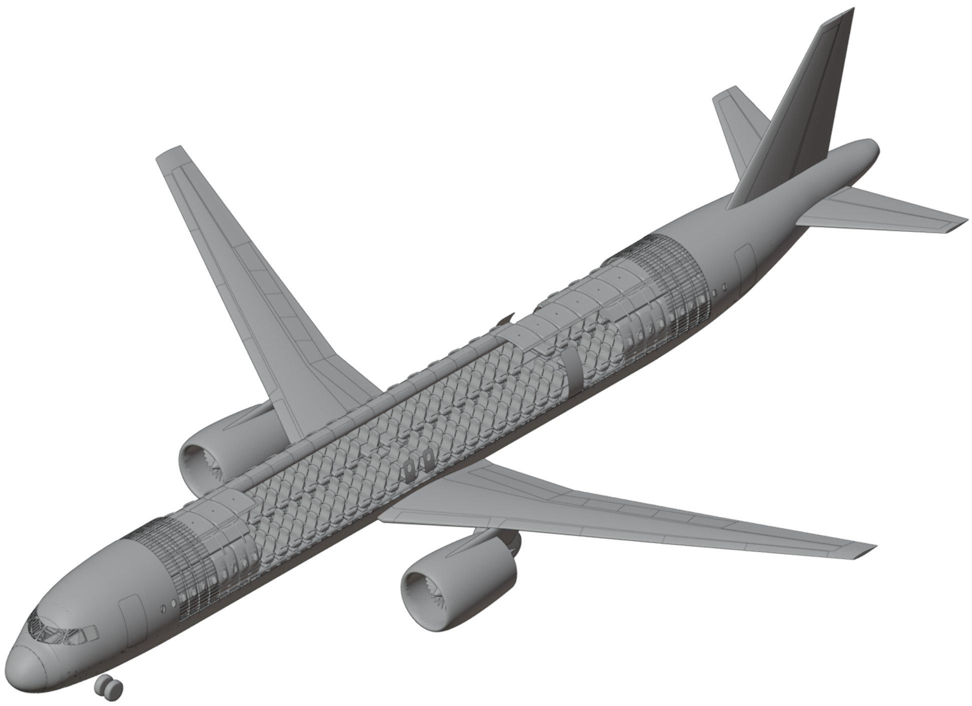


Figure 14 – Cut view of a detailed geometry model of structure and cabin for interactive cabin visualization.

