Deutscher Luft- und Raumfahrtkongress 2023
DocumentID: 610297

TOWARDS MULTI-FIDELITY SIMULATIONS IN THE CONTEXT OF THE VIRTUAL ENGINE

M. Schuff*
* German Aerospace Center (DLR), Institute of Test and Simulation for Gas Turbines,
Am Technologiezentrum 5, 86159 Augsburg, Germany

Abstract

Multi-fidelity simulations couple methods of different degrees of detailing levels and by extension also disciplines. The individual level of fidelity for a specific computational domain is chosen for the best compromise between accuracy, the potential to model effects, and computational costs.

Depending on the point of view, the term *multi-fidelity simulation* can be ambiguous towards the actual simulation goal and used methodology. Typically, the components of a system are divided into the involved disciplines, such as aerodynamics and structural mechanics, and each define their own domain of investigation. Those domains each include different discretization schemes, boundary conditions and solver methods. In order to allow coupling between those domains, an adequate description of the connections and interfaces is required. The typical holistic engine analysis approaches such as preliminary design, cycle analysis, or performance computation are derived from a geometry-based description of the individual components. In contrast, a domain-based description of the simulation topology is introduced which is built upon ontologies. It serves as the foundation to couple different domains across disciplines and fidelity levels.

The current work focuses on the built-up of the description of 3D high-fidelity simulations' setup and coupling, especially on high-performance clusters, in light of a future connection with the lower fidelity levels. A distinction of general approaches of different multi-fidelity scenarios is pointed out within the context of gas turbine simulation. Current efforts in implementation of multi-fidelity simulations focus on strategies for zooming and on the other hand, the approach towards coupling of 1D network simulations of secondary air systems with CFD.

Keywords

Multi-fidelity; Coupling; Virtual Engine

NOMENCLATURE

Abbreviations

0D	Zero-dimensional
1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
ADAPT	Assessment and Digitalization of forthcoming Propulsion Technologies (DLR project)
BC	Boundary condition
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CSM	Computational Structural Mechanics
DLR	Deutsches Zentrum für Luft- und Raumfahrt
FSI	Fluid structure interaction
GTlab	Gas Turbine Laboratory (software)
HPC	High pressure compressor
HPT	High pressure turbine
LPC	Low pressure compressor
LPT	Low pressure turbine

1. INTRODUCTION

Gas turbines, especially aero engines, have become one of the most complex machinery due to the interaction of every single component and the engine being in synthesis, which requires ever more multi-disciplinary considerations. Synthesis means, that a little change at one point influences all other components and forces them to adapt and form a new equilibrium of the system. Initially, the turbomachinery is designed by defined input, i.e. required thrust, dimensional restrictions etc. In the pre-design phase, an initial geometry is proposed that contains e.g. the stage load distribution, a selection of airfoils and so on, but also goes into cycle analysis and results in mass flow rates and pressure ratios. Based on the initial design, numerical analysis is performed for a single component or groups of components. The system

doi: 10.25967/610297

is divided into different computational domains and each domain is treated by a discipline-owned numerical technique or method. In the classical approach, the first divide is into fluid dynamics and structural mechanics. The whole approach analyzes the system by steadily increasing the fidelity level, getting more detailed in every step. In this way, the detail of the analysis can be adjusted to the level needed, to get either a better understanding of the physical phenomena happening at component level, or to propose an optimization of the geometry. Overall, this is more or less a one-directional approach towards the higher levels of fidelity.
While it is nowadays possible to run large 3D setups with a fully coupled CFD-CSM approach, the computational effort for such simulations is still too high for everyday usage in engineering design, especially on whole engine level [1]. One way to overcome this difficulty, is to treat each domain with the individual level of fidelity that is chosen for the best compromise between accuracy or the potential to model effects and computational costs.

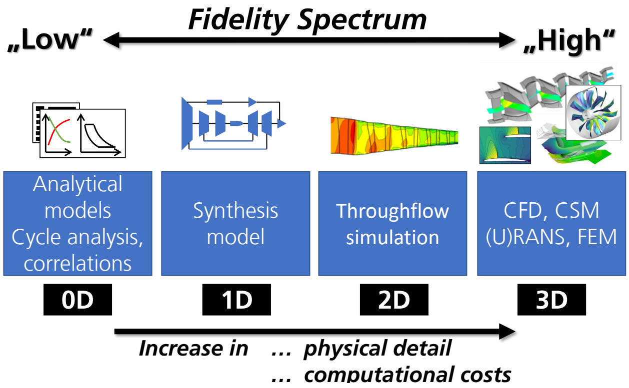

FIG 1. The fidelity spectrum based on dimensionality

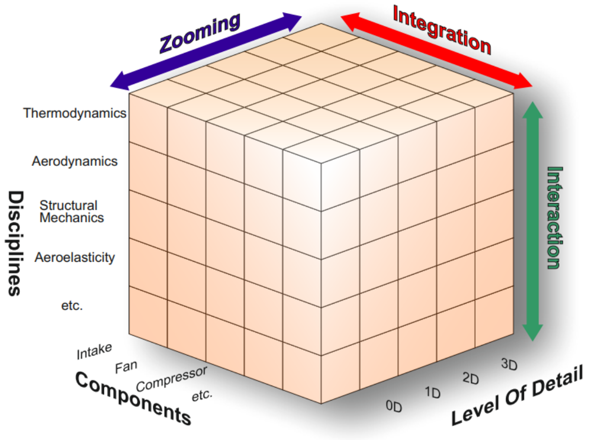

FIG 2. Building blocks of a complex system (adapted from [4], [5])

