# CAENexus™ Abstract Modeling Streamlines CAE Processes Improving Productivity and Engineering Collaboration

Enabling designers to initiate dependable simulations while relieving analysts from tedious routine work

CAE Connected™

Novus Nexus Inc.
Simulate Better With US

**Novus Nexus Inc.**

143 Cadycentre #351
Northville, MI 48167
USA

www.novusnexus.com

# *CAENexus Abstract Modeling Streamlines CAE Processes Improving Productivity and Engineering Collaboration*

*Facing an ever-growing worldwide competition, manufacturers in all industries are challenged to develop new and improved products as quickly and cheaply as possible. At the same time, their products and systems involved are growing in complexity. To address this challenge, increasing the use of virtual tests/simulations earlier in the design cycle is a key factor to remain competitive and to develop products in a timely manner that perform as desired. Unfailingly, automated processes that work reliably, efficiently, and that are quick to implement will be important to get the full benefit from virtual tests. Here, Novus Nexus' CAENexus comes into play by offering a new pre-processing approach based on abstract modelling technology. CAENexus reliably connects design and simulation worlds, reduces non-productive analyst work, speeds up simulation processes, ensures the use of best practices, and preserves valuable corporate knowledge and expertise.*

*Challenge to build and develop new products faster and cheaper, yet better*

# Introduction

A growing international competition is leading to shorter product life cycles with ever more functional and smarter products aimed to address, for example, IoT or digital twin requirements. Development processes are getting more complex while development organizations are asked to reduce development time and cost.

Another fact to consider is that when analysts retire they are often difficult to replace one-to-one, either due to required cost savings or because appropriate resources are hard to find.

*Shorter product life cycles generating more functional and smarter products require the use of more and earlier simulations*

One way to address these development challenges is the use of more and earlier simulations to better understand product performance for well-informed design decisions. At this time, though, and for miscellaneous reasons, many companies do not take full advantage of the possibilities of virtual testing. The motives for this situation usually include one or several of the following topics:

- Uncertainty about the Return on Investment (ROI)
- Lack of suitable resources
- Doubts about the trustworthiness of simulation results (e.g. are they always comparable, especially if done by different people)
- Fear of simulation results not being available in time for design decision making
- High effort to reliably transfer models between different tools (e.g. CAD model transfer to CAE solvers)
- Company culture favoring fear of change and risk avoidance

*But many companies still do not take full advantage of simulations*

While the CAENexus products can't take away fear of change or a risk avoidance attitude, they have proven to deliver excellent ROIs, significantly increase the efficiency of existing teams as well as ensure dependable, comparable, and timely results - independent of the users involved - for better decision making. Due to the concept and methodology behind CAENexus, CAD models are ready for simulation without conversion to another format as needed by other pre-processors. This enables a smooth CAD to CAE process using intelligently designed CAD models as the sole geometry source, which avoids CAD translation issues and makes the desired, robust automation process possible.

# CAENexus Business Value

While CAENexus implements a new concept based on abstract modeling, it has already proven its value at existing customers in the development of construction and forestry equipment, home and kitchen appliances, and automotive air conditioning systems, to name a few. Automating CAE processes with abstract models is easier, faster, and more reliable than other approaches, such as using templates or scripting, because abstract models are geometry independent and the model creation is similar to simulation set-ups in conventional preprocessors.

The main business benefits that have been achieved from deploying CAENexus in product development processes are as follows:

- Significantly enhances the productivity and efficiency in product development by automating the CAD to CAE workflow.
    - Frees up simulation specialists from non-productive tasks, such as CAD model import and clean-up, and simulation setup (estimated to take up 70% of analysts' time when working conventionally) thereby allowing them to focus on more meaningful result reporting and sharing for better and faster design decisions.
    - Enables designers to initiate simulations whenever they create new geometry versions without having to wait for an analyst to be available for model set-up.
    - Aids **easier, faster, and more reliable automation** thereby making more simulation tasks suitable for highly efficient automated processes.
- Ensures dependable, comparable results and consistent use of best CAE practices.
    - Allows for design decisions to be made on reliable simulation results independent of who was involved or where the simulation was performed.
    - Results in better performing, more competitive products that cause less recall and warranty issues.
- Preserves valuable corporate know-how related to CAE practices in abstract models.
    - Helps to avoid knowledge and expertise losses due to individual analysts retiring or joining other companies.
    - Supports continuous application and improvement of CAE best practices.

