Collaborative Aircraft Engine Preliminary Design using a Virtual Engine Platform, Part A: Architecture and Methodology

S. Reitenbach*, M. Vieweg†, R. Becker‡, C. Hollmann§, F. Wolters¶, J. Schmeink‖, T. Otten**, M. Siggel††
German Aerospace Center (DLR), Cologne, Germany, D-51147

As in many other industries, the sector of aircraft engines and gas turbines is also undergoing a change towards digitalization. The intention is to make digital technologies applicable over the entire life cycle of the product and thus improve planning, design, construction, assembly, operation, and maintenance. Intelligent digitalization technologies like the digital thread or digital twin will drastically change engineering and construction processes. Consequently, the preliminary aircraft engine design must also be embedded into the context of digitalization. As part of the projects PEGASUS and PERFECT, the German Aerospace Center (DLR) has started the development of the virtual engine platform GTlab (Gas Turbine Laboratory). Its modular architecture ensures a high degree of usability, expandability, and flexibility for the design and assessment of innovative next generation engine and gas turbine concepts. The purpose of this paper is to present the most important aspects of the GTlab framework and how they contribute to meet the requirements of preliminary aircraft engine design in the context of digitalization. A central topic is the digital representation of the engine system, which is realized by a central data model approach. This includes the geometric description of all engine components, as well as additional data such as thermodynamics, aerodynamics, structural characteristics and mass breakdown. In addition, the central data model enables an efficient management of the intricate data flow and the extensive amount of data transferred between the different disciplines and fidelity levels during the aircraft engine design. Further functionalities of the GTlab framework include the automated generation of 3-D geometries by means of a CAD kernel interface, the acquisition of material data via a material database and a standardized gas model interface. Besides the core functionalities, GTlab includes three major modules for the preliminary aircraft engine design from 0-D-performance up to 3-D. The detailed collaborative predesign proces by means of the framework is presented in part B [1], exemplary for a ultra high bypass turbofan suited to a middle of the market aircraft configuration.

I. Nomenclature

API	=	Application Programming Interface
CAD	=	Computer Aided Design
CFD	=	Computational Fluid Dynamics
CSG	=	Constructive Solid Geometry
EIS	=	Entry Into Service
FEM	=	Finite Element Method
GTlab	=	Gas Turbine Laboratory
GUI	=	Graphical User Interface
PEGASUS	=	Preliminary Gas Turbine Assessment and Sizing
PERFECT	=	Preliminary design and evaluation of future engine concepts

*Research Associate, Institute of Propulsion Technology, Linder Hoehe, 51147 Cologne, Germany, stanislaus.reitenbach@dlr.de

†Research Associate, Institute of Propulsion Technology, Linder Hoehe, 51147 Cologne, Germany, maximilian.vieweg@dlr.de

‡Research Associate, Institute of Propulsion Technology, Linder Hoehe, 51147 Cologne, Germany, richard.becker@dlr.de

§Research Associate, Institute of Propulsion Technology, Linder Hoehe, 51147 Cologne, Germany, carsten.hollmann@dlr.de

¶Research Associate, Institute of Propulsion Technology, Linder Hoehe, 51147 Cologne, Germany, florian.wolters@dlr.de

‖Research Associate, Institute of Propulsion Technology, Linder Hoehe, 51147 Cologne, Germany, jens.schmeink@dlr.de

**Research Associate, Institute of Propulsion Technology, Linder Hoehe, 51147 Cologne, Germany, tom.otten@dlr.de

††Research Associate, Simulation and Software Technology, Linder Hoehe, 51147 Cologne, Germany, martin.siggel@dlr.de

PLM	=	Product Lifecycle Management
UML	=	Unified Modeling Language
UHBR	=	Ultra-High Bypass Turbofan

II. Introduction

New digital technologies open up numerous opportunities and create new business models covering the entire aviation industry. From airports, airlines, suppliers and aircraft manufacturers, the aircraft engine sector is also affected by the digitalization. This concerns all stages of an engine's life and development cycle. According to [2], the life cycle of an engine can be divided into six stages (Fig. 1). The design process itself can be further subdivided into the three disciplines preliminary design, detail design, and full production design. The field of digitalization supports these processes by providing innovative technologies in the context of the virtual product.

The digital engine refers to the virtual representation of all geometric and physical characteristics of the engine using computer-aided design tools and numerical simulation methods. It supports the entire virtual engine design along the preliminary design phase up to the detailed design covering all disciplines such as thermodynamics, aerodynamics, and structural mechanics. Furthermore, it is the foundation for the virtual certification, which is one of the long-term objectives of the digitalization of the aviation sector [3]. Costs and effort are to be reduced by eliminating the need for real certification tests. However, the simulation techniques must be further developed and interfaces have to be standardized in order to support or even replace this time-consuming and well-regulated process with virtual methods.

In addition to the design, development and certification of an engine, the remaining stages of the life cycle can also be digitalized. The digital replica of the physical product and all its components from production to disposal is called the digital twin [4].

In order to productively utilize the entire range of digital technologies, the continuity and traceability of all relevant data must be guaranteed. The digital thread covers all stages of an engine's life cycle. All decisions, procedures and results should be fed into the digital thread. It represents a comprehensive and consistent source of information. Thus, all relevant data can be completely traced back and can be used in every phase of the product life cycle.

The main focus of this paper is the preliminary design in the context of digitalization. Although the knowledge of the final product is very limited in this stage, decisions have an enormous influence on the subsequent stages of the life cycle especially in terms of design, space and fixed costs [2, 5]. Therefore, digitalization plays an significant role in preliminary engine design and its innovative methods could lead to substantial improvements in the entire design process.

There are various approaches for the virtual preliminary design of engines in research and industry. Pratt & Whitney Canada initiated the development of an integration and optimization methodology named PMDO (Preliminary Multi-Disciplinary Optimization) to contribute to strategic decision support [6]. The PMDO methods enable fast conceptual design and evaluation of different alternative engine concepts. Another approach to the predesign topic was carried out by Schaber [7]. The GTSDP gas turbine predesign tool enables the integration of thermodynamics, gas

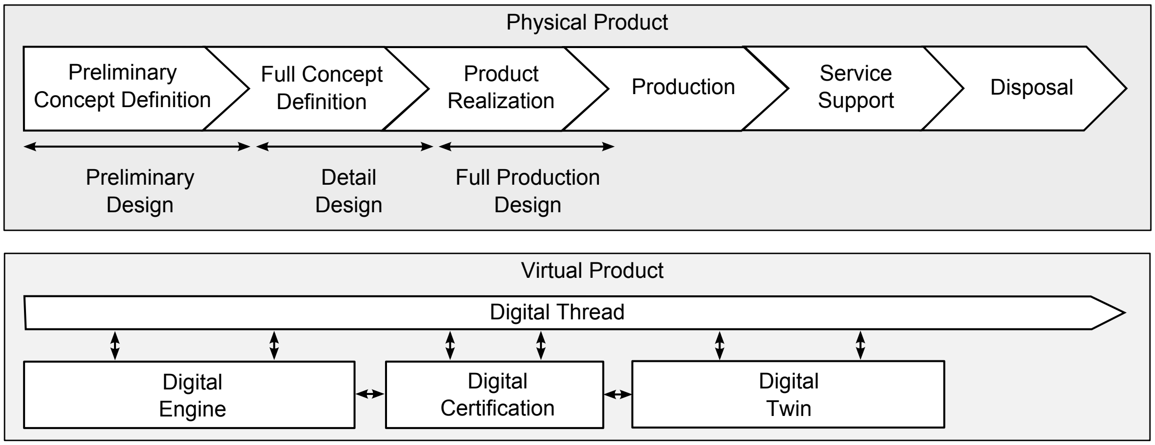

Fig. 1 Life cycle of an engine in terms of digitalization.

