

Article

Model-Based Engineering Process Automation from Design to Manufacturing of Fiber Composite Helicopter Structures Using Graph-Based Design Languages

Claudia Schopper ^{1,2,*}, Dominik Schopper ^{1,2}, Maximilian Holland ³, Julian Dinkelacker ² , Julian Schuster ³ and Stephan Rudolph ¹

¹ Institute of Aircraft Design (IFB), University of Stuttgart, 70569 Stuttgart, Germany; rudolph@ifb.uni-stuttgart.de (S.R.)

² IILS Ingenieurgesellschaft für Intelligente Systeme mbH, 72818 Trochtelfingen, Germany; julian.dinkelacker@iils.de

³ Fraunhofer Institute for Casting, Composite and Processing Technology IGCV, 86159 Augsburg, Germany; maximilian.holland@igcv.fraunhofer.de (M.H.); julian.schuster@igcv.fraunhofer.de (J.S.)

* Correspondence: claudia.schopper@ifb.uni-stuttgart.de

Abstract

The design and manufacturing of carbon-fiber-reinforced polymer (CFRP) structures in aerospace require balancing high structural performance with cost-efficient, reproducible production. Conventional design and planning methods are often fragmented across disciplines, causing data discontinuities and limited traceability. This paper introduces a graph-based design language (GBDL) information architecture that integrates CFRP design and manufacturing within a unified, model-based framework. The approach formalizes engineering knowledge through process ontologies and graph-based data models linking geometry, material, tooling, and process parameters in a consistent, machine-interpretable form. Each step, from geometry derivation and structural design to prepreg hand lay-up and automated fiber placement, is represented within a shared design graph to ensure data consistency, transparency, and automated assessment of lead time, labor, cost, waste, and energy consumption. Although current implementations address selected use cases with partially automated interfaces, the architecture establishes a scalable foundation for full interoperability. A helicopter-frame case study demonstrates the applicability and adaptability of the method.

Keywords: graph-based design languages; manufacturing cost estimation; CFRP manufacturing; design-manufacturing integration



Academic Editor: Khamis Essa

Received: 10 February 2026

Revised: 17 March 2026

Accepted: 18 March 2026

Published: 26 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The research reported here is based on the results of the COBAIN research project within the framework of the German national aviation research (“Luftfahrtforschungsprogramm”, for short: “LuFo”) in collaboration with Airbus Helicopters Deutschland GmbH (AHD), the University of Stuttgart, IILS mbH, Fraunhofer IGCV and DLR. Over the past decade, the use of carbon-fiber-reinforced polymers (CFRP) has become firmly established in industrial manufacturing, particularly in the aerospace sector, due to their excellent ratio of weight to mechanical performance (over 50% aircraft primary design product is CFRP) [1]. Nevertheless, their production still poses significant challenges: the highly customized and often complex geometries of components can lead to process-related defects

and quality variations [2]. Consequently, many manufacturing steps still rely heavily on manual labor, making cost estimation and reproducibility difficult.

To address these issues, research is increasingly focusing on approaches to automating CFRP production, including automated lay-up technologies such as automated fiber placement (AFP) [3–6] as well as broader automation approaches for fiber-reinforced polymer manufacturing [7–10], which also includes the automation of material handling [11] or the automation of the machining and postprocessing of composites [12,13]. These developments are also increasingly adopted in industrial practice to improve process stability and production efficiency [3]. While efforts to automate CFRP production frequently concentrate on automating the manufacturing processes themselves, such as process execution, monitoring, and quality control, there are few known initiatives aimed at automating the design processes for manufacturing planning such as component design and process optimization. However, the design and manufacturing of fiber-reinforced composites involve a multitude of interdependent parameters, process steps, and data sources. Achieving both high structural performance and cost-efficient, reproducible production demands an integrated, data-driven approach both in its implementation and in its design process, which still remains a major challenge in composite manufacturing [14]. Fragmented toolchains and isolated disciplines often lead to breaks in data flow and hinder the evaluation of how design choices affect manufacturing effort, recurring costs, or material efficiency.

In recent years, model-based engineering and digital modeling concepts have gained attention as means to ensure consistency and traceability throughout a design process and across product life cycles. Within this context, graph-based design languages (GBDLs) have emerged as a promising approach for representing engineering knowledge in a formal, machine-readable and executable way [15]. By using graphs as a central data model, GBDLs enable the integration of heterogeneous data sources, the representation of process ontologies, and the automation of repetitive reasoning and calculation tasks. Several works have demonstrated the potential of GBDLs for design automation in the context of manufacturing planning [16,17] and composite structures [18,19]. The so-called design compiler used to define and execute GBDL in this work is the Design Cockpit 43 v4.0.24 (DC43® v4.0.24v4.0.24).

This paper addresses this gap by presenting a GBDL-based modeling framework for the end-to-end integration of the design and manufacturing of CFRP helicopter structures. The approach captures the relevant process steps from geometry derivation and structural design to production planning using prepreg hand lay-up and automated fiber placement. Each step is represented by an ontology that systematically decomposes the workflow into main and subprocesses and assigns the relevant parameters and calculation rules. The resulting design graph serves as a common exchange format across tools and data formats, thereby ensuring interoperability with established standards such as CPACS.

The contribution of this work is threefold. First, it demonstrates how GBDLs can be used to model the entire process chain of CFRP component design and manufacturing in a consistent and extensible manner. Second, it provides a system architecture for linking geometry, material, tooling, and process parameters in a way that supports the automated calculation of evaluation criteria. This enables the rapid evaluation of numerous product variants depending on various modifications throughout the product lifecycle “at the push of a button”. The modification options include changes to component geometries, material selection, or the scope of available production processes. The generated product variants can then be compared in terms of evaluation parameters such as lead time, labor effort, recurring costs, waste, and energy consumption. This may allow for the identification of promising product variants that could not be further explored if each evaluation task had to be performed manually for every product variant due to capacity constraints. And

third, it illustrates the application of the approach through a case study of a helicopter frame, showing how this digital integration strategy can also be used to automate CFRP production tasks.

2. Framework and Integration Strategy

2.1. Graph-Based Design Languages

Graph-based design languages (GBDLs) provide a formal framework for integrating the entire process chain within a consistent modeling environment. In a GBDL, processes, data, and dependencies are represented as nodes and edges in a graph, which serves as the central data model [15]. It is important to note that GBDLs in general and in the application in this paper, are structured as follows: First the modeled domain is structured into an ontology with a class diagram, lifted from the Unified Modeling Standard (UML) [20]. It describes all objects (classes/nodes) of a domain and their relations (associations/edges) inside of the graph. The second aspect is the formulation of rules of possible graph-transformations of the predefined ontologies that describe the engineering logic. The rules are set inside a program, the so-called production system, to enable a machine-executable rule-based generation of a central data model, the design graph. This structure enables the systematic decomposition of complex manufacturing workflows into main and subprocesses, while ensuring that geometry, material, tooling, and process parameters remain consistently linked across all stages. By combining ontological representations with rule-based calculation modules, GBDLs support automated reasoning, interoperability with external tools, and rapid evaluation of design and manufacturing scenarios.

2.2. Process Overview

The integration approach is structured around a process map, shown in Figure 1, that captures the sequence of design and manufacturing tasks together with their required data flows. Each process is defined by its input and output parameters, the data formats used, and its interdependencies with other processes. This mapping forms the methodological basis for linking autonomous design languages in an end-to-end workflow.

Graph-based design languages are employed as the central integration mechanism. Each design language models a specific domain task—such as target definition, geometry design, or manufacturing planning—while the design graph acts as a common exchange format. This ensures that results can be seamlessly transferred between modules, independently of the external tools or file formats involved. Standardized interfaces such as CPACS are used to bridge to the established engineering software, enabling the exchange of load cases, material properties, or geometric data.

The framework has been exemplified using the design and manufacturing of a helicopter-frame structure. In this context, design languages were developed for both prepreg hand lay-up and automated fiber placement. Each technology-specific module incorporates user inputs, database access for material and process data, and rule-based evaluation of process times, costs, waste, and energy consumption. By capturing these steps in the ontology and linking them through the design graph, the overall process chain can be consistently modeled and adapted to different CFRP component types.

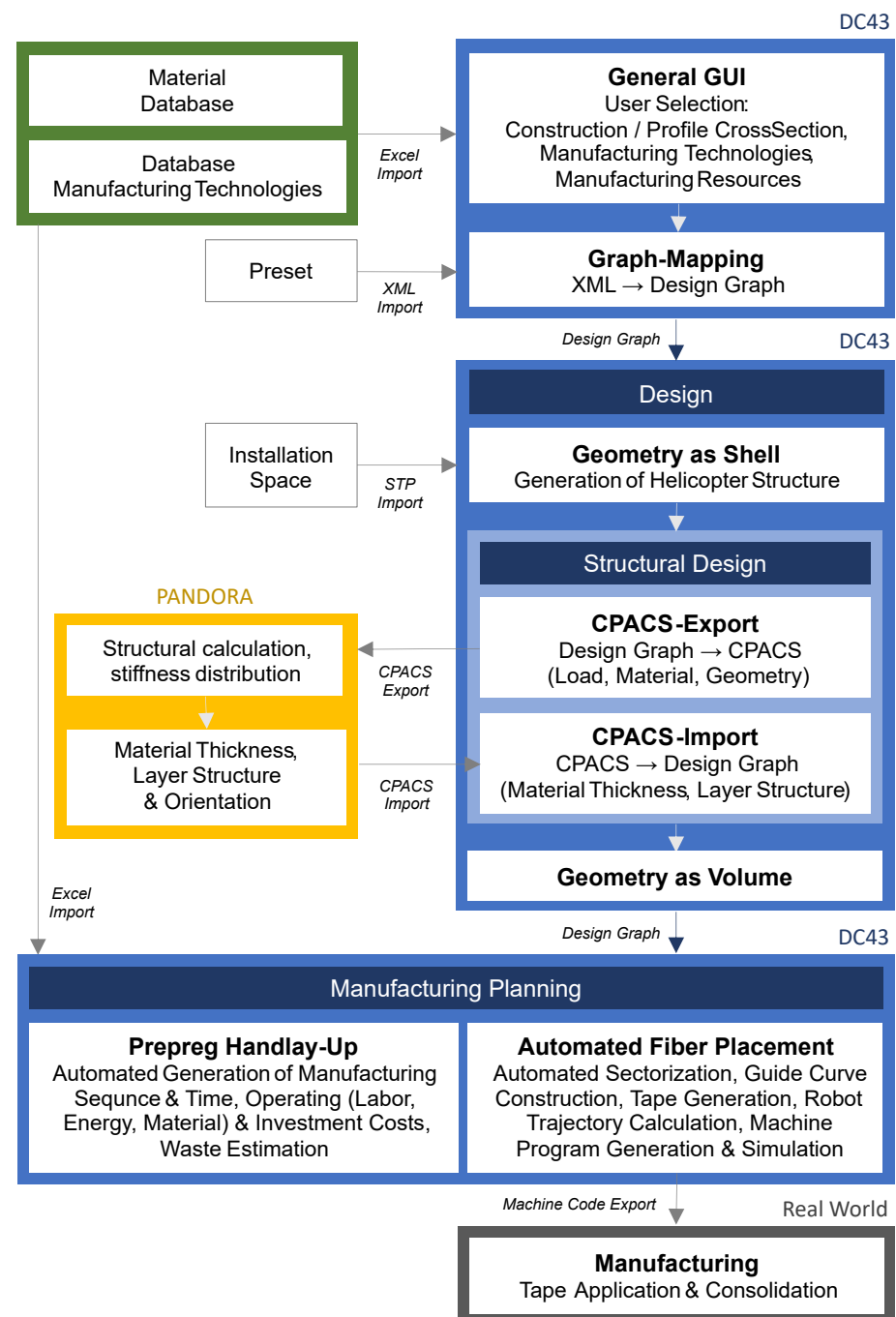


Figure 1. Schematic overview of the automated engineering process. Boxes represent applications, and arrows represent the file exchange format between the applications.

