Abstract Modeling – Robust Automation of CFD Processes Made Easy

Authors:

Amol Patil
Bruce Webster, PhD
Davis Evans
Karlheinz Peters
Santosh T. Patil

Novus Nexus, Inc., Northville, MI, United States
http://www.novusnexus.com

Summary:

Facing an ever-growing worldwide competition, manufacturers in all industries are challenged to develop new and improved products as quickly and economically as possible. At the same time, their products and systems involved are growing in complexity. Increasing the use of virtual tests throughout all design cycles is a key factor to remain competitive and to develop products that perform as desired in a timely manner. Automated processes which work reliably, efficiently, and that are quick to implement will be important to get the full benefit from virtual tests. Even more so, when such automation democratizes simulation by empowering designers to initiate dependable virtual test themselves. Abstract modelling technology offers a new and elegant pre-processing approach to achieve these goals. Abstract modelling reliably connects design and simulation worlds, reduces non-productive analyst work, speeds up robust simulation processes, ensures the use of best practices, and preserves valuable corporate knowledge and expertise.

This paper will explain what abstract modelling is, including differences and commonalities with traditional pre-processors, its unique approach to automation, and how it can be used for "simulation apps". Two key aspects of an automated abstract modelling workflow are enabling designers to initiate dependable simulations while relieving analysts from tedious routine work.

Further topics include how abstract modelling helps to ensure the trustworthiness of simulation results (e.g. are they always comparable, especially if done by different people), reduce the impact from a lack of availability of CFD specialists, and make simulation results available in time for critical design decisions.

Keywords:

Abstract Modeling, CAE Automation, CAD, CFD, Democratization, Pre-Processing, Simulation, Simulation Application

1 Introduction

While most people have heard about abstract art, the concept of abstract modeling related to 3D-simulations is unknown to many analysts. What role can abstraction play, when we are designing parts or assemblies whose specific geometries determine how a product behaves?

Traditional art and simulation have a common goal: both attempt to capture objects as realistically as possible. Art uses paintings or sculptures, simulation virtual models for their representation of reality. The closer the results are to real life, the better. But as soon as we bring "abstract" into the equation, art and simulation no longer have the same goals. Abstract art does not attempt to represent an accurate depiction of a visual reality but instead uses shapes, colors, forms and gestural marks to achieve its effect. [1] Being an art discipline on its own, abstraction frees artists from objective reality and gives freer rein to their imagination. Simulation on the other hand is always tied to reality, usually products under development and their behavior, which means abstract modeling must ultimately consider real geometries and be able to relate to them. With this being the case, why abstract modeling?

As mentioned before, 3D-simulations are based on specific geometries and an analyst will get many variations to perform similar tests until the most suitable option is found. It also means that s/he needs to set-up the same simulation again and again for each geometry - a tedious, non-productive task slowing down the execution of simulations and taking away time that would be better spent on higher value assignments, such as creating more meaningful reports to better support decision making. Abstract modeling addresses this inefficiency (and more) by facilitating very easy creation of reusable simulation set-ups to robustly automate pre-processing and enable designers to start high fidelity simulations without becoming an expert. Unlike abstract art, abstract modeling's goal is to support simulating real objects, but the first step involves a separation of simulation set-up and real geometries using place holders instead - hence the set-up is done in an abstract way.

2 Business Pressures, Simulation's Potential and Challenges

Fast innovation while still delivering high quality products is important for manufacturers in all industries to compete in today's global economy. Applying virtual tests/simulations earlier and using them more frequently in all design cycles is an essential factor to remain competitive and to develop such products in a timely manner. As mentioned before, automated processes that work reliably, efficiently, and that are quick to implement will be key to get the full benefit from virtual tests.

Unfortunately, the reality in many organizations is different. Design engineers lack timely access to CFD and other CAE tools. Analysts spend 30% – 70% of their time preparing simulations with repeated set-ups - a delay that often slows down when designers receive feedback on the performance of their models, which can cause schedule slippages and quality issues.

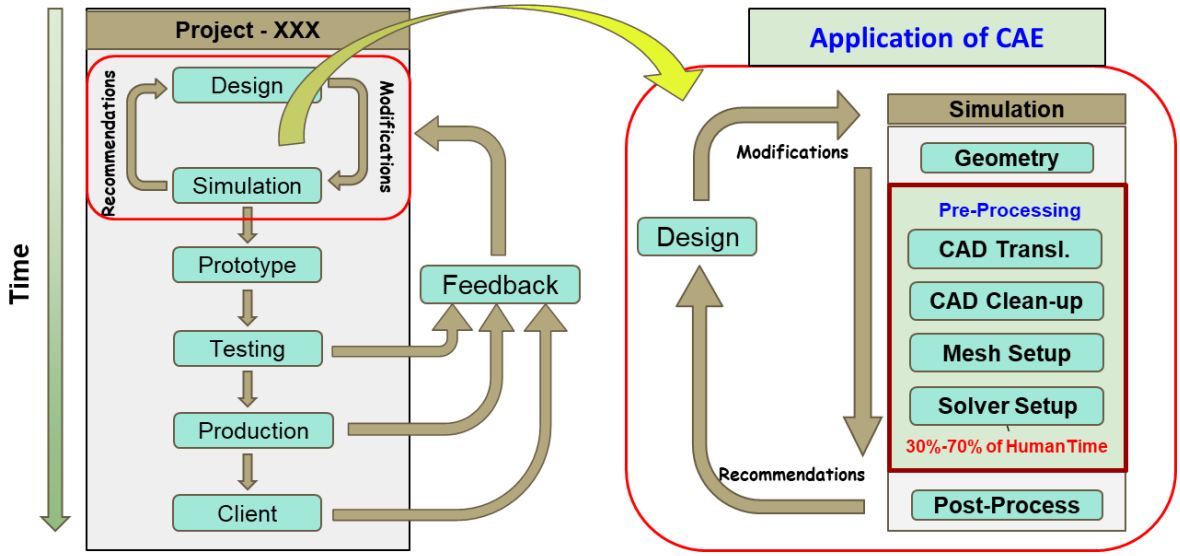

Figure 1: Product Life Cycle – Simulation Centric

According to a survey by Tech-Clarity, an independent research firm, 96% of their respondents see advantages if their designers have direct access to simulation tools. [2] Expected benefits include:

- Earlier detection of problems
- Possibility to reduce the number of prototypes
- Less rework
- More innovative designs

But at the same time, 65% of those respondents believe design engineers do not perform as many simulations as they should. Direct access to simulation tools is hampered by:

- Lack of simulation expertise
- Complexity of simulation tools, too hard to use
- Simulation turn-around time considered too long

The survey above confirms two important points. First, simulation is recognized as a key technology for companies to remain competitive because it makes development processes more efficient, creates superior products and enables more innovation, faster. Second, simulation is not used to the degree it should be. Lack of resources and lack of knowledge are holding companies back from exploiting the full potential of simulation. While automation of simulation processes could help a lot, it is not used to the degree necessary. But why?

Let us look at common automation methods, where they are used and how abstract modeling compares to those. The main methods are scripting, drag-and-drop environments (e.g. PIDO systems and simulation user environments, which often also require extensions applying scripts) and model-based templates found with CAD integrated solutions.

Simulation automation most often involves some kind of scripting. Writing scripts requires an understanding of all potential geometry related topologies to correctly apply simulation parameters as well as how to operate all engineering software products through the script. When working with complex geometries and/or physics, those scripts quickly become complicated, taking a long time to create and making them difficult to maintain. Abstract modeling on the other hand automates the “CAD to solver” process without any scripting, neither by the user nor behind the scenes. Simply put, automation based on abstract modeling is much easier and faster to realize than scripting: the effort is roughly equivalent to manually setting up a single simulation for a specific geometry.

Model-based templates are created for a specific geometry with its underlying topology; the templates are re-usable for modified geometries without user interaction if the topology does not change. Once the topology changes the user must manually update the simulation set-up. This again is different with abstract modeling: the abstract model (meaning the simulation set-up contained therein) is geometry and topology agnostic enabling automatic simulation processes for any conceivable geometry variation.