2. THE VIRTUAL ENGINE

In light of the so-called digitization, also known as the digital transformation, such terms as *virtual product* and *digital twin* have arisen. While the definition of these terms is very ambiguous, the basic idea behind all interpretations might be found in the expressed wish for a better exchange and closer collaboration between disciplines. The ultimate goal is the development of methodologies and techniques to gain more insight in and an improved understanding of the product. The term and classification of a *virtual aircraft* has been outlined by Risse [2]. The *virtual engine* is the equivalent in aero engine development. It aims to enable the handling of future challenges, such as reduction of environmental impact, but also cost effectiveness.
Within DLR, a group of experts from different disciplines and departments was formed (Fachausschuss Virtual Engine, abbr. FAVE, German for “Expert Committee Virtual Engine”) to accompany and moderate the process for the aero engine research aspirations [3]. In this context, the term has been defined as:

> *“The Virtual Engine is a flexible platform to answer cross-discipline topics and problems specific to aero engines via the usage of multi-fidelity tools of digitalization.*
> *The goal is to enable the highest possible level of detail (i.e. 3D high-fidelity methods) for each component.*
> *The intended use is a simulation- and data-based description of gas turbines, test rigs or parts of it over the whole product life cycle and in interaction with the respective environment.”*

Based on this, there is no monolithic system or “one-to-rule-them-all” platform which handles all arising and involved tasks. However, dedicated tools, solvers, and platforms for specific purposes exist. Many different aspects of engine research from prelimary design over optimization up to large-scale simulations contribute to the overall virtual engine. In the long-term, an effort to connect real-world data will be pursued.
In a first coordinated step driven by the expert committee, a DLR-internal project was initiated called *ADAPT* (“Assessment and Digitalization of forthcoming Propulsion Technologies”). The effort aims to harmonize the tool infrastructure and emphasize future collaborative research. Within this project, the built-up of a platform for multi-fidelity simulations is currently pursued.

3. FIDELITY LEVELS

When defining multi-fidelity, the question arises what the different levels of fidelity actually are. For an engineer who’s focus is on analytical cycle analyses, high-fidelity might be everything that cannot be solved analytically anymore. On the other hand, a CFD engineer working with RANS simulations could declare fidelity levels by the refinement degree of the numerical grid. Thus, the term “fidelity level” is very ambiguous and requires a definition what actually shall be done.

The present work coarsely classifies fidelity levels as shown in Fig 1. This classification might not be an unanimous opinion among every researcher, because it is very much not comprehensive of all involved tasks. The declaration of the fidelity levels by the dimensionality of its solution techniques divides the problem into an engineering decision: which method does one apply to solve a specific question?
A hypothesis: there are two worlds of engineering in the context of this work. One comes from the direction of preliminary design and their techniques typically solve the numerical problem within seconds or minutes (executable on a small personal computer or workstation). The other world applies simulation setups which require hours or even days to complete. The usage of high-performance computation clusters (HPC) is most often desired. Based on the walltime which the method typically takes to deliver results, the whole tool infrastructure is built around. However, in recent years, this clear separation has eroded. Both sides want to integrate methods from the other world to improve certain aspects of their own solution, but with a different focus. One side wants to include the improved accuracy and potential to model effects into their results. The other side wants to improve accuracy by getting more specific boundary conditions for an operating condition, but accepts the fact that the data comes from a source with lesser details than the own solution technique.

4. MULTI-FIDELITY SCENARIOS

Multi-fidelity scenarios can be visualized with the help of the "cube of complex systems" shown in Fig 2 which is an adapted version of the one presented by Lytle [4] resp. Reitenbach et al. [5] and based on the principle of Claus et al. [6]. While the term "coupling" most often is associated with inter-disciplinary methods (here named as interaction), in this paper's context it refers to coupling different bricks along all axes.
The following examples are focused on fluid dynamics. They represent the hitherto projected use cases and thus, will be the first multi-fidelity scenarios to be investigated. We keep in mind that other disciplines like structural mechanics, fluid-structure interaction (FSI), or acoustics, might also benefit from setting up multi-fidelity scenarios.

4.1. Zooming

A prominent application of a multi-fidelity method is the so-called "zooming". In the overall engine performance analysis during preliminary design, the usage of generic component characteristics based on a thermodynamic model is common practice. As such generic characteristics cannot account for the actual geometries, serious deviations between the computed and actual operating characteristics can arise, especially in off-design conditions. The zooming approach utilizes or replaces the lower-fidelity data of the thermodynamic cycle model by higher-fidelity procedures such as throughflow simulation or 3D-CFD [7]–[9]. Different strategies for coupling and integrating those higher-fidelity simulations are investigated by Pachidis et al. [10] or Schmeink and Schnoes [11]. Their distinction basically divides into:

- In the *direct approach*, the low- and high-fidelity simulations are carried out in sequential order, see Fig 3. This technique integrates the aerodynamic simulation into the thermodynamic model: CFD simulations are used as a kind of sub-simulation in the performance synthesis tool. The process can be very slow in terms of required walltime, what might be prohibitive.
- The *indirect approach* will run a set of predefined CFD simulations resulting in a characteristics map that replaces the map from the thermodynamic model, see Fig 4.
- A more advanced method, known as the *iterative scaler approach*, will require less pre-simulated high-fidelity computations. The generic map is