dynamics, structural mechanics, acoustics, weight and costs into a multidisciplinary predesign process. Based on the approach of Schaber, MTU Aero Engines has pushed the development of the multidisciplinary design system MOPEDS [8]. An important aspect of the system is its modular architecture. The methods of the integrated disciplines are called up via adapted interfaces and contribute to the extensibility and maintainability of the overall system. The University of Cranfield developed the modelling and management tool TERA with the aim of enabling multidisciplinary optimization of aircraft engines [9]. The method was used in the EU project VITAL for the design of an engine considering economic and ecological risks [10]. A heavyweight among the engine simulation systems is the object-oriented simulation environment NPSS [11]. The development of the software is the result of cooperation between NASA, various industrial partners, universities and other governmental institutions. The object-oriented design enables a modular process structure in which the various process components can be exchanged at any time. An almost complete summary of currently existing predesign procedures for aircraft engines in industry and research is given by Kupijai [12].

In addition to the field of engines and stationary gas turbines, there are also various procedures and processes in the virtual preliminary design of the aircraft. Böhnke et al. present a procedure for the automated conceptual preliminary design of aircraft using a distributed design environment [13]. The communication of the individual simulation tools is carried out by means of the central data model CPACS (Common Parametric Aircraft Configuration Schema) [14].

The present paper introduces technologies for the collaborative preliminary design of noval aircraft engine configurations developed during the DLR projects PEGASUS and PERFECT. The general architecture of this highly iterative, multidisciplinary and multi-fidelity process is described. Furthermore the virtual engine platform GTlab is introduced to demonstrate the potential to improve the collobarative preliminary design process through the application of modern software and digitalization technologies. The results of an engine design, an Ultra-High-Bypass Turbofan (UHBR) model, based on the introduced methods are presented in part B [1].

III. Predesign Process and Collaborative Architecture

As part of the projects PEGASUS and PERFECT the DLR developed a multidisciplinary predesign process for novel engine concepts. The general procedure is based on [15] and is further extended by the preliminary propulsor design. It is a highly iterative, multidisciplinary and multi-fidelity design process involving several simulation tools and experts from various departments at different DLR locations.

The collaborative process architecture which is inspired by the AGILE project [16, 17] is divided into three domains (Fig. 2). A data server represents the neutral domain and includes the digital engine representation based on the central data model and shared resources (e.g. material database). The tools and procedures associated with the individual disciplines such as engine performance, conceptual design, component aerodynamics, and mechanical design are available via the service domain. The data exchange is realized via the neutral domain using the interfaces of a central data model API. Finally, the administration domain is responsible for the management of the entire design process chain. An integrator monitors the process, merges data and initiates additional design iterations.

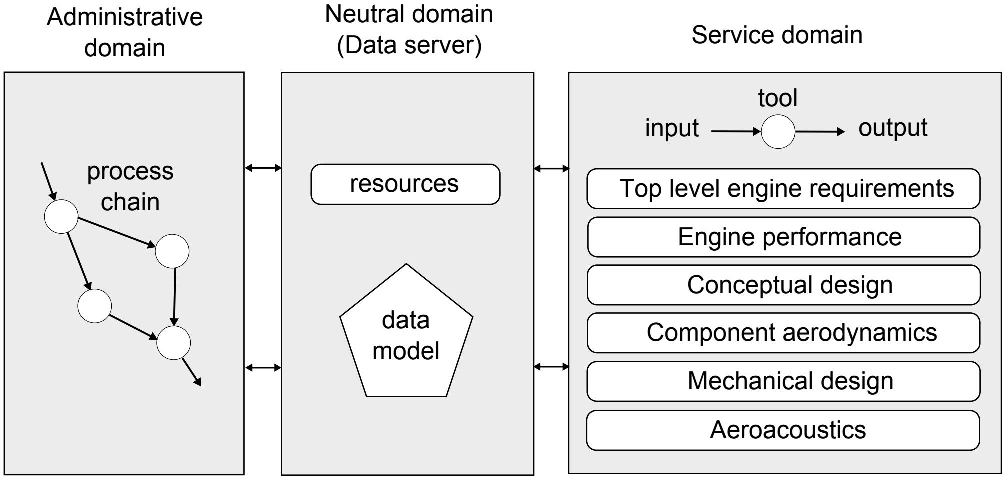

Fig. 2 Collaborative predesign architecture.

IV. Data Modeling

A major challenge of multidisciplinary engine design is the management and transfer of data throughout the entire design process. Usually, the individual design tasks are handled by teams with the appropriate expertise. These teams can be part of different organizational units or even different companies. This leads to the fact that the exchange of data and the communication between the individuals involved can easily become complex and confusing. Furthermore, the handling of the input and output data of the utilized tools can differ considerably. Another challenge is the readability, reusability and traceability of the data across the entire design process as well as the continuity of the data across different levels of detail.

In order to meet all these requirements of data management, GTlab relies on the approach of a central data model. The philosophy is based on the basic aspects of the design of complex systems according to [18] and can be projected onto the entire engine system as shown in Fig. 3. The various components of an engine together form the integration axis. Disciplines such as thermodynamics, aerodynamics and structural mechanics form the coupling axis and extend the system into the second dimension. Finally, the third dimension is defined by the fidelity axis, which takes into account the different levels of detail. The resulting data cube constitutes the digital representation of the engine system. Within the multidisciplinary predesign, it is desirable to be able to move freely within this data cube in order to access or transfer required information.

The concrete development and implementation of the central data model is based on the unified modeling language (UML) [19, 20]. UML is an established, object-oriented, and standardized modeling language which is ideally suited for specification, description, documentation, and visualization of complex systems like aero-engines and gas-turbines.

The structure of the individual components of an engine and their general relationships within the overall system were identified in order to obtain a detailed overview of the entire engine system. A simplified UML representation for a compressor system is shown in Fig. 4. Inside the UML diagram, each element represents an engine component or subcomponent. Elements can have their own attributes or inherit attributes from other elements to avoid duplicate definitions. In addition, elements can be associated with other elements to build an assembly. The multiplicity indicates the number of subcomponents that can be available in an assembly. The parameters and attributes of an element are standardized and their functionality clearly specified. Calculation results can be stored independently of the fidelity level and are therefore accessible in any phase of the design process. The advantages of the application of a central data model as infrastructure for the exchange of data during multi-fidelity simulation were demonstrated in Refs. [21–23].

