AIAA SciTech Forum
7-11 January 2019, San Diego, California
AIAA Scitech 2019 Forum

10.2514/6.2019-2225

Toward the Realization of a Highly Integrated, Multidisciplinary, Multifidelity Design Environment

Dean E. Bryson*
Aerospace Systems Directorate, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433

Robert Haimes†
Aerospace Computational Design Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139

John F. Dannenhoffer, III‡
Aerospace Computational Methods Laboratory, Syracuse University, Syracuse, NY 13244

Conceptual Design is generally performed using historical data and low-order numerical models due to their computational expedience and low threshold of input information. However, performance predictions provided by these tools sometimes fail to capture critical effects due to a lack of interdisciplinary coupling, insufficient fidelity within a single fidelity, or the omission of a discipline altogether. In the best case, this inadequacy leads to missed design opportunities, but also may result in costly, late-stage design corrections or the production of a vehicle with limited capability in the worst case. While recent efforts have made progress toward bringing higher-order, physics-based predictions forward in the design process, use of these tools has been hindered by the modeling tools typically employed. The Computational Aircraft Prototype Syntheses research program attempts to provide a modeling infrastructure that enables agile, physics-based design by analysis, decoupling the availability of certain analysis tools from the stage of design. This manuscript describes a shift in thinking about vehicle modeling and geometry definition and its integration with meshing and analysis, and provides examples of applications in the literature. Ultimately, multiple, consistent, analysis-specific geometries should be outputs of a unifying *design model*, and should play an active role throughout the entire analysis process.

I. Introduction

Despite advances in the ability to predict the performance of aircraft using physics-based, numerical simulations, the traditional design process has largely failed to leverage these tools in Conceptual Design. Early designs are based largely upon historical data and low-order tools that sacrifice realism of physics to achieve the speed and computational efficiency required for large design studies and seldom include multidisciplinary couplings. Analysis realism increases in lockstep with design milestones, corresponding to the solidification of decisions that enable and justify the investment required to produce more representative models. These models are typically based upon a traditional (but generally parametric) computer-aided design (CAD) geometry representation. The reliance upon geometry models that are time-consuming both to produce and process into analysis inputs further inhibits the adoption of higher-fidelity tools in Conceptual Design.

As new, highly integrated vehicle concepts and technologies diverge from their predecessors, low-order tools frequently fail to screen unsatisfactory designs or to leverage interdisciplinary phenomena that lead to superior vehicles. For example, Allison *et al.* [1] demonstrate that optimal designs based on low-fidelity tools may be infeasible when evaluated with high-fidelity analyses, and *vice versa*, and that the individual fidelity optima are not necessarily near each other. Davies *et al.* [2], Alyanak and Allison [3], and Alyanak *et al.* [4] further highlight the technology opportunities afforded and design failures avoided by improving the realism of simulations in Conceptual Design. The importance of having confidence in the future performance of a vehicle early in design is underscored by the finding that 75% of life cycle cost is locked in at the point of concept down selection [5], after which the cost of defect correction increases rapidly.

---

*Research Aerospace Engineer, Design and Analysis Branch (AFRL/RQVC), Senior Member AIAA
†Principal Research Engineer, Department of Aeronautics and Astronautics, Member AIAA
‡Associate Professor, Mechanical and Aerospace Engineering, Associate Fellow AIAA

The design of highly integrated vehicles requires a design and analysis environment that is commensurately integrated. At any phase of the design process, engineers must have the ability and flexibility to integrate and couple disciplinary tools across a spectrum of fidelities to accurately predict vehicle performance. Higher-order analyses may require design details that have not yet been decided. However, we are of the opinion that designers should be enabled to perform these analyses at the earliest possible moment with the best information at hand, and to determine sensitivities to free design parameters. The associated vehicle representations will mature alongside the design. Barriers to this vision are not solely cultural; the present mode of operation is born out of the significant investment of hands-on time to create the geometric models, meshes, and inputs required for various analyses. These challenges are exacerbated by the fact that vastly different representations of the geometry are required by different disciplines and even different models within the same discipline. The disparity of these representations also inhibits the interdisciplinary transfer of information for coupled design.

The realization of a highly integrated design environment requires not only an evolution of toolsets, but also an evolution in our thinking about modeling for analysis and design. We believe Computational Aircraft Prototype Synthesis (CAPS) [6] provides an example for a new approach. It is being developed with a vision for a more fluid design process, providing greater freedom to analyze configurations with tools spanning the spectra of disciplines and fidelity levels at any stage of design maturity. Rather than focusing on the model as single geometric representation of a vehicle, we propose a model that gives rise to multiple consistent, geometric realizations of the same aircraft configuration. These views are augmented with attribute information, or meta-data, that describe both what the geometry represents and how it behaves in analysis. This geometry is not simply a stepping-off point for analysis, but rather it plays an active, central role throughout the process of design by analysis.

This manuscript describes the philosophy and approach taken in CAPS, which is being developed by Massachusetts Institute of Technology and Syracuse University under contract with the Air Force Research Laboratory. While the predecessor paper [6] goes into more of the implementation details (which have evolved over time), here we strive to maintain a higher-level perspective on some of the ideas we believe enable automated multifidelity, multidisciplinary analysis for design. After describing current challenges and introducing some preliminary terminology to orient the reader, this paper describes the central concept of the *design model* and its relationship to geometry. It then segues into the use of geometric meta-data to enable the automation of analysis meshing and pre-processing, and the integration and configuration of automated analyses into useful workflows. Finally, after providing a sampling of applications enabled by CAPS, the paper summarizes key enablers to design by multifidelity, multidisciplinary design by analysis and provides perspective on barriers to widespread adoption.

II. Challenges in Geometry for Design and Automated Meshing

There are many barriers to achieving the vision of a highly integrated design tool, both technological and cultural as previously mentioned. Technologically, the source of many difficulties is in the representation and use of geometry in the design process. The tools at hand do not lend themselves to the type of process envisioned, and in attempting to make the best use of the tools available, process bottlenecks have been created, further constraining the process. Our hope is that by creating tools suited to design by analysis, the required cultural changes will naturally evolve.

A. Geometry for Design

The current use of geometry in most design processes is *ad hoc*, at best, and usually contains a host of different tools. It is common to have a Conceptual Design tool that *sizes* the design (*i.e.*, sets some basic parameters) but does not realize 3D geometry — it merely suggests the shape. Each discipline may have early-stage specialized tools that can build 3D parametric models consistent with their geometric needs but without full regard to the larger multidisciplinary design at play. There are some disciplines that may remain “geometry free” (*e.g.*, a mesh in use may be morphed) during design. Only at some later stage in the process is there an attempt to put it all together (usually in a general geometry generation system where the parametrics may be lost or must be recreated), which requires great effort and is quite error prone. The end result is that the final realized geometry may be different than that analyzed, begging the question: “how can this be the optimal design?”

There is a clear need for a single, multidisciplinary/multifidelity geometry modeling system that can be used throughout the early phases of design and can support the complex geometric requirements of the later stages. Ideally, this system should be able to generate geometries that don’t need to be fixed or repaired so it is suitable for automation within the envisioned design process. Also, when done, the geometry system should be able to output the designed geometry so that a commercial CAD system can import the model without loss of accuracy in the shapes represented.

Relegating commercial CAD to the end of the design process may be surprising, but its focus has been, and is, manufacturing, not analysis. Attempting to use these systems within the early phases of design is partially to blame for the current state of affairs. These are rather large, monolithic software systems, difficult to connect to and use as a service (*i.e.*, not have a user in front of a graphical user interface). An effective geometry system for analysis needs to be able to provide its output seamlessly to the rest of the design process without translation.