3. Application to CFRP Component Design

This section introduces the first GBDL tasked with the automatic generation of a CFRP helicopter structure component. The procedure is demonstrated using the example of a helicopter fuselage frame. Even more so than when creating component geometry manually, the automated process requires a process to be defined that strictly adheres to design specifications and guidelines. This means that new design rules must be found and implemented for different types of components. In the scope of the project, a similar approach was implemented for shell-shaped components. For the overall structure, as shown in Figure 1, the process of geometry design is split into three steps. An initial geometry derivation from the installation space, given from a preliminary design step,

is created as a shell. Then a structural analysis based on load case is performed, using an interface to the external solver PANDORA. From the imported calculation results a volumetric final geometry is created. Throughout this process the design graph is used as the central data model, handling the derived representation in the CAD and CPACS format.

3.1. Design of a Helicopter Structure Fuselage Frame

This chapter introduces the design language for generating the component geometry in form of a CAD model of fuselage frames. Figure 2 shows the ontology or the associated class diagram. This class diagram is exemplary for all GBDLs in this paper and aims to show the formalization of a domain, with the relevant concepts, relationships and parameters, into an ontology. Arrows with an empty tip signify an inheritance relationship and blue arrows an association with a given name and multiplicity in square brackets. The design incorporates different profile specifications such as I-, C- and Ω -Profiles which have specific parameters and construction rules, defined in the design language. These are typical designs in helicopter-frame construction that were proposed by AHD as design variants. The schematics and respective parametrization are shown in Figure 3. The interface to the structural analysis, described in more detail in Section 3.2, is given by the Fuselage Data-Exchange Profile. A visualization of the different profiles in the final design are depicted in Figure 3. An algorithmic procedure of generic and profile-specific design rules derive a component geometry (first as shell and then as solid) completely automatically from the specification of an installation space at any position of the helicopter.

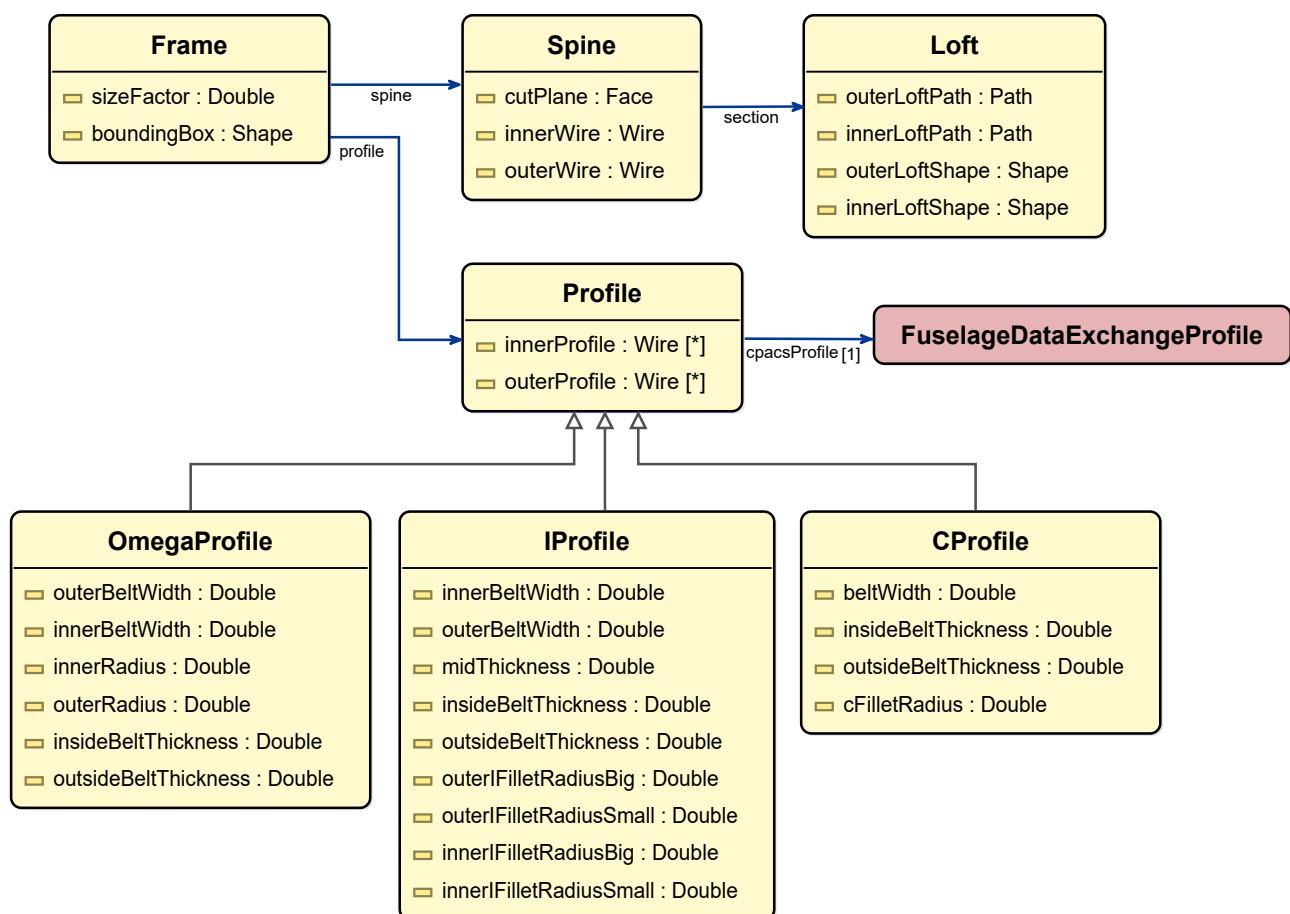


Figure 2. Ontology or class diagram for the frame design. The arrows in the figure follow the UML standard. Numbers in square brackets indicate multiplicity, where the * symbol denotes a multiple multiplicity greater than 0.

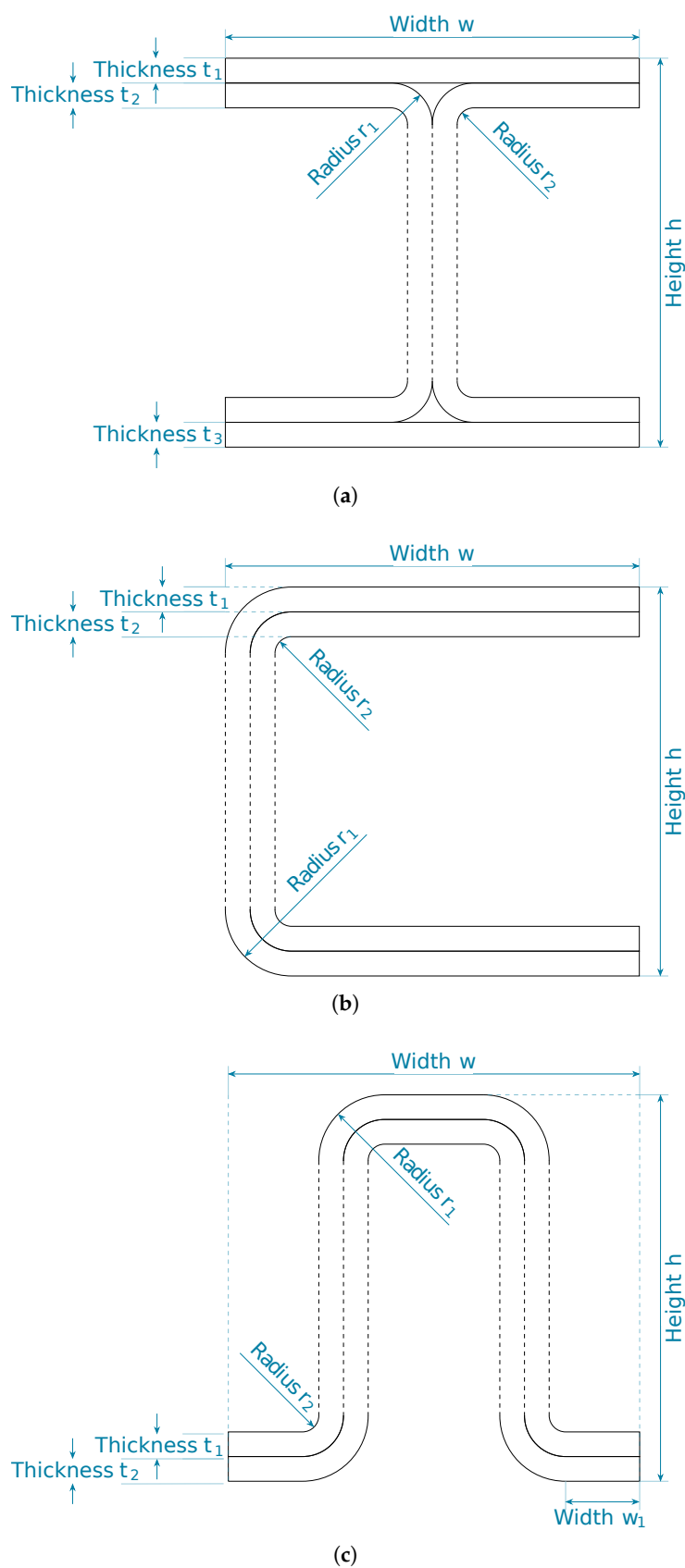


Figure 3. Different types of frame profiles including the geometric design conditions (translated from [21]). Shown are: (a) “T”-shaped frame profile; (b) “C”-shaped frame profile; and (c) “Ω”-shaped frame profile. Dashed lines have adjustable lengths.

For the automated design of the fuselage frame, all desired boundary conditions must be specified. This concerns the envelope geometries of the fuselage cell and the inner cabin of the helicopter, as the installation space available for the frame is limited between these surfaces. The two boundary surfaces mentioned are imported into the design language in the standardized geometry exchange format STEP ing (More information on the STEP exchange format can be found in [22]). To enable variant design, the wire-frame structure of the design zone with which the design language will then operate, must first be developed using a reverse engineering process. For this purpose, a segment is cut out of the helicopter fuselage at a defined point in a multi-stage process using a so-called “electronic knife” and is replaced by a suitable wire-frame model that can be manipulated in the design language. The cutting out of the construction zone from the fuselage and cabin geometry is shown schematically in Figure 4.