The abstract definition of the central data model allows to store different types of blade parameterizations. A very powerful variant is the BladeGenerator parameterization, which was developed at DLR [24]. It features a high-degree

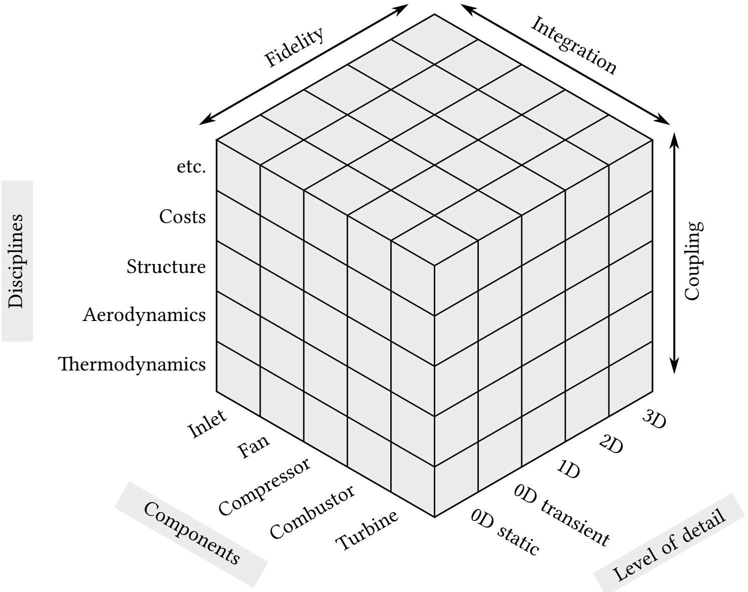

Fig. 3 Basic aspects of designing complex systems.

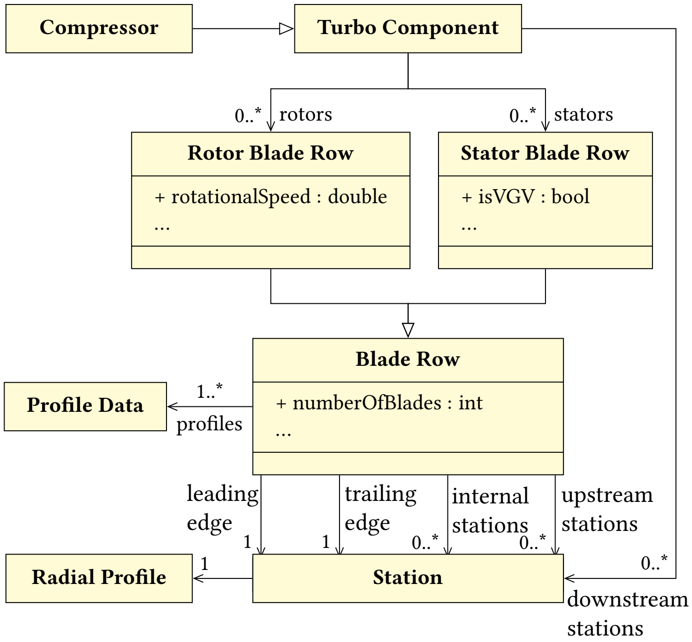

Fig. 4 Simplified UML representation of a compressor system.

of freedom in global and local geometry modification as well as the design of curvature distributions to influence shock-boundary layer interactions. Since the parameterization supports automated blade optimization processes for fans, compressors and turbines using three-dimensional (3-D) computational fluid dynamics (CFD), the geometry provided in the preliminary design can then be used in tools with higher fidelity without subsequent modifications.

The software implementation of the UML-based data model is realized by a C++ API providing a variety of predefined functions to access the data sets for browsing, reading and writing required data.

During the entire preliminary design, the engine model passes through a large number of different design phases and therefore undergoes a considerable amount of modifications. For this purpose, the API of the central data model provides a revision management system in order to be able to identify exactly what was changed, when it was changed, and why the change was made. As shown in Fig. 5, a new design branch can be opened at any time in order to pursue different design decisions. Additionally, a data set can be serialized via the central data model API and processed within external revision management systems or product lifecycle management (PLM) software to be included in the digital thread of a product.

The modular structure of the data model allows the subsequent extension of the data structures so that new engine components and disciplines can be added.

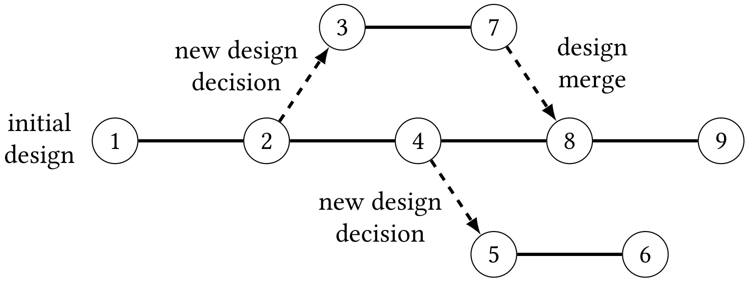

Fig. 5 Revision management system of central data model API

V. Framework

GTlab's software architecture is based on the three-tier architecture concept, to provide benefits by modularizing the user interface, logic, and data storage (Fig. 6). The modularization of the application into different layers gives the development teams the possibility to develop, maintain and improve the software faster than the development of a single code base [25]. A single layer can thus be updated with minimal effort and impact on the other layers. Furthermore, by using different layers, reliability and availability can be increased by hosting different parts of the application on different physical systems.

The data layer is responsible for storing and loading data. Within GTlab, this layer is based on the central data model described above. It provides standardized interfaces for accessing and modifying data generated during the preliminary engine design. Furthermore, the data layer can be protected from direct access if, for example, it is located on a secured network. Therefore, the interaction must take place strictly via the application or the presentation layer.

The logic that controls the core functions of the framework is included in the application layer. In addition to the possibility of uniform storage and distribution of resources, an internal process control environment is also provided.

The presentation layer is the front end of the framework and consists of the graphical user interface (GUI) and the post processing environment. It is applied by the user to control all operations within the virtual engine platform and to modify and visualize stored data sets of the data layer.

GTlab offers an API for the development of software modules to extend the framework. It uses the inversion of control principle to increase modularity of the application and make it extensible without modifying the code of the framework [26]. Standardized interfaces enable the integration of new or existing predesign processes into a uniform environment for the overall system design including consistent data management, resource management, graphical representation, and post-processing. The modules follow the plug-in concept and are integrated by the framework during runtime. The extensibility refers to all three layers of the application, so that the structure of the central data model, the simulation procedures as well as the graphical user interface can be extended.

Since GTlab is written in the programming language C++ using the cross-platform application framework Qt [27], it enables a platform-independent access regardless of the operating system.