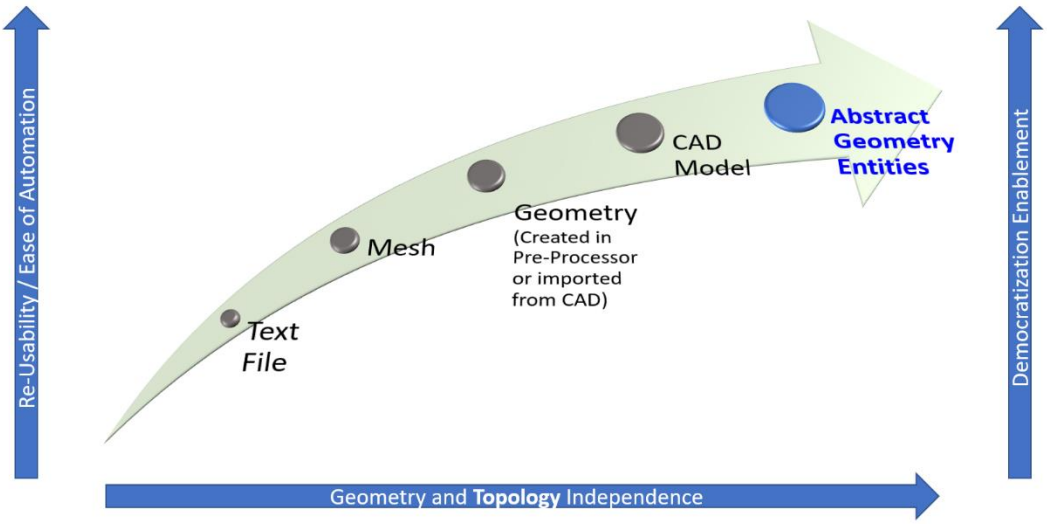

Figure 2: Simulation Parameter Connection

In the following chapters we will demonstrate how abstract modeling can help development organizations to overcome these automation challenges as well as enable designers to initiate dependable simulations whenever they have finished a new product version. Abstract modeling based CAENexus (FluidNexus for CFD) is used for UI-screenshots as an example of how abstract modeling can be implemented.

3 A Closer Look at Abstract Modeling

3.1 Why Abstract Modeling

The core notion of abstract modeling is to approach simulations in the following way: to make the set-up reusable and ready for automation, to broaden the base of users who can initiate reliable simulations, and to always deliver comparable results. An important aspect here is that it becomes fast and easy for an analyst to create these reusable simulation set-ups in the form of abstract models. This has been achieved by retaining a similar user experience as known from traditional pre-processors avoiding the need to define the automated process through time-consuming, complex scripting. In other words, abstract modeling makes it possible to automate specific simulations in hours or days versus weeks or months when using scripting or other conventional methods.

How is this done? Setting up a CFD simulation abstractly requires considering potential geometries and using placeholders for the various elements of any given geometry. In the case of the CAENexus family, these placeholders are called “classes”. Like geometry, those classes have dimensions (2D or 3D for CFD) and are usually named by users according to their material or function. CAD parts or their faces refer to classes defined in an abstract model via text attributes, which allows re-using such abstract models with any geometry.

CFD analysts are often performing the same type of CFD simulation on many varying geometry instances. When setting up a CFD analysis, for example when calculating internal flow rates, there are both common physical and material aspects between product variations, and there are differences between them also, e.g. in geometry and possibly functionality. The two simple models in Figure 3 below demonstrate this.

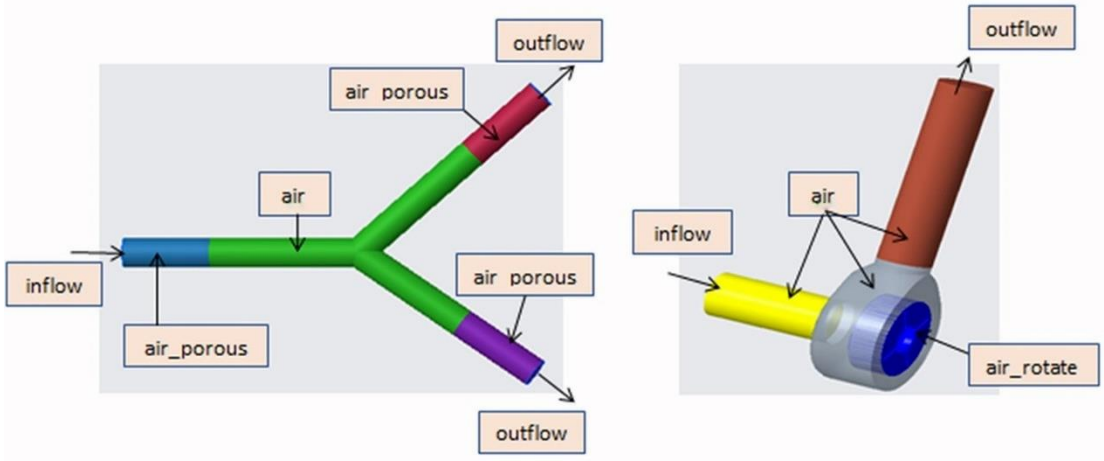

Figure 3: Geometry Examples

In these two models, each part has air (spaces) and some other material which require element set attributes. They differ in one having porous material, the other a rotating device. A user needs to also apply material parameters, inflow, outflow, and/or wall conditions, etc. in both models to perform a simulation. In a traditional simulation environment, the analyst must repeat those set-ups every time the geometry model changes. This process is not only time consuming but could also lead to non-comparable results if the simulation set-up is not done in a consistent way, e.g. when mesh strategies or selected turbulence models change between simulation runs.

Methods like scripting or templates represent established possibilities to automate CFD and other CAE processes, but the effort to create, test and maintain those scripts is very high. Complex geometries and related changes are hard to support and not always possible. CYON Research published a white paper [3] in which they grouped CAE automation into three categories: straightforward, difficult and hairy. The automation approaches evaluated were mesh-based templates, model-based templates and

abstract CAE-modeling. In their analysis, all approaches can handle straightforward cases, difficult cases require either model-based templates or abstract modeling, while abstract modeling is the only option for automating hairy cases. In summary, abstract modeling provides the only approach that can automate in all cases.

Unlike templates, abstract models are agnostic to specific shapes and topologies. Using classes and their relations (further explained below) as geometry placeholders, the effort to create an abstract model is comparable to the effort of setting up a simulation for a normal CAD part, which is significantly faster than what is needed for a script or template. Abstract models are easy to understand, maintain and expand, making them a natural tool to preserve, share and consistently employ simulation best practices. Abstract modeling-based automation frees analysts from repetitive work and is highly reliable, which ensures that each simulation is done right, independent of who initiates it, when or where. Such automation therefore helps with democratizing simulation by enabling CAD designers without CFD know-how to initiate dependable simulations. Other beneficial effects include an overall higher efficiency of simulation processes, simulation results that are always comparable, as well as a reduction of development time and cost.

Abstract models are flexible in that they can be created containing a larger, more general group of potential classes/materials than needed for a given individual part. When such a broader-use abstract model is combined with a CAD model that refers to a subset of available classes, only those classes/materials specified on the CAD model plus their related physics will be considered for the simulation input.

In summary, abstract modeling confers several advantageous features that lead to cost and time savings in product development:

a. A simpler, faster way to reliably automate the process from CAD to solver input
b. Increasing analyst capacity for value adding tasks
c. Enabling CAD designers to initiate dependable simulations with always comparable results
d. Best-Practice/Knowledge capturing as a management tool

3.2 Abstract Modeling Ingredients

3.2.1 Abstract Model

An abstract model is made up of abstract classes with their “child items”, abstract relations with their “child items”, and all related CAE attributes. The term “child items” refers to specific subsets within the abstract model. There is NO specific geometry associated with abstract classes or relations and their child entities. That is why you do not see any geometry in the abstract model user interface.

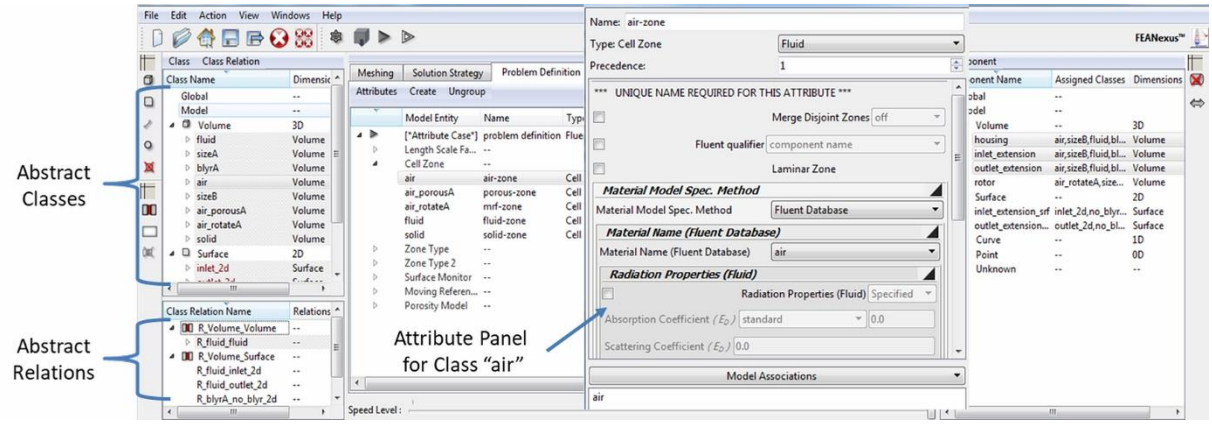

Figure 4: CAENexus Abstract Model User Interface

A relation between any two abstract classes represents the common entity that these classes share. Relations are useful when a user wants to specify a CAE attribute on such a common entity. For example, if a user wants to specify an interior boundary condition on the common face between two fluid objects, then the relation between their classes allows to easily specify such an interior attribute.

CAE attributes are specified on abstract classes, relations, or their child items based on where they would later be applied on any real geometry.