B. Automated Meshing

In a design process, any task that requires an individual in the loop will be a bottleneck. How is it possible to perform a design study that may examine thousands of geometries where each needs attention during meshing to prepare to run an analysis?

The current state of automated meshing is driven by two factors: (1) the mesh topologies that are acceptable for the analysis at hand, and (2) the state and form of the input geometry. As far as the mesh topologies, there are techniques where triangles (for surfaces) and tetrahedra (for volumes) can be generated without intervention for geometries of any complexity. There are also some meshing algorithms that can apply structured collections of elements (quadrilaterals and hexahedra) for shapes that can be simply decomposed.

The real difficulty is the import of geometry. For analyses that require watertight geometry (*i.e.*, computational fluid dynamics, CFD), the repair of broken geometry is very time consuming and potentially introduces deviations from the original design intent. Non-watertight geometry comes about through the necessary use of tolerances when writing common geometry file formats, with the result being that geometric faces that should be adjacent in fact do not share a common edge. Other errors introduced include self-intersecting edges and faces. All of these geometric failures inhibit the automatic generation of analysis meshes. Many times the pre-processing of the geometry is the pacing item and requires intervention by an individual interactively using a geometry “repair” tool. There are two fundamental reasons that this is necessary: (1) the geometry transmission file formats are not robust and (2) the geometry is not commensurate with the analysis of interest.

III. Computational Aircraft Prototype Syntheses

A. Terminology

When discussing a design system that encapsulates many disciplines there is bound to be some clashing of terms. This is oddly true for the closely related fields of geometry and meshing where the same terms have entirely different meanings. A geometry model in the form of a Boundary Representation (BRep) has two fundamental components: geometry entities and topology that *holds* the model together. The geometry components consist of points, analytic curves and surfaces. The topology model has Nodes (which contain points), Edges (which contain curves and are bounded by Nodes), Loops that are made up of ordered collections of Edges and associated orientations, Faces (which contain surfaces trimmed by Loops), Shells that are collections of Faces with orientations, and finally Bodies. Note the capitalization of the topological entities is intentional and differentiates the BRep meaning from the terms node, edge, and face often used in discrete geometry and meshing.

B. The Design Model and Design Intent

A critical enabler of the CAPS philosophy is the specification of geometry via the *design model*. A key concept of the design model is that it does not encode geometry itself, but rather geometry is an output—a use-specific realization, or *view*—of the model. The design model embodies the concepts of various components in a configuration (*e.g.*, wing, fuselage, control surface hinge line, structural layout, illustrated in Fig. 1), the operations used to construct these components, and the design parameters that drive them. The behavior of the design model in response to parametric changes is described by the *design intent*. The design intent must describe the design model in detail sufficient for all uses throughout the entire design process. It largely embodies the information traditionally captured in Conceptual Design, but this data needs to be maintained throughout the entire design process (through traditional Preliminary and Detailed Design phases). The design model and intent are encoded in the scripting language of OpenCSM (Open-source Constructive Solid Modeler) [7], the geometry subsystem upon which CAPS is built; OpenCSM is built upon EGADS (Electronic Geometry Aircraft Design System) [8] and OpenCASCADE, in turn. As CAPS itself has only a programming application programming interface (API), design models may be generated, manipulated, and viewed in the Engineering Sketch Pad (ESP) [9], a predecessor program. ESP is specifically constructed to allow for the *translation* of the design

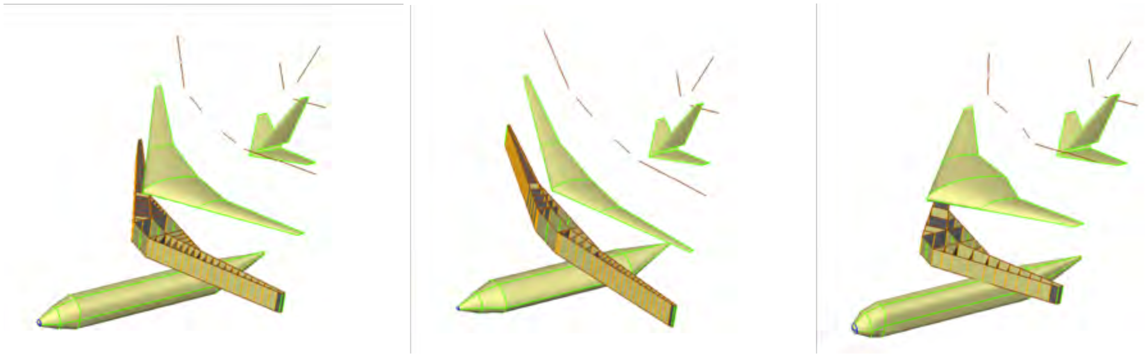


Fig. 1 Parametric variations of conceptual components embodied in a transport design model (components are separated for clarity, but are coincident in the actual model). Control surfaces are represented by their hinge lines, shown above the wing. Note that components vary consistently with respect to shared design parameters, which may include topological structure changes (*e.g.*, number of ribs).

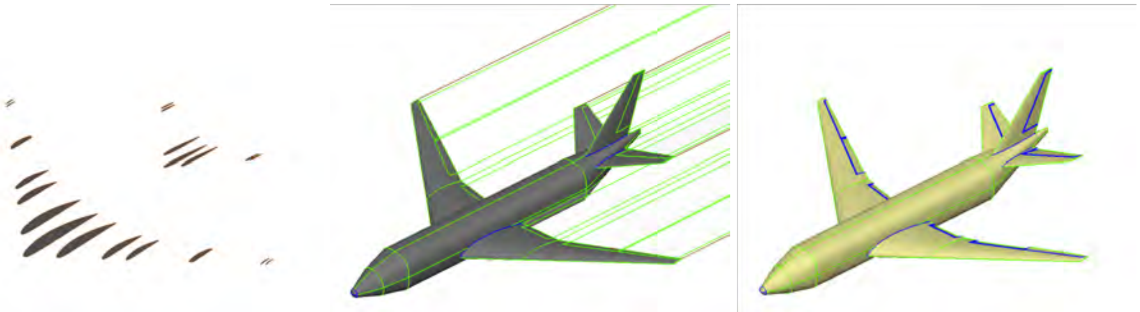


Fig. 2 Views of an aircraft design model for three levels of aerodynamic analysis: vortex lattice (left), potential flow (middle), CFD (right).

intent into the design model.

Note that the design model should not be confused with the term “master model”, though there are some similarities. The design model is specifically used to generate parametric multifidelity and multidisciplinary geometries for the direct use by analyses (not manufacturing).

The utilization of a design model rather than a traditional geometry model permits the generation of multiple, mathematically consistent views of geometry for multifidelity, multidisciplinary applications from a single specification. (Note that here, “multifidelity” is used to denote analyses within a discipline, or a fixed set of coupled disciplines, that require different geometric representations due to modeling assumptions or level of detail.) In doing so, the burden of reinterpreting or reverse engineering a geometric representation is eliminated, as are the associated errors. In generating different geometric views, the design model specifies how various components interact and combine to produce the required result.

Examples of multifidelity views for aerodynamic analyses of an aircraft are provided in Figs. 2 and 3, spanning from a typical representation for a vortex lattice method to a traditional CFD solution of the Navier-Stokes equations. Generating these views makes use of the wing, fuselage, tail, and control surface component definitions. For the vortex lattice representation, airfoil sections are extracted from the wing at locations where the leading and trailing edges break, as well as at the boundaries of control surfaces. This representation is common to several vortex lattice codes; however, an alternative view modeling the wing mid-surface could also be envisioned. For the potential flow model, the outer mold line (OML) of the wing is used, upon which the control surface boundaries are scribed. The control surfaces need not be separated from the wing because the target analysis simulates deflections by altering the surface normal using a transpiration boundary condition. An infinitely thin wake sheet is also added, extending from the trailing edge to the farfield boundary. For the CFD model, the control surfaces are actually separated from the wing, introducing a small gap, the size of which is a design parameter.