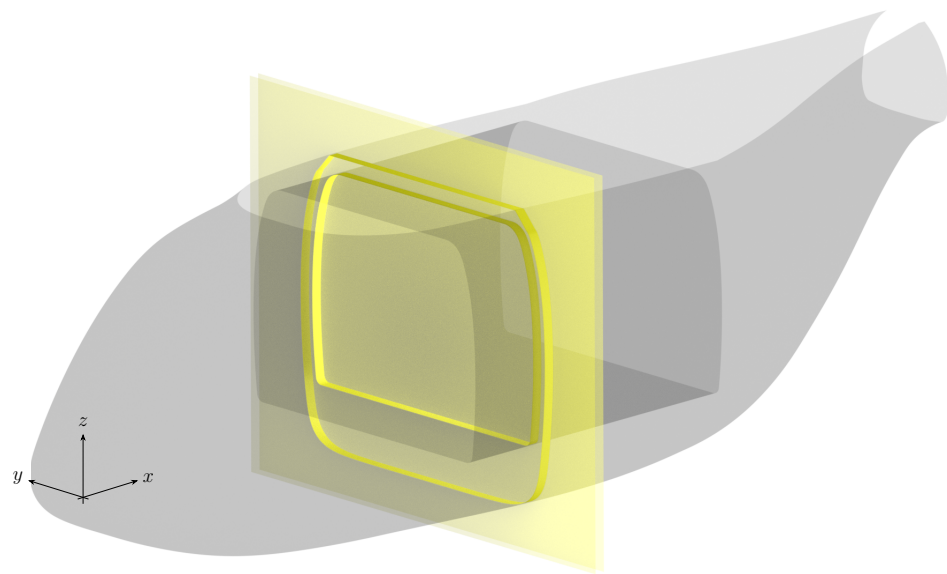


Figure 4. Cutting out the construction zone of the helicopter frame from the cell geometry using an “electronic knife” (from [21]).

The cutting planes of the “electronic knife” at the leading and trailing edge of the construction zone and the resulting fuselage and cabin cut segments are highlighted semi-transparently in yellow. From the cut out construction zone a wire-network is derived with further cuts of the installation space geometry to guide the construction of specific profile-type frames. To account for irregular input geometries, an intermediate processing step is applied to the cut wires to guarantee a continuous shape of the curve. This is important to avoid errors in the subsequent algorithmic geometry generation. The wire-frame models of the algorithmically generated profiles are shown in Figure 5.

For other designs, a somewhat more complex, individually adapted wire network is required. For this reason, the wire meshes are derived individually each time at runtime in the problem context. The wire mesh is used in profile-specific rules stored in the design language to design the shell model of the frame. This shell geometry is then used in a structural calculation, from which a material thickness is calculated. The interface, and execution of, the structural analysis is described in detail in Section 3.2. The wire-frame is subsequently used to construct the solid body of the frame. In the automated geometric calculation, the wire-frame serves specifically as a boundary curve and as a guide curve for several sweep and extrusion operations. Figure 3 shows all three implemented profile types. Despite their different forms, they share common inherited parameters, such as width, height and thickness. They also share the general design idea, where the lower and

upper parts are only dependent on the profile parameters, the center elements (dotted lines in Figure 3) can vary to adapt to the derived installation space.

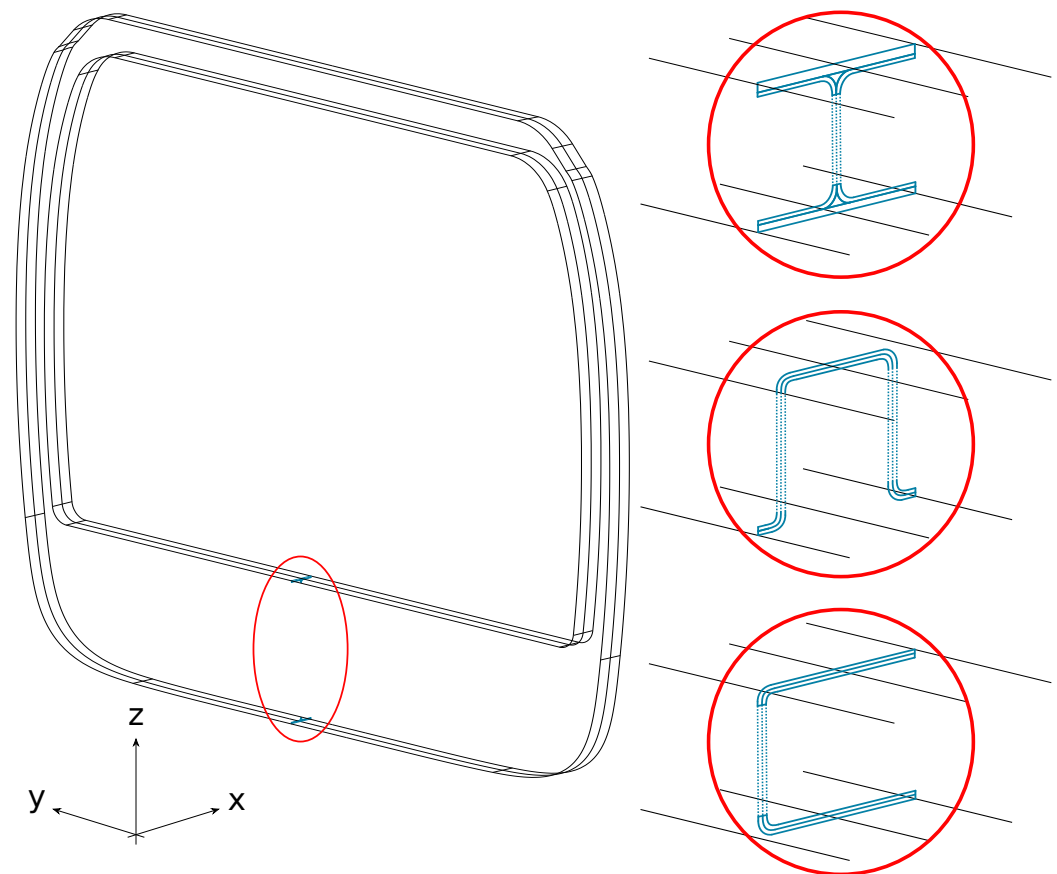


Figure 5. Wire-frame models for the frame profiles incorporated into the construction zone (partly from [21]). From top to bottom: “I”-Profile, “Ω”-Profile, “C”-Profile. Dashed lines have adjustable lengths.

The result of the algorithmic generation of frame geometry, depending only on user input for the desired profile type and parameters and frame position along the x-axis, is shown in Figure 6. A typical configuration would have multiple frames along the cabin to provide sufficient stiffening properties for the shell structures. With the methodology presented here, it is also possible to form several frames at different locations in one design run, if desired also in different designs, materials and manufacturing processes. To illustrate these statements, Figure 6 shows three different frames in the fuselage at different locations and in different profile designs from one program run.

The three profile types defined in this example (see Figure 3) allow three topologically different variants of frames to be defined by the design language. In addition, the parameters of the individual profile shapes can be varied almost arbitrarily in theory, which leads to an increase in the number of variants.

For the generation of the frames shown in Figure 6, the profile parameters were largely identical. In Figure 7 detailed views of the different frame construction methods are shown. To emphasize the profile shape, a section was placed in the upper area, where the profiles are shown in cross-section in Figure 3.

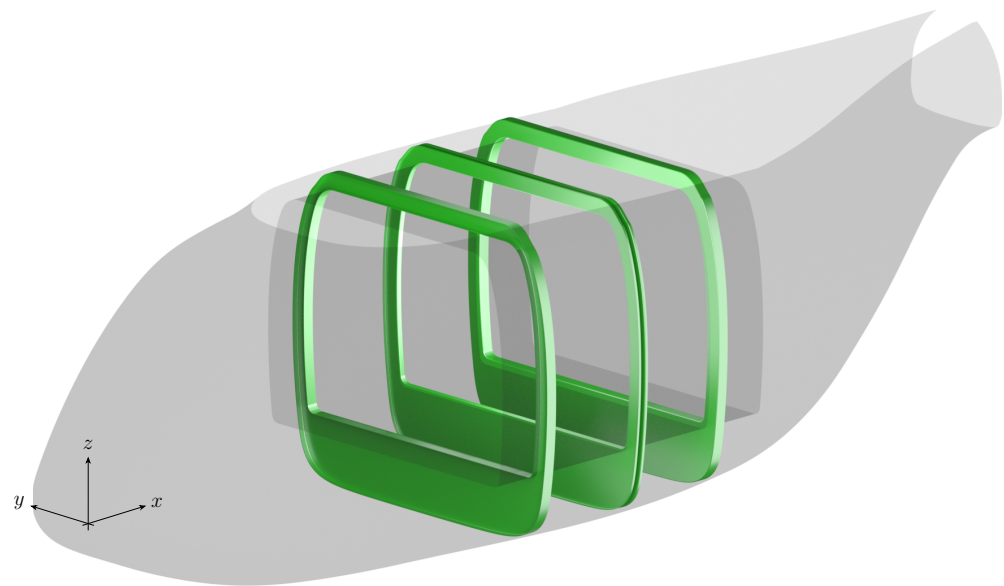


Figure 6. Different variants of helicopter frames at different positions within the fuselage geometry (from [21]). From front to back: “I”-Frame, “ Ω ”-Frame, “C”-Frame.

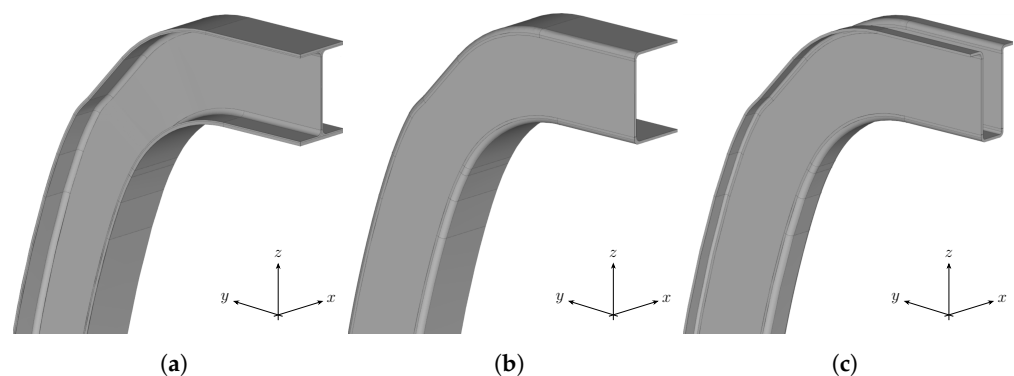


Figure 7. Detailed view of the various fully automatically generated frame variants (from [21]). Shown are: (a) “I”-Frame. (b) “C”-Frame. (c) “ Ω ”-Frame.

The use of design languages for the construction of the fuselage frame brings a high degree of flexibility to the design. Different types of construction can be easily examined and weighed against each other. In addition, changes in the boundary conditions—for example, the position of the frame in the helicopter structure, the material used, the manufacturing process or the profile geometry—can be transferred fully automatically to a new product. The total duration of the automated generation sequence of a frame is less than one minute for each profile design on a mobile workstation from 2023. (This is, specifically, the model: ThinkPad T14s, 2023, CPU: AMD Ryzen 7 PRO 6850U, 2701 MHz, 8 cores, 16 logical processors, RAM: 32 GB.) The immense gain in speed compared to the classic design process by hand in CAD software means that the future development of design tasks dependent on geometry synthesis can be carried out much more efficiently thanks to faster testing, but also that significantly more variants can be calculated in a shorter time. In combination with a downstream simulation, such as a mechanical stress analysis, this results in a very efficient way of weighing up a comprehensive selection of variants against each other and evaluating them with regard to one or more assessment parameters.