VI. Process Management

Today's product development has become very complex and increasingly relies on the assistance of software systems. Engineers use a variety of software tools to design and simulate products. However, it is often the case that the individual software processes have to be connected with each other and various parameters and results have to be transferred from

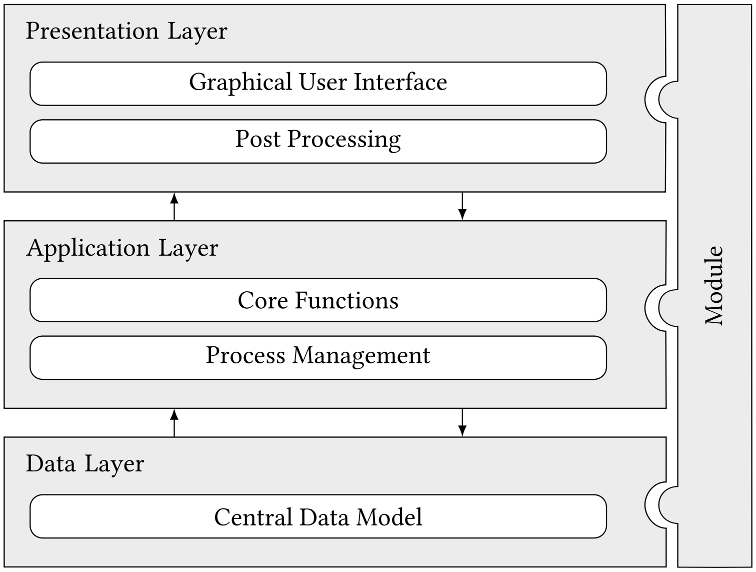

Fig. 6 Software architecture of GTlab based on the three-tier architecture concept.

one tool to another. Automation of all modeling and simulation tools enables rapid identification of optimal system performance, increasing process efficiency and significantly reducing failure rates.

GTlab provides a process control environment for the automated execution of design processes. Different process elements and calculation routines can be interconnected to form complex process chains. Each process element has access to the central data model described above in order to read and write the required data. This ensures data integrity throughout the design process and ensures that each element is always working on the latest representation of the engine model. The efficiency of the data exchange depends significantly on the number of interfaces and is therefore the critical factor for the flexibility of a design system. Using a central data model for the exchange of input and output quantities within a design process, the number of interfaces can be significantly reduced compared to the direct coupling of process elements (Fig. 7).

In order to find an optimal solution to a design problem, optimizer routines are usually used. GTlab's process control environment provides a generic optimizer interface that can be applied to solve optimization problems independently of the optimization algorithm. The central data model allows the optimization problem to be formulated universally and solved with a user-selectable optimization algorithm. The object-oriented interface can easily be extended by new optimization routines. In addition to optimizers, GTlab also includes basic process elements such as parameter studies, loops and convergers.

The major challenge of process control environments is the standardized integration of tools to make them available within the application. In order to be able to integrate calculation logic, design procedures or external calculation tools, GTlab provides a container with its own interface to the internal process control environment and enables access to all core functions of the GTlab framework. To handle input and output data, the container enables direct access to the central data model. An unique identification and integrated versioning of the individual container implementation ensures that the information is available to the process and thus ensures traceability. This is of particular importance in order to reproduce how a result was generated and which simulation method was involved.

Within the collaborative engine design it is often required to execute tools on different machines. Whether by the need of computational capacity and performance (e.g. cluster) or by the dependence on a certain operating system. Consequently, the connection of several tools to one process may not be possible. For this purpose, the internal process control environment of GTlab provides a functionality to find, select, and transparently execute remote tools in a platform-independent way. An additional application layer instance of GTlab can be outsourced to another system to enable the tools to execute correctly. The provided tools can then be identified and executed remotely via the process control environment. The remote interface is based on the distributed, workflow-driven integration environment RCE, which addresses various aspects of collaborative design [28]. Another important benefit of the distributed execution is the ability to optimize resource consumption and processing speed through load balancing. When a particular tool is present on more than one device, the interface automatically provides necessary load information for those devices. The process control environments can then use this information to transparently select the most appropriate instance for execution.

VII. CAD Interface

3-D CAD models are an important foundation for the digital engine and the digital twin. The primary use of GTlab is the design of aircraft engines and gas turbines. As written before, the engine design is given as a parametric

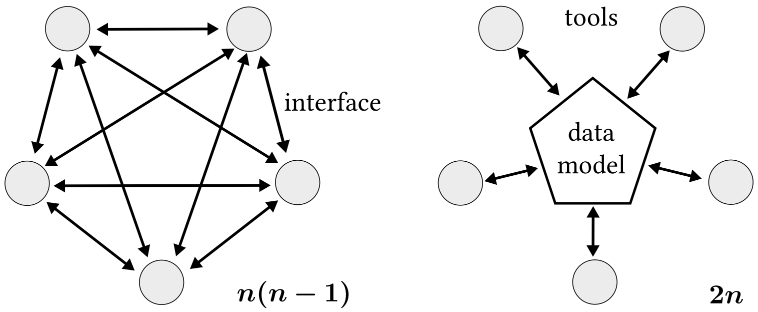

Fig. 7 Number of interfaces - comparison between direct tool coupling and usage of a central data model.

description within the central data model. To link the design process to further analyses, higher-fidelity geometry modeling, and the production eventually, a CAD interface has been developed. It transforms the parametric design into CAD geometries, in particular Non-Uniform Rational B-Splines (NURBS)-based curves, surfaces, shells, solids, and parts. These CAD geometries are de facto the standard in the engineering industry and are supported by a large range of software, including mesh generators, CFD and FEM simulations, or design software. The central data model contains all necessary information to automatically generate the different engine geometries in 2-D or 3-D, even at different fidelity levels (Fig. 8). The CAD interface uses the experience gained during the development of the geometry library TiGL [29], which is used for the geometry generation in the preliminary aircraft design at DLR [30–32]. This includes surface modeling such as section skinning or curve network interpolation, as well as the geometry export into common data formats.

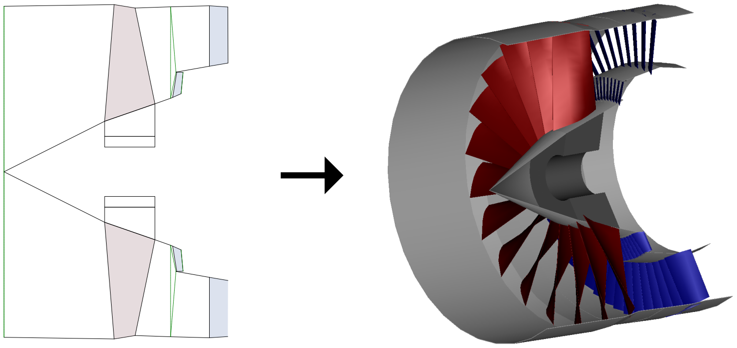

Fig. 8 Automated generation of 3-D CAD geometry based on central data model

The CAD interface contains shape factory functions and classes, including primitive shapes, e.g. shape of revolutions, prisms, compounds, or boolean operation to support constructive solid geometry (CSG). Higher level shape factories build gear boxes, blades – by interfacing BladeGenerator [24] for the modeling, blade foots, fuel injectors, combustion chambers, disks, shafts, ducts, and nacelles, based on the central data model parametrization. All shapes use watertight surfaces so that they can be transformed into solid geometries. This has the advantage that physical properties such as volume and moment of inertia tensor can be calculated for each geometry. It therefore allows an accurate estimation of the mass of a single component or assemblies in the early design process. Functions to compute these physical shape properties are also part of the CAD interface. To exchange these geometries with external programs, the CAD interface contains exports functions to common file formats, including IGES, STEP (ISO 10303 [33]), and COLLADA (COLLAborative Design Activity).