Fig. 4 illustrates a view intended for structural finite element analysis (FEA) of a wing. To create this geometry,

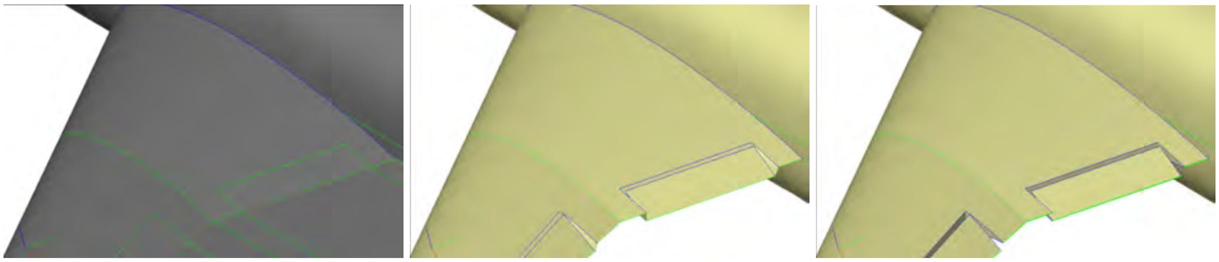


Fig. 3 Three different trailing edge treatments for aerodynamic analysis: sharp trailing edge with control surface outlines (potential flow, left), sharp trailing edge with control surface gaps filled (Euler CFD, middle), blunt trailing edge with open control surface gaps (Navier-Stokes CFD, right).

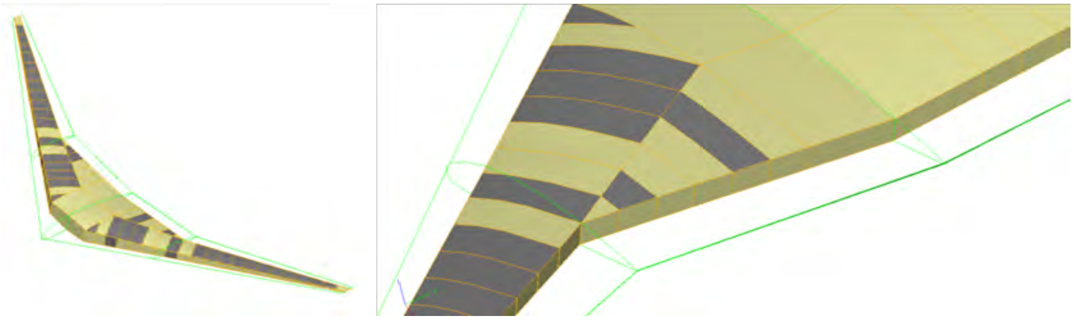


Fig. 4 A geometry view for finite element analysis of a built up wing structure. Divisions on the upper surface indicate locations of internal structure.

the structural layout and wing components were intersected to produce a conformal inner mold line (IML). The wing component was then hollowed, leaving only the outer shell, and unioned with the internal structure. Note that the BRep Faces forward of the leading edge spar and aft of the trailing edge spar are shown transparent. This indicates that they are marked to be ignored by the FEA, effectively modeling only the wing box. (Marking of geometry, or attribution, will be described later in more detail.) It is key that both the structural view and the aerodynamic views containing the OML in Fig. 2 stem from the same wing component specification. It not only ensures consistency, but the geometry subsystem recognizes this common ancestry and can leverage it to enable conservative transfer of field data between models (also described later in more detail).

The geometry subsystem has several other features that enhance CAPS capabilities. First, all surfaces are trimmed (and are represented as Faces), and all closed geometries are watertight (based on the BRep topology, not floating-point precision). This feature is significant because it permits automated meshing without the laborious process of geometry cleanup and repair found in traditional model building. Second, the geometry subsystem is analytically differentiated, with the exception of a few operations, namely fillets and chamfers, which are approximately differentiated by finite differences. This feature is being leveraged in ongoing work to develop shape sensitivities for structural FEA [10] and fully analytic adjoint sensitivities for CFD. It should be noted that currently, adjoint sensitivities are typically analytic down to the level of the surface mesh, but sensitivities of surface geometry are traditionally approximated by non-analytic means, such as finite differences. This has problems in determining the appropriate step size and tracking points between geometric instances.

C. Automation of Meshing and Pre-processing for Analysis

Attributes (or meta-data) are a pivotal component of any automated and coupled framework. In a real sense, they provide a convenient place to hold data associated with the use of the model. The automated meshing and pre-processing (generation of analysis inputs) for analysis hinges on the attribution of geometry to describe how it should be used by downstream analyses. EGADS [8] provides a rich suite of attribute types (character strings, vectors of integers or reals, coordinate systems, *etc.*). Any number of uniquely named attribute name/value pairs may be put on BRep objects. Attributes placed on topology components within the geometry subsystem are persistent in that operations on attributed

BReps do not change the attribution, except when the operation destroys or creates new elements (Faces, Edges, and Nodes). The geometry subsystem also automatically maintains its own set of attributes that describe the geometric properties (*e.g.*, coordinate ranges, area, volume, centroid) and history of entities to enable capabilities such as analytic differentiation. These attributes are available to the modeler, but require no intervention to be maintained.

Attribution may capture a wide range of information, and the modeler has freedom to implement any desired schema. However, it is recommended that user attributes within CAPS are separated into two categories: describing what the geometry is and describing how it should be used for analysis (which, in broad terms, includes meshing). This dichotomization has three purposes: (1) to ease the mental burden of attribution on the modeler as complexity increases (geometric detail and analysis disciplines and fidelities), (2) to simplify the identification of BRep Edges and Nodes that are otherwise difficult to isolate, and (3) to mitigate the potential overloading and conflict of attributes for different analyses operating on the same geometry. This distinction also supports the typical separation between vehicle designer (configurator) and disciplinary practitioner (analyst).

In the initial approach taken by CAPS, attributes for analysis were applied as part of the geometric build specification. In a relatively low complexity application [11, 12] using two levels of aerodynamic representation and a single structural representation, extensive attribution was required to specify boundary conditions, loads, and material properties for structural FEA models. Particular challenges were the unique identification of BRep Edges and Nodes to be attributed for boundary conditions and the selection of Faces that were part of a larger surface for applying a gradation of material properties. In the initial implementation, unique identification numbers based on the order of creation were used to isolate Faces, Edges, and Nodes. Unfortunately, these identifiers varied with changes in parameters or build procedure. Alternatively, Faces could be identified by projecting a vector to find the nearest surface (though this could be problematic depending on the complexity of the geometry). However, an analogous operation for Edges and Nodes is not obvious, and the projection is subject to numerical tolerances (though it was found to be robust enough in that particular application). Overcoming these challenges presented a potentially overwhelming workload for an individual modeler, and distributing the work amongst a team would only add confusion.

The two-stage approach to attribution is designed to address these challenges. During the geometry build process, attributes identify geometric entities as the design model produces them. These attributes help the configurator and analysts locate and identify specific geometric entities. For example, during the build, BRep Faces representing the skins, spars, and ribs may be tagged as such, along with information such as indexing (1, 2, 3, ...) or location (root, upper surface, *etc.*). Then, Faces may be identified as “upper skin,” Edges as “adjacent to root rib and spars,” or Nodes as “adjacent to lower skin, spar 1, and rib 5” (Fig. 5). A subset of Faces might also be selected as “lower skin adjacent to rib 3 and rib 4.” Attributes may be applied as broadly as an entire BRep Body or as narrowly as a specific element of the BRep topology defined by the build process. It is sometimes still necessary to identify an Edge by its identification number during the build process. For example, the adjacency of the upper and lower surfaces is not sufficient to disambiguate the leading and sharp trailing edges of a wing. In this case, identifying both Edges immediately after the wing is created is straight forward and not prone to error in practice.