3.2. Structural Design: CPACS Processing, Export and Import

The structural pre-design of the helicopter air-frame is carried out using the design tool PANDORA (“Parametric Numerical Design and Optimization Routines for Aircraft” [23]) developed at DLR. PANDORA enables the structural design and optimization of the layer structure (material thickness, layer orientation, etc.) of composite materials using FEM calculations. As per the data-exchange format between the graph-based design language and the PANDORA v1.0 software, the CPACS file format is to be used. The abbreviation CPACS stands for “Common Parametric Aircraft Configuration Schema” [24] and is an XML-based, open-source data-exchange format for aircraft data, which was also developed at DLR. In order to use the geometries and material parameters generated in the presented design language in PANDORA, an interface to the CPACS data format version 3.4 was implemented, which supports both the import and export of CPACS data.

The geometries and materials of the helicopter air-frame are modeled in the design language for geometrical design with their own ontology (see Section 3.1). A converter was developed in addition to the CPACS interface, which extracts the relevant information, like profile geometry, installation space and material selection, from the design language for geometrical design and translates it into a CPACS-compliant ontology. The CPACS file generated by the interface can then be read into PANDORA. After optimization in PANDORA, the CPACS file is augmented with the optimized layer structure information. The CPACS file is then returned and imported by the interface, and the converter transfers the relevant information (layer structure, profile thickness, etc.) back into the ontology of the design language. Currently, only monolithic structures are integrated in the component geometry design and structural design. The design language for production planning described later, however, also takes sandwich structures into account. The incorporation of sandwich structures here could easily be achieved by extending the design language models. This requires an extension of the ontologies and the definition of additional rules for geometry generation and CPACS export and import.

Figure 8 shows a part of the class diagram for the structural design and the CPACS converter. For better comprehensibility, selected classes in the ontology are linked to the corresponding geometric characteristics in the geometry visualization using arrows. The geometric representation of the fuselage in the generated CPACS file can be visualized using the TiGL Viewer v3.2.3 [25] software.

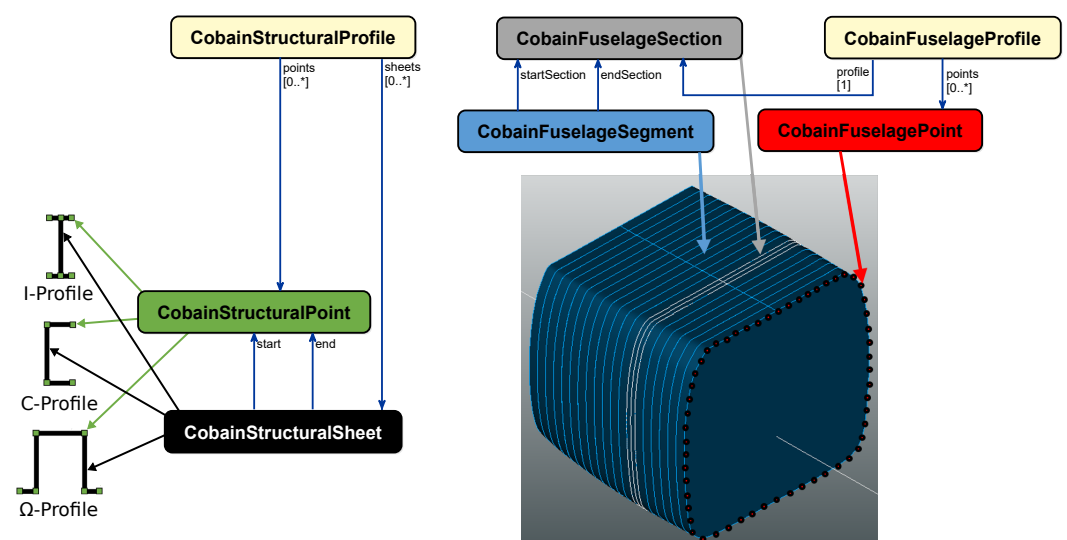


Figure 8. Selection of CPACS interface class diagram with visual representations. According to the UML standard, arrows represent associations. Numbers in square brackets indicate multiplicity, where the * symbol denotes a multiple multiplicity greater than 0.

The data required for the optimization in PANDORA consists of the composite materials to be used, the geometric and structural representation of the components, and the forces acting on the structure. To define the helicopter geometry in the CPACS data format, the outer skin of both the helicopter fuselage and the helicopter cabin is cut several times in the y-z plane. The resulting profiles are defined using points (CobainFuselagePoint), approx. 200) and can be transformed (moved, rotated, scaled) as required. Overall, a fuselage consists of several segments (CobainFuselageSegment), which in turn consist of a start section and an end section (CobainFuselageSection). The sections are each represented by a profile (CobainFuselageProfile). A profile definition can therefore be used for several sections.

The structural elements of the fuselage are represented by frames and stringers, which can reference different section profiles (CobainStructuralProfile) (I, Omega and C). To display these profiles in CPACS format, the structural elements are cut in the plane normal to the guide curve of the structural element, and the skeleton line of the resulting section profile is extracted. This skeleton line is divided into individual sections (CobainStructuralSheet), which are then also represented using points (CobainStructuralPoint) (with start and end points). The profiles calculated in this way are extruded along a guide curve on the outer skin for representation in 3D (for frames at a fixed x-position, i.e., radially along the outer skin; for stringers along the x-direction from the front of the helicopter to the rear of the helicopter). Figure 9 shows the CPACS-compliant wire-frame model prepared for export to CPACS. In the design language, the converter applies transformation rules that map the developed ontologies and class diagrams for structural geometry into CPACS-compliant models. This allows the automatic creation of interface-compatible geometry representations for export.

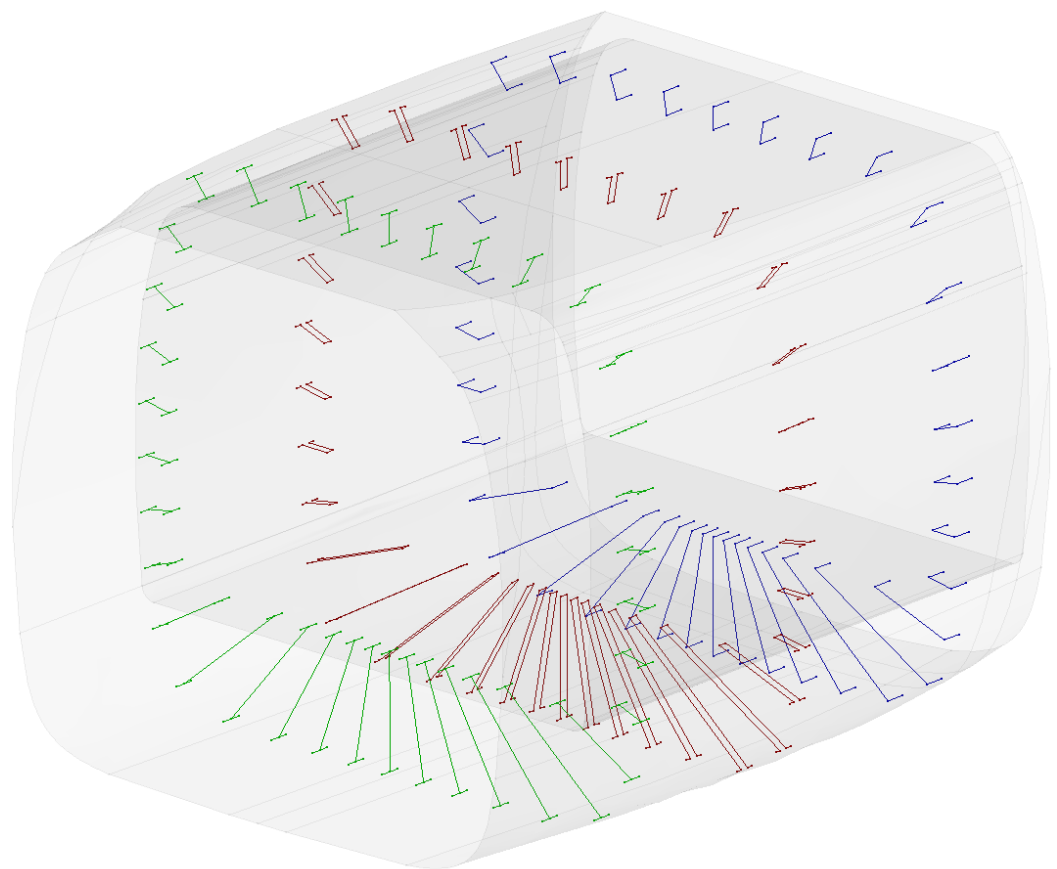


Figure 9. Automatically generated profiles of “I”-Frame (green), “Ω”-Frame (red), and “C”-Frame (blue) in the design language for CPACS export.

Figure 10 shows the design graph after the design language has been executed. The blue nodes represent the points derived from the converter during preprocessing (see “Part”). Internally, all geometric entities, including each derived point, are saved as a separate geometric part. Each cluster represents a section for the structure of the body geometry. These are represented internally as assemblies. The green nodes represent the associated CPACS points (see “CobainFuselagePoint”) in the interface-compliant representation. The entire interface-compliant model is represented as a graph with colored (in the figure partly hidden) points. In the same way, the structural elements such as the frame are also represented graphically in the data model.

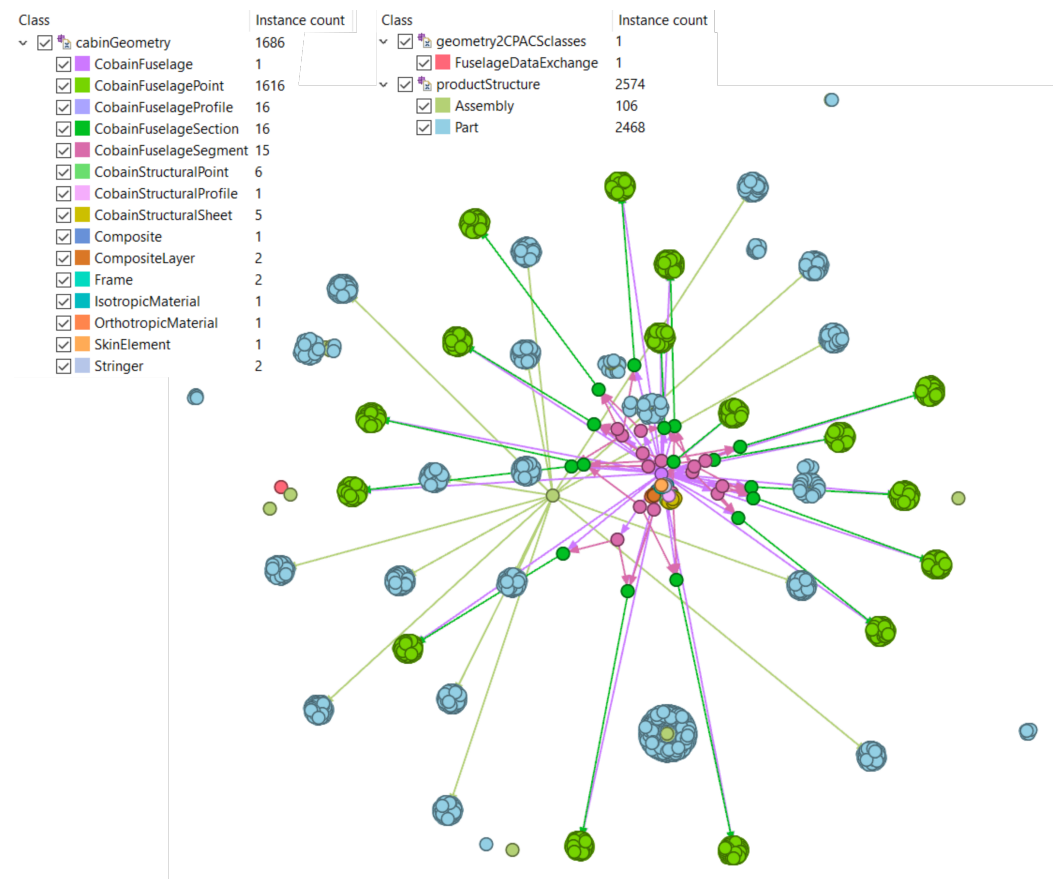


Figure 10. Design graph after execution of the design language for the CPACS export of the fuselage.