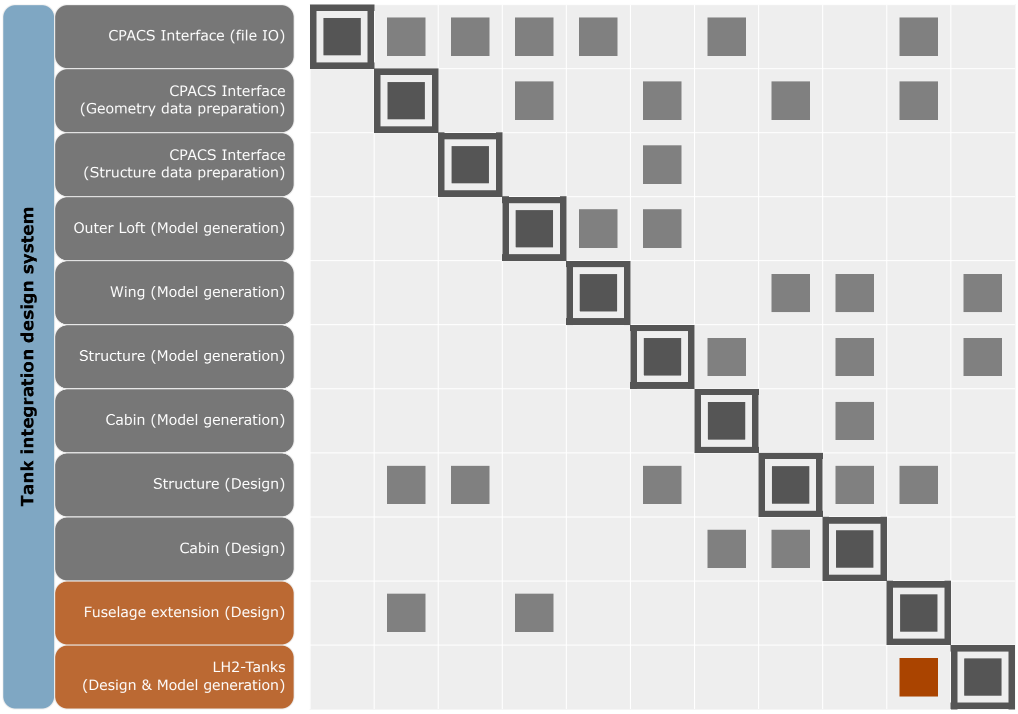


Figure 15 – N² chart of the liquid hydrogen tank integration design system.

The tank design is formulated in terms of the available space between the bulkhead and the HTP front spar, as illustrated by the incoming connection from the wing and structure model generation rule sets. These positions can be manipulated via the change in fuselage length Δl. On the other hand, the tank must provide sufficient space for a given amount of fuel resulting in a target volume $\hat{V}_{LH2}$. The tank volume V_{LH2} is a function of both the change in fuselage length and detailed geometric parameters given in section 3.4:

$$V_{LH2}\left(\Delta l, l_{cc}, r_{fillet}, t_{insulation}, t_{wall}, n_{tanks}\right). \tag{1}$$

To simplify the problem, the detailed tank parameters are fixed, which reduces the design problem to

$$V_{LH2}\left(\Delta l\right) = \hat{V}_{LH2}. \tag{2}$$

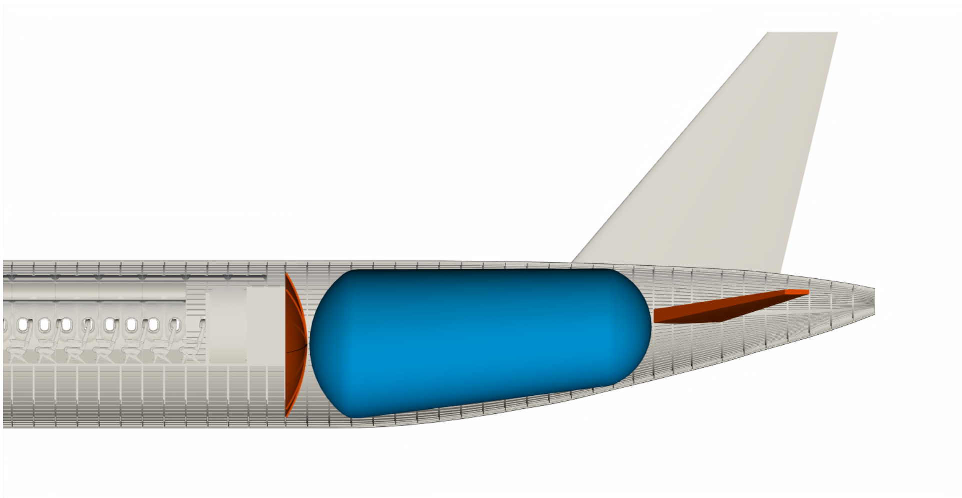
The relationship $V_{LH2}\left(\Delta l\right)$ can be expressed in FUGA by resetting the value for Δl in the data repository, invalidating all of its descendants, and re-evaluating the system to receive the updated volume. The problem can then be solved for the necessary Δl for the fuselage by finding the scalar root for

$$g\left(\Delta l\right) = V_{LH2}\left(\Delta l\right) - \hat{V}_{LH2} = 0 \tag{3}$$