After the geometry is built, the identification attributes are leveraged to apply information for analysis pre-processing.

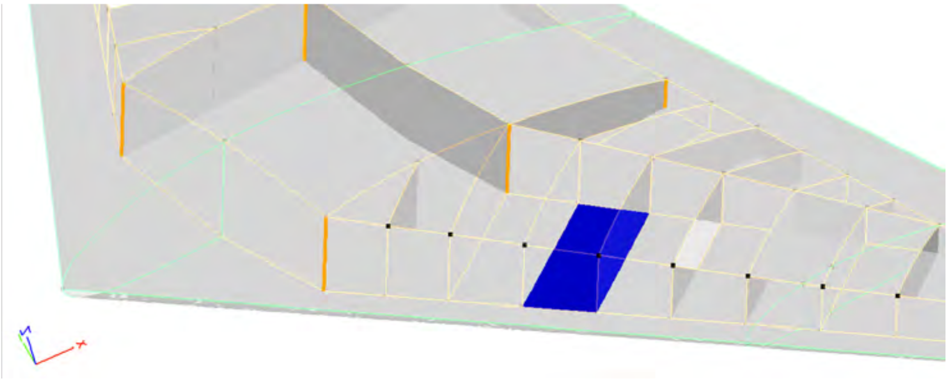


Fig. 5 Specific geometric entities may be isolated using geometric identification attributes, such as “Face on left lower surface adjacent to spars 1 and 2 and ribs 3 and 4” (shown in blue), “Edges adjacent to both the root rib and wing spars” (orange lines), and “Nodes adjacent to ribs, spar 1, and left upper surface” (black dots).

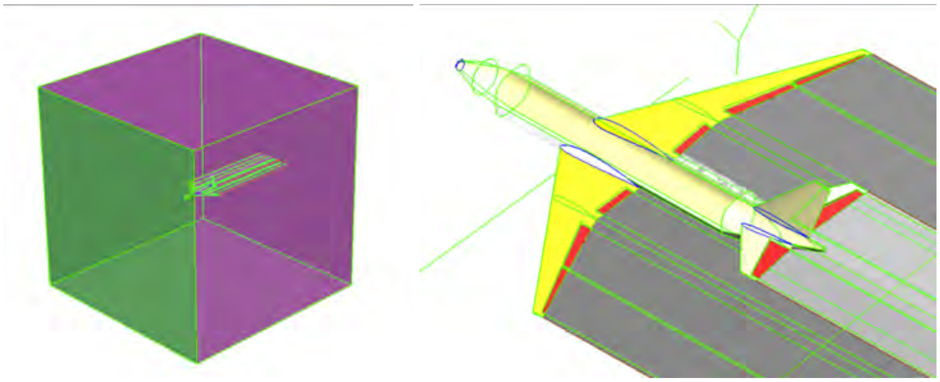


Fig. 6 Analysis attributes to specify boundary conditions for a potential flow analysis: inflow (green), outflow (magenta), outer mold line (yellow), control surfaces (red), wake (gray).

Fig. 6 illustrates the attribution of geometry to apply boundary conditions for a potential flow analysis. In our structures example, material properties may be applied to Faces marked "upper skin." The same attributes may also be use to specify skin thickness design variable patches. Boundary conditions may be applied to the Edges "adjacent to root rib and spars" to fix the model in space or to permit free flight. A point load or concentrated mass might be applied to the Node "adjacent to lower skin, spar 1, and rib 5." It is critical to note that the particular analysis attributes are specific to the use case, but not to a particular realization of the design parameters. In our example, the analysis attribution scheme is specific to a model of the skin, spar, and rib surfaces, but remains valid as the vehicle OML changes or even as the internal structural topology changes (so long as the indexing scheme is defined accordingly).

Analysis pre-processing occurs within an analysis interface module (AIM). AIMs are dynamically loaded at CAPS runtime (and are shared objects in Linux and the Mac or dynamically loaded libraries using Windows). This technique allows for the extendability of CAPS without building in all of the possible connections to analysis and meshing packages. It also allows for the use of proprietary or sensitive software without exposing the details to an *Open Source* world.

Each analysis package integrated with CAPS has one (or more) unique AIM(s). AIM developers are responsible for providing the functionality to write the formatted input required to perform a particular analysis, as well as to parse the associated output. If the meshing/analysis software has its own API, then that can be used directly so that no file input/output is required. During pre- and post-processing, the AIM leverages the attributed geometry, allowing it to have access to the appropriate model representation and to associate analysis information (*e.g.*, properties, results) with specific portions of the geometry via the attributes.

In some instances, an AIM may generate a mesh for its particular analysis. In other cases, an analysis AIM may be linked to a second AIM exposing a dedicated meshing package, with a specific parent/child relationship. In both cases, the analysis attribution may also be used to specify mesh controls. The placement of attributes depends on the manner in which the algorithm operates. In some instances, Faces are identified to provide an approximate element size for the resulting mesh. In other cases, particular Edges may be identified to specify a node distribution to seed the mesh generation. In yet other instances, controls on element spacing may be applied to Nodes to guide the algorithm in producing smooth, high-quality meshes along Edges and Faces.

D. The CAPS Executive and the CAPS Data Manager

CAPS is not a framework in and of itself, but provides an infrastructure for multifidelity, multidisciplinary modeling that can be integrated within a framework. The role of a CAPS executive is to organize and configure particular AIM instances into workflows, drive the selection of design and analysis parameters, and trigger the actual execution of analyses within the design environment. As the CAPS API is written in C, C programs are a natural implementation. More frequently, however, scripts have been written using the pyCAPS Python implementation [13], or more recently the gCAPS extension (*ibid.*) exposing the capability of the Air Force Research Laboratory, Aerospace Systems Directorate, Multidisciplinary Science and Technology Center's distributed computing environment [14–16]. Implementations contributed by other collaborators may be envisioned, as well.

It is within the executive that AIM instances are configured, and actual analysis information is specified and associated with attributed geometry. For example, a symmetry boundary condition may be applied to geometry marked

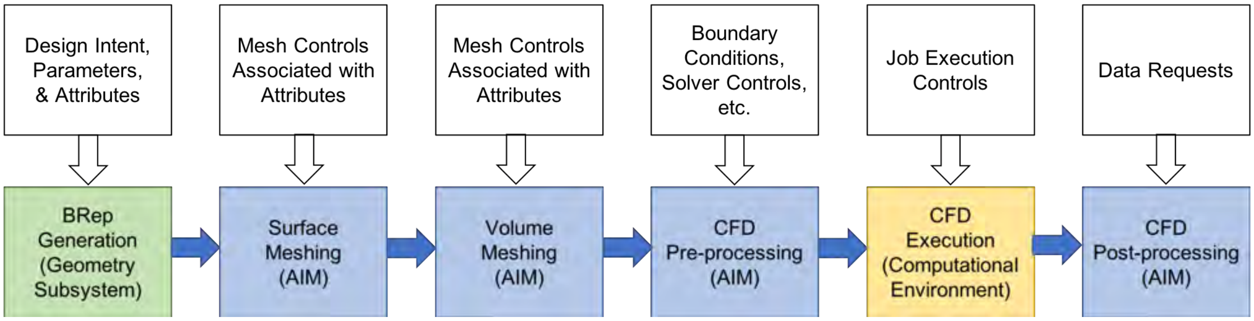


Fig. 7 A sample workflow for CFD analysis. Upper blocks indicate information provided by the user to configure the workflow, and lower blocks illustrate the division of effort into different AIMs and subsystems.

with a specific attribute value, or similarly material properties and gauges may be specified for a different attribute. The executive also specifies information that does not correlate with geometry, such as angle of attack, Mach number, governing equations, trim conditions, or solution controls like number of iterations or time step size. Within the AIM implementations, care has been taken to provide commonality of naming across similar analysis packages to allow easy interoperability.