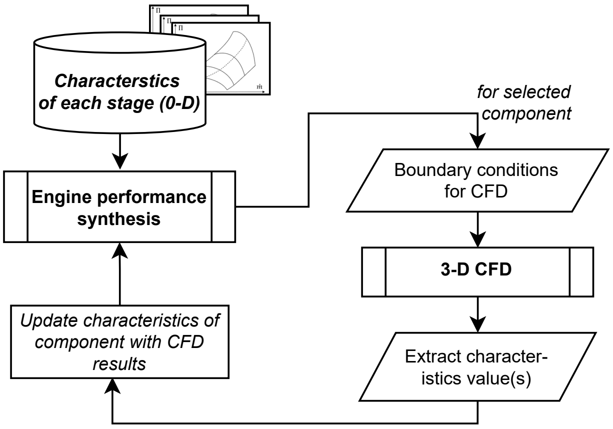

FIG 3. Zooming approach with direct integration of aerodynamic model into the thermodynamic analysis

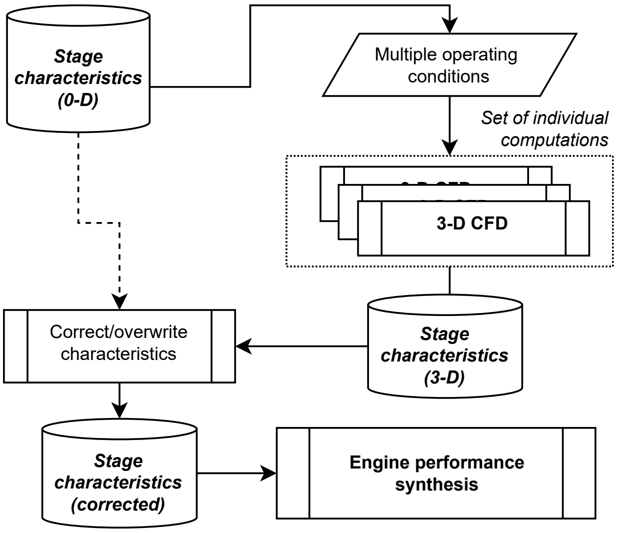

FIG 4. Zooming approach using pre-simulated CFD results to replace or scale the characteristics of the thermodynamic model

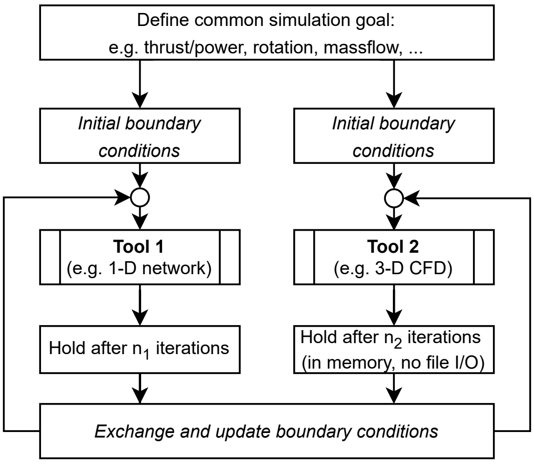

FIG 5. Parallel execution of multiple solvers of (potentially) different fidelity level and iterative coupling strategy

scaled to fit the CFD results, often only for a smaller region within the full characteristics map.

Overall, this enables the engine design process to use actual component geometries and also allows for optimization of the whole engine [9]. The important part of this approach is: the same component is simulated at different fidelity levels, thus dealing mostly along on axis of the “cube”.

4.2. Coupling of Components at Different Fidelity Levels

For each component of a large system the fidelity level might be chosen individually. This approach is similar to classical inter-disciplinary scenarios, such as FSI, where field variables have to be exchanged (and maybe interpolated) and converged at the domain boundaries. Various tools or methods are carried out in parallel as a single distributed simulation setup and exchanges occur after a dedicated amount of inner iterations of each tool, see Fig 5.

A prospected application is the coupling between lower- and higher-fidelity methods in the context of the secondary air system. Secondary air systems are typically analyzed by connecting 0D elements into a 1D network model [12]. Not every single element may have good correlations or equations which allow an adequate representation as a 0D element, but higher-order tools such as CFD can represent the physical effects. Coupling between the elements of different fidelity levels and the appropriate coupling strategies will be discussed in a later publication.

4.3. Utilizing Low-Fidelity to improve High-Fidelity Input

A scenario that includes some of both aspects from above, is the implementation of a newly designed component into an existing system, e.g. a different fan geometry or a new high pressure turbine. In another scenario, switching a system to hydrogen combustion will lead to a significantly increased cooling effort [13] and requires the secondary air system to be modified. In both cases, a detailed analysis of the new or affected component is necessary with the whole system equilibrium being relevant. The non-affected components can be simulated by a thermodynamic synthesis tool which has been trimmed by another multi-fidelity strategy before-hand.

5. SIMULATION TOPOLOGY

5.1. Geometry-Based Description of Gas Turbine

Based on the workflows in preliminary design, the gas turbine is described by geometrical features. It includes components such as fan, booster, compressor, ducts, combustor, turbine, and so forth. Most likely the description is not purely CAD modeling, but has a high degree of parametrization. This includes stations along the flow path, which resemble the boundary of each component, stages, etc. Stations can furthermore describe the general shaping of the blades and vanes, which is supplemented by airfoil definitions with scaling and rotation of those. Thus, aerodynamic surfaces are defined. Further descriptions include information about disk shapes, bleed ports, or detailing such as fillets and cooling channels or holes. In this manner, a full geometrical description of the gas turbine is achieved with varying detail based upon the intended purpose. These workflows and the data management are typically included in what might be called “digital representation system” with a central data model. Within DLR, this platform is called “Gas Turbine Laboratory” (GTlab). Its core and many features are developed at the DLR-Institute of Propulsion Technology [5] and is used and extended by partners within DLR and outside [14]. Many low-fidelity processes, such as thermodynamic models and performance analysis already have a high degree of implementation towards the GTlab framework [15], [16]. When going to high-fidelity processes, data out of GTlab can be utilized to generate grids and meshes as well as input for the simulations. However, those processes are only loosely connected to its data base or origin. A data feedback is very limited and mostly depends on the know-how of the engineer, resp. researcher.