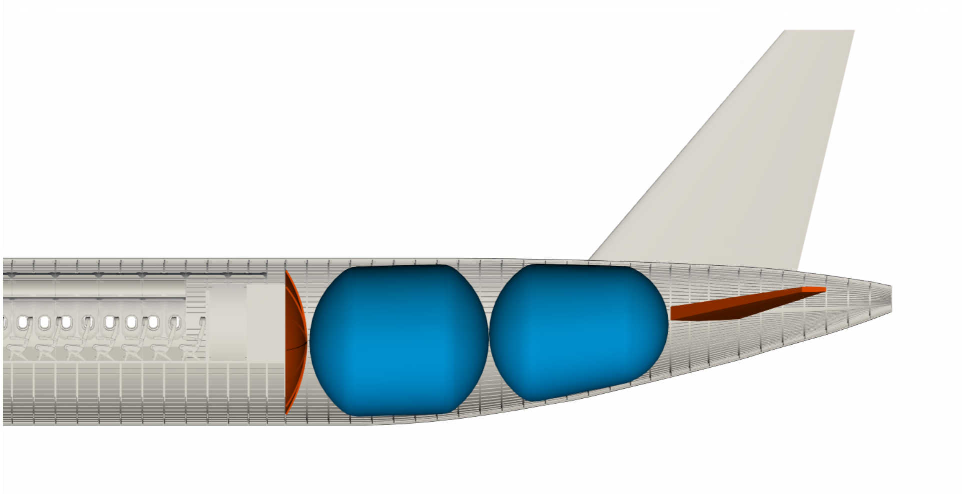
using e.g. the secant method. In this way, the feedback connection between the fuselage and tank design rule sets, which is marked in orange in figure 15, can be integrated into the system and resolved. In this study, the secant method implemented in SciPy [45] is applied.

The resulting geometry models are shown in figure 16, assuming a design with one, two or three tanks. For a hypothetical short-range design mission, a target volume $\hat{V}_{LH2} = 55m^3$ is assumed, based on reference data by Brewer [18] and Burschyk et al. [42].

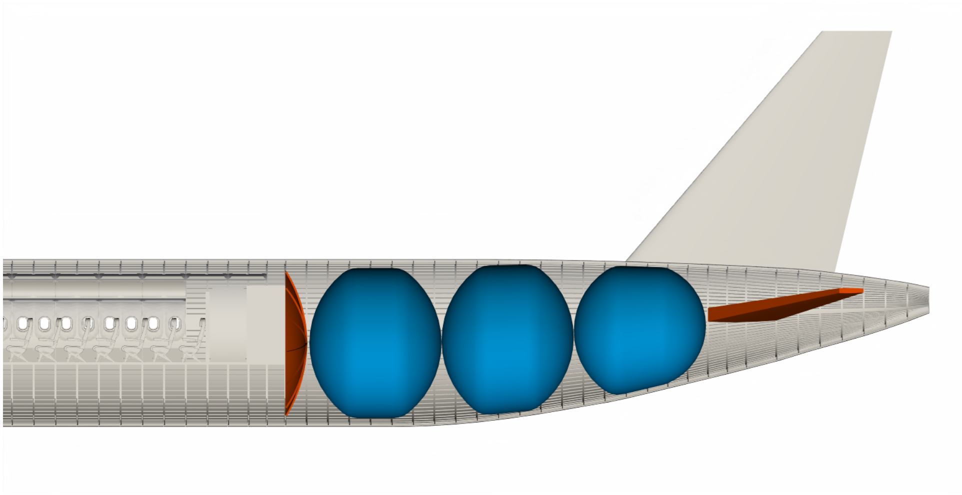
A summary of the distribution of the individual tank volumes and lengths is provided in figure 17.



(a) Integration result for $n_{tanks} = 1$



(b) Integration result for $n_{tanks} = 2$



(c) Integration result for $n_{tanks} = 3$

Figure 16 – Detailed geometry output for layouts with integrated LH2 tank.