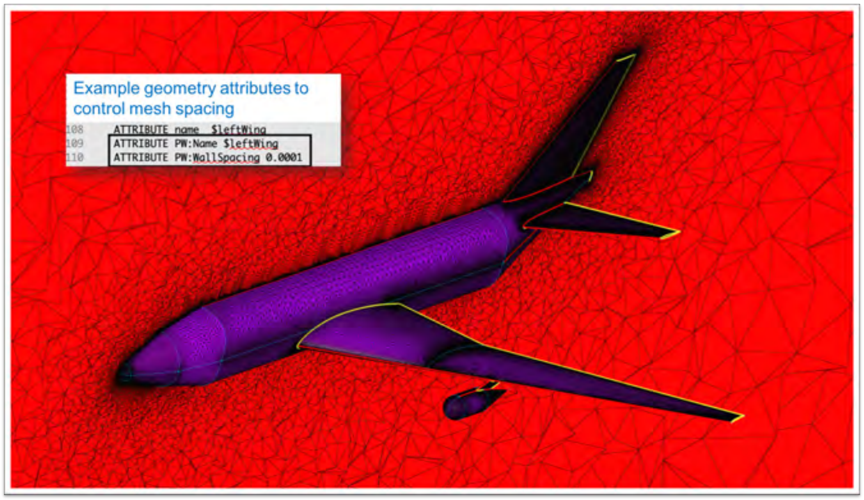
Multiple AIM instances combine to create workflows. A common example appears in three dimensional CFD, consisting of three AIMs: a surface mesher, a volume mesher, and a flow solver (Fig. 7). First, the surface mesh is generated on the geometric boundaries, making use of the attributes controlling element size. The surface mesh then seeds the volume mesh, which is finally given to the flow solver AIM, which specifies additional information about the solution process, including boundary conditions, equation sets to solve, and convergence requirements. A second common workflow appears in aeroelasticity, where the same CFD solver may be coupled with a structural FEA solver. In this case, the workflow consists of transferring pressures from the aerodynamics solution to the structural solver, and displacements from the structural solution back to the aerodynamics solver. Currently, the aerodynamic solver must be capable of deforming its own mesh in response to the structural displacements.

CAPS tracks the status of the executing workflow by the state of *value objects*. Value objects reflect the available inputs and outputs from either the geometry subsystem or individual analyses. When linked, these objects create a hierarchy where the change in one value invalidates its *downstream* usage. A *dirty* object requires the rebuilding of geometry or rerunning of AIMs so that the overall CAPS status can be made *clean*. In the process, the flow of information between coupled AIMs and the geometry subsystem also occurs. In doing so, the geometry is persistent and plays an active role. In the aeroelasticity example, the geometry serves as a vehicle for interdisciplinary data transfers, aided by attribution and the stemming of the multiple geometric views from a single design model. The surfaces that are conceptually similar (*e.g.*, the wing upper surface) between the structural and aerodynamic views may be attributed as a common bound for sharing data. Using the mathematical definition of the geometry as a conduit, displacements and pressures on dissimilar analysis meshes may be automatically transferred, using either interpolative or conservative schemes to avoid the accumulation of errors.

IV. Applications in the Literature

Many samples in the literature highlight the broad capabilities and applicability of CAPS that have evolved over its development. Many of these examples [10, 13, 17–19] were organized into a special session on CAPS at the 2019 AIAA SciTech forum in San Diego, California. Several others [20–23] were presented as companion papers in related sessions.

This first sampling of papers overview core technologies upon which specific applications are often built. Durscher and Reedy [13] describe the implementation of the pyCAPS Python interface and related gCAPS extension in Groovy and Java. pyCAPS simplified the construction and configuration of workflows using CAPS, and gCAPS extends pyCAPS with the goal of integrating CAPS within a distributed, heterogeneous computing environment. Karman and Wyman [20] demonstrate the use of geometric attribution to control automated unstructured meshing within Pointwise (Fig. 8). It should be noted that similar work is also being performed by Marcum at Mississippi State University using the advancing front, local reconnection (AFLR) libraries (Fig. 9). Docampo-Sanchez and Haimes [21] describe recent advances in unstructured quadrilateral meshing being developed under the CAPS effort (Fig. 10), targeted at structural analysis. Another capability developed under the CAPS effort is the ability to create high-quality representations from



Example geometry attributes to control mesh spacing

```
    ATTRIBUTE name  $leftWing
   ATTRIBUTE PW:Name $leftWing
   ATTRIBUTE PW:WallSpacing 0.0001
```

Fig. 8 Attributed, water tight geometry enables high-quality, automated meshing within Pointwise (courtesy S. Karman, Pointwise, Inc.[20]).

point cloud data, as presented by Jia and Dannenhoffer [24] and Dannenhoffer [25]. The ability to couple adjoint-based design environments with geometric sensitivities to user design parameters has been researched by Dannenhoffer and Haimes [26].

The CAPS effort has also brought about advances in the area of structural modeling. Drela *et al.* [17] introduce research into the modeling concept of the hybrid shell model (HSM). Goals of HSM are to provide a faster, simpler approach to structural modeling for early design by lumping properties into an outer shell, which may naturally be coupled with aerodynamic analyses to produce an aeroelastic capability. Canfield *et al.* [10] utilize the analytic derivatives of the geometry subsystem to produce structural shape sensitivities from a traditional finite element solver. Gandhi *et al.* [18] describe their use of the design model to rapidly produce parametric, full-vehicle structural models for global and localized failure analyses (Fig. 11). Kao *et al.* [27] also used CAPS-generated models for physics-based weight estimation and sizing of a wing and fuselage, both in isolation and in combination.

The high-quality, multifidelity geometry and automated analysis pre-processing have brought about a new capability for design, analysis, rapid prototyping, and experimentation. The geometry representations used for analysis naturally extend to solid bodies that drive additive manufacturing processes. Donovan and Allison [28] present force coefficient predictions and measurements for a fighter configuration using two levels of CFD analysis and a printed wind tunnel model. Similarly, Pankonien and Reich [29] demonstrate the production of printed aeroelastic models that may be affordably, intentionally tested to flutter failure (Fig. 13), albeit initially using traditional CAD geometry. This work motivated Pankonien *et al.* [23, 30] use the multifidelity nature of CAPS geometry to design, analyze, and test active conformal control surfaces (Fig. 12). First, the mathematical description of the internal structure is checked for geometric feasibility; second, a shell model of the structure is constructed to analyze its aeroelastic performance; finally, a thickened, solid model is produced to additively manufacture a specimen for aeroservoelastic wind tunnel testing. Pankonien *et al.* [22] would later proceed to use CAPS to vary the scale, mass, and structural properties as a precursor to scaling of low-cost aeroelastic models. Pankonien *et al.* [31] further make a novel use of attribution to create speckle patterns for digital image correlation using multimaterial printing for structural components (Fig. 14). This technique shows promise for accelerating specimen preparation and evaluating the performance of non-traditional structural topologies.