5.2. Simulation Network

The now described *domain-based topology* aims to supplement the geometrical model from above, by giving the description of the simulation domains and their inter-connection. In other words, the volume between the surfaces and stations of the geometrical model is discretized for simulation purposes. Also, an abstract base for describing the coupling of those domains and disciplines is established.

Fig 6 depicts such a topology for a generic front part of an engine, consisting of a fan stage and a simple low pressure compressor (sometimes also referred to as booster) and building a network of connected CFD

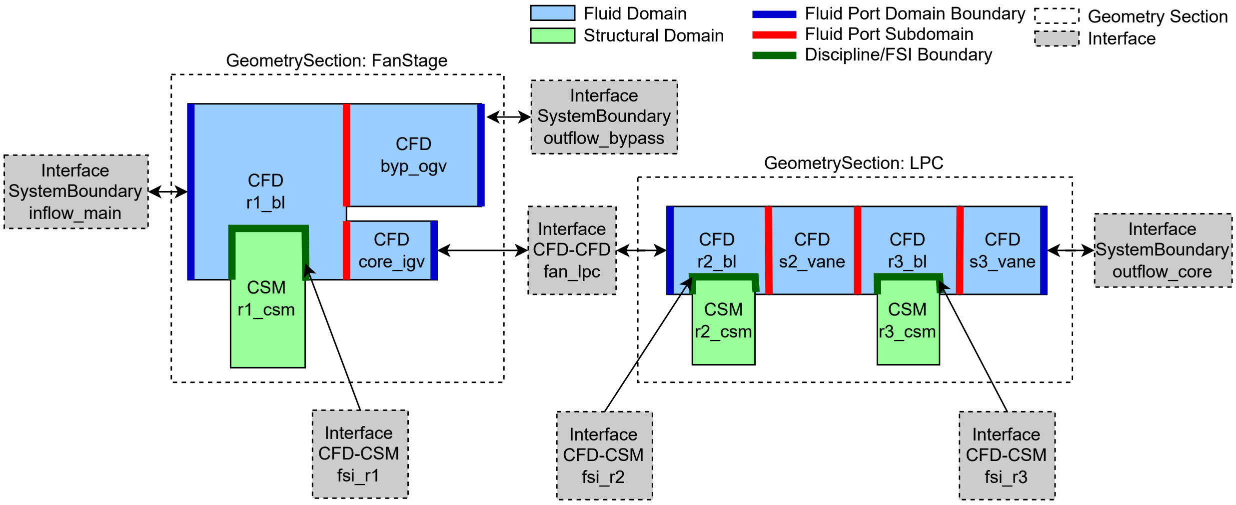

FIG 6. Domain-focused view of a generic simulation network of high-fidelity simulations of fan and booster

and CSM domains. In this paper, only a generic and simplified geometry is presented. By extension and in the same manner, the complexity of the simulation network could be extended to full engine scope.
The connections between the (sub)domains are labeled as “boundaries” and “interfaces”. These declarations will be explained in the following.

5.3. Ontology of Simulation Network

Ontologies can help to comprehend a complex system. Acc. to Gruber [17]:

> *“An ontology is an explicit specification of a conceptualization.”*

Later, Strassner [18] adds:

> *“Ontologies have their root in philosophy. (...)*
> *In computer science and knowledge engineering, ontologies define theories of what exists. (...)*
> *Ontologies offer a formal mechanism for defining a better understanding of facts.”*

In the presented context, ontologies can be interpreted as an extension of the classical class- and object-oriented description of systems. They add semantics for the connections between objects. Those semantics can then be further used for reasoning of relationships. Thus, objects have relations to each other, which can be expressed by graphs. Objects become data nodes and each node can also have properties (a set of variables in key-value arrangement).
Fig 7 depicts the fan stage geometry section in an ontological view of the geometry section. Note that the interface labeled “fan_lpc” only hints the connection to a second fluid port which is located in the next geometry section. The building blocks are outlined in the following.

5.3.1. Top Level

- GeometrySection: This is the top-level building block. The geometry section combines domains which are reasonably in the same region, which can be interpreted as component level. Thus, the geometry section might be equivalent to the components fan, LPC, HPC, combuster, HPT, and LPT. As seen for the LPC, two independent CSM meshes, one for each rotor, is included in the same section. Note that structural domains of the vanes are neglected/omitted in this example. Both CSM simulations could be activated as coupling partners of the CFD mesh in an FSI scenario. Typically this will mean that one geometry section has exactly one CFD mesh/domain. This is not exclusive of more CFD domains being in one geometry section.

5.3.2. Domains

- CfdDomain: A CFD domain contains exactly one CFD simulation setup. In general, CFD domains can be divided into subdomains – one for each blade row. For generality, a single-row setup contains exactly one subdomain.
- CfdSubdomain: In turbomachinery context, CFD meshes mostly can be divided into subdomains which correspond to blade rows or ducts. Many simulation scenarios will start with a steady CFD computation for the full mesh, but the subsequent investigations concentrate on a small set of subdomains for discipline-owned methods (examples: flutter and forced response analysis in aeroelasticity). Thus, a subdomain will encompass a certain blade row in the primary flow path. Inlets and outlets of each subdomain are defined as FluidPort. Surfaces of blades or vanes are defined by a CfdSurface. Both are described below.
- CsmDomain: The CSM domain is the equivalent part for the structural mechanics. Less detail is currently given here, as the focus is on aerodynamic simulations in this paper.