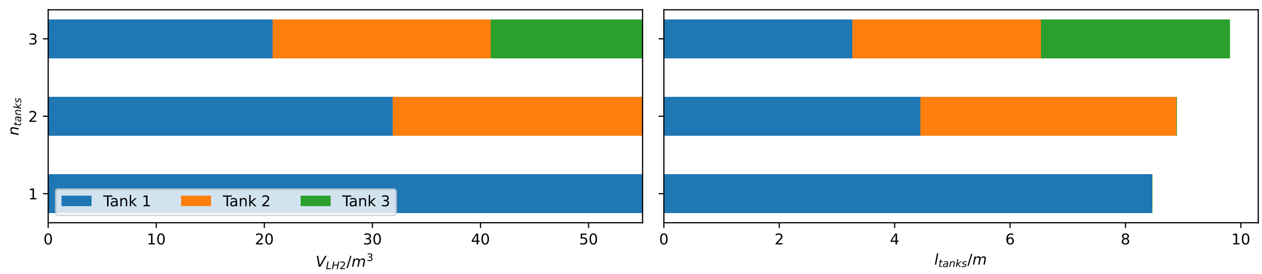


Figure 17 – Comparison of tank volume and length for the layouts given in figure 16.

While the accumulated volumes amount to the prescribed $55m^3$ in all cases, the distribution of the volume among the tanks, if multiple tanks are installed, is biased towards the front. This is due to the assumption of evenly divided tank lengths, which are also reflected in the figure. In combination with the decreasing cross-sectional area of the fuselage towards the rear, it causes the tank volumes to decrease. Thus, a criterion for the relative tank sizes based on an even distribution of the volume may in fact be preferable for the sake of improved redundancy. The cumulative tank lengths show a clear advantage to having the least number of tanks possible, since less space is lost due to intermediate walls.

The above results show a consistent geometric solution for a tank integration. This is not necessarily synonymous with a feasible engineering solution. A fuselage length increase of up to $10m$ will inevitably have substantial impact on the overall aircraft design, requiring changes e.g. in the wing and empennage sizing, or the landing gear. Therefore, it is important to iterate the results with a preliminary design synthesizer, such as openAD before making a final assessment of the design.

5 Conclusion

In this paper, a knowledge-based methodology has been presented, which can be applied to generate product details necessary to supply geometry models for analyses for different disciplines or at different levels of fidelity. Using CPACS as an aircraft product data repository and a comprehensive set of design and model generation rules, provided by the KBE tool FUGA, it has been shown, that consistent geometry models tailored to specific fidelity requirements of various use cases can be provided.

Furthermore, the concept of rule sets has been introduced, which can be used to structure the available rules and deploy them according to the design task at hand. New rule sets can be added to expand the scope of the design capabilities, or incorporate different product architectures. The feasibility is demonstrated by expanding an outside-in fuselage design process with new rule sets to solve the problem of integrating a liquid hydrogen tank into the fuselage.

The results show, that the knowledge-based system implemented in FUGA can help integrate design and geometry generation tasks. They also show, that geometrically feasible solutions do not always correspond to valid engineering solutions. For instance, for the hydrogen tank example, better integration, both with preliminary aircraft design and hydrogen tank detail design will be necessary to provide a reliable design proposal. The value of FUGA in this scheme is, that it can provide consistent suggestions for the respective experts to evaluate.

Furthermore, improvements can be made to the knowledge-base, not only by adding new rule sets, but also by expanding the existing rule sets. For example, adding the wing-fuselage intersection areas to the structural modeling rule set will further improve the geometry model for the GFEM generation, whereas an implementation of a fully automatic cabin layout generator based only on the number of passengers will enable more complex tank integration scenarios, where the tank size can be traded with the fuselage length and the number of passengers.

With its extensible modular architecture and support for multi-fidelity geometry, FUGA has the potential to become a knowledge hub for highly integrated multidisciplinary design and analysis processes of the fuselage and cabin, providing product data ranging from a basic LOPA to complex on-board system layouts. In this way, it can provide the basis for applications as diverse as finite element analysis for vibro-acoustics, manufacturing simulation and human-centered assessment using VR. As a result, further research is ongoing, both to develop stable automatic interfaces to analysis tools and to formulate further existing design knowledge, e.g. on on-board systems or manufacturing processes, in terms of FUGA rule sets.

Contact Author Email Address

jan-niclas.walther@dlr.de

Copyright Statement

References

[1] Agte, J. et al. “MDO: assessment and direction for advancement—an opinion of one international group”. In: *Structural and Multidisciplinary Optimization* 40.1-6 (Apr. 2009), pp. 17–33. DOI: 10.1007/s00158-009-0381-5.