Multiple studies in the development of multifidelity surrogate modeling and design optimization techniques have leveraged the modeling capabilities of CAPS. Bryson [11] and Bryson and Rumpfkeil [12] demonstrate an optimization algorithm that accelerates the aero-structural shape and sizing optimization of a vehicle by combining data from low-order aeroelasticity models with Euler CFD coupled to a modal structural solver. CAPS not only produced multiple representations of the aerodynamic and structural models, but also managed the transference of mode shapes from the structural solution to the CFD mesh. Bryson and Rumpfkeil [32] also use the multifidelity aerodynamics models to highlight an improved approach to combining data sources in polynomial chaos expansions. Clark *et al.* [33] compare several surrogate modeling techniques to approximate vehicle performance, including CFD aerodynamics and variable-cycle engine modeling, to support vehicle design based on mission effectiveness rather than traditional

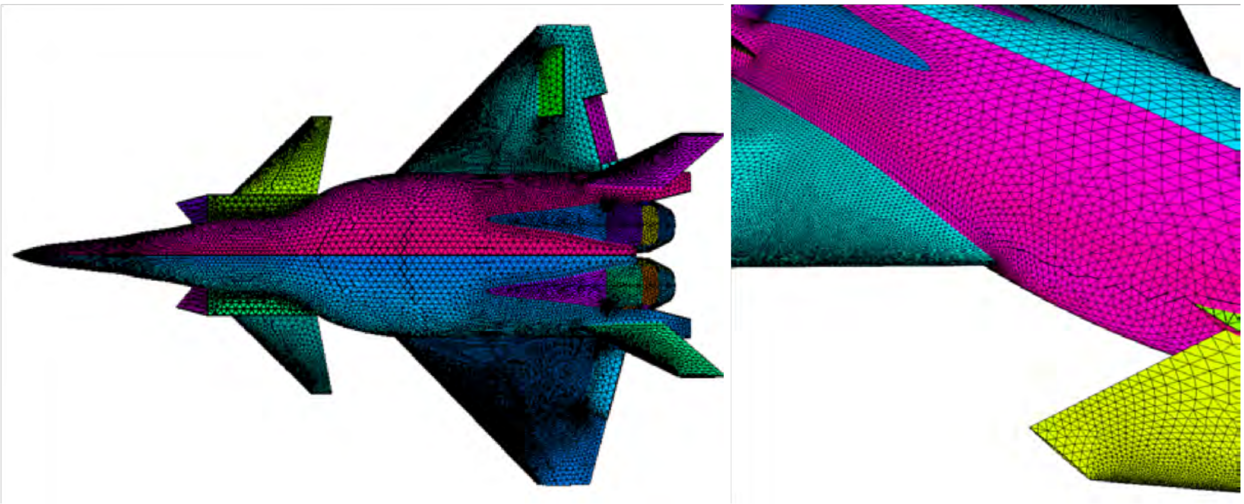


Fig. 9 Through tight integration with geometry via an Analysis Interface Module, AFLR is able to refine meshes based on surface curvature.

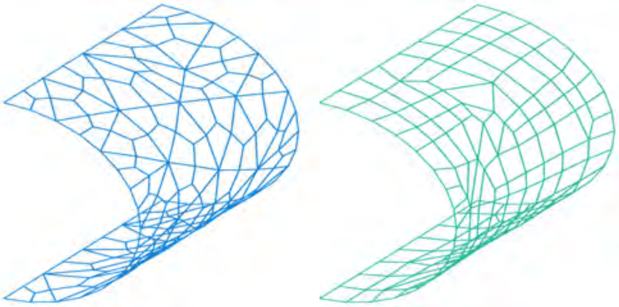


Fig. 10 Unstructured quadrilateral meshes may be obtained by splitting triangular meshing, regularizing the vertices, and optimizing the element angles and sizes.

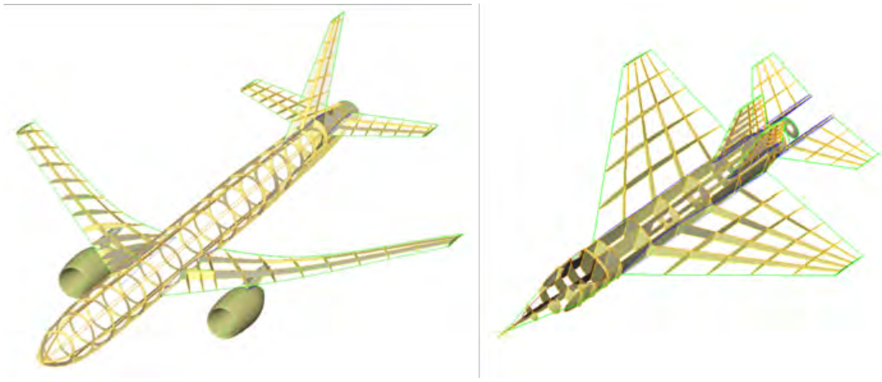


Fig. 11 The design model drives the creation of parametric models of conformal, internal structure, which may be used to perform both global stress analysis and local failure analysis [18].

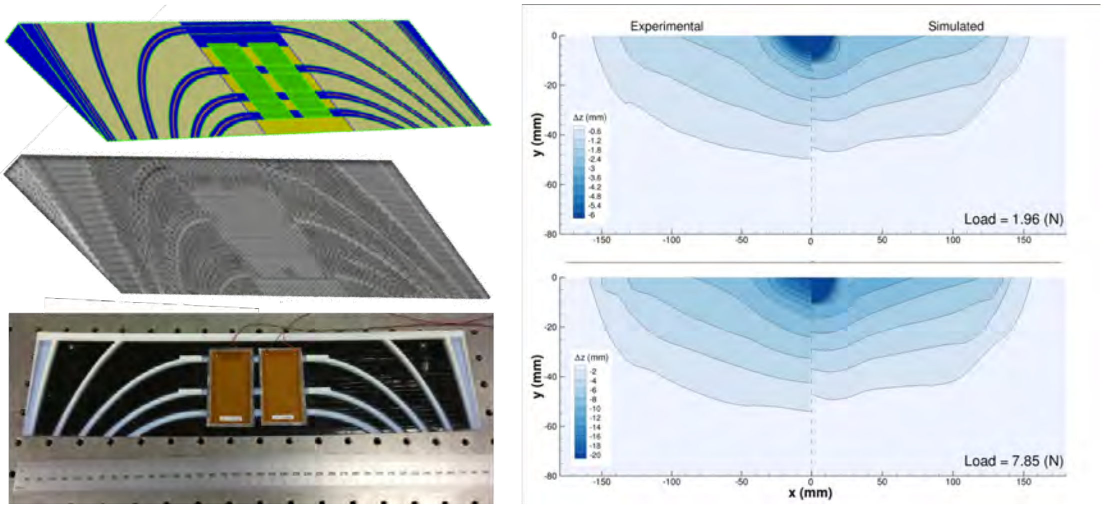


Fig. 12 From a single design model of a conformal control surface (top left), shell models for aeroelastic analysis (middle left) and solid models for multimaterial 3D printing (bottom left) may be automatically produced, accelerating comparison of simulation and experiment (right) [30].