3.2.2 CAE Enabled CAD Model

To connect an abstract model with real geometry a user needs to define string parameters on 3D and 2D parts using the attribute system of the CAD software. These string parameters are called SCLASS* and SCOMP, which are keywords for CAENexus. SCLASS* stands for “Simulation CLASS” and SCOMP stands for “Simulation COMPonent”. The text or numerical values of SCLASS* string parameters must be identical to the abstract class names defined in an abstract model.

Figure 5 shows one of the models tagged in PTC Creo® with SCLASS* and SCOMP parameters.

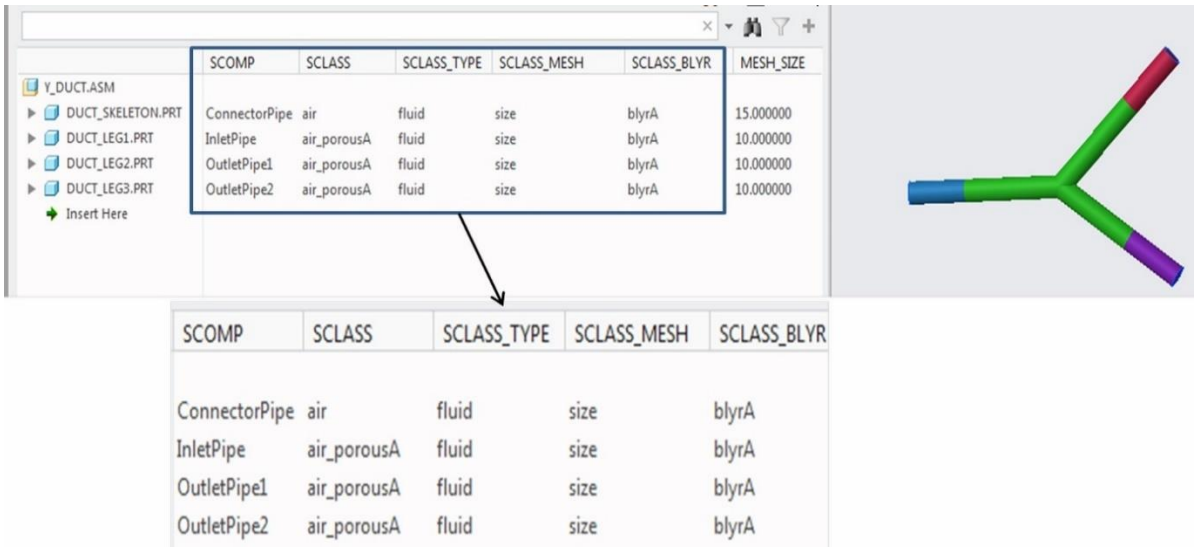

Figure 5: Example of CAE String Parameters in a CAD model

The SCOMP parameter values are unique for each part and represent component names. The related SCLASS* parameters in this example are SCLASS, SCLASS_TYPE, SCLASS_MESH, SCLASS_BLYR, etc. The values of the SCLASS_TYPE category could be fluid, solid, void etc. while the values of SCLASS indicate the specific material involved, e.g. air, air_porousA etc. The values of SCLASS_BLYR represent attributes related to a boundary layer and are labeled as blyrA. As mentioned before, these string values represent mainly physics aspects of the simulation, common across CFD applications.

3.2.3 Simulation Model – Connecting Abstract Models with CAD Models

To create the simulation input, an abstract model with its defined abstract classes, relations, and related attributes is combined with the desired CAD model. The result of this combination, in CAENexus terminology, is called the “Simulation Model” and is generated automatically by the software. The simulation model represents the state where abstract classes and their relations connect to real geometry, with attributes being transferred from abstract entities to real geometric entities.

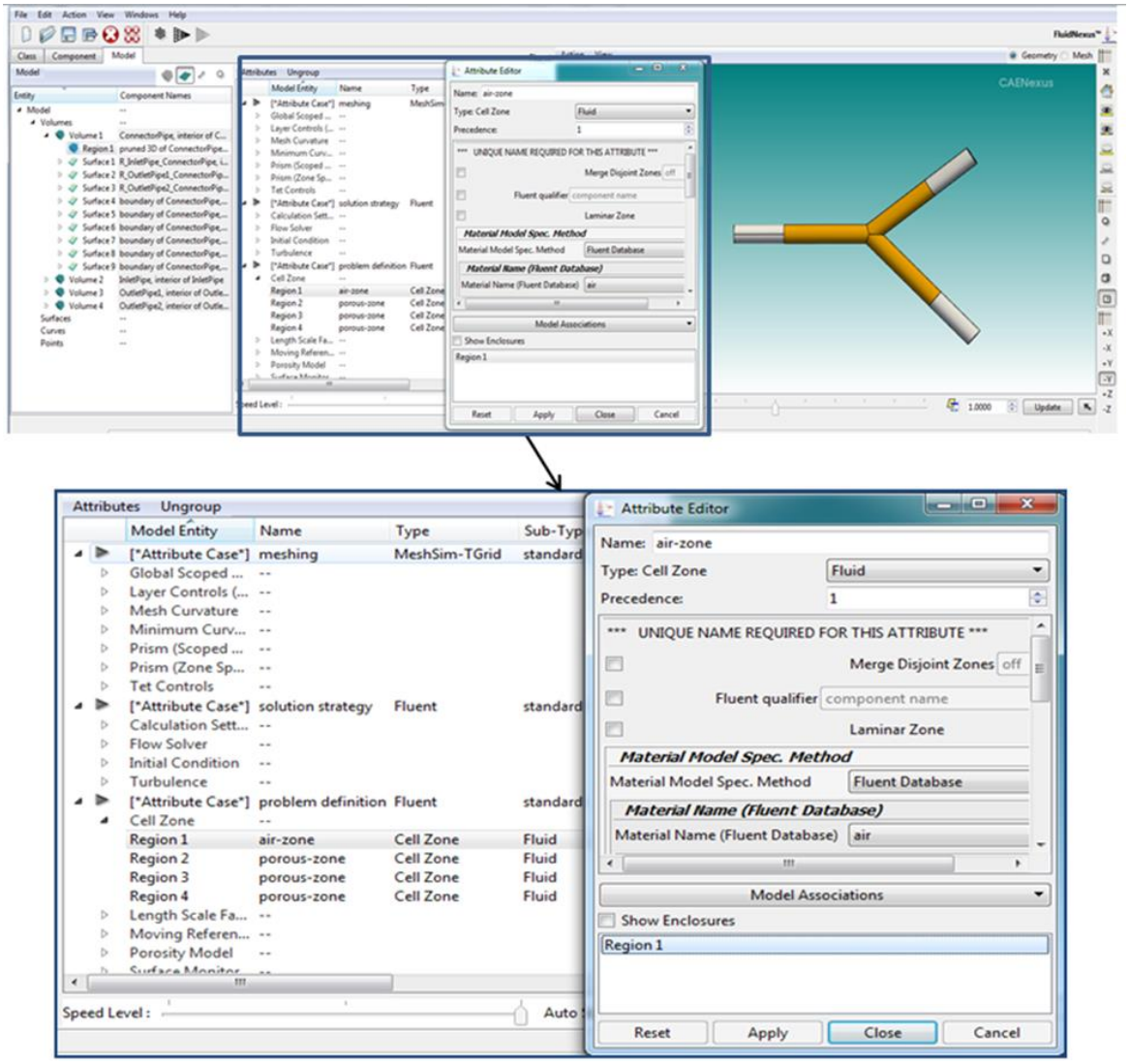

Figure 6: Simulation Model Geometry 1, Y-Duct

The simulation model shown in Figure 6 highlights geometry (shown in yellow) with the “air” class and the cell zone fluid attribute transferred from the “air” class to Region 1 (real geometry of “air-zone”). Similarly, each class, class relation, and component with their child items get connected with the geometry. Also, attributes defined on each abstract entity get transferred to real geometry.

If the same abstract model is combined with other CAD models having SCLASS* parameter values as air, size, blyrA etc., the respective simulation model will show the geometry associated with those classes.

The next simulation model shown in Figure 7 below was created by combining the same abstract model from before with a different CAD model. It highlights geometry with the “air” class and the continuum solid attribute transferred from the “air” class to real geometry.