[2] Siedlak, D. J. L., Pinon, O. J., Schlais, P. R., Schmidt, T. M., and Mavris, D. N. “A digital thread approach to support manufacturing-influenced conceptual aircraft design”. In: *Research in Engineering Design* 29.2 (Sept. 2017), pp. 285–308. DOI: 10.1007/s00163-017-0269-0.

[3] Singh, V. and Willcox, K. E. “Engineering Design with Digital Thread”. In: *AIAA Journal* 56.11 (Nov. 2018), pp. 4515–4528. DOI: 10.2514/1.j057255.

[4] Sarojini, D., Xie, J., Cai, Y., Corman, J. A., and Mavris, D. “A Certification-Driven Platform for Multidisciplinary Design Space Exploration in Airframe Early Preliminary Design”. In: *AIAA AVIATION 2020 FORUM*. American Institute of Aeronautics and Astronautics, June 2020. DOI: 10.2514/6.2020-3157.

[5] Torrigiani, F. et al. “An MBSE Certification-Driven Design of a UAV MALE Configuration in the AGILE 4.0 Design Environment”. In: *AIAA AVIATION 2021 FORUM*. American Institute of Aeronautics and Astronautics, July 2021. DOI: 10.2514/6.2021-3080.

[6] Kroll, N. et al. "DLR project Digital-X: towards virtual aircraft design and flight testing based on high-fidelity methods". In: *CEAS Aeronautical Journal* 7.1 (Dec. 2015), pp. 3–27. DOI: 10.1007/s13272-015-0179-7.

[7] Ciampa, P. D. et al. "Streamlining Cross-Organizational Aircraft Development: Results from the AGILE Project". In: *AIAA Aviation 2019 Forum*. American Institute of Aeronautics and Astronautics, June 2019. DOI: 10.2514/6.2019-3454.

[8] Goertz, S. et al. "Overview of Collaborative Multi-Fidelity Multidisciplinary Design Optimization Activities in the DLR Project VicToria". In: *AIAA AVIATION 2020 FORUM*. American Institute of Aeronautics and Astronautics, June 2020. DOI: 10.2514/6.2020-3167.

[9] Liersch, C. M. and Hepperle, M. "A distributed toolbox for multidisciplinary preliminary aircraft design". In: *CEAS Aeronautical Journal* 2.1-4 (Aug. 2011), pp. 57–68. DOI: 10.1007/s13272-011-0024-6.

[10] Nagel, B. et al. "Communication in Aircraft Design: Can We Establish a Common Language?" In: *28th Congress of the International Council of the Aeronautical Sciences (ICAS), Brisbane, Australia*. 2012. URL: https://elib.dlr.de/134586/.

[11] Alder, M., Moerland, E., Jepsen, J., and Nagel, B. "Recent Advances in Establishing a Common Language For Aircraft Design With CPACS". In: *Aerospace Europe Conference 2020*. Jan. 2020. URL: https://elib.dlr.de/134341/.

[12] Petsch, M., Kohlgrüber, D., and Heubischl, J. "PANDORA - A python based framework for modelling and structural sizing of transport aircraft". In: *MATEC Web of Conferences* 233 (2018). Ed. by Kontis, K. and Pantelakis, S. DOI: 10.1051/matecconf/201823300013.

[13] Klimmek, T., Schulze, M., Abu-Zurayk, M., Ilic, C., and Merle, A. "cpacs-MONA – An independent and in high fidelity based MDO tasks integrated process for the structural and aeroelastic design for aircraft configurations". In: *IFASD 2019 - International Forum on Aeroelasticity and Structural Dynamics*. June 2019. URL: https://elib.dlr.de/128099/.

[14] Wunderlich, T. F., Daehne, S., Reimer, L., Schuster, A., and Brodersen, O. "Global Aero-Structural Design Optimization of More Flexible Wings for Commercial Aircraft". In: *AIAA AVIATION 2020 FORUM*. American Institute of Aeronautics and Astronautics, June 2020. DOI: 10.2514/6.2020-3170.