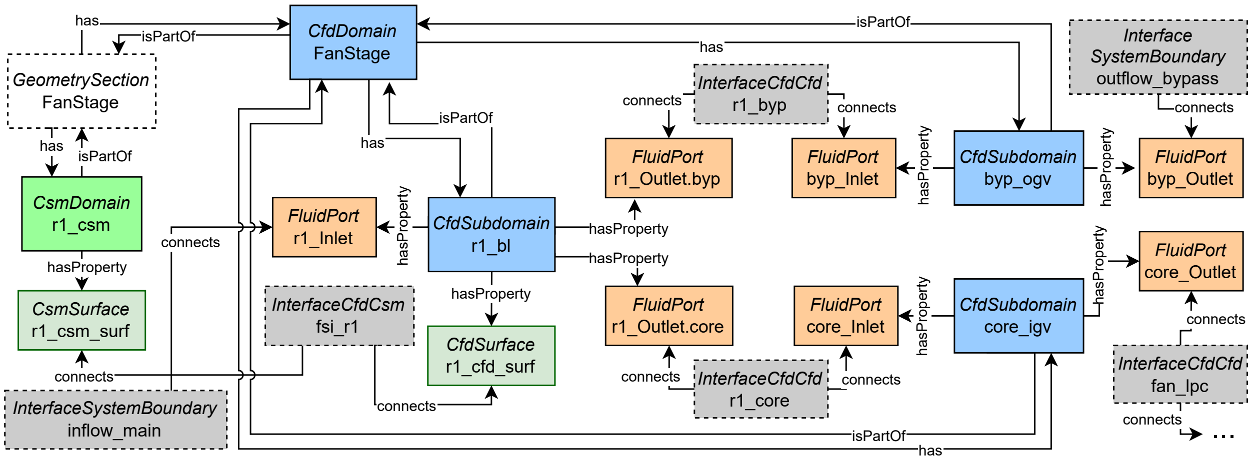

FIG 7. Ontology-based description of the geometry section “FanStage” from the simulation network

5.3.3. Boundaries

- Boundary: Boundaries represent the edge of a simulation domain. A boundary is the abstract definition of FluidPort, CfdSurface, and CsmSurface. In the context of simulation, a boundary will exchange information through an interface.
- FluidPort: A fluid port describes a boundary in CFD which is considered to be permeable for the fluid (in contrary to walls/surfaces). Each CFD subdomain has at least an inlet and one outlet (if not considered as a plenum). Inlet and outlet direction are important information in the simulation as the boundary conditions have to be applied accordingly.
- CfdSurface: A CFD surface is a boundary of the CFD mesh with wall conditions and thus the contrary to a fluid port. In analogy, an interface will handle the information flow towards other domains. If the CFD surface is not connected to an interface, the wall can be considered adiabatic (no exchange of field variables) and rigid (no deformations/deflections).
- CsmSurface: Analogous to the CFD surface, but for the structural domain.

5.3.4. Interfaces

- Interface: In the abstract definition, an interface connects boundaries of two domains. The interface will only tell which boundaries are connected and declare the type of connection, e.g. CFD-CFD, CFD-CSM, etc. There is no actual implementation of the interface given in the pure topological description. The implementation has to be done at the workflow level which sets up the individual simulation.
- InterfaceCfdCfd: Two fluid ports of CFD domains are connected. When the connection is established across the border of one CfdDomain, the CFD-CFD interface becomes relevant for the coupling of two separate CFD simulations in a simulation network. On the other hand, connecting two CfdSubdomain within the same CfdDomain will typically be an internal information of the CFD solver. This intra-domain information becomes relevant when the CFD mesh shall be divided into two or more independent meshes. Typical implementations are of the type “mixing plane” or “zonal”.
- InterfaceCfdCsm: Typically, this can be found in FSI setups. Note that at this point no technique for the actual coupling strategy is given here, too. As above, this strategy is part of the workflow for the simulation setup.
- InterfaceSystemBoundary: A boundary can be at the edge of the network. This is typically an inflow or outflow of the system, e.g. air intake or outlet nozzle. In general, system boundaries at the edge of the network are the entry points for boundary conditions provided by the user. Therefore, an InterfaceSystemBoundary connects to one Boundary within the network and the other end is “left open” for the workflow to specify the values. In case of CFD, fluid ports at the system boundary will likely connect to either inflow with total conditions, or outflow with a static back pressure or desired mass flow rate. In CSM, the system boundaries typically are fixed or floating bearings.

5.3.5. Remarks

This paper is intentionally focused on 3D simulation and aerodynamics reflecting the first use cases for the multi-fidelity scenarios. As more disciplines and models with lower fidelity levels will be included in the future, the topological description has to be extended. A suitable example is the usage of 3D-CFD elements connected to a 1D network model of the secondary air system. Although the interfaces are labeled with CFD or CSM above, in this example the ontological description will properly stay the same as no information about the coupling technique is included.

5.4. Implications of Topology for Simulation Workflow

The simulation topology only provides the foundation for a tool which checks for consistency. Some interfaces must be specified for a simulation not to be un-