The CAD interface uses the CAD kernel OpenCASCADE [34] as its modeling back end. OpenCASCADE is an open source C++ framework that already provides modeling of primitives, boolean operations, data export, and visualization. However, the CAD interface is designed to be independent from its back end, by providing an abstraction layer around the back end-specific data types, classes and functions. This makes it possible to replace the OpenCASCADE back end with e.g. a commercial kernel.

VIII. Core Modules

GTlab includes three major modules to support the multidisciplinary preliminary aircraft engine design. The functionalities provided by the modules are not only available to the design process, but can as well be interfaced with additional modules. The following chapters give a rough overview of the functionalities.

A. Performance Interface

Performance simulations are employed for the thermodynamic cycle design and analysis. They are dealing with the synthesis of the individual component behavior for the derivation of overall performance data (e.g. like specific thrust and thrust specific fuel consumption) of the entire engine with respect to operating limits (e.g. like temperatures and speeds). Performance studies can be conducted for the entire flight envelope and are essential for the interaction of the components in the overall system. Performance activities support simulation, experiments, flight tests and engine condition monitoring and are therefore indispensable over the entire engine life cycle. They can be seen as a backbone for aero-engine design and development projects, from the initial predesign phase through product realization and verification to in-service support. For this reason, the GTlab framework provides a performance interface for conducting performance studies and enabling the provision of performance data in other domains and vice versa to analyze the impact of information feedback on the cycle.

The engine performance calculation is based on an one-dimensional flow approach representing thermodynamic states at multiple stations along the flow path of the engine. Hereby a station defines a state in between consecutive components. A functional relationship maps the input information of a component to a set of output variables. The relationship may consist of algebraic equations representing a physical model or an empirical correlation. For turbo components, these correlations are typically represented by interpolation tables or performance maps. Within initial performance studies, real component characteristics are typically not available to research organizations. Therefore, suitable performance maps for the turbo components are used and scaled during design to match defined component design targets. However, this procedure is an approximation and depends on the quality and suitability of the standard map within the underlying use case. Thus, the GTlab framework has set up a map collection based on the resource management system that provides a database of multiple turbo component maps that represent typical component characteristics and operating ranges from existing designs. These maps can be improved or replaced easily during the overall system design process by higher fidelity simulations or experimental data.

For computation of the thermodynamic state, a gas model is applied. Typically the working fluid in performance applications is modelled as a perfect gas, whereas the heat capacity is a function of temperature only. For the flexible use of any fuel within GTlab, the data on gas properties and heat release are stored in tables, which are combined in an exchangeable gas performance map. This map displays the data of thermodynamic functions such as heat capacity, enthalpy, entropy function, isentropic exponent etc. as well as the temperature rise (more precisely the outlet temperature of the combustion chamber) as a function of the fuel-to-air ratio, water-to-air ratio, temperature and inlet temperature. It can also be extended on the basis of various modeling methods. Due to the generic interface of the data table any data sources (e.g. correlations like provided by Walsh and Fletcher [35] or chemical calculation programs like CANTERA [36]) can be used to generate the data. For a representation of the consistent gas model across the fidelity levels, the gas performance maps are stored in a fuel collection and provide maps for different types of fuels and chemical compositions. Similar to the map collection, the gas models are thus available to all simulations in GTlab.

The general model layout with its components and the corresponding connections can be conveniently defined by drag and drop via the model editor. It follows the modular architecture of the simulation environment to enable the simulation of arbitrary engine concepts. An example of a model within the model editor is shown in Fig. 9. The engine setup is stored in the central data model. The results of the performance simulation are provided on a operating point basis. For this purpose, the data are stored in a two-dimensional table for operating points and performance parameters. Beside the data provided by the components, station data and the overall engine parameters, additional user defined variables can be created by means of the expression interface. The expressions are evaluated during each performance run and can be also be used within the performance calculation.

The performance run can be controlled by means of the process control environment, providing single point design and off-design capabilities. The single point calculation can be assigned to process elements such as loops or optimizers. Thus, any series of operating points such as parameter studies or multi-operating-point iterations can be performed.

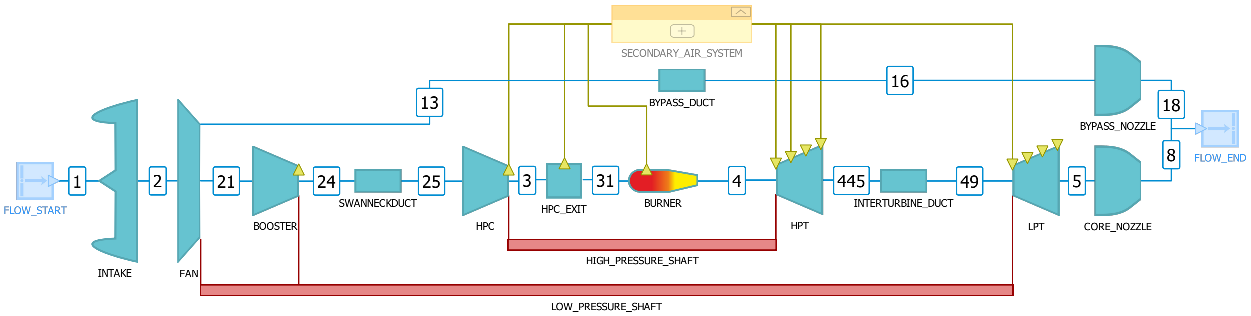

Fig. 9 Thermodynamic cycle schematic of turbofan engine model.

Within the performance cycle design, the basic engine concept is defined and an initial engine performance model is derived, based on the top level aircraft requirements. Therefore, parametric studies are performed that take into account both design and off-design calculations to ensure that certain operating limits are not exceeded and that all power requirements are satisfied. This procedure is repeated during the iteration of the engine design to update the parameters of the initial model whenever more accurate data is available from a more detailed model. Later during the engine life cycle, the performance model can be used as a digital representation for cycle analysis studies, test verification and condition monitoring (cf. [23, 37]).

With respect to the synthesis of all component modules, performance has to deal with a number of nonlinear equations. However, the numerical scheme is applied internally to ensure the conservation of mass, momentum and energy as well as additional constraints.

In principle, the performance interface is designed to enable the exchange of performance solvers. This means that any kind of performance program can be linked to the GTlab framework using the described interfaces. Due to the modular design, however, the interfaces can also be extended and adapted as required.

At DLR, the internal performance code DLRp2 is used within the framework of GTlab. It provides modern calculation methods for design and off-design simulations as well as for stationary and transient simulations. It basically consists of a component library, utility libraries and a solver library for the synthesis of the engine as well as the methods for performing calculations. The program is based on the generally accepted practice that an external solver (e.g. Newton-Raphson) is used to solve the equation system. For more details on the performance code, see [38].