[15] De Marco, A. *JPAD, A Java Application Programming Interface for Aircraft Design*. en. 2018. DOI: 10.5281/ZENODO.2064963.

[16] Siggel, M., Kleinert, J., Stollenwerk, T., and Maierl, R. "TiGL: An Open Source Computational Geometry Library for Parametric Aircraft Design". In: *Mathematics in Computer Science* 13.3 (July 2019), pp. 367–389. DOI: 10.1007/s11786-019-00401-y.

[17] Walther, J.-N., Hesse, C., Biedermann, J., and Nagel, B. "High Fidelity Digital Cabin Mock-Up based on Preliminary Aircraft Design Data for Virtual Reality Applications and Beyond". In: *AIAA AVIATION 2021 FORUM*. American Institute of Aeronautics and Astronautics, July 2021. DOI: 10.2514/6.2021-2775.

[18] Brewer, G. D. *Hydrogen Aircraft Technology*. CRC Press Inc., 1991. ISBN: 0-8493-5838-8.

[19] Silberhorn, D., Atanasov, G., Walther, J.-N., and Zill, T. “Assessment of Hydrogen Fuel Tank Integration at Aircraft Level”. In: *Deutscher Luft- und Raumfahrtkongress 2019*. Sept. 2019. URL: https://elib.dlr.de/129643/.

[20] Hartmann, J. and Nagel, B. “Eliminating Climate Impact From Aviation - A system level approach as applied in the framework of the DLR-internal project EXACT”. In: *Deutscher Luft- und Raumfahrtkongress 2021*. Sept. 2021. URL: https://elib.dlr.de/148552/.

[21] Chapman, C. B. and Pinfold, M. “Design Engineering — A need to rethink the solution using KBE”. In: *Applications and Innovations in Expert Systems VI*. Springer London, 1999, pp. 112–131. DOI: 10.1007/978-1-4471-0575-6_9.

[22] La Rocca, G. “Knowledge Based Engineering Techniques to Support Aircraft Design and Optimization”. PhD thesis. TU Delft, 2011. URL: http://resolver.tudelft.nl/uuid:45ed17b3-4743-4adc-bd65-65dd203e4a09.

[23] Scherer, J. and Kohlgrüber, D. “Fuselage structures within the CPACS data format”. In: *Aircraft Engineering and Aerospace Technology* 88.2 (Mar. 2016), pp. 294–302. DOI: 10.1108/aeat-02-2015-0056.

[24] Walther, J.-N., Hesse, C., Alder, M., Biedermann, J. Y.-C., and Nagel, B. “Erweiterung der Kabinenbeschreibung im CPACS-Luftfahrzeugdatenschema zur Anbindung Detaillierter Analysen”. In: *Deutscher Luft- und Raumfahrtkongress 2021*. Sept. 2021. URL: https://elib.dlr.de/147064/.

[25] Open Cascade. *Open Cascade Website*. 2022. URL: https://www.opencascade.com/.

[26] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A. “OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization”. In: *Structural and Multidisciplinary Optimization* 59.4 (Mar. 2019), pp. 1075–1104. DOI: 10.1007/s00158-019-02211-z.

[27] Pate, D. J., Gray, J., and German, B. J. “A graph theoretic approach to problem formulation for multidisciplinary design analysis and optimization”. In: *Structural and Multidisciplinary Optimization* 49.5 (Oct. 2013), pp. 743–760. DOI: 10.1007/s00158-013-1006-6.

[28] Gent, I. van. “Agile MDAO Systems: A Graph-based Methodology to Enhance Collaborative Multidisciplinary Design”. PhD thesis. 2019. DOI: 10.4233/uuid:c42b30ba-2ba7-4fff-bf1c-f81f85e890af.

[29] Hagberg, A. A., Schult, D. A., and Swart, P. J. “Exploring Network Structure, Dynamics, and Function using NetworkX”. In: *Proceedings of the 7th Python in Science Conference*. Ed. by Varoquaux, G., Vaught, T., and Millman, J. Pasadena, CA USA, 2008, pp. 11–15. URL: https://www.osti.gov/biblio/960616.