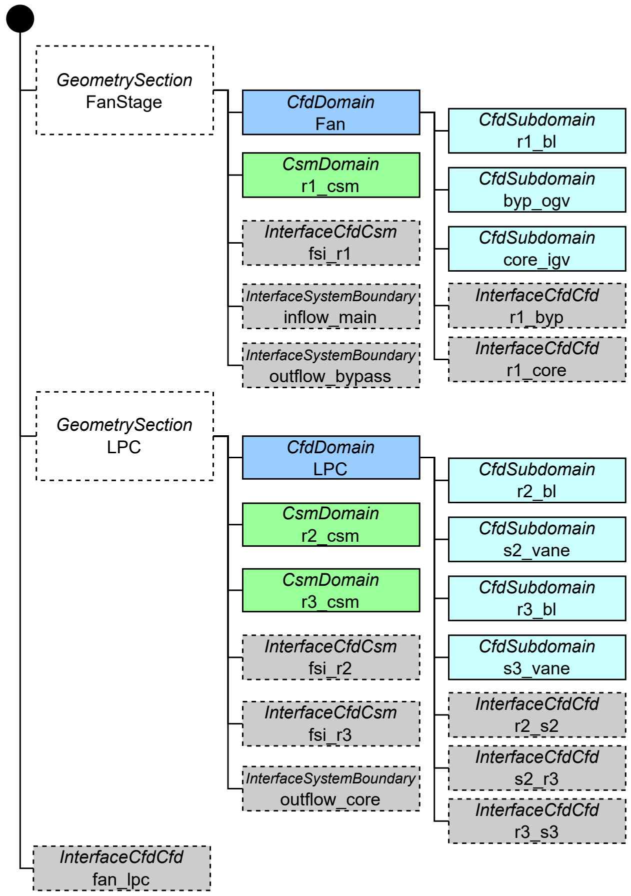

FIG 8. Hierarchical view of the generic high-fidelity network of fan and booster (without inlet and outlet boundaries)

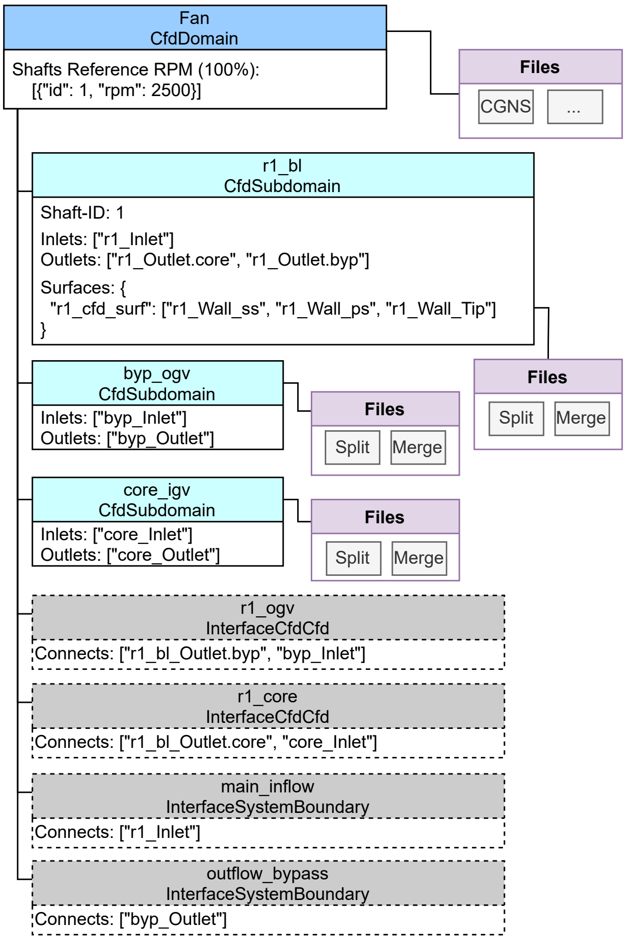

FIG 9. Detailed view of the CFD domain for the fan stage

physical, e.g. inflow total conditions etc. Other interfaces can be left untreated. The topology does not define further what happens in this case. An example is the pure CFD simulation which neglects an interaction with the structure. As described for the InterfaceCfdCsm above, in this case the CfdSurface becomes adiabatic and rigid.
If integrated into a stand-alone tool, the topology can be utilized to pre-process and provide the inputs, i.e. meshes and boundary conditions, for interdisciplinary coupling tools, such as *FlowSimulator* (DLR tool) [19] or *preCICE* (open source tool) [20]. Stemming from a collaborative effort as mentioned above, the topology is planned to be integrable towards data management tools, especially the one included in the GTlab framework.

6. SIMULATION PLATFORM

6.1. Hierarchical View of Simulation Network

After having introduced a domain-based simulation topology, the questions arises, where and how the information is stored. Hierarchical file systems define how data is structured. The presented generic simulation network is depicted in a hierarchical structure in Fig 8.

Some information is missing here because of the implementation which has been done. In the ontology, some nodes are connected to other nodes via a “has property” relation. This implies the linked information are not actual objects, but rather node properties, e.g. boundaries (fluid ports, surfaces) are identifier strings. Furthermore, to which boundaries the interfaces connect are also identifier strings and thus node properties.

6.2. Data Nodes

The elements from Fig 8 are defined as *data nodes*. Each data node is sorted in a parent-child relationship. Properties and files can be attached to each data node. This now becomes very specific to the actual used methods, tools and scenarios. For the CFD domain of the fan stage, an examples is given in Fig 9.

6.2.1. Node Properties

The CFD domain stores a reference speed of the shaft. This information is linked in the rotating subdomain representing the rotor. Inlet and outlets are properties (see above) of each subdomain and listed with an identifier name. The blade surface is linked to identifier names within the CFD mesh. The identifier “Blade” can then be used in the CFD-CSM interface (not depicted here). The subdomains are connected by a CFD-CFD interface which links the identifier names of the fluid ports.

6.2.2. Files

The CFD domain represents a CFD setup and thus, a mesh is attached; in this case, it is a CGNS file. Further files might be included for this node, such as simulation control files. Each subdomain contains files called "split" and "merge", which are tool-specific files for mesh partitioning of the individual subdomain.

6.3. Data Tree for Simulation Setup and Results

Two data trees will be used for simulation purposes, called *base* and *results*. *Base* stores the general configuration, and *results* contains the actual simulation results. This is demonstrated for the example of simple steady-state CFD computations, which use the same configuration but different boundary conditions. Note that the example is not fully representative of all steps required to set up, run, and analyze a CFD simulation of a gas turbine component, but shall convey an idea of the process workflow.