The design language enables the automated generation of fuselage frame geometries based on installation space, profile type, and boundary conditions. Wire-frame and shell models are derived, which are subsequently exported via the CPACS interface for structural analysis and converted into solid representations. This establishes a continuous workflow from geometry definition to structural optimization and, later, to manufacturing process modeling (see Figure 1).

The transition from geometry to manufacturing, however, cannot yet be performed fully automatically for all cases. For the presented frame structures, a rule-based subdivision into plies was implemented, while for shell-like components an empirical method based on curvature and semi-finished material width was applied. These methods allow for a first estimation of cut pieces and lay-up sequences, but do not fully reflect industrial practice. This limitation is inherent to the preliminary design stage, where detailed ply books and lay-up strategies are not yet available. The proposed approach therefore provides approximate, rule-based estimations at an early stage, while the manufacturing design language is capable of handling more detailed component descriptions once available.

In this way, the methodology ensures overall continuity between design and manufacturing, while at the same time acknowledging that the levels of detail represented in the two domains differ. The presented bridging approach illustrates how early design data can already be leveraged for manufacturability assessments, even if the transition is not yet fully optimized.

4. Application to Manufacturing Planning

The final step of the integrated process framework is manufacturing planning. In this section, two alternative methods are modeled in their respective design languages: prepreg hand lay-up and automated fiber placement (AFP). Both are fully integrated into the overall data flow and are designed to import automatically generated geometries together with additional information from the design graph. The two methods demonstrate the application of the integrated process model in complementary ways.

The modeling of the prepreg hand lay-up process shows that even established, manual manufacturing methods can benefit from digital formalization. By embedding expert knowledge into an ontology, the model enables early assessment of manufacturing consequences already during the design phase. This allows designers to flexibly respond to variations in component geometry, material choice, and process configuration, while ensuring that manual processes are systematically integrated into the overall workflow.

In contrast, the AFP model represents a further step toward full digital continuity. Here, the process is not only evaluated digitally but directly translated into a complete digital representation of manufacturing. From this, machine trajectories and NC code are generated and can be executed on the production system. This demonstrates how the integrated framework accommodates both traditional manual processes and highly automated digital manufacturing strategies.

4.1. Prepreg Hand Lay-Up

The prepreg hand lay-up process remains one of the most widespread manufacturing methods for fiber-reinforced polymer components in aerospace. Its attractiveness lies in the achievable structural quality, the suitability for small and medium production volumes, and its flexibility for producing complex geometries. However, the process is highly labor-intensive and involves a large number of interdependent steps that significantly affect the production process time, cost and resources.

To address this challenge, a dedicated design language was developed to model the prepreg hand lay-up process within the graph-based design language (GBDL) framework. The purpose is not to generate detailed production schedules but to enable rapid, reliable estimates of key evaluation criteria during early design stages. The design evaluation criteria to be represented in the model are specified as lead time, labor time, recurring costs (subdivided into material, labor, auxiliary, and energy costs), waste, and energy consumption. These estimates allow designers to understand trade-offs between geometry, material, and process choices, and to make better-informed decisions when exploring alternatives. In this way, the model bridges the gap between conceptual design and realistic production constraints.

The design language incorporates information from the component geometry as well as recorded data from the actual manufacturing process. This includes resources such as workers, tools, and materials, as well as process-specific parameters like draping speeds, curing times, and similar factors.

In order to process the geometry algorithmically within the module, it must be represented through a structural and geometrical decomposition. The underlying ontology is shown in Figure 11 and illustrates the hierarchical partitioning of the part into preforms,

which are further divided into subpreforms or substructures. The substructures in turn consist of several layers (plies), with specific emphasis on the top and bottom face sheets. Each layer is composed of multiple oriented cuts derived from semi-finished materials. In addition, special forms such as inserts (sandwich cores or gusset fillers) are included. The translation of component geometry into this structure is currently performed manually, since no rule-based approaches for the design of CFRP components are used in industrial practice, and their configurations are highly individual. Simplified approaches for automated ply derivation were implemented within the scope of the project but are not published here. The relevant information for the manufacturing process can be derived from the geometric and structural data stored in the graph, i.e., dimensions, number of layers, cuts, special components etc. Another relevant aspect is how easy or difficult the draping of individual cut pieces is, which can vary significantly depending on the complexity of its shape. Since this has a major impact on both processing time and cost, a complexity factor was introduced based on the proportion of strongly curved areas ($>0.11/m$).

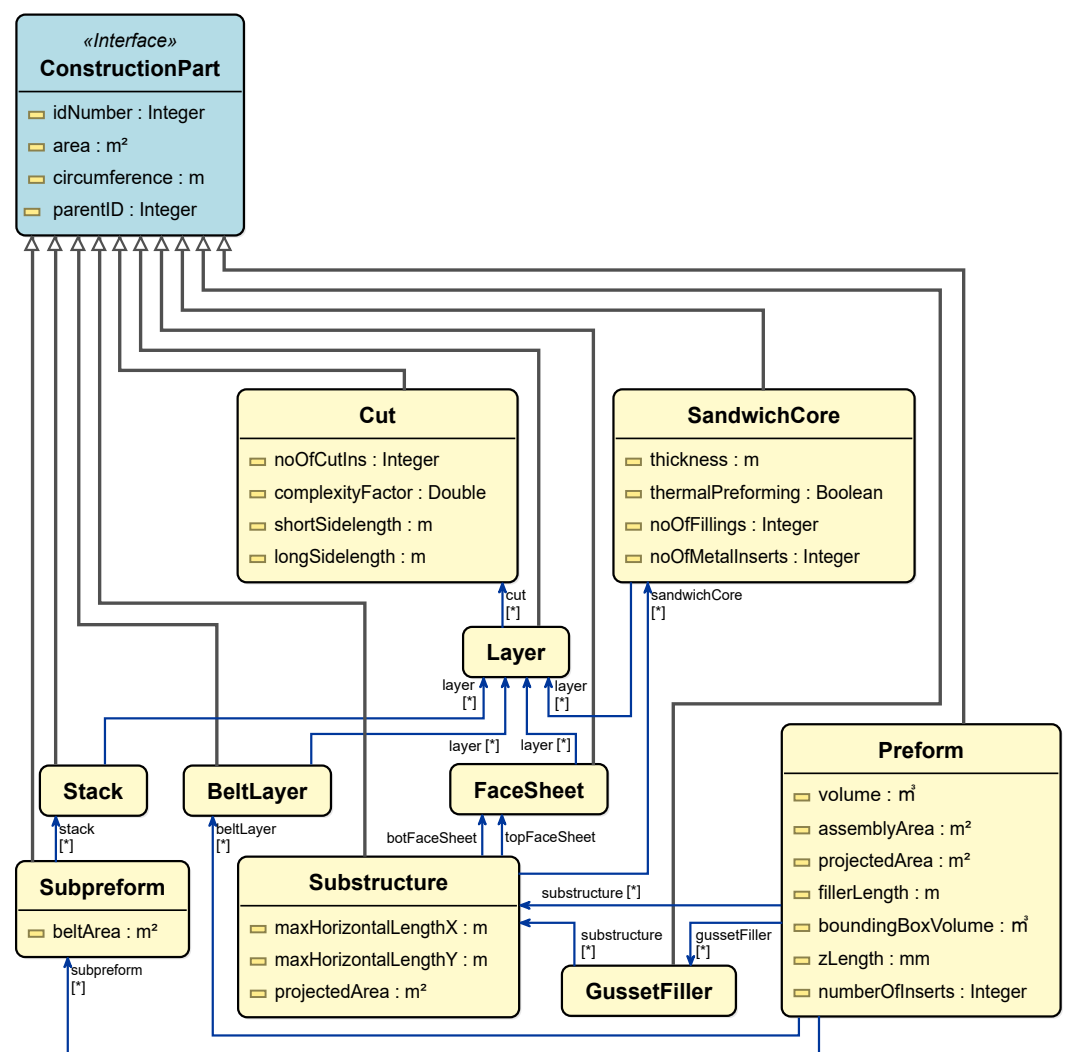


Figure 11. Ontology of the construction parts of the design language for prepreg hand lay-up. The arrows in the figure follow the UML standard. Numbers in square brackets indicate multiplicity, where the * symbol denotes a multiple multiplicity greater than 0.

The modeling approach is based on an ontology that systematically decomposes the overall workflow into main and subprocesses. Each subprocess contains the relevant parameters and their relations to the evaluation criteria within this design language. Calculating the evaluation criteria at the subprocess level enables a higher level of detail in the results

and precise expert-based estimations of the parameters. The definition and estimation of the parameters are carried out in collaboration with production experts and are based on the specific manufacturing workflow at AHD. The process parameters for the prepreg process are embedded in the models of the design language. Geometry and tool parameters originate from the imported geometry or the corresponding forming tool. Machine, operator, and material parameters are input values that can be dynamically adjusted according to user selection and are centrally available in the databases. Parameters are organized into six categories:

- Geometry parameters, such as surface area, ply thickness, and curvature, which are derived from imported geometry.
- Tool parameters, including the size of forming tools, mold surface characteristics, and the length of vacuum channels, estimated from generic tooling guidelines.
- Material parameters, which describe weight, cost, curing cycles, etc., stored in a central database.
- Machine parameters, covering capacities and specifications of cutting tables, autoclaves, and auxiliary equipment.
- Worker parameters, capturing the variability of human work, for example, worker speed, handling efficiency, or level of training.
- Process parameters, which represent general assumptions about the workflow, such as compacting intervals, adhesive film usage, or quality control steps.

An example of this breakdown into subprocess steps is shown in Figure 12 for the laminating subprocess.