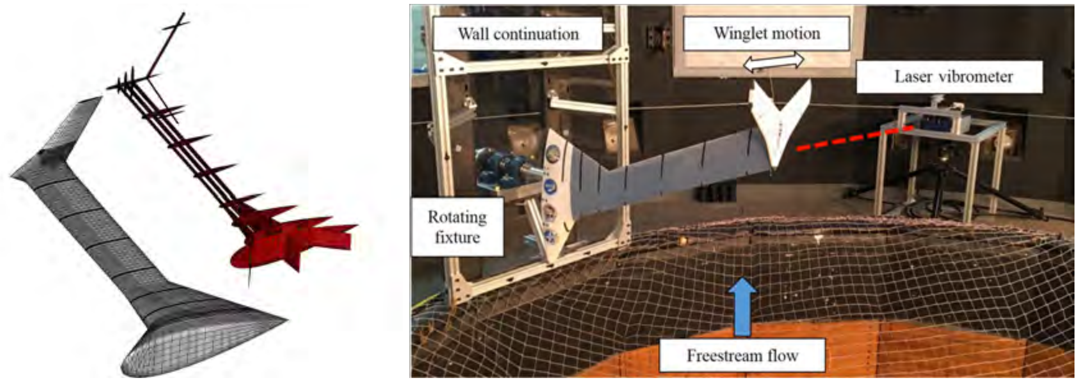


Fig. 13 Attributed, parametric models of outer mold line and internal structure (left) facilitate aeroelastic scaling studies [22], which extend to the rapidly producing affordable, 3D-printed, aeroelastic wind tunnel models [29] (right) that may be tested to failure.

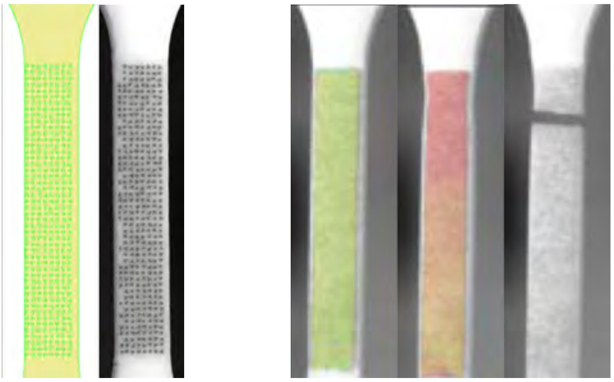


Fig. 14 Attributed, printer-ready geometry enables multi-material 3D printing of built-in digital image correlation speckle patterns (left) to accelerate experimentation and permit measurement of internal strain fields (right) [31].

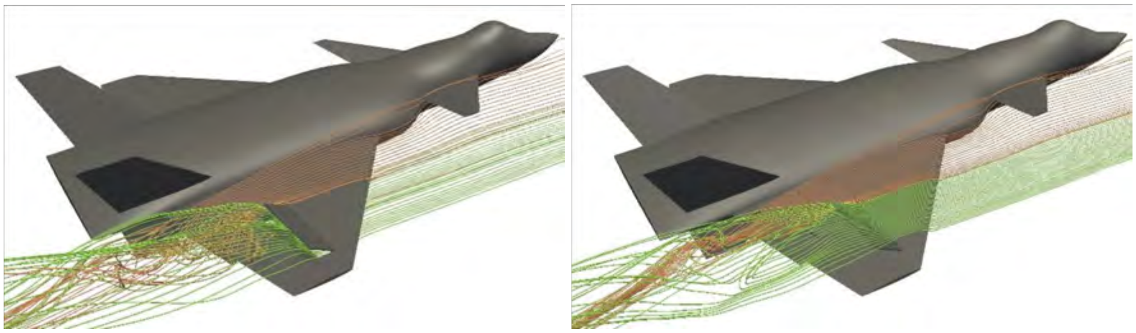


Fig. 15 Parametric geometry with automated meshing and analysis pre-processing enables physics-based prediction of stability derivatives with innovative control effectors [35]. Note the deflection of the inboard trailing edge flap and the flow-through of the spoiler-slot-deflector (left) versus the spoiler (right).

engineering requirements. Allison, Alyanak, and Bhagat [34] utilized early CFD meshing capabilities to improve the modeling of nozzle installation effects for advanced propulsion modeling, though the CAPS effort was in its infancy at the time.

CAPS has also been used to feed the stability and control analyses of advanced aircraft. Bhagat, Allison, and Alyanak [35] parametrically-define several conventional and innovative control effectors for a fighter configuration (Fig. 15). Using the design model and analysis AIMs, different control suites may be quickly generated and analyzed using body-fitted CFD to calculate stability derivatives for varying angles of attack and side slip. Meckstroth [19] also develops a parametric model supporting shape optimization of a fighter to minimize control power required.

V. Conclusion and Future Work

We believe that a shift in the role and representation of geometry plays a central role in enabling an environment capable of rapid, multifidelity, multidisciplinary design. Most significantly, we advocate for constructing design models driven by design intent and conceptual elements that comprise a vehicle, rather than a producing a single—albeit typically parametric—view of geometry, which is traditionally approached from a manufacturing rather than analysis mindset. Such a design model enables the construction of multiple, analysis-specific views from a single specification, eliminating the necessity and ambiguity of reinterpreting geometry for different purposes. From this representation, it naturally extends that geometry serves as a conduit for transferring data between coupled analyses. Hence, the geometry should play an active role throughout the analysis process extending beyond analysis pre-processing.

We have also found persistent attribution of geometry to be a critical element of the design environment. When coupled with trimmed, watertight geometry, attributes drive the automated generation of analysis meshes and inputs, removing a bottle neck that precludes using many high-fidelity analyses in the early design process. Ultimately, the attributes provide a linkage between geometry and non-geometric information required for analysis. However, in our experience, the application of analysis attributes in the midst of the design model specification can present more of a conceptual burden than advantage, particularly for structural models requiring identification of BRep Edges and Nodes. A more practicable approach is to attribute geometry with its conceptual purpose as it is built, and to apply analysis-specific attributes to the analysis-specific views after their construction. This dichotomy has the advantage of supporting the typical separation of the modeler, or configurator, from the analyst.

Underpinned by the attributed design model, we have produced and demonstrated the CAPS infrastructure to manage the flow of information between the geometry subsystem, various analysis interface modules (AIMs), and the environment driven by an external design process. Beyond producing the design model, the user's primary interaction with CAPS is in the configuration and coordination of AIMs within an executive process. The AIMs themselves perform pre- and post-processing, having a one-to-one mapping with a particular analysis package. The actual execution of an analysis within the computational environment is managed external to the AIM by design, as to permit interoperability across a wide range of environments.

Aside from the shift of cultural mindset required, perhaps the greatest barrier to adoption of CAPS technology is learning to script the design model. While the Engineering Sketch Pad (ESP) provides a native viewer with built-in script editor, the graphical process entails selecting model operations from a menu and auto-generating the corresponding script. In our experience, users typically construct the design model within ESP by iterively making small script modifications and viewing the results. To users accustomed to graphical geometry layout and perhaps uninitiated in computer programming, producing a design model script can be a daunting process. One step we have recently taken to lower this barrier is the introduction pre-coded analysis view generators. When loaded, these scripts take geometry attributed under a convention and produce the representations required by certain analyses. Similarly, a library of typical components with a predefined design intent could be envisioned, allowing users to construct vehicle models with a building block approach.

Looking more outwardly, widespread adoption will require the incorporation of these modeling philosophies into industry-standard modeling packages and design frameworks. Our hope is that by demonstrating continued success, the ideas espoused by CAPS will attain broad acceptance to advance the state of design by multifidelity, multidisciplinary analysis.

VI. Acknowledgment

This work was funded by the Air Force Research Laboratory, Aerospace Systems Directorate, under the "Computational Aircraft Prototype Syntheses" effort with Dean Bryson as the technical monitor. Approved for public release; distribution is unlimited. Case number 88ABW-2018-5906, cleared 27 Nov 2018.

References

[1] Allison, D. L., Morric, C. C., Schetz, J. A., Kapania, R. K., and Watson, L. T., "Reevaluating conceptual design fidelity: An efficient supersonic air vehicle design case," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 230, No. 3, 2016, pp. 581–598.