B. PreDesign Interface

The predesign interface module provides many useful tools for the multidisciplinary engine design. A 2-D-based plot visualizes component geometry data sets based on the central data model representation in a uniform tool-independent manner. Changes to the geometry have a direct influence on the visualization and thus allow the design process to be performed interactively. This is particularly beneficial for the conceptual design, since the computation times of the simulation tools are comparably short.

The positioning of the individual components is decisive for the composition of the overall engine system. Particularly, when individual geometry changes lead to a displacement of the overall geometry, as is the case for a rubber engine. Using an internal constrain solver, the component geometries can be aligned within GTlab based on definable geometrical constraints. Changes in geometry cause the solver to search for a configuration of the geometric variables so that all constraints are met. This functionality allows the automated arrangement of all components of an engine during the design process without the need for manual adjustment.

In addition to the 2-D plot, the predesign interface also provides an OpenGL-based 3-D renderer using the aforementioned internal CAD interface. Again, changes to the data set result in a direct update of the 3-D geometry displayed in the 3-D-plot. Furthermore, the user interface enables the interaction with the displayed 3-D geometries.

The availability of material data is an important aspect in the design of the engine, particularly with regard to structural mechanics. Built on the internal resource management system of the GTlab framework, the predesign interface provides an uniform way for retrieving material data which is stored in the standardized data format MatML. MatML is an extensible markup language developed especially for the interchange of materials information [39]. Via a corresponding API, the processes and tools connected to the framework can retrieve required material data and use them in their calculation routines without the need for an own implementation. In addition, the API allows materials to be filtered according to specific requirements, so that only materials with appropriate attributes can be selected. Material properties such as modulus of elasticity, tensile strength, yield strength, density or thermal expansion can be determined as a function of temperature. A detailed documentation of the MatML notation can be found in [40].

C. Conceptual Design Toolset

The thermodynamic cycle design based on overall system performance simulation naturally does not require the specification of concrete component geometries and at best provides cross-sectional areas at component entry and exit for the subsequent component design. Already this makes it a great challenge to achieve an appropriate geometrical component design from scratch that complies with the thermodynamic specifications given by the underlying cycle. However, since there is commonly no direct coupling between thermodynamic cycle and detailed component design, it is not unlikely that thermodynamic assumptions are made which cannot be realized within the aerodynamic or structural component design. The later these inconsistencies are detected and the further the specific component and overall engine design progresses, the less reversible are potentially made mistakes during the early engine design phase. Therefore, an early identification of fundamental mistakes in the design process of aero engines is essential. To ensure this, an interactive coupling between thermodynamic specification and geometrical component design is required.

Especially in the early design phase of an engine, component design parameters may change quite frequently. To still enable a high degree of interactivity between thermodynamic and geometric design of a component, the application of methods with short response time is essential. However, the detailed component design commonly starts on mid fidelity level and normally requires input and expert knowledge which cannot be provided by the performance engineer responsible for the cycle design. This type of tools is therefore considered unsuitable for carrying out conceptual design studies. Rather simple, quick and stable methods are required which provide information on the basic feasibility and initial component dimensions based on very limited input.

In order to close the gap between thermodynamic cycle definition and sophisticated component design, the GTlab-Sketchpad module provides component based calculation routines for initial geometry generation which are mainly based on the performance specifications of the thermodynamic cycle and only require a small number of additional input parameters. The implemented calculation procedures use functionalities provided by both the performance and the predesign interface and are encapsulated in component-based calculation modules, enabling direct read and write access to the central data model. The generated component geometry is directly mapped to the corresponding data model parametrization of the appropriate component which automatically enables a direct 2-D and 3-D visualization without the need of additional implementations.

The GTlab-Sketchpad module library offers calculation routines for the aerodynamic/aerothermal design of the main engine components. This includes the fan, axial compressors, combustor, axial turbines, and intercomponent ducts. Also calculation modules for the structural design of blade foots and disks are provided.

The calculation methods for all turbo components are based on a meanline approach which obviously deals with strong simplifications but allows a very quick evaluation and therefore interactivity. Besides the approximation of required stage and blade numbers as well as component annuli, the meanline approach in connection with aerodynamic loss correlations also allows a rough estimation of stage and component efficiencies and blade geometries comprising leading and trailing edge angles, chord length and profile thickness. By application of a simplified radial equilibrium approach, the predictions can be easily extended from the meanline to values at hub and tip of the according blade row. The combustor design on the other hand is based on empirical correlations from design data of existing combustor systems and can be matched with cross sections at compressor outlet and turbine inlet if available.

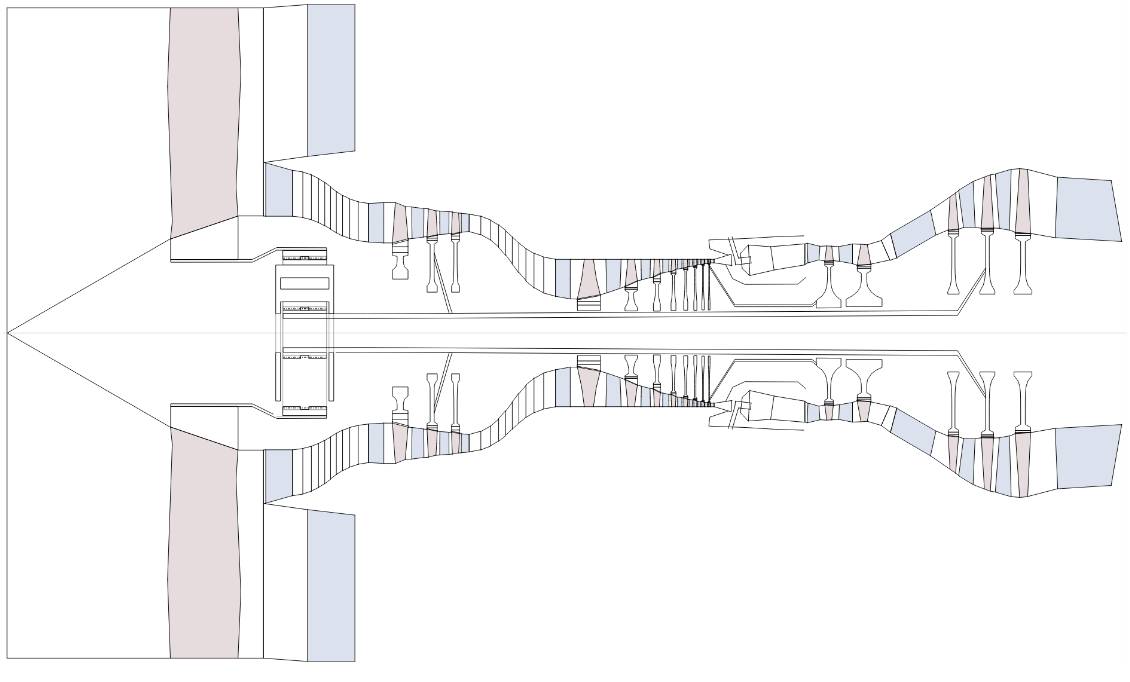

Fig. 10 Generic geared turbofan engine conceptual design cross section.