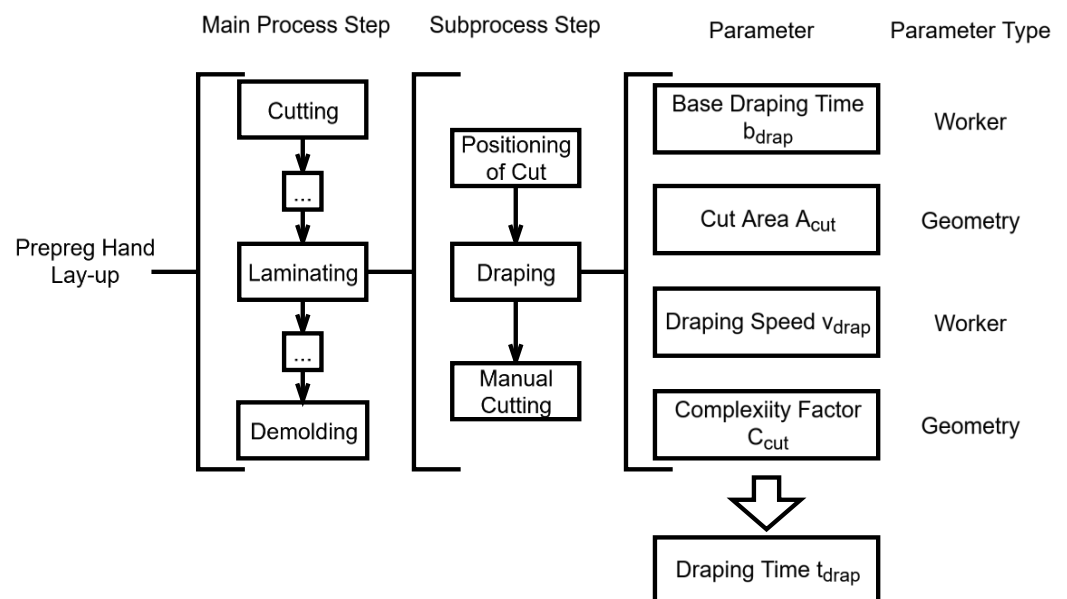


Figure 12. Schematic breakdown of the production process using the example of laminating.

The lamination step itself consists of the subprocess steps positioning the cut piece, draping, and manual cutting. Based on the parameters an empirical relation of lead time for the subprocess step draping was derived by experts. For this example, the equation is as follows:

$$t_{drap} = K_{cut} \cdot (b_{drap} + A_{cut} \cdot v_{drap}), \quad (1)$$

Here, t_{drap} denotes the lead time to be calculated for the subprocess step. The parameters b_{drap} and v_{drap} represent the base draping time and the draping speed of the worker. From the geometry, the surface area A_{cut} and the complexity factor K_{cut} of the cut

piece currently being processed are required. The sum of the individual lead times of the subprocess steps constitutes the total lead time of the lamination process step. Analogous to this example, a complete specification of all parameters and their empirical equations is provided for the tasks within the prepreg process.

The calculation of process steps follows the real manufacturing sequence in chronological order. The processing order in the model reflects actual production, ensuring, for example, that the lamination of the third ply is only computed after the second. This is necessary for later analyses of potential parallelization. The specific manufacturing sequence in the prepreg hand lay-up process strongly depends on the component geometry. For instance, the process steps of lamination and compacting occur with varying repetitions depending on the component, since each cut piece requires a separate lamination step. Compacting is carried out after every five prepreg plies. Thus, preceding and subsequent process steps change, and the specific process sequence is dynamically adapted by the model according to the component geometry. In addition, process variations are possible within individual steps, such as the decision to use reusable vacuum hoods versus disposable vacuum bags, or adding optional steps like polishing, adhesive films, and quality control. The program flow chart of the dynamically created process flow is shown in Figure 13.

The process strongly depends on forming tools. Since prepreg plies are laminated directly onto the mold, tool parameters (e.g., surface or vacuum channel length) and availability influence the workflow. Instead of detailed component-specific models, generic tools are used based on AHD's guideline for lay-up fixtures, from which relevant parameters are derived. Autoclave cycles are modeled using supplier curing cycle specifications (heating/cooling rates, dwell times, curing temperatures), rather than detailed thermal simulations.

When execution of the process model is complete, the compiler aggregates the outputs from all subprocesses into the graph. These results are provided at two levels of detail. At the subprocess level, times, costs, energy use, and waste are reported for each individual activity, such as draping a ply, compacting a stack, or curing in the autoclave. This granularity allows engineers to identify which steps dominate overall production time or cost and to explore potential optimization strategies. At the overall level, the model aggregates values for the entire component. Lead time, labor effort, recurring costs, total energy demand, and cumulative waste are reported as key indicators. This dual-level reporting enables both quick feasibility assessments and detailed investigations of bottlenecks or cost drivers. The results can also be exported in Excel format for further analysis in established workflows. This makes it possible to combine model outputs with cost accounting systems, scheduling tools, or sustainability assessments already in use in industrial settings.

A central advantage of the graph-based approach is that the design graph retains all intermediate values. This means that recalculations with modified parameters—such as a different material, an alternative tooling concept, or adjusted operator speeds—can be carried out rapidly without reconstructing the entire model. As a result, the hand lay-up design language supports efficient iteration and enables systematic exploration of alternative manufacturing scenarios during early design phases.

The design language model for the prepreg hand lay-up process was validated using real-world data from AHD's production. Qualitatively speaking, the predictions from the design language were quite close to the real-world data. Quantitative statements cannot be published here, as they are subject to the NDA with AHD. Since the translation of component geometries into the CFRP structure is currently performed manually, further deviations are to be expected due to the simplifications inherent in the modeling process when the automated, rule-based translation of the component geometry into the CFRP structure is incorporated.

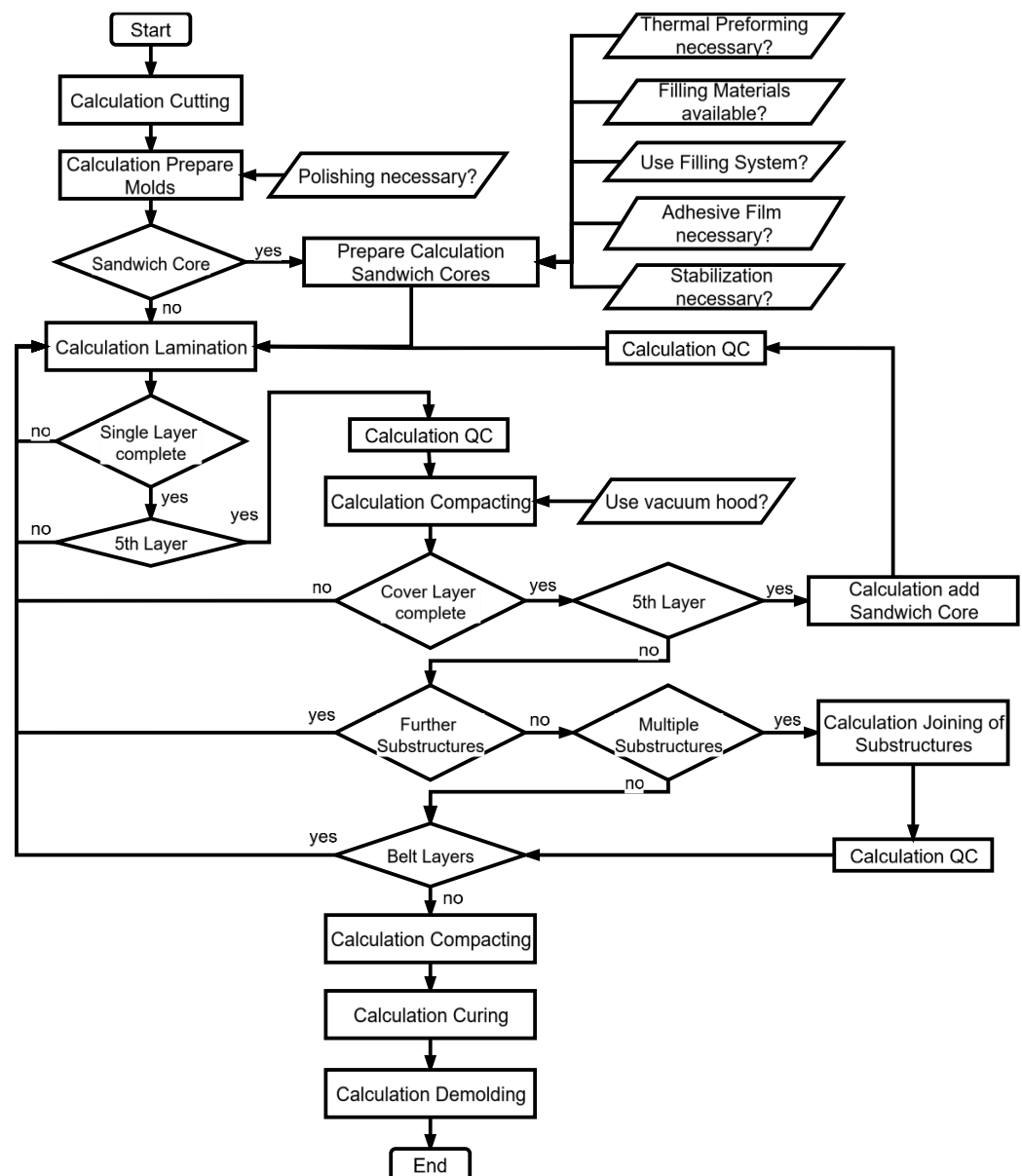


Figure 13. Program flow chart for calculating the process steps for prepreg hand lay-up.

4.2. Automated Fiber Placement

Automated fiber placement (AFP) is a fully automated alternative to hand lay-up. It is a state of the art technology for manufacturing stiffened structures such as monolithic [26] or sandwich composite shells [27]. It can process thermoset CFRP materials, as well as thermoplastic matrices such as Low-Melt PAEK (LMPAEK). The choice of material, shape, fiber trajectories and process parameters has major implications on the quality [28,29] and cost [30] of the manufactured component. Integration of design and manufacturing of AFP is thus a frequent research topic [31]. The process chain has been modeled in-depth with a graph-based design language at the Fraunhofer Institute of Casting, Composite and Processing Technology (IGCV) in Augsburg, Germany. Process planning can range from a global model of the entire process chain to very detailed models of individual processes. Graph-based design languages allow one to capture different levels of abstraction consistently. In the presented approach, a global model of the AFP process chain is generated through a sequence of model transformation rules. In addition, a detailed model of the automated lay-up process itself is implemented. The detailed model serves as an input for

an automated machine programming workflow, which results in an executable NC code which is ultimately tested in a real-world experiment at Fraunhofer IGCV.

NC program generation for AFP requires a multi-staged workflow. The domain-specific CAD definition uses shells, wires and splines to define the product. This definition is stored in an abstract design graph, as displayed in Figure 14. An Afp component consists of multiple plies, each with an orientation and thickness. It has a tool surface, represented by a shell geometric entity. Every ply has a zone which defines its shape through a boundary wire which must lie on the tool surface. In addition, every zone has a reference curve defining its 0° fiber direction on the curved tool surface. In the presented example, all plies have the same zone and thus identical outer dimensions. However, the model does allow for every ply to have an individual thickness, orientation and zone. The superposition of all zones results in the global boundary of the preform, termed ‘Manufacturing Edge Of Part’ (MEOP).