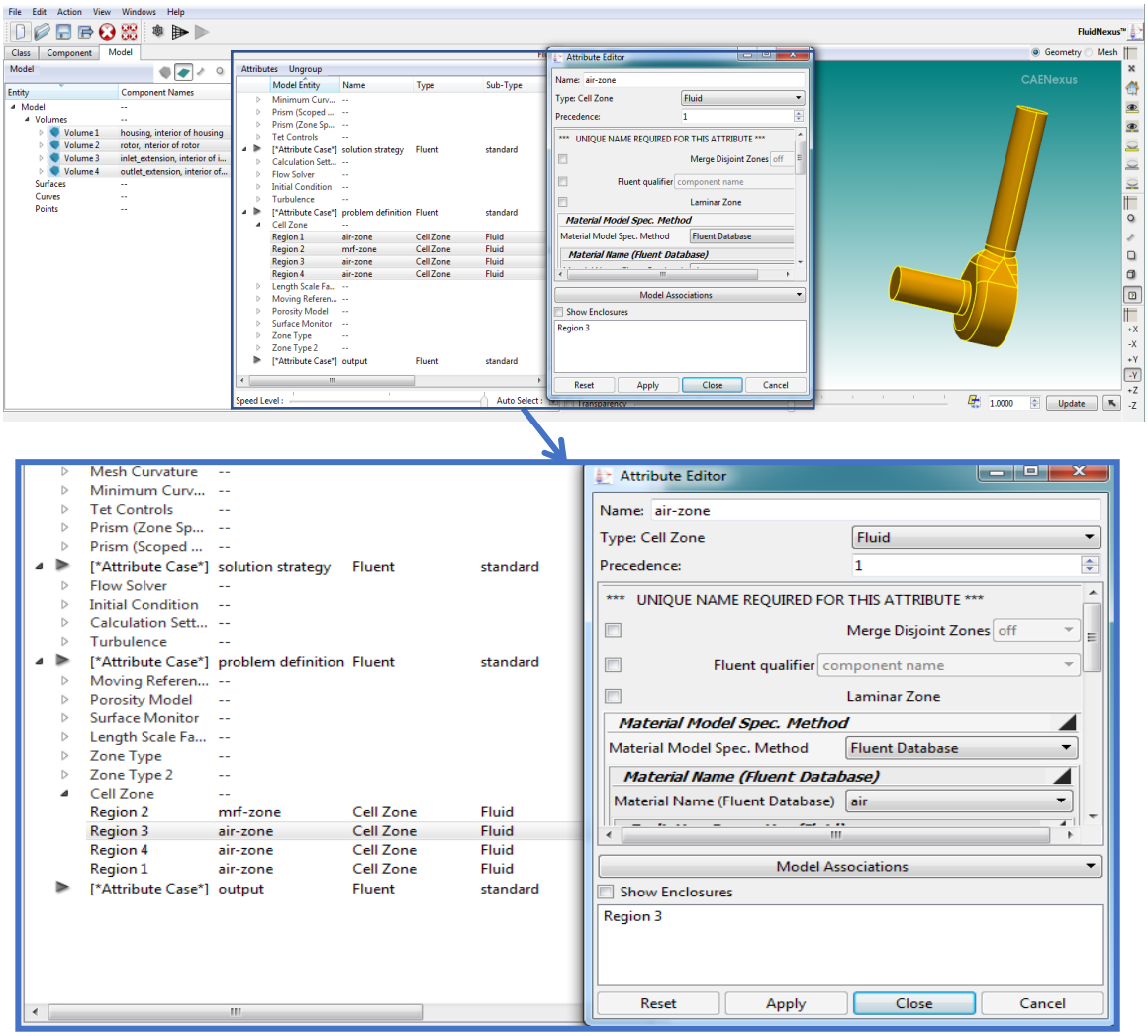

Figure 7: Simulation Model Geometry 2, Pump

From the two examples above, we can note the following:

a. The abstract model contains a class “air”, independent of a specific geometry.
b. When an abstract model gets combined with a CAD model having the “air” string parameter on one or several parts, a simulation model gets created where the class “air” is assigned to all respective parts. The attributes on an “air” class get transferred to real geometry as well.
c. The process explained above for one class and one attribute set is representative of CAENexus’ ability, based on abstract modelling, to automatically generate simulation models that map all the classes and attributes contained in an abstract model onto the real geometry of any CAE ready CAD model.

3.3 Accessing Simulation Parameters from CAD

As shown before, a single, well defined abstract model can work with a plurality of CAD models of varying shapes and complexities without having to be modified by the user. But how can such an abstract model handle changes of parameters that are usually specified as class attributes, for example, different angular velocities or mesh sizes? An obvious possibility is to change the attribute value in the abstract model itself, but there is also a more flexible way that does not require editing the abstract model. Instead, users can define simulation parameters via the string parameters on the CAE ready CAD model in order to modify the default parameters in the abstract model for specific cases. In the following CAD model, a user has specified simulation parameters as “MESH_SIZE” and “ANG_VEL_Z”.

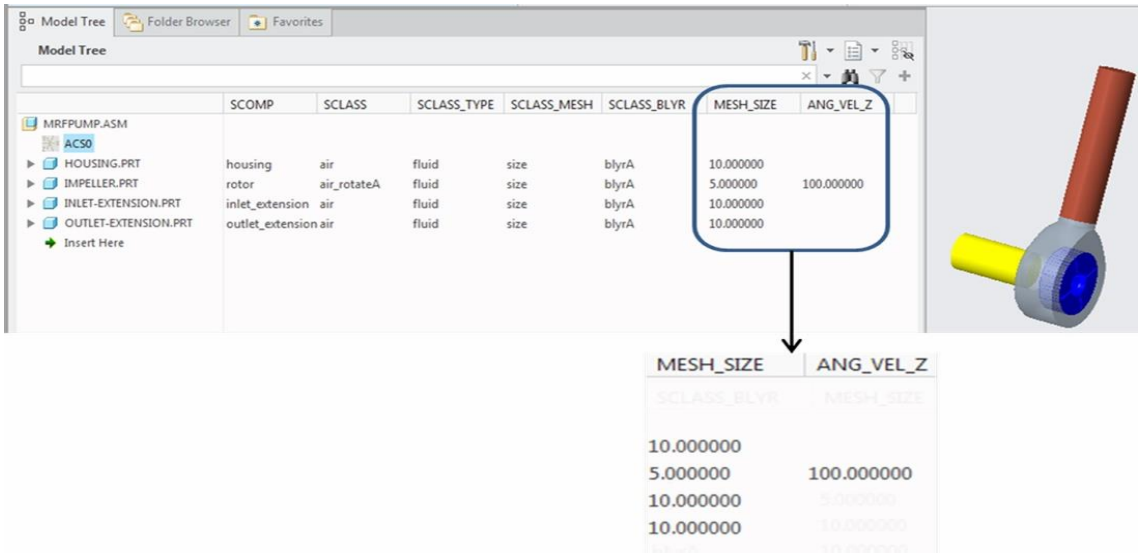

Figure 8: Attributes Defined on CAD

CAENexus will check the CAD model for parameters of specific attributes that, in the abstract mode, have been specified by a "doubleParamFromCAD" expression using names as specified on the CAD model. If specific attribute parameters are found on the CAD model, the default values in the abstract model are replaced by the values from CAD. If not, then the default values from the abstract model are used.

3.4 Abstract Modeling Automation

Abstract modeling automation performs the reliable creation of meshes and all necessary solver input files without involvement of an analyst. To facilitate this, there are two distinctive roles involved.

a. CAE Engineer: CAE engineers author the abstract model itself, creating all necessary abstract classes, relations, child entities, and applying CAE physics attributes.
b. CAD Engineer: CAD engineers will create CAE-ready CAD models and usually add the necessary SCLASS* and SCOMP string parameter values. Adding string parameter values can also be done by a CAE engineer with access to the CAD system.

Once an abstract model has been created and tested, a vast number of simulations can be run seamlessly by simply pairing it with CAE-ready CAD models. The analyst's pre-processing effort is reduced to the one-time creation of an abstract model which serves as the core of the CAENexus' automated three-step process: generate simulation model, generate mesh model, and export solver deck.

In case of CAENexus/FluidNexus, the robustness of this three-step process is enhanced through the direct use of the CAD system for mesh generation. Avoiding all CAD conversions eliminates geometry translation issues and clean-up efforts, also time consuming, non-productive tasks.

Based on FluidNexus users' feedback, abstract modeling methodology and automation significantly improves simulation process efficiency and reliability which enables them to perform more virtual tests with a given number of solver licenses.