Subsequent to the initial gas path design, GTlab core functionalities such as the integrated material database and CAD-kernel interface allow a direct derivation of important quantities for structural-mechanical design e.g. weight, center of gravity or moment of inertia for single engine parts but also assemblies. This information can be directly used within the initial structural design of blade foots, disks and shafts. An almost complete overview over all calculation modules included in GTlab-Sketchpad and the applied methodologies is given in [41]. A possible resulting geometry from the Sketchpad process is illustrated in Fig. 10.

Since each component parameterization can be translated directly into a 3-D geometry, the exact volume and, using a known material, the exact weight can always be determined with high accuracy. Nevertheless, additional component parts such as accessories which have a considerable influence on the component weight in general, remain unconsidered during the initial engine design. For this reason the weight estimation for each component can also be performed by means of correlations based on methods introduced by [42] and [43].

IX. Conclusions

In the context of this paper, the virtual engine platform GTlab was presented which is a feasible approach towards virtual engine design in the context of digitalization. The integrative aspect by utilizing advanced software technologies helps to master the challenges of collaborative and multidisciplinary engine design.

The digital representation of the engine system using the presented central data model approach allows for an efficient management of the intricate data flows during design processes and thus helps to handle the extensive amount of transferred data between different disciplines and fidelity levels in a standardized manner. This is associated with a considerable potential for improvement in terms of error prevention and time optimization in the exchange of input and output quantities. The highly flexible and tool-independent methodology enables the system to be seamlessly extended by additional disciplines, components and simulation procedures.

Although the principles of engine design and aircraft design may differ, the introduced data model approach is similar to the methods of the collaborative aircraft design using CPACS. Therefore, the cooperation between the engine and aircraft design departments at DLR intends to be further intensified in the future. Concerning the technical aspects of this cooperation, the main focus is the consolidation of relevant data structures of both methodologies.

The modular structure of the virtual engine platform allows the adaptation to varying design processes while simultaneously retaining all core functionalities and the standardized data management. Functionalities such as the integrated 3-D CAD interface, the intelligent resource management system, uniform plotting and post processing routines provide a useful infrastructure for a wide range of applications in multidisciplinary preliminary design of aircraft engines.

GTlab does not intend to reinvent the simulation procedures, but to create a framework in which modern multidisciplinary methods and digitalization technologies significantly improve the application of these procedures in the context of preliminary engine design.

Although the general functionality to execute CFD processes within GTlab has already been demonstrated [22, 23], the extension of the interfaces to the detailed engine design including aero-structure coupled simulations is one of the next objectives and will be addressed in the DLR internal projects SimBaCon (Simulation Based Certification) and VirTriP targeting the digitalization of the aircraft engine and stationary gas-turbine sector.

Information on latest developments and projects relating to GTlab can be accessed in [44].

Acknowledgments

The authors would like to thank the GTlab development team, the associates from the Institute of Propulsion Technology, Structures and Design, Simulation and Software Technology, Air Transportations Systems, and the entire PEGASUS and PERFECT project team for the help and collaboration enabling the present study.

References

[1] Vieweg, M., Hollmann, C., Reitenbach, S., Schnoes, M., Behrendt, T., and Krumme, A., "Collaborative Aircraft Engine Preliminary Design using a Virtual Engine Platform, Part B: Application," *AIAA SciTech Forum*, 2020.

[2] Jones, M., Bradbrook, S., and Nurney, K., "A Preliminary Engine Design Process for an Affordable Capability," Tech. Rep. No. ADP014191, Rolls-Royce PLC, Bristol, United Kingdom, 2003.

[3] Tabaste, O., “A Route Toward Virtual Certification of Aircraft,” *Proceedings of 27th Congress of the International Council of the Aeronautical Sciences*, Nice, France, 2010.

[4] Glaessgen, E., and Stargel, D., “The Digital Twin Paradigm for Future NASA and US Air Force Vehicles,” *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Honolulu, Hawaii, 2012. Paper No. AIAA 2012-1818.

[5] Briceno, S. I., Laughlin, M. B., and Mavris, D., “A Virtual Simulation Platform for the Design, Testing, and Verification of Unmanned Aerial Vehicle Designs,” *Simulia Community Conference*, 2014.

[6] Brophy, F., Mah, S., and Turcotte, J., “Preliminary Multi-Disciplinary Optimization (PMDO) an Example at Engine Level,” Tech. Rep. No. RTO-EN-AVT-167, NATO AVT Educational Notes RDP, RTO, 2010.

[7] Schaber, R., “Numerische Auslegung und Simulation von Gasturbinen,” Ph.D. thesis, Technische Universität München, Munich, Germany, 2000.

[8] Jeschke, P., Kurzke, J., Schaber, R., and Riegler, C., “Preliminary Gas Turbine Design Using the Multidisciplinary Design System MOPEDS,” *Journal of Engineering for Gas Turbines and Power*, Vol. 126, No. 2, 2004, pp. 258–264.

[9] Ogaji, S., Pilidis, P., and Hales, R., “TERA - A Tool for Aero-engine Modelling and Management,” *Second World Congress on Engineering Asset Management and the Fourth International Conference on Condition Monitoring*, Harrogate, United Kingdom, 2007.

[10] Bretschneider, S., Arago, O., and Staudacher, S., “Architecture of a Techno and Environmental Risk Assessment Tool Using a Multi-modular Build Approach,” *18th International Symposium on Air-Breathing Engines*, Bejing, China, 2007. Paper No. ISABE-2007-1103.

[11] Lytle, J., Follen, G., Naiman, C., and Evans, A., “Numerical Propulsion System Simulation (NPSS) 1999 Industry Review,” *NASA Glenn Research Center*, 2000. Paper No. NASA/TM-2000-209795.

[12] Kupijai, P., “Ein Beitrag zur automatisierten Triebwerksvorauslegung,” Ph.D. thesis, Technische Universität Cottbus, Cottbus, Germany, 2014.

[13] Böhnke, D., Nagel, B., and Gollnick, V., “An Approach to Multi-fidelity in Conceptual Aircraft Design in Distributed Design Environments,” *IEEE Aerospace Conference*, 2011.

[14] Nagel, B., Böhnke, D., Gollnick, V., Schmollgruber, P., Rizzi, A., La Rocca, G., and Alonso, J. J., “Communication in Aircraft Design: Can we Establish a Common Language,” *28th International Congress Of The Aeronautical Sciences*, 2012.

[15] Reitenbach, S., Krumme, A., Behrendt, T., Schnös, M., Schmidt, T., Hönig, S., Mischke, R., and Mörland, E., “Design and Application of a Multidisciplinary Predesign Process for Novel Engine Concepts,” *Journal of Engineering for Gas Turbines and Power*, Vol. 141, No. 1, 2019.