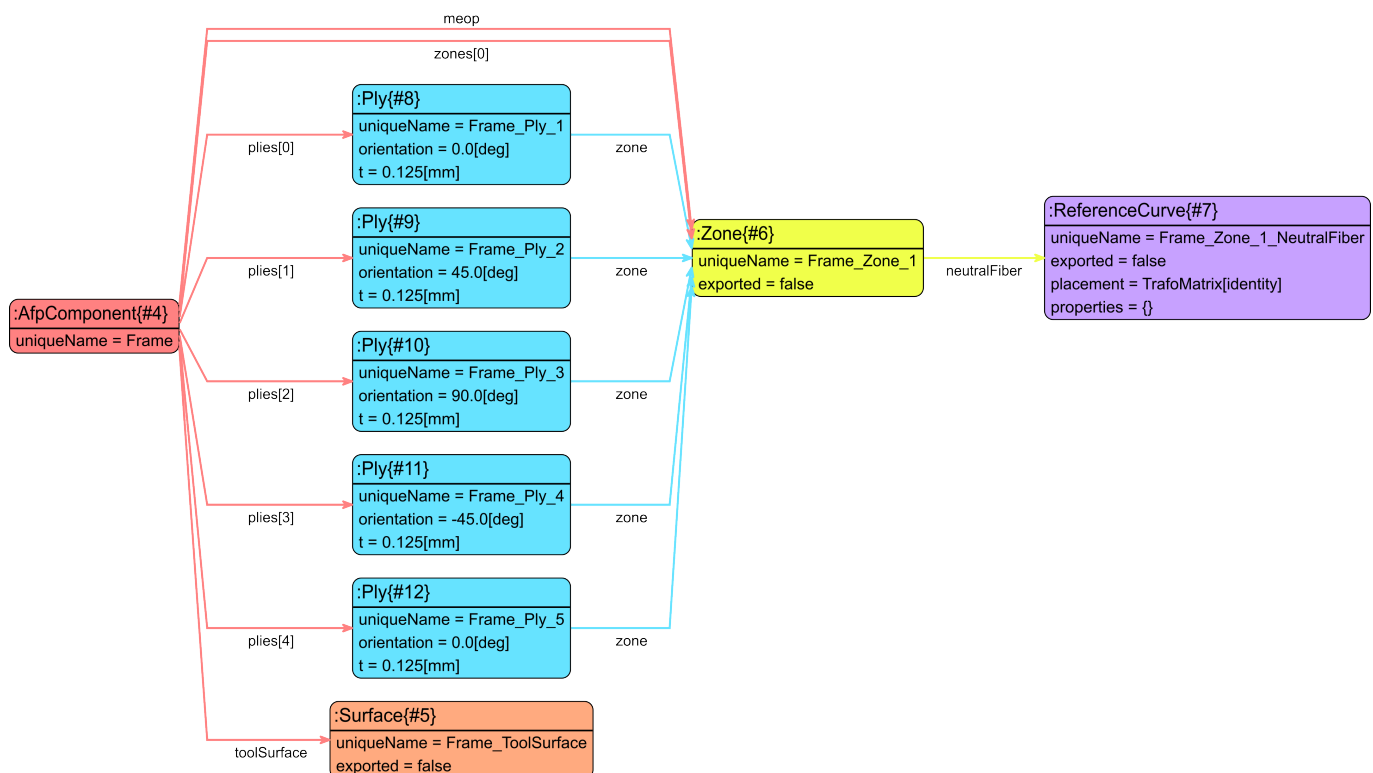


Figure 14. Design graph of an AFP component. Arrows indicate linked nodes. Numbers in square brackets indicate multiplicity.

The generation of individual guide curves for the lay-up process faces multiple challenges in the case of doubly curved surfaces. These include:

- Fiber angle deviation: The fiber angle is only well defined at the intersection of the guide curve and the reference curve.
- Fixed tow width: Tows have a constant width, so that guide curves within one ply must be parallel. But this may result in a high fiber angle deviation. Otherwise, triangle gaps and overlaps appear to compensate for inclined guide curves, affecting the structural performance [32].

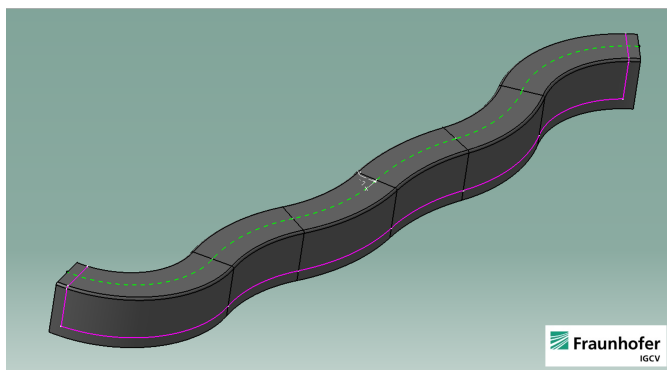
Minimizing the fiber angle deviation and gaps/overlaps at the same time is not possible. Instead, different sectorization and guide curve construction strategies can be applied to achieve a compromise. The following rules have been implemented:

- The surface is partitioned into several sectors, each with its individual guide curve.

- Every sector has one guide curve per fiber direction (e.g., 0° , 45° , -45° , 90°). All tows within the sector are placed in parallel.

As a result, gaps and overlaps form at the interface of sectors but not within one sector. Fiber deviations occur outside of the global reference curve and the guide curve of every sector. Smaller sectors tend to reduce fiber angle deviation, but increase gaps/overlaps. These rules are implemented in a modular macro workflow in CATIA V5. They are executed after exporting the DC43 v4.0.24 design graph into step files and an annotation text file, and importing it into CATIA, cf. Figure 15a. The sectorization and guide curve construction is tested with an s-shaped trial tool. In Figure 15b, the resulting sectors are colored light blue, and their guide curves (one per orientation, four orientations per sector) are colored magenta. In this example, the component is partitioned into 11 sectors.

(a) 3D CAD model of AFP component



(b) Generated sectors and guide curves

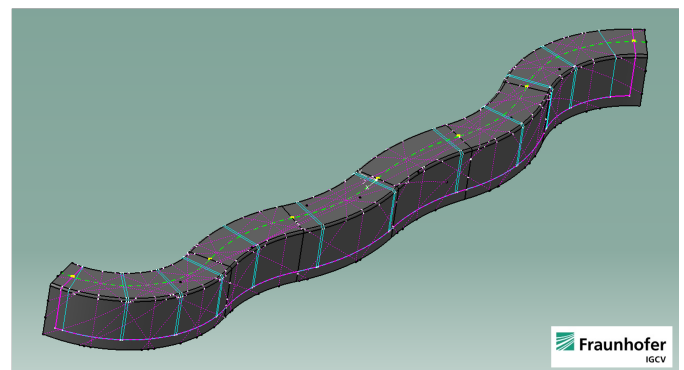


Figure 15. Sectorization and guide curve construction for automated fiber placement. Displayed elements include MEOP (magenta, full line), neutral fiber (green, dashed), sector boundaries (blue), and generated guide curves per sector (magenta, dashed).

Based on the guide curves, the tow paths and NC machine code are generated using the Coriolis CATfiber platform. The resulting NC code is verified with a machine simulation and consequent lay-up experiments at the Coriolis C1.2 fiber placement machine at Fraunhofer IGCV in Augsburg. Figure 16 shows the lay-up of one sector in the CATfiber machine simulation and the lay-up experiment in the lab.

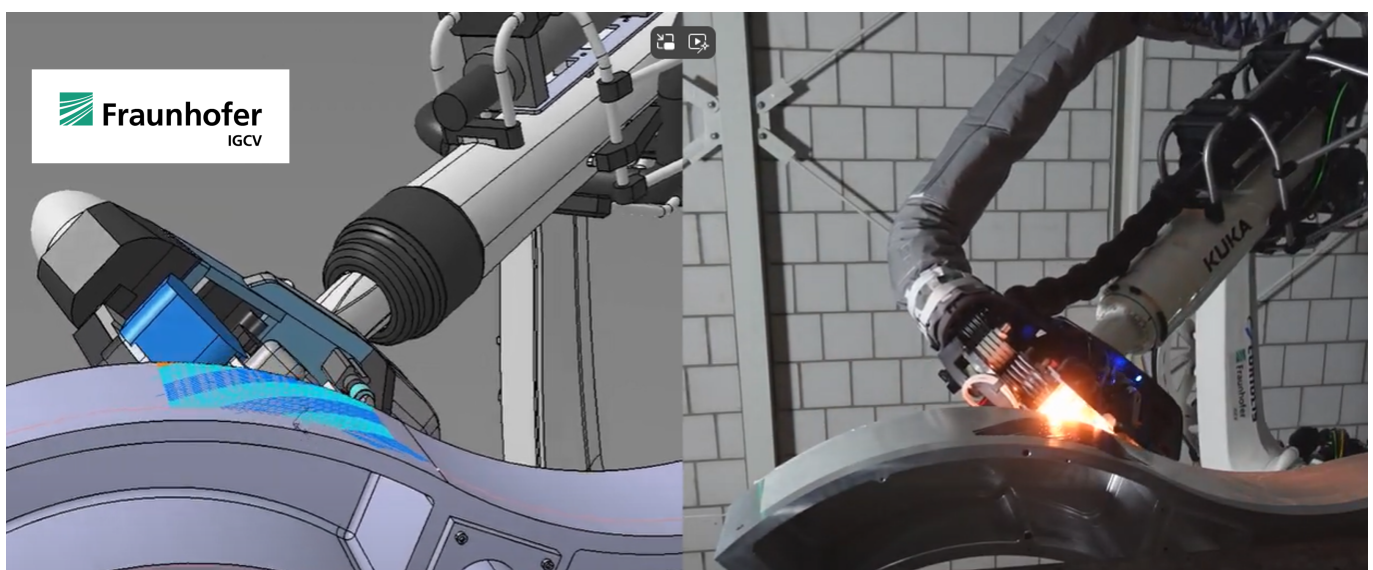


Figure 16. AFP tape simulation and lay-up.

Verification specifically includes lay-up trials of different sectors and orientations. In Figure 17, the lay-up of three sectors can be seen. The leftmost has a 45° orientation, and the two other sectors to the right have 90° orientations, although their inclination is different due to the curved reference curve. On the top of the tool, a 0° ply is laid along the reference curve. The trial runs demonstrate that it is possible to automatically generate a real-world machine control code based on a design graph.



Figure 17. AFP lay-up experiment at Fraunhofer IGCV.

5. Conclusions

This paper has presented a graph-based modeling framework that integrates the design and manufacturing of CFRP helicopter structures. The approach captures the geometry, material, tooling, and process parameters within an integrated design language, enabling the dynamic generation of manufacturing process chains at runtime. By linking relevant subprocesses with evaluation criteria such as lead time, labor effort, costs, energy consumption, and waste, the framework provides a consistent and extensible foundation for assessing production alternatives already during early design stages.

The current contribution is primarily methodological. The focus lies on demonstrating how heterogeneous inputs can be systematically represented, interconnected, and evaluated through rule-based process modeling. While detailed empirical validation and large-scale benchmarking are outside the scope of this work, the framework is intentionally designed to accommodate such data once available. Its modular structure ensures that additional process knowledge, expert data, or industrial case studies can be incorporated seamlessly to fit a manufacturer's individual manufacturing process. To give a specific example, the design language for manufacturing planning presented in this paper is limited to prepreg hand lay-up and automated fiber placement due to the nature of the research project; however, it can be easily extended to other novel production processes, such as the use of carbon nanotubes [33] or the machining of CFRP components [12,13]. However, this always requires that both the material parameters for the component design and the process parameters for production planning are known and can be quantitatively integrated into the model.