[30] Kahn, A. B. “Topological sorting of large networks”. In: *Communications of the ACM* 5.11 (Nov. 1962), pp. 558–562. DOI: 10.1145/368996.369025.

[31] Siggel, M., Stollenwerk, T., Kleinert, J., Gruber, B. M., and Maierl, R. *TiGL - The TiGL Geometry Library*. 2022. DOI: 10.5281/ZENODO.858650.

[32] Lazzara, D., Haimes, R., and Willcox, K. “Multifidelity Geometry and Analysis in Aircraft Conceptual Design”. In: *19th AIAA Computational Fluid Dynamics*. American Institute of Aeronautics and Astronautics, June 2009. DOI: 10.2514/6.2009-3806.

[33] Böhnke, D. "A Multi-Fidelity Workflow to Derive Physics-Based Conceptual Design Methods". PhD thesis. Technische Universität Hamburg-Harburg, 2015. URL: `https://elib.dlr.de/97905/`.

[34] Dorbath, F. "A flexible wing modeling and physical mass estimation system for early aircraft design stages". en. PhD thesis. TU Hamburg, 2014. DOI: `10.15480/882.1159`.

[35] Munjulury, R. C., Staack, I., Berry, P., and Krus, P. "A knowledge-based integrated aircraft conceptual design framework". In: *CEAS Aeronautical Journal* 7.1 (Nov. 2015), pp. 95–105. DOI: `10.1007/s13272-015-0174-z`.

[36] Jonge, R. L. A. de. "Development of a Knowledge-Based Engineering Application to Support Conceptual Fuselage Sizing and Cabin Configuration". MA thesis. TU Delft, Jan. 2017. URL: `http://resolver.tudelft.nl/uuid:59d18aa3-dbcb-4252-b1cb-eaf13f0059bf`.

[37] Woehler, S., Atanasov, G., Silberhorn, D., Fröhler, B., and Zill, T. "Preliminary Aircraft Design within a Multidisciplinary and Multifidelity Design Environment". In: *Aerospace Europe Conference 2020*. Apr. 2020. URL: `https://elib.dlr.de/185515/`.

[38] Walther, J.-N. and Ciampa, P. D. "Knowledge-based automatic Airframe Design using CPACS". In: *Transportation Research Procedia* 29 (2018), pp. 427–439. DOI: `10.1016/j.trpro.2018.02.038`.

[39] Walther, J.-N., Kocacan, B., Hesse, C., Gindorf, A., and Nagel, B. "Automatic cabin virtualization based on preliminary aircraft design data". In: *CEAS Aeronautical Journal* (Jan. 2022). DOI: `10.1007/s13272-021-00568-w`.

[40] Diamond, S. and Boyd, S. *CVXPY: A Python-Embedded Modeling Language for Convex Optimization*. 2016. DOI: `10.48550/ARXIV.1603.00943`.

[41] European Aviation Safety Agency. *Certification specifications and acceptable means of compliance for large aeroplanes CS-25 - Amendment 23*. 2022. URL: `https://www.easa.europa.eu/document-library/certification-specifications/cs-25-amendment-23`.

[42] Burschyk, T., Cabac, Y., Silberhorn, D., Boden, B., and Nagel, B. "Liquid hydrogen storage design trades for a short-range aircraft concept". In: *Deutscher Luft- und Raumfahrtkongress 2021*. Sept. 2021. URL: `https://elib.dlr.de/148344/`.

[43] Geuzaine, C. and Remacle, J.-F. "Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities". In: *International Journal for Numerical Methods in Engineering* 79 (Sept. 2009), pp. 1309–1331. DOI: `10.1002/nme.2579`.

[44] Fuchs, M., Beckert, F., Biedermann, J., and Nagel, B. "A collaborative knowledge-based method for the interactive development of cabin systems in virtual reality". In: *Computers in Industry* 136 (Apr. 2022), p. 103590. DOI: `10.1016/j.compind.2021.103590`.

[45] Virtanen, P. et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature Methods* 17.3 (Feb. 2020), pp. 261–272. DOI: `10.1038/s41592-019-0686-2`.