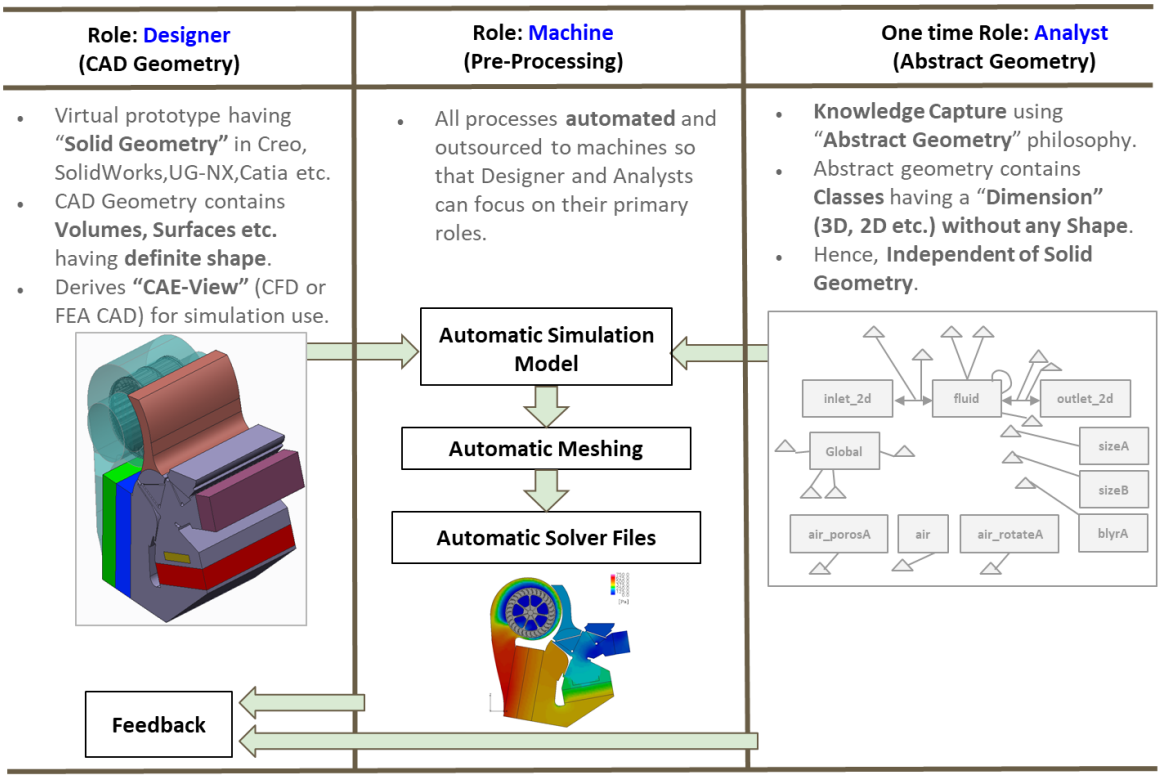

Figure 9: Automatic Three-Step Process

3.5 Simulation Driven Design

FluidNexus users have experienced excellent efficiency gains by starting with a CAD representation of the CFD geometry instead of the manufacturing version of their product. They derive the manufacturing CAD model only after the CFD driven optimization and validation are done.

4 Versatility of Abstract Models – Use Cases

In the previous chapter we have learned the basics of abstract modeling and seen how one abstract model was used to facilitate an automatic CFD process involving two geometrically different, but simple CAD models as examples. Let's now look at a scenario containing more complex devices.

4.1 HVAC Subassembly

The first use case is a performance analysis for a HVAC sub-assembly. Some things have not changed compared to the previous simple models, e.g. air is still the fluid involved. But the number of parts has grown, and the model contains, among others, two elements: an evaporator and a heater, both characterized as porous material. The abstract model for this case therefore contains related classes, air_porousA and air_porousB, with their respective physical parameters.

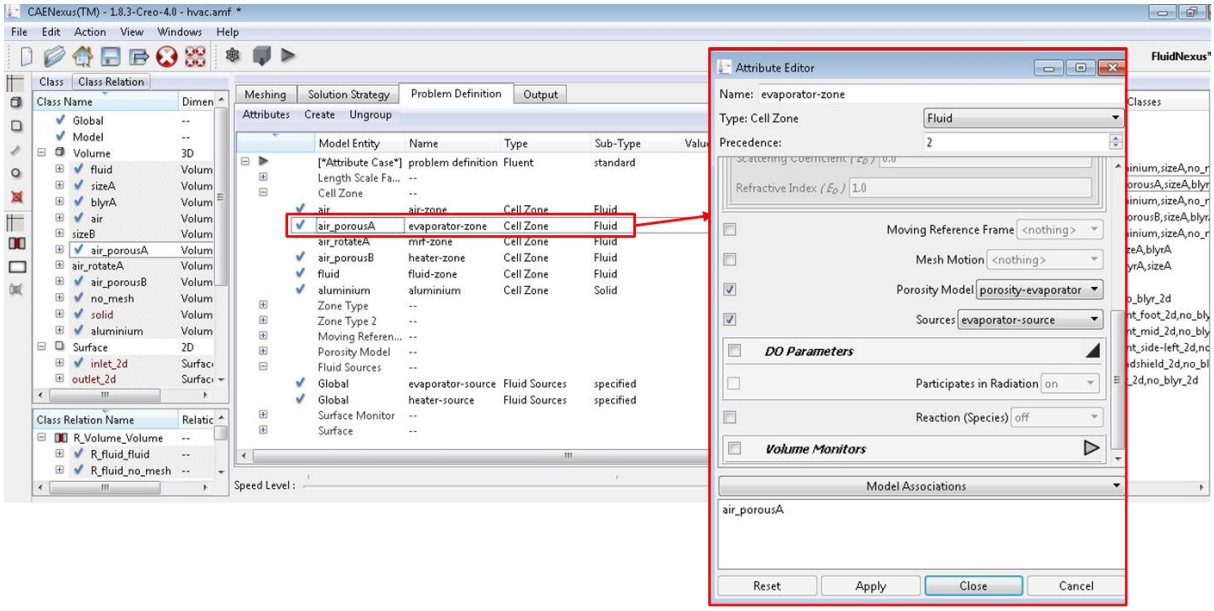

Figure 10: Abstract Model HVAC Simulation

While the abstract model shown in Figure 10 contains more classes than the one initially used for the simple models in Chapter 3, it can nevertheless by used with the Y-duct as long as the physical/material parameters for air_porousA are identical. FluidNexus will always consider only those classes it finds on the CAD model to prepare the simulation set-up. This means, that it is possible to prepare an abstract model for a complex system and then use that same model for simulations of the overall assembly and sub-assemblies – always applying the same simulation strategy and creating comparable results.

As explained for the simple models, our HVAC CAD model just needs to contain matching text strings in order to combine with the abstract model to then create the simulation model, mesh model, and the full solver input deck.

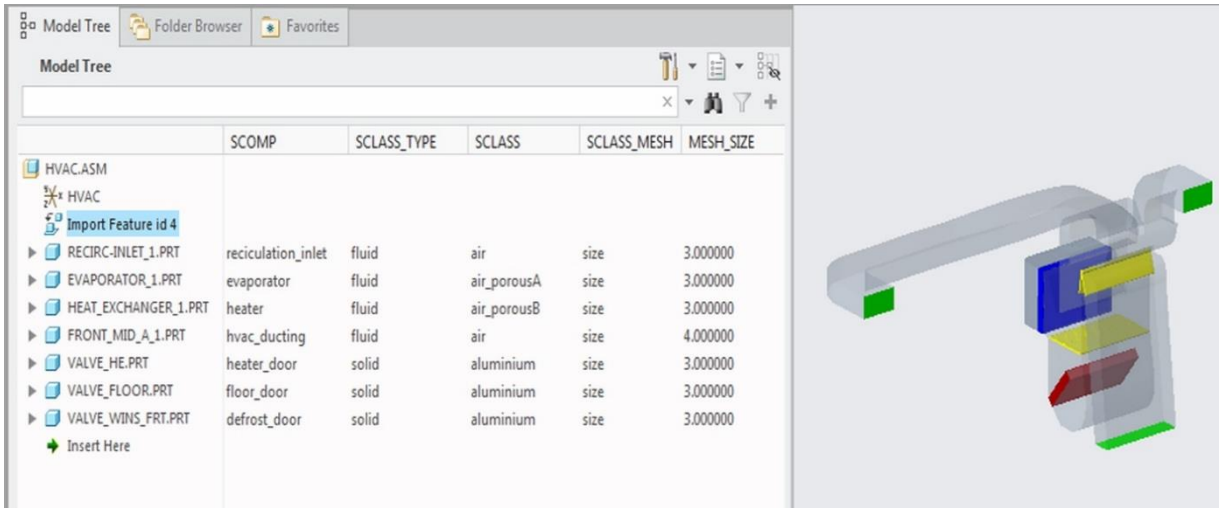

Figure 11: Tagged HVAC CAD (Symmetry-) Model

Then, CAD and abstract models are combined as before to create the simulation model, where all simulation attributes are transferred to the geometry, shown below in Figure 12.

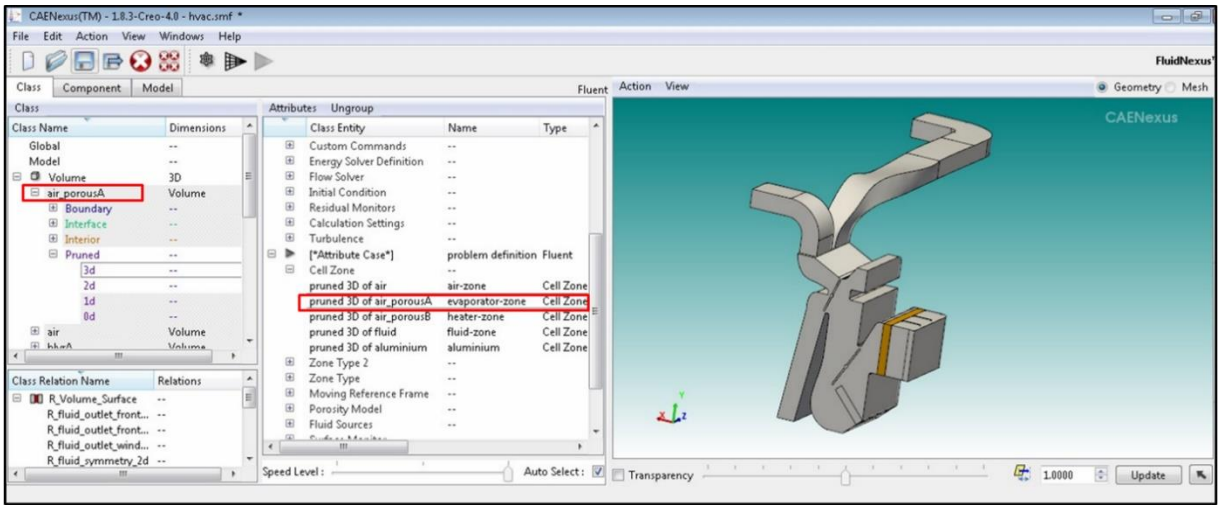

Figure 12: Simulation Model with Geometry of Class “air_porousA” Highlighted

Working with complex systems requires the possibility to efficiently check that all boundary conditions, material properties, etc. are applied to all faces and volumes where they are needed. FluidNexus offers flexible possibilities to review the set-up per component, class or model (single faces and volumes). Usually, an analyst does this with several different CAD models to ensure a robust automatic process afterwards.