[16] Ciampa, P. D., and Nagel, B., “The AGILE Paradigm: The Next Generation of Collaborative MDO,” *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017. Paper No. AIAA 2017-4137.

[17] Ciampa, P. D., and Nagel, B., “AGILE the Next Generation of Collaborative MDO: Achievements and Open Challenges,” *2018 Multidisciplinary Analysis and Optimization Conference*, 2018. Paper No. AIAA 2018-3249.

[18] Claus, R. W., Evans, A., Lylte, J., and Nichols, L., “Numerical Propulsion System Simulation,” *Computing Systems in Engineering*, Vol. 2, No. 4, 1991, pp. 357–364.

[19] Rumbaugh, J., Jacobson, I., and Booch, G., *The Unifed Modeling Language Reference Manual*, Addison-Wesley Object Technology Series, Addison-Wesley, Boston, MA, USA, 1999.

[20] Object Management Group (OMG), “Unified Modeling Language,” `http:/http://www.uml.org/`, 2019. Accessed: 2019-11-16.

[21] Reitenbach, S., Schnös, M., Becker, R.-G., and Otten, T., “Optimization of Compressor Variable Geometry Settings Using Multi-fidelity Simulation,” *Proceedings of ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*, 2015. ASME Paper No. GT2015-42832.

[22] Klein, C., Reitenbach, S., Schoenweitz, D., and Wolters, F., “A Fully Coupled Approach for the Integration of 3D-CFD Component Simulation in Overall Engine Performance Analysis,” *Proceedings of ASME Turbo Expo 2017: Turbine Technical Conference and Exposition*, 2017. ASME Paper No. GT2017-63591.

[23] Klein, C., Wolters, F., Reitenbach, S., and Schönweitz, D., "Integration of 3D-CFD Component Simulation Into Overall Engine Performance Analysis for Engine Condition Monitoring Purposes," *Proceedings of ASME Turbo Expo 2018: Turbine Technical Conference and Exposition*, 2018. ASME Paper No. GT2018-75719.

[24] Voss, C., and Nicke, E., "Automatische Optimierung von Verdichterstufen," *AG Turbo COOREFF-T FKZ*, 2008. Fachlicher Abschlussbericht Forschungsvorhaben, FKZ:0327713B, AG Turbo COOREFF-T.

[25] Gallaugher, J. M., and Ramanathan, S. C., "Choosing a Client/Server Architecture: a Comparison of Two- and Three-tier Systems," *Information Systems Management*, Vol. 13, No. 2, 1996, pp. 7–13.

[26] Fayad, M., and Schmidt, D. C., "Object-oriented Application Frameworks," *Communications of the ACM*, Vol. 40, No. 10, 1997, pp. 32–38.

[27] The Qt Company, "Qt - Cross-platform Software Development for Embedded & Desktop," `https://www.qt.io`, 2019. Accessed: 2019-11-16.

[28] Seider, D., Basermann, A., Mischke, R., Siggel, M., Tröltzsch, A., and Zur, S., "Ad hoc Collaborative Design with Focus on Iterative Multidisciplinary Process Chain Development Applied to Thermal Management of Spacecraft," *Fourth CEAS Air and Space Conference*, 2013.

[29] Siggel, M., Kleinert, J., Stollenwerk, T., and Maierl, R., "TiGL – An Open Source Computational Geometry Library for Parametric Aircraft Design," Tech. Rep. arXiv:1810.10795 [cs.CG], ArXiV, October 2018. URL `https://arxiv.org/abs/1810.10795`, submitted to *Mathematics in Computer Science*.

[30] Kroll, N., Abu-Zurayk, M., Dimitrov, D., Franz, T., Führer, T., Gerhold, T., Görtz, S., Heinrich, R., Ilic, C., Jepsen, J., et al., "DLR Project Digital-X: Towards Virtual Aircraft Design and Flight Testing Based on High-fidelity Methods," *CEAS Aeronautical Journal*, Vol. 7, No. 1, 2016, pp. 3–27.

[31] Goertz, S., Ilic, C., Jepsen, J., Leitner, M., Schulze, M., Schuster, A., Scherer, J., Becker, R., Zur, S., and Petsch, M., "Multi-Level MDO of a Long-Range Transport Aircraft Using a Distributed Analysis Framework," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2017. Paper No. AIAA 2017-4326.

[32] Liersch, C., Huber, K., Schütte, A., Zimper, D., and Siggel, M., "Multidisciplinary Design and Aerodynamic Assessment of an Agile and Highly Swept Aircraft Configuration," *CEAS Aeronautical Journal*, Vol. 7, No. 4, 2016, pp. 677–694.

[33] Pratt, M. J., "Introduction to ISO 10303 — The STEP Standard for Product Data Exchange," *Journal of Computing and Information Science in Engineering*, Vol. 1, No. 1, 2001, pp. 102–103.

[34] Open CASCADE, "Open CASCADE Technology, 3D Modeling and Numerical Simulation," `https://www.opencascade.com`, 2019. Accessed: 2019-11-16.

[35] Walsh, P., and Fletcher, P., *Gas Turbine Performance*, American Society of Mechanical Engineers, 2004. Second Edition, ISBN 978-0791802113.

[36] Goodwin, D. G., Moffat, H. K., and Speth, R. L., "Cantera: An Object-oriented Software Toolkit for Chemical Kinetics, Thermodynamics, and Transport Processes," `https://www.cantera.org`, 2019. Version 2.2.0, Accessed: 2019-11-16.

[37] Wolters, F., Becker, R., Schnell, R., and Ebel, P.-B., "Engine Performance Simulation of the Integrated V2527 - Engine Fan," *54th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*, 2016.

[38] Becker, R., Wolters, F., Otten, T., and Nauroz, M., "Development of a Gas Turbine Performance Code and its Application to Preliminary Design," *Deutscher Luft- und Raumfahrtkongress DLRK*, 2011. Paper No. DLRK2011-241485.

[39] Kaufman, J. G., and Begley, E. F., "MatML: A Data Interchange Markup Language," *Advanced Materials and Processes*, Vol. 161, 2003.

[40] Begley, E., and Begley, E., *MatML Version 3.0 Schema*, US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2003.

[41] Becker, R. G., Reitenbach, S., Klein, C., Otten, T., Nauroz, M., and Siggel, M., "An Integrated Method for Propulsion System Conceptual Design," *Proceedings of ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*, 2015. ASME Paper No. GT2015-43251.

[42] Sagerser, D. A., Lieblein, S., and Krebs, R. P., "Empirical Expressions for Estimating Length and Weight of Axial-Flow Components of VTOL Powerplants," Tech. Rep. No. NASA TM X-2406, NASA Lewis Research Center, Cleveland, OH, USA, 1971.

[43] Klees, G., and Fishbach, L., *Aircraft Engine Weight Estimation Method*, Society of Allied Weight Engineers, Los Angeles, CA, USA, 1978. SAWE-Paper No. 1248.

[44] DLR Institute of Propulsion Technology, "GTlab - Gas Turbine Laboratory - Project Page," `https://www.gtlab.de`, 2019. Accessed: 2019-11-16.