[2] Davies, C., Stelmack, M., Zink, S., De La Garza, A., and Flick, P., "High Fidelity MDO Process Development and Application to Fighter Strike Conceptual Design," *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, Indianapolis, IN, 2012.

[3] Alyanak, E., and Allison, D., "Multi-Parameter Performance Evaluation, the Next Step in Conceptual Design Concept Assessment," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, Kissimmee, FL, 2015.

[4] Alyanak, E., Kolonay, R., Flick, P., Lindsley, N., and Burton, S., "Efficient Supersonic Air Vehicle Preliminary Conceptual Multi-Disciplinary Design Optimization Results," *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, Indianapolis, IN, 2012.

[5] Council, N. R., *Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Systems Acquisition*, The National Academies Press, Washington, DC, 2008. doi:10.17226/12065, URL `https://www.nap.edu/catalog/12065/pre-milestone-a-and-early-phase-systems-engineering-a-retrospective`.

[6] Alyanak, E., Durscher, R., Haimes, R., Dannenhoffer, J. F., III, Bhagat, N., and Allison, D., "Multi-Fidelity, Gemoetry-Centric Multi-Disciplinary Analysis for Design," *AIAA Modeling and Simulation Technologies Conference*, American Institute of Aeronautics and Astronautics, Washington, DC, 2016.

[7] Dannenhoffer, J. F., III, "OpenCSM : An Open-Source Constructive Solid Modeler for MDAO," *51st AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Grapevine, TX, 2013. doi:10.2514/6.2013-701.

[8] Haimes, R., and Drela, M., "On the Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design," *50th AIAA Aerospace Scienes Meeting*, American Institute of Aeronautics and Astronautics, Nashville, TN, 2012.

[9] Dannenhoffer, J. F., III, and Haimes, R., "Generation of Multi-fidelity, Multi-discipline Air Vehicle Models with the Engineering Sketch Pad," *54th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2016.

[10] Canfield, R., Alnaqbi, S., Durscher, R., Bryson, D., and Kolonay, R., "Shape Continuum Sensitivity Analysis using ASTROS and CAPS," *AIAA SciTech*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[11] Bryson, D. E., "A Unified, Multifidelity Quasi-Newton Optimization Method with Application to Aero-Structural Design," Ph.D. dissertation, University of Dayton, Dayton, OH, 2017.

[12] Bryson, D. E., and Rumpfkeil, M. P., "Aeroelastic Design Optimization using a Multifidelity Quasi-Newton Method," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, Kissimmee, FL, 2018.

[13] Durscher, R., and Reedy, D., "pyCAPS: A Python Interface to the Computational Aircraft Prototype Syntheses," *AIAA SciTech*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[14] Sobolewski, M., and Kolonay, R., "Service-Oriented Life Cycles for Developing Transdisciplinary Engineering Systems," *Transdisciplinary Lifecycle Analysis of Systems – Proceedings of the 22nd ISPE Inc. International Conference on Concurrent Engineering, CE 2015*, Advances in Transdisciplinary Engineering, Vol. 2, edited by R. Curran, N. Wognum, M. Borsato, J. Stjepandic, and W. J. C. Verhagen, International Society of Productivity Enhancement, Inc., IOS Press, 2015, pp. 541–551.

[15] Kolonay, R. M., "A Physics-Based Distributed Collaborative Design Process for Military Aerospace Vehicle Development and Technology Assessment," *International Journal of Agile Systems and Management*, Vol. 7, No. 3/4, 2014, pp. 242–260.

[16] Burton, S. A., Alyanak, E. J., and Kolonay, R. M., "Efficient Supersonic Air Vehicle Design using the Service-Oriented Computing Environment (SORCER)," *International Journal for Simulation and Multidisciplinary Design Optimization*, Vol. 5, No. A25, 2014, pp. 1–11.

[17] Drela, M., Galbraith, M., and Allmaras, S., "Hybrid Shell Model for Aeroelastic Modeling," *AIAA SciTech*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[18] Gandhi, V., Joe, J., Dalir, H., and Dannenhoffer, J., "Rapid Generation of Parametric Aircraft Structural Models," *AIAA SciTech*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[19] Meckstroth, C., "Parameterized, Multi-fidelity Aircraft Geometry and Analysis for MDAO Studies using CAPS," *AIAA SciTech*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[20] Karman, S., and Wyman, N., "Automatic Unstructured Mesh Generation with Geometry Attribution," *AIAA SciTech*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[21] Docampo-Sanchez, J., and Haimes, R., "Smooth Quad Meshing via Vertex Regularization and Curvature Driven Optimization," *AIAA SciTech*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[22] Pankonien, A. M., Durscher, R. J., and Bhagat, N., "Parameteric 3D-Printable Flutter Model for Constrained Aeroelastic Scaling," *AIAA SciTech*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[23] Pankonien, A., Durscher, R., and Bhagat, N., "Aeroelastic Trade Study of a Conformal Control Surface for a Printable Wind Tunnel Model," *AIAA SciTech*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[24] Jia, P., and Dannenhoffer, J., "Generation of Parametric Aircraft Models from a Cloud of Points," *54th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2016.

[25] Dannenhoffer, J. F., III, "The Creation of a Atatic BRep Model Given a Cloud of Points," *55th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Grapevine, TX, 2017.

[26] Dannenhoffer, J. F., III, and Haimes, R., "Using Design-Parameter Sensitivities in Adjoint-Based Design Environments," *55th AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Grapevine, TX, 2017.

[27] Kao, J. Y., White, T. L., Leonard, M. S., Reich, G. W., and Burton, S. A., "Structural Sizing of Aircraft Wings and Fuselages in Conceptual Multidisciplinary Design Processes," *2018 Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, Atlanta, GA, 2018.

[28] Donovan, A. B., and Allison, D. L., "Multidisciplinary Design Including Geometry-Enabled Wind Tunnel Fidelty Level Analysis," *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, Denver, CO, 2017.

[29] Pankonien, A. M., and Reich, G. W., "Multi-Material Printed Wind-Tunnel Flutter Model," *AIAA Journal*, Vol. 56, No. 2, 2018, pp. 793–807. doi:10.2514/1.J056097.

[30] Pankonien, A. M., Durscher, R., Bhagat, N., and Reich, G. W., "Multi-Material Printed Trailing Edge Control Surface for an Aeroservoelastic Wind Tunnel Model," *2018 Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, Atlanta, GA, 2018.

[31] Pankonien, A. M., Durscher, R., Deaton, J. D., Piedel, A., and Bhagat, N., “Multi-Material Printing Integral Digital Image Correlation Patterns,” *2018 Aerodynamic Measurement Technology and Ground Testing Conference*, American Institute of Aeronautics and Astronautics, Atlanta, GA, 2018.

[32] Bryson, D. E., and Rumpfkeil, M. P., “All-at-Once Approach to Multifidelity Polynomial Chaos Expansion Surrogate Modeling,” *Aerospace Science and Technology*, Vol. 70, 2017, pp. 121–136. doi:10.1016/j.ast.2017.07.043.

[33] Clark, D. L., Jr., Allison, D. L., Bae, H., and Forster, E., “Metamodeling for Effectiveness Based Aircraft Design Under Uncertainty,” *2018 Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, Atlanta, GA, 2018.

[34] Allison, D., Alyanak, E., and Bhagat, N., “High Fidelity, Nonlinear, Integrated Nozzle Installation Effects for Numerical Propulsion System Simulation,” *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, Kissimmee, FL, 2015.

[35] Bhagat, N., Allison, D. L., and Alyanak, E., “Geometry Driven High Fidelity Stability Derivatives Obtained Using Automated CFD Analysis Process,” *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, Dallas, TX, 2015.