The helicopter fuselage frame example illustrates the feasibility of the approach and serves as a proof of concept for its applicability to complex aerospace components. By showing how dynamic process models can be automatically created from design inputs, the

study highlights the potential of graph-based design languages to bridge the gap between conceptual design and realistic production planning.

Future work will therefore concentrate on extending the database of process parameters with empirical data from industrial applications, refining the complexity factors for draping, and validating the estimated evaluation criteria against real-world production scenarios. In this way, the proposed framework lays the methodological foundation for a more comprehensive model-based integration of design and manufacturing in the aerospace industry.

Author Contributions: Conceptualization: S.R. and C.S.; software and validation: C.S., D.S., J.D., M.H. and J.S. writing—original draft: C.S. and M.H.; writing—review editing: J.D. and S.R.; funding acquisition: S.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research is based on the results of the COBAIN research project, conducted within the framework of the German National Aviation Research Program (“Luftfahrtforschungsprogramm”, LuFo), part of the Federal Ministry for Economic Affairs and Climate Action (BMWK), grant numbers: 20W1908B (IILS mbH), 20W1908D (IFB), and 20W1908E (IGCV).

Data Availability Statement: The datasets presented in this article are not readily available due to confidentiality agreements with the industry partner. Requests to access the datasets should be directed to the corresponding author.

Acknowledgments: We gratefully acknowledge the German Federal Ministry of Economic Affairs and Climate Action (BMWK) for the funding. We would like to thank Airbus Helicopters for providing the opportunity of participation and the information to comprehend the helicopter design and manufacturing process during the project.

Supported by:



on the basis of a decision
by the German Bundestag

Conflicts of Interest: Authors Maximilian Holland and Julian Schuster were employed by Fraunhofer Institute of Casting, Composite and Processing Technology IGCV. Author Julian Dinkelacker was employed by IILS Ingenieurgesellschaft für Intelligente Lösungen und Systeme mbH. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GBDL	Graph-Based Design Language
CFRP	Carbon-Fiber-Reinforced Polymer
CPACS	Common Parametric Aircraft Configuration Schema
STEP	Standard for the Exchange of Product model data
AFP	Automated Fiber Placement

References

- Patole, R.; Ambhore, N.; Agrawal, D. Carbon Composites in Aerospace Application—A Comprehensive Review. *Mater. Int.* **2023**, *5*, 12. [CrossRef]
- Toyoda, D.; Wärmefjord, K.; Söderberg, R. Challenges in Geometry Assurance for Composites Manufacturing. *J. Comput. Inf. Sci. Eng.* **2023**, *23*, 051007. [CrossRef]
- Brasington, A.; Sacco, C.; Halbritter, J.; Wehbe, R.; Harik, R. Automated fiber placement: A review of history, current technologies, and future paths forward. *Compos. Part C Open Access* **2021**, *6*, 100182. [CrossRef]
- Zhang, L.; Wang, X.; Pei, J.; Zhou, Y. Review of automated fibre placement and its prospects for advanced composites. *J. Mater. Sci.* **2020**, *55*, 7121–7155. [CrossRef]
- Yamaguchi, K.; Phenisee, S.E.; Chen, Z.; Salviato, M.; Yang, J. Ply-drop design of non-conventional composites using Bayesian optimization. *arXiv* **2020**, arXiv:2007.08757. [CrossRef]
- Pourabdollah, P.; Finger, L.; Alhourani, R.; Frerich, T.; Höfer, R.; Gehlhoff, F.; Kriegesmann, B. Automated composite design generation and optimization using variable angle tow plies for aerospace applications. *Compos. Struct.* **2025**, *372*, 119511. [CrossRef]
- Frketic, J.; Dickens, T.; Ramakrishnan, S. Automated manufacturing and processing of fiber-reinforced polymer (FRP) composites: An additive review of contemporary and modern techniques for advanced materials manufacturing. *Addit. Manuf.* **2017**, *14*, 69–86. [CrossRef]
- Swery, E.; Hans, T.; Bultez, M.; Wijaya, W.; Kelly, P.; Hinterhoelzl, R.; Bickerton, S. Complete Simulation Process Chain for the Manufacturing of Braided Composite Parts. *Compos. Part A Appl. Sci. Manuf.* **2017**, *102*, 378–390. [CrossRef]
- Deden, D.; Frommel, C.; Glück, R.; Larsen, L.; Malecha, M.; Schuster, A. Towards a fully automated process chain for the lay-up of large carbon dry-fibre cut pieces using cooperating robots. In Proceedings of the SAMPE Europe 2019. SAMPE Europe Conference 2019, Nantes, France, 17–19 September 2019.
- Jayasekara, D.; Lai, N.; Wong, K.H.; Pawar, K.; Zhu, Y. Level of automation (LOA) in aerospace composite manufacturing: Present status and future directions towards industry 4.0. *J. Manuf. Syst.* **2021**, *62*, 44–61. [CrossRef]
- Björnsson, A.; Jonsson, M.; Johansen, K. Automated material handling in composite manufacturing using pick-and-place systems—A review. *Robot. Comput.-Integr. Manuf.* **2018**, *51*, 222–229. [CrossRef]
- Esch, P.; Frommknecht, A.; Niemeier, W. Integrierte Komplettbearbeitung von CFK-Struktur- und Großbauteilen. *ATZproduktion* **2020**, *7*, 10–17. [CrossRef]
- Esch, P.; Gebhardt, A.; Tiedje, O.; Frommknecht, A. Integrated Machining, Quality Inspection and Sealing for CFRP Components. In *Advances in Automotive Production Technology—Theory and Application*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 377–385. [CrossRef]
- Campbell, F.C. *Manufacturing Processes for Advanced Composites*; Elsevier: Amsterdam, The Netherlands, 2003.
- Vogel, S.; Rudolph, S. Complex System Design with Design Languages: Method, Applications and Design Principles. *arXiv* **2018**, arXiv:1805.09111. [CrossRef]
- Kiefer, J.; Breckle, T.; Stetter, R.; Manns, M. Digital assembly planning using graph-based design languages. *Procedia CIRP* **2018**, *72*, 802–807. [CrossRef]
- Kübler, K.; Schopper, D.; Riedel, O.; Rudolph, S. Towards an Automated Product-Production System Design-Combining Simulation-based Engineering and Graph-based Design Languages. *Procedia Manuf.* **2020**, *52*, 258–265. [CrossRef]
- Braiger, J.; Baur, J.; Gugliuzza, J.; Rudolph, S.; Carosella, S.; Middendorf, P. Design Automation of Fibre Composite Parts via Graph-Based Design Languages. In *Proceedings of the Advances in Automotive Production Technology—Digital Product Development and Manufacturing*; Holder, D., Wulle, F., Lind, J., Eds.; Springer: Cham, Switzerland, 2025; pp. 344–352.
- Holland, M.; Paul, N.; Linder, C.; Elsafty, H.; Ernis, G.; Geinitz, S. Model-driven Approach for integrated Design and Process Planning of Fiber Composite Aerostructures. In Proceedings of the Society for the Advancement of Material and Process Engineering (SAMPE Europe Conference), Madrid, Spain, 3–5 October 2023.
- Object Management Group (OMG). *Unified Modeling Language: Superstructure*, Version 2.5; Object Management Group (OMG): Needham, MA, USA, 2015. Available online: <http://www.omg.org/spec/UML> (accessed on 9 February 2026).
- Schopper, D. Round-Trip-Engineering im Modellbasierten Ingenieurentwurf. Ph.D. Thesis, Institut für Flugzeugbau, Stuttgart, Germany, 2024. [CrossRef]
- Anderl, R.; Trippner, D. (Eds.) *STEP—Standard for the Exchange of Product Model Data: Eine Einführung in die Entwicklung, Implementierung und Industrielle Nutzung der Normenreihe ISO 10303 (STEP)*; Teubner: Stuttgart, Germany, 2000.
- Petsch, M.; Kohlgrüber, D.; Heubischl, J. PANDORA—A python based framework for modelling and structural sizing of transport aircraft. *MATEC Web Conf.* **2018**, *233*, 00013. [CrossRef]
- Deutsches Zentrum für Luft- und Raumfahrt e.V. CPACS: *Common Parametric Aircraft Configuration Schema*, Version 3.4; DLR Institute of System Architectures in Aeronautics: Hamburg, Germany, 2022. Available online: <https://www.cpacs.de/> (accessed on 9 February 2026).

25. Siggel, M.; Kleinert, J.; Stollenwerk, T.; Maierl, R. TiGL: An Open Source Computational Geometry Library for Parametric Aircraft Design. *Math. Comput. Sci.* **2019**, *13*, 367–389. [CrossRef]
26. Castanie, B.; Azoti, W.; Crouzeix, L.; Bello, A.; Taborda, R.P.; Mahmood, A.; Viste, A. Review of monolithic composite laminate and stiffened structures in aeronautic applications. *Compos. Part C Open Access* **2025**, *17*, 100585. [CrossRef]
27. Zenker, T.; Jennemann, F.; Wegmann, M. Automated Fiber Placement Based Manufacturing of Carbon Fiber Reinforced Sandwich Helicopter Sideshells. In Proceedings of the Aerospace Europe Conference, Bordeaux, France, 25–28 February 2020. Available online: <https://publica-rest.fraunhofer.de/server/api/core/bitstreams/ac6fd346-580b-42a8-b583-48f1fb347bf5/content> (accessed on 9 February 2026).
28. Seefried, K.; Bezerra, R.; Drechsler, K. Prediction of consolidation quality in laser-assisted AFP and effect of blank quality on stamp forming. *J. Reinf. Plast. Compos.* **2023**, *43*, 1067–1083. [CrossRef]
29. Scheiterlein, K.; Seitenglanz, P.; Wegscheider, D.; Etlijani, O.; Drechsler, K. Qualification Methodology for Automated Fiber Placement and Concept for its Industrialization. In Proceedings of the SAMPE Europe Conference, Belfast, UK, 24–26 September 2024. Available online: <https://www.nasampe.org/store/viewproduct.aspx?id=25366875> (accessed on 9 February 2026).
30. Morse, L.; Mallardo, V. Bottom-up manufacturing cost optimisation of composite aircraft structures: Manual layup vs. automated layup. *AIP Conf. Proc.* **2023**, *2848*, 020027. [CrossRef]
31. Noevere, A.; Collier, C.; Harik, R. Integrated Design and Manufacturing Analysis for Automated Fiber Placement Structures. In Proceedings of the Digital Library: Sampe 2019, Charlotte, NC, USA, 20–23 May 2019.
32. Zenker, T.; Bruckner, F.; Drechsler, K. Effects of defects on laminate quality and mechanical performance in thermoplastic Automated Fiber Placement-based process chains. *Adv. Manuf. Polym. Compos. Sci.* **2019**, *5*, 184–205. [CrossRef]
33. Zecchi, S.; Cristoforo, G.; Piatti, E.; Torsello, D.; Ghigo, G.; Tagliaferro, A.; Rosso, C.; Bartoli, M. A Concise Review of Recent Advancements in Carbon Nanotubes for Aerospace Applications. *Micromachines* **2025**, *16*, 53. [CrossRef] [PubMed]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.