The *base* data tree in Fig 10 builts upon the previously shown data trees, omitting most of the content for the purpose of clarity. Inlet conditions are added as a special sort of connected data. Thus, a simulation will be linked to a CFD domain PLUS the defined inlet condition. Simulations with the same inlet conditions are typically clustered under a characteristics map (e.g. compressor map).

The *results* data tree in Fig 11 mirrors the base on the highest level with a data node for each geometry section's results (only one in this case). On the next level, many nodes of the type <u>OperatingPoint</u> follow. The operating point bundles the boundary conditions for a specific simulation, such as inlet condition, rate of rotation(s), back pressure or mass flows etc. The configuration and individual simulations are now defined for a workflow, that uses exactly this kind of input.

After a simulation has been carried out, the results are added to the corresponding operating point, here called <u>CfdResults</u>. In the given example, only one CFD simulation has been carried out so far. Meta information about the CFD results include the solver type, the solver's version, a dedicated solver mode, etc. The attached files are a CGNS with the field/volume data, and some more files created by post-processing steps, e.g. extracted surfaces of the blades and vanes. Again, this list of properties and files is by far not exhaustive for every possible scenario.

The node <u>CfdResults</u> is not the only child an operating point may have. For example, structural mechanics add their own results node(s). When using CFD results (e.g. surface pressure and temperatures) as an input, this linkage can be added to the topology resp. ontology. By extension, a workflow management tool can check those two trees for the required inputs and if results are already available.

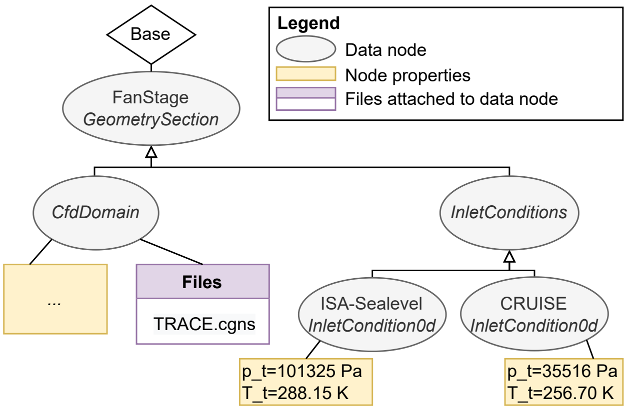

FIG 10. Datatree for the configuration setup (simplified)

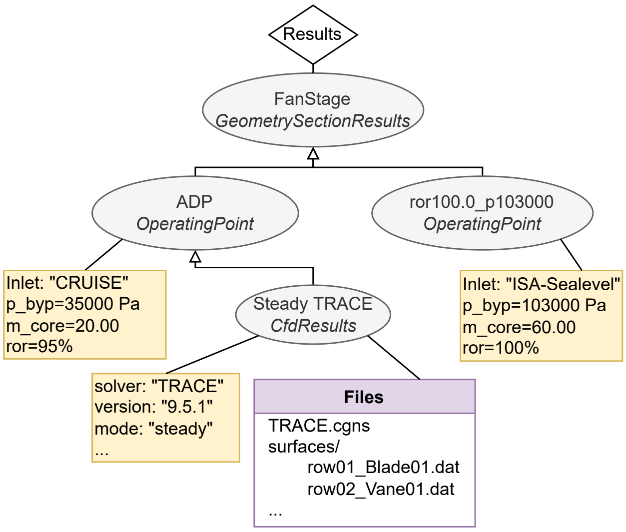

FIG 11. Datatree of the simulation results (simplified)

The structure in Fig 10 and Fig 11 reflects the input and output of the workflow once the CFD domain and its mesh are defined:

- ***<u>ALL</u>*** simulations use the configuration of the CFD domain and the mesh attached to it.
- ***<u>SOME</u>*** simulations use the same inlet condition, making it a connected feature of the interface for the inlet, defined by the operating condition.
- Exactly ***<u>ONE</u>*** simulation uses a specific set of *operating conditions* (boundaries conditions).

By definition, two simulations with the exact same input must always deliver the same result. This makes ***<u>ONE</u>*** simulation unique. If not, some setting change was not included in the definition (solver settings, workstation used, compiler used, etc.). To which extend this might be relevant is up to the user.

As an example: using the same boundary conditions but a different set of solver settings (e.g turbulence model) could be desired. In an isolated perspective, this contradicts the last bullet point since more than one simulation is based on the same operating conditions. In the intended way of the simulation platform, the solver settings belong to the CFD domain directly. Thus, a new CFD domain node has to be created if solver settings are changed. This might become unhandy in future scenarios. An other way might be to define specific solver settings directly for the individual simulation node. In that scenario, the last bullet point has to be adapted as following: "(. . .) uses a specific *combination* of operating conditions *and solver settings*".

7. SUMMARY AND OUTLOOK

The *Virtual Engine* platform within DLR currently lacks a convenient capability to include high-fidelity processes into the overall processes. As a supplemental to the geometry-based description of the gas turbine, a domain-based simulation topology is introduced, allowing to express multi-layered and connected simulations. A connection towards the central data model, as commonly used in preliminary design already, is planned in the near future. Furthermore, the description of simulations based on an ontology allows provenance methods to be invoked what will be part of another publication.