Once an analyst has finished checking the simulation set-up, mesh and solver input deck creation are done simply and automatically, with two clicks of a button. An example of the automated surface mesh generation follows in Figure 13.

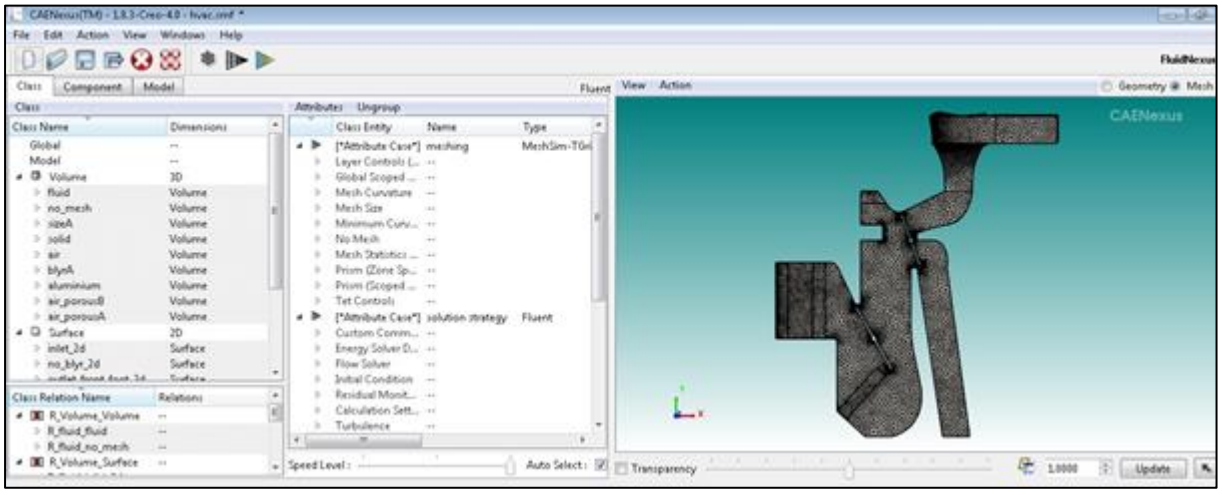

Figure 13: HVAC Mesh Model

Once the abstract model has been tested and validated, it can be used to investigate multiple HVAC performance outcomes under different operating conditions, or even different geometry designs. By simply editing the CAD model in geometry and/or string parameters as desired, these new investigations are also run with the automated simulation processes (simulation model, mesh generation, and solver input deck generation).

Further, initiating the automatic CAD to solver process for multiple simulation interrogations of different iterations can be performed at once through a single command line. Details regarding name(s) and locations of the abstract and CAD models are defined in a text file, and then CAENexus executes the same automatic process for each one. An example of the results of this process is shown below, showing different geometry orientations and physics problems solved for the same HVAC system.

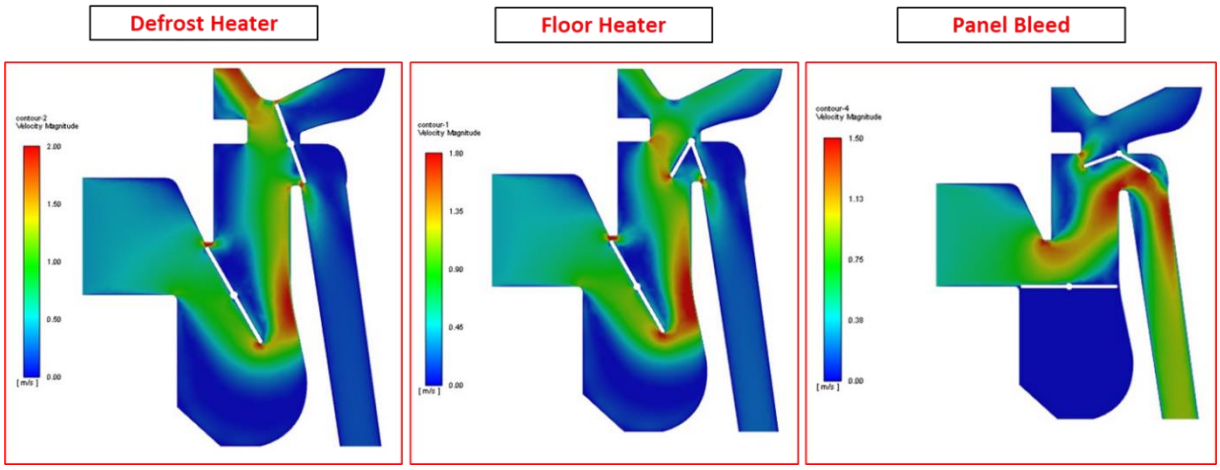

Figure 14: Velocity Contours for Multiple Operating Conditions

4.2 Fuel Cell

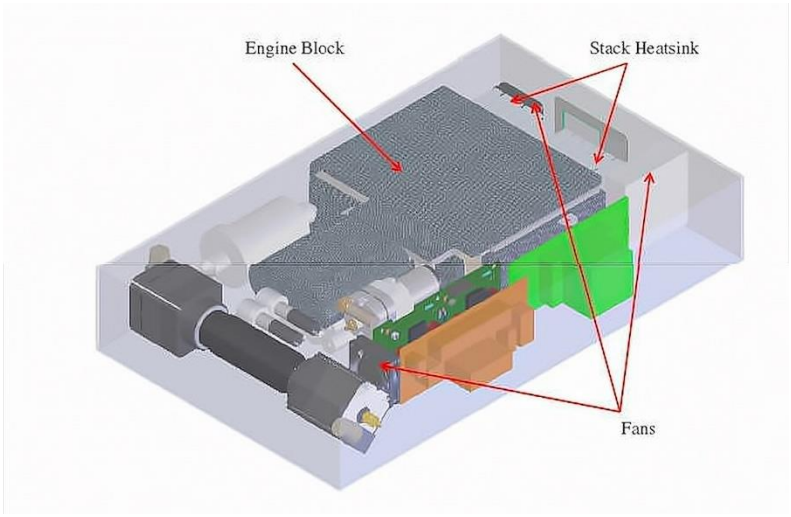

Figure 15: Fuel Cell Assembly

The second use case is a cooling simulation for a fuel cell. Compared to the HVAC sub-assembly, the number of parts has grown even more, porous media is not needed. Instead, this system uses fans at multiple locations and with potentially different performance characteristics.

An abstract model for this case requires additional classes for heatsinks and their properties, fans and their properties, etc. These classes can simply be added to our previous abstract model, making it more versatile for a larger number of geometries. It would also be possible to author a new abstract model, one solely dedicated for fuel cell cooling type problems, if desired.

Figure 16 below shows some of the additional classes and class relations, including a multitude of "air_rotate" classes representing different fan options. Defining these fan options in the abstract model like this allows us to simulate the efficiency of various fans through a simple plug-and-play method. To easily test these different fan options, we need only to update the string parameter on the CAD model, and the simulation model will be automatically generated using the desired fan option for the simulation set-up.

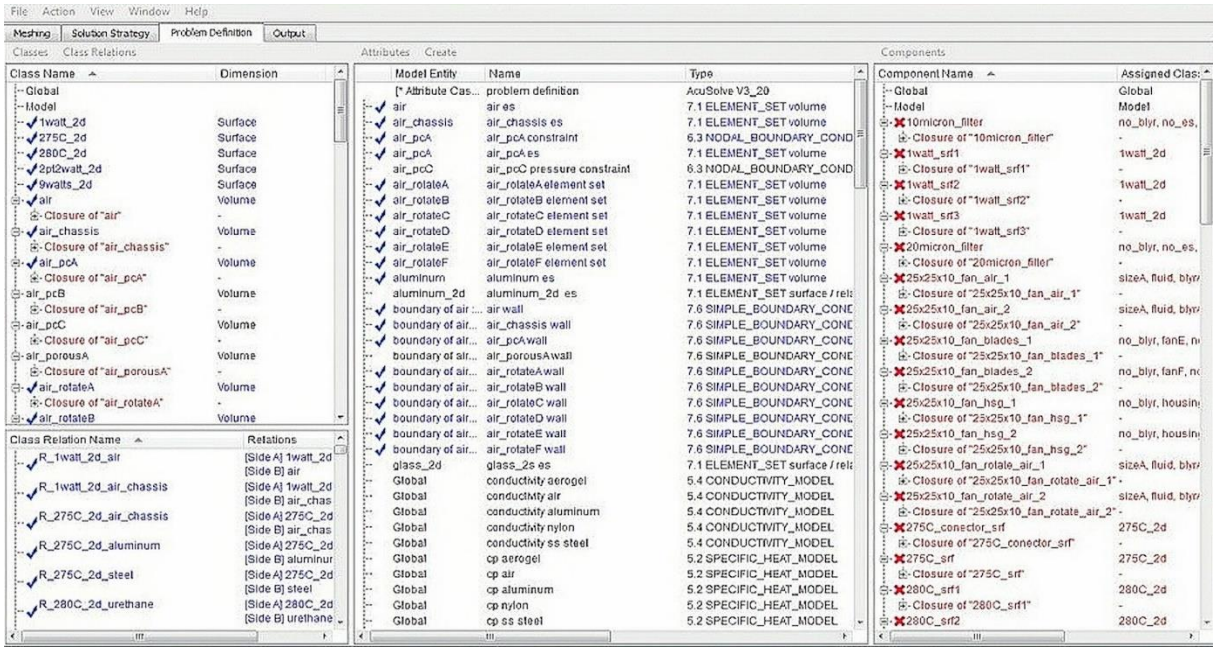

Figure 16: Abstract Model Fuel Cell Cooling

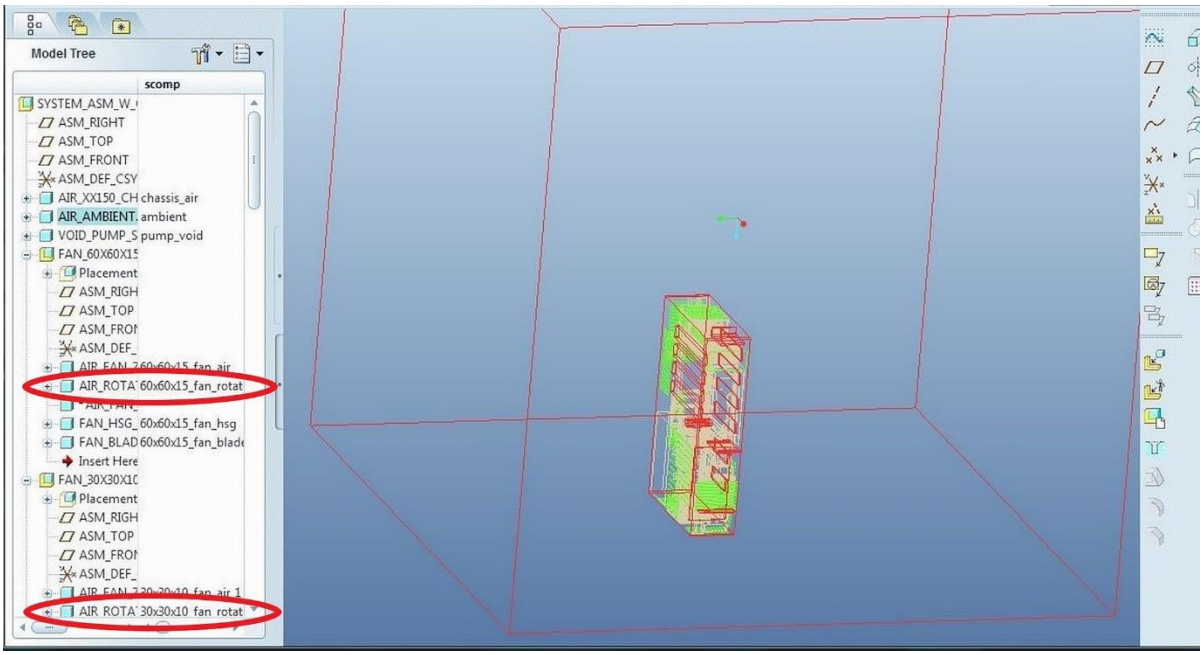

Figure 17: Simulation Ready CAD Model of Fuel Cell

As before, ensuring that the abstract model accurately maps onto different versions of complex systems, like for the fuel cell in this case, is called “burning-in” the abstract model. CAENexus benefits from its flexible capabilities to check that all parameters are correctly defined to facilitate the burn-in process. Volumes or faces can be selected via classes, class-relations, components, model entities or by clicking on a part in the geometry window.

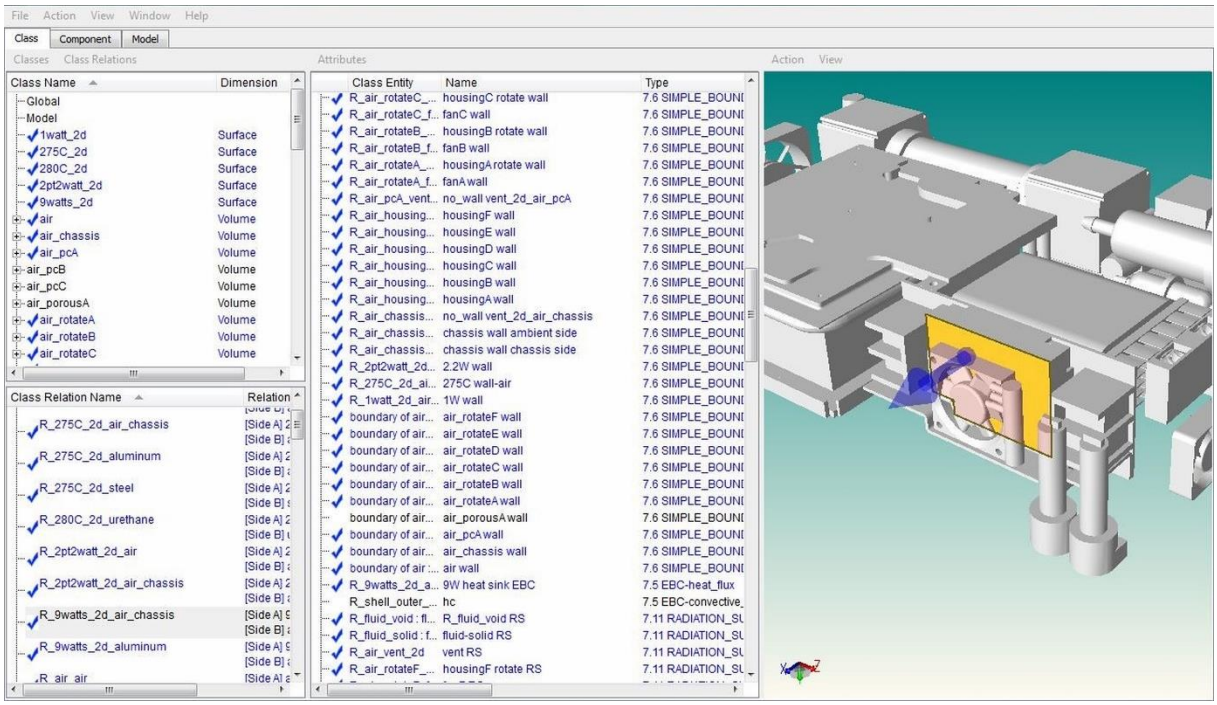

Figure 18: Simulation Model "Class View" – Face Selected by Class Relation

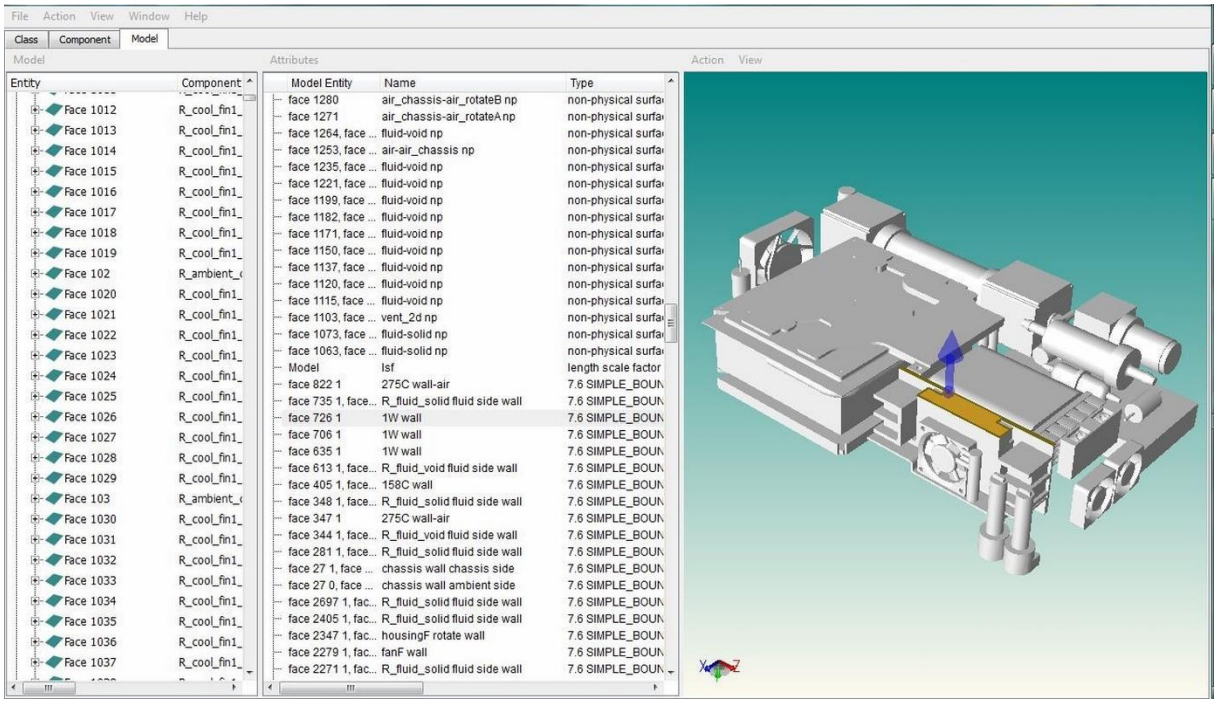

Figure 19: Simulation Model "Model View" – Individual Face Selected in Model Tree

Once the abstract model burn-in is done, mesh and solver input deck creation are done as explained in the HVAC section.

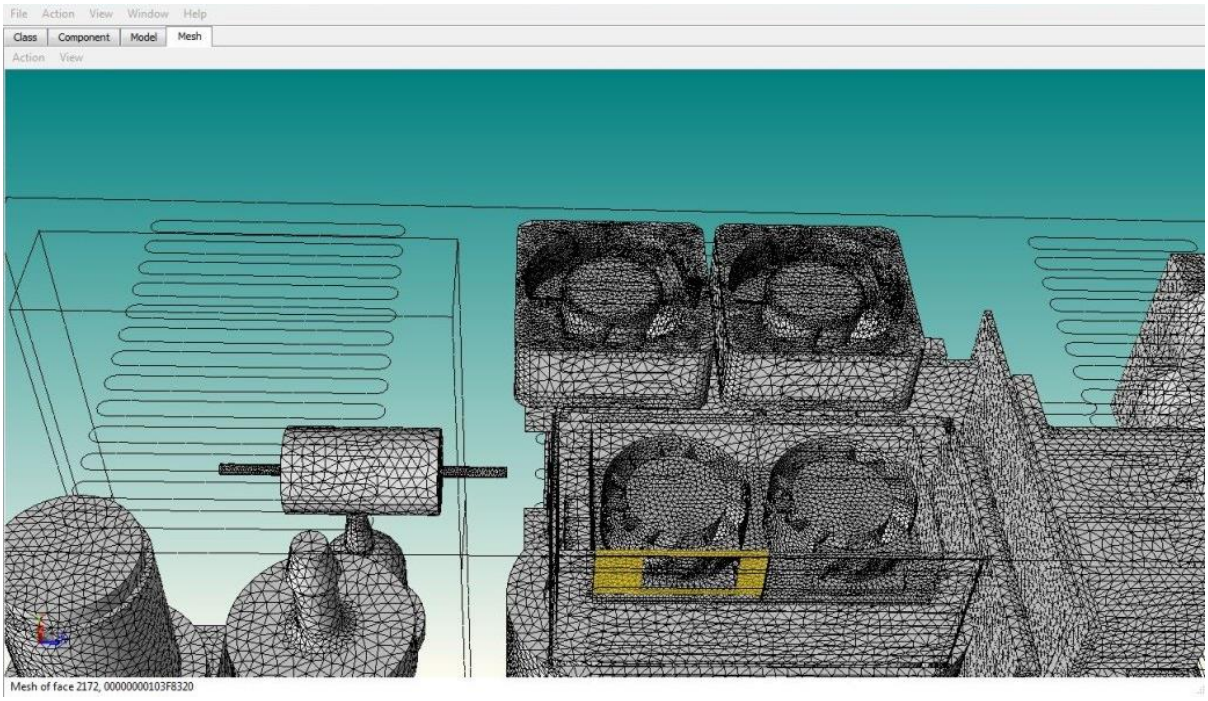

Figure 19: Mesh Example

The abstract model used for this fuel cell cooling case contains classes like “air_porousA”, “air_rotate” and could therefore still be used for our simple geometries, in principle also for the HVAC case. This means it is possible to have one abstract model for varying simulations. It is advisable though, to not combine use cases that are exceedingly different and would require too many classes not shared between the use cases. In those situations, it is better to work with separate abstract models, as they are faster to create and burn-in.

5 Abstract Modelling for Simulation Applications

So far, we have seen how abstract modelling facilitates a far better use of simulation resources and democratizes simulation. What makes all of this possible is the re-usability of abstract models due to their capability to work with any geometry. That same capability makes abstract modeling an ideal foundation for the front-end implementation of simulation applications. Simulation applications are growing in popularity as they enable non-analysts to perform simulations for a specific, precisely defined problem.

CFD related simulation applications could address a variety of problem types, such as external aerodynamics (e.g. virtual wind tunnel), fluid mixing, pump design, electronics cooling, etc. The standard utilization of abstract models is already close to the front-end functionality of a simulation app, in that it also enables non-simulation specialists to run and benefit from simulations. Typically, a simulation app has:

- an application specific user interface (e.g. browser based) including the possibility to upload CAD models,
- a process to automatically prepare all solver input files based on user defined parameters (e.g. desired wind speed), geometry, and simulation set-up,
- the solver(s) to perform simulations, and
- a facility to report results.

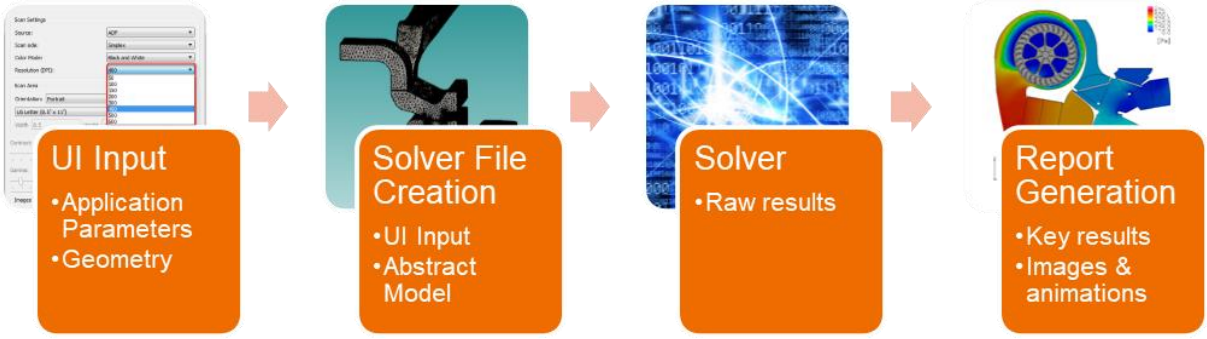

Figure 20: Simulation App Components

An abstract modeling-based process that automatically prepares all solver input files inside a simulation app would take advantage of abstract modeling's standard functionality of reliably handling diverse geometries. However, managing varying forms of fine-tuned user input in the same simulation app could be accomplished via minor extensions, such as using simple Python scripts. Like standard pre-processing automation, reliable simulation apps will be easier and faster to implement through abstract modeling technology.

6 Conclusions

This article demonstrates that the automation of CAD-to-solver processes can be done in a straight-forward, easy way, using an approach similar to traditional simulation set-ups with which analysts are already familiar. The key difference is generating the set-up independently from any specific geometry through the use of placeholders in the form of abstract classes and therefore, making the set-up reusable.

While abstract art is liked by some and not cared for by others, abstract modelling technology offers a unique combination of benefits that should be of interest to all product development organizations. Those benefits include:

- ⇒ Democratization of simulation allowing better use of simulation resources by empowering CAD designersto start dependable simulations and support design decisions as models evolve
- ⇒ Automation of simulation pre-processing becomes easier, faster and more robust than with other methods
- ⇒ Systematic capturing and re-use of simulation know-how and best practices
- ⇒ Consistent, comparable results, independent of where or by whom simulations are performed
- ⇒ Vastly improved efficiency of CAD to CAE solver input process through robust automation

At a time of growing worldwide competition and increasing product complexity, abstract modeling saves significant time and cost while helping users to create optimized products. Development organizations using this technology gain the capacity for more meaningful reports to improve the quality of design decisions, are able to overcome crucial losses of knowledge or expertise when experienced simulation specialists leave for another employer or retire, and ensure a much better return from their investment in engineering software.

7 References

[1] https://www.tate.org.uk/art/art-terms/a/abstract-art
[2] TechClarity Infographic: "Making Products More Competitive by Empowering Design Engineers", 2019
[3] Cyon Research White Paper: "Classes of MCAE Software: Clarifying the Market" 2008, 23 pages
[4] Novus Nexus White Paper: "CAENexus™ Abstract Modeling Streamlines CAE Processes Improving Productivity and Engineering Collaboration" 2019, 15 pages