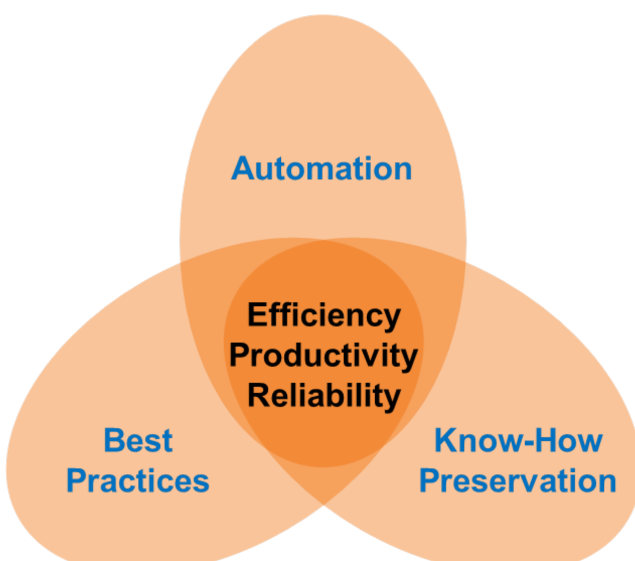


*Figure 1: CAENexus Benefits*

***CAENexus Value***

*Enhances productivity and efficiency in product development*

*Ensures dependable, comparable results and consistent use of best CAE practices*

*Preserves valuable corporate know-how related to CAE practices*

***Easier, faster, and more reliable automation than other methods***

# Abstract Modeling – the Secret Recipe

## What is Abstract Modeling

CAE analysts are often performing the same type of CFD simulations with varying geometry models. When setting up a CFD analysis there are common things between geometry models and there are differences between them as shown in the two simple models in Figure 2 below.

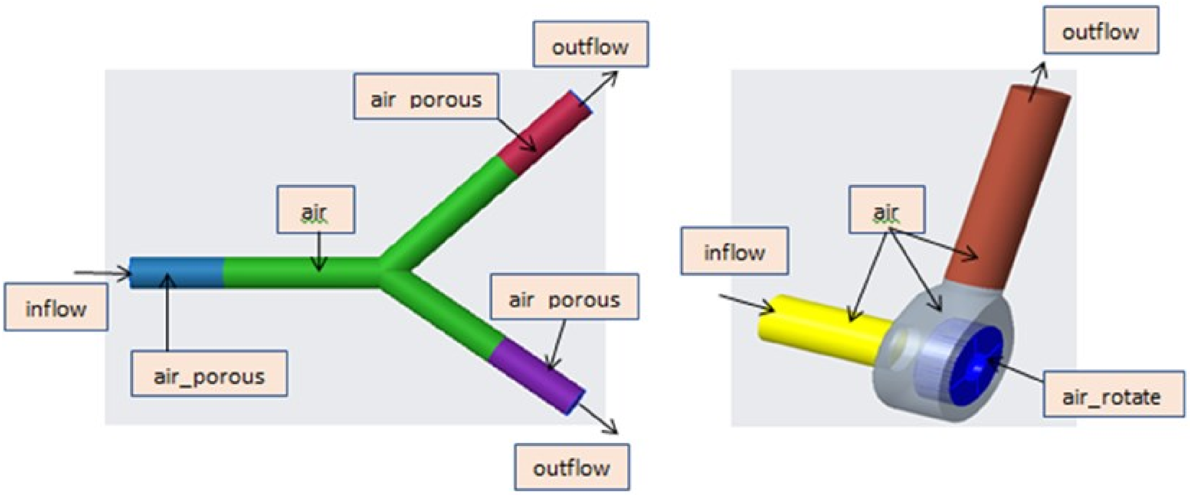


*Figure 2: Geometry Examples*

In these two models, each part has air or some other material that requires element set attributes. A user also needs to apply inflow, outflow, and wall in both models to perform a simulation.

The geometry of each model is different and each model has different parts. For example, the model on the left has parts like inlet pipe, outlet pipe, connector pipe, etc. The model on the right has parts like rotor, housing, inlet extension, housing, etc.

Abstract modeling makes use of similarities (same physics) through the re-use of simulation set-up parameters (attributes/behaviors) and automatically applies them to different CAD models (individuated parts and their shapes).

In summary, the secret recipe of abstract modeling is to use abstract entities, called classes, which function as placeholders representing physics intents defined with attributes. These classes with their attributes are then automatically connected to the appropriate geometries. To enable this, CAD models and their parts have parameter strings matching the names of classes used in the abstract model. Thereby, abstract modeling technology demonstrates several advantageous features. Abstract models are:

a. geometry independent because CAE attributes like initial conditions, material parameters, etc. are specified on abstract classes and NOT on specific geometry or mesh,
b. reusable because abstract classes are common across all models,
c. a knowledge capturing and management tool because the CAE analyst's knowledge is stored in the abstract model file as re-usable, best-practice simulation settings with all related parameters.

*Abstract Modeling makes use of the similarities between all virtual tests, such as the same underlying physics. Simulation parameters like attributes or behaviors are repeatedly and automatically applied to different CAD models.*

*Geometry independent*

*Highly reusable*

*Knowledge capturing and management tool*

## How Abstract Modeling Works

### Abstract Model

In CAENexus, users can create abstract classes and relations between classes as shown in Figure 3. 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 the contact on the common face between two solid objects then the relation between their classes allows to easily specify the contact attribute.

*Classes and class relations are placeholders for real geometry*

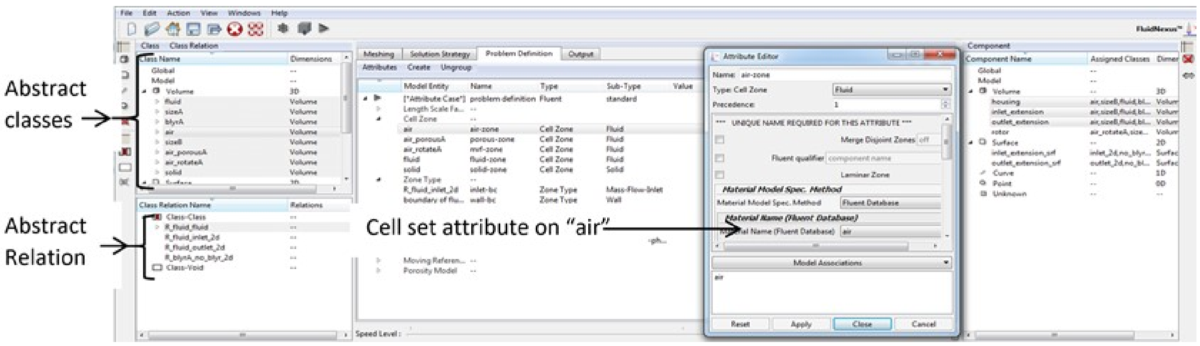


*Figure 3: CAENexus Abstract Model User Interface*

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 and will be explained in detail further on. 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.

### 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 values of SCLASS* string parameters must be identical to the abstract class names defined in an abstract model.

*String parameters on CAD models enable combination of abstract models with geometry*

The model in Figure 4 is tagged in PTC Creo® with SCLASS* and SCOMP parameters.

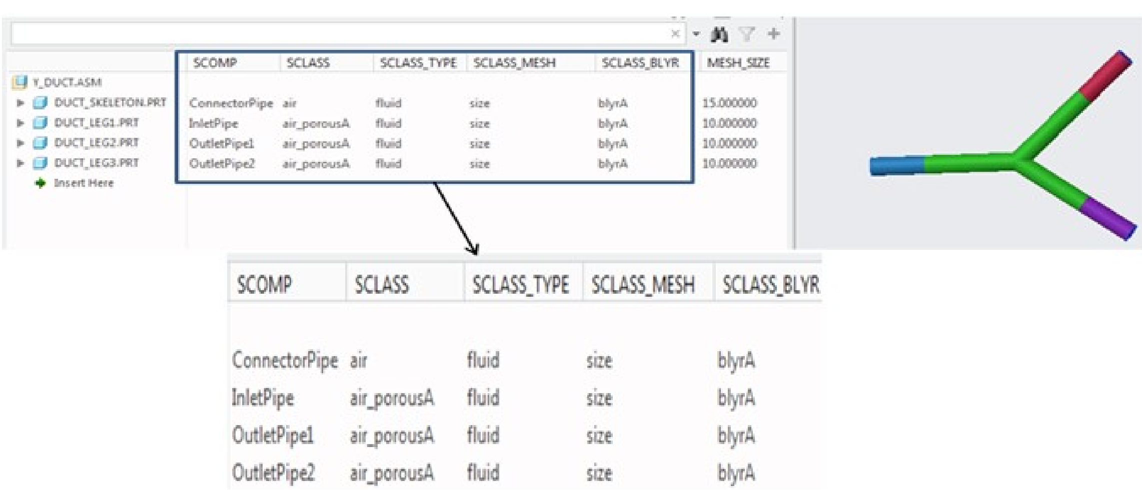


*Figure 4: Example of CAE String Parameters in a CAD model*

The SCOMP parameter values are unique for each part and represent component names. In our example, these are connectorPipe, InletPipe, OutletPipe1 as shown in the above figure. The related SCLASS* parameters are SCLASS, SCLASS_MESH, SCLASS_TYPE, SCLASS_BLYR etc. The values of SCLASS_TYPE are 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 are blyrA etc. As mentioned before, these string values represent physics aspects of the simulation, common across CFD applications.

## Connecting Abstract Models with CAD Models

When an abstract model (with 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”. The simulation model represents the state where abstract classes and components relate to real geometry and attributes are transferred from abstract entities to real geometric entities.

*Combining an abstract model with a CAD model creates a “Simulation Model”*

*In a Simulation Model, abstract classes and components relate to real geometry and attributes are transferred from abstract entities to real geometric entities*