In prospect of running complex systems with different levels of fidelity, the foundation, respectively the underlying infrastructure is currently developed. In the current state only the high-fidelity level with a focus on CFD has received a proper description. In terms of CFD-CFD coupling, this will avoid forcing all CFD meshes into one very large simulation setup in order to avoid sequential execution of the simulations. An exchange of field variables is desired during the simultaneous running of all simulations to achieve convergence in less walltime. This utilizes the full potential of highly parallelizable applications on supercomputers. It has been identified that there are mainly two worlds to express the topology in the context of gas turbine simulations (geometry-based and domain-based). The current work aims to close the gap in-between. In order to achieve the mid- to long-term goal of a true virtual engine platform, both worlds have to speak a common language. Future work will also show how both topologies are using each other to make use of the full potential of a virtual engine platform.

Contact address:

matthias.schuff@dlr.de

References

[1] Keskin, A. and Fox, R., "The road to a digital twin for fan-blade off and thermally enabled structural analyses of aero engines", in *ASME Turbo Expo 2022*, 2022. DOI: 10.1115/gt2022-84296.

[2] Risse, K., "Das virtuelle Flugzeug", *DLGR-Mitgliedermagazin*, vol. 3/2018, pp. 14–17, 2018, ISSN: 0173-6264. [Online]. Available: https://www.dglr.de/fileadmin/daten-dglr/vernetzen/fachbereiche/l2/DGLR_FB2_VirtualAircraft.pdf.

[3] DLR, "Compute before flight", *DLRmagazine*, vol. 170, pp. 10–13, May 2022, ISSN: 2190-0108. [Online]. Available: https://www.dlr.de/en/media/publications/magazines/2022_dlrmagazine-170-compute-before-flight.

[4] Lytle, J. K., "The numerical propulsion system simulation: A multidisciplinary design system for aerospace vehicles", in *14th ISABE*, NASA/TM-1999-209194, 1999.

[5] Reitenbach, S., Vieweg, M., Becker, R.-G., et al., "Collaborative aircraft engine preliminary design using a virtual engine platform, part A: Architecture and methodology", in *AIAA Scitech 2020*, 2020. DOI: 10.2514/6.2020-0867.

[6] Claus, R., Evans, A., Lytle, J., and Nichols, L., "Numerical Propulsion System Simulation", *Computing Systems in Engineering*, vol. 2, no. 4, pp. 357–364, 1991, ISSN: 0956-0521. DOI: 10.1016/0956-0521(91)90003-N.

[7] Turner, M. G., Reed, J. A., Ryder, R., and Veres, J. P., "Multi-Fidelity Simulation of a Turbofan Engine With Results Zoomed Into Mini-Maps for a Zero-D Cycle Simulation", in *ASME Turbo Expo 2004*, 2004. DOI: 10.1115/GT2004-53956.

[8] Pachidis, V., Pilidis, P., Talhouarn, F., Kalfas, A., and Templalexis, I., "A fully integrated approach to component zooming using computational fluid dynamics", *Journal of Engineering for Gas Turbines and Power*, vol. 128, no. 3, pp. 579–584, 2004. DOI: 10.1115/1.2135815.

[9] Hollmann, C., Reitenbach, S., Schönweitz, D., and Wolters, F., "A fully coupled approach for the integration of 3D-CFD component simulation in overall engine performance analysis", in *ASME Turbo Expo 2017*, 2017. DOI: 10.1115/GT2017-63591.

[10] Pachidis, V., Pilidis, P., Texeira, J., and Templalexis, I., "A comparison of component zooming simulation strategies using streamline curvature", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 221, no. 1, pp. 1–15, 2007. DOI: 10.1243/09544100JAERO147.

[11] Schmeink, J. and Schnoes, M., "Automated component preliminary design and evaluation in the overall engine using fully coupled approaches", in *ASME Turbo Expo 2022*, 2022. DOI: 10.1115/gt2022-80634.

[12] Woelki, V. D., "Modulated secondary air systems for enhanced off-design operation of stationary gas turbines and aero engines", Ph.D. dissertation, TU Berlin, 2022.

[13] Chiesa, P., Lozza, G., and Mazzocchi, L., “Using hydrogen as gas turbine fuel”, *Journal of Engineering for Gas Turbines and Power*, vol. 127, no. 1, pp. 73–80, 2005. DOI: 10.1115/1.1787513.

[14] Reitenbach, S., Hollmann, C., Schmeink, J., et al., “Parametric datamodel for collaborative preliminary aircraft engine design”, in *AIAA Scitech 2021*, 2021. DOI: 10.2514/6.2021-1419.

[15] Becker, R.-G., Wolters, F., Nauroz, M., and Otten, T., “Development of a gas turbine performance code and its application to preliminary engine design”, in *DLRK 2011*, 2011.

[16] Vieweg, M., Reitenbach, S., Hollmann, C., et al., “Collaborative aircraft engine preliminary design using a virtual engine platform, part B: Application”, in *AIAA Scitech 2020*, 2020. DOI: 10.2514/6.2020-0124.

[17] Gruber, T. R., “A translation approach to portable ontology specifications”, *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993, ISSN: 1042-8143. DOI: 10.1006/knac.1993.1008.

[18] Strassner, J., “3.4 - knowledge engineering using ontologies”, in *Handbook of Network and System Administration*, Bergstra, J. and Burgess, M., Eds., Amsterdam: Elsevier, 2008, pp. 425–455, ISBN: 978-0-444-52198-9. DOI: 10.1016/B978-044452198-9.50018-5.

[19] Reimer, L., “The FlowSimulator—a software framework for CFD-related multidisciplinary simulations”, in *NAFEMS European Conference*, 2015.

[20] Chourdakis, G., Davis, K., Rodenberg, B., et al., “preCICE v2: A sustainable and user-friendly coupling library”, *Open Research Europe*, vol. 2, no. 51, 2022. DOI: 10.12688/openreseurope.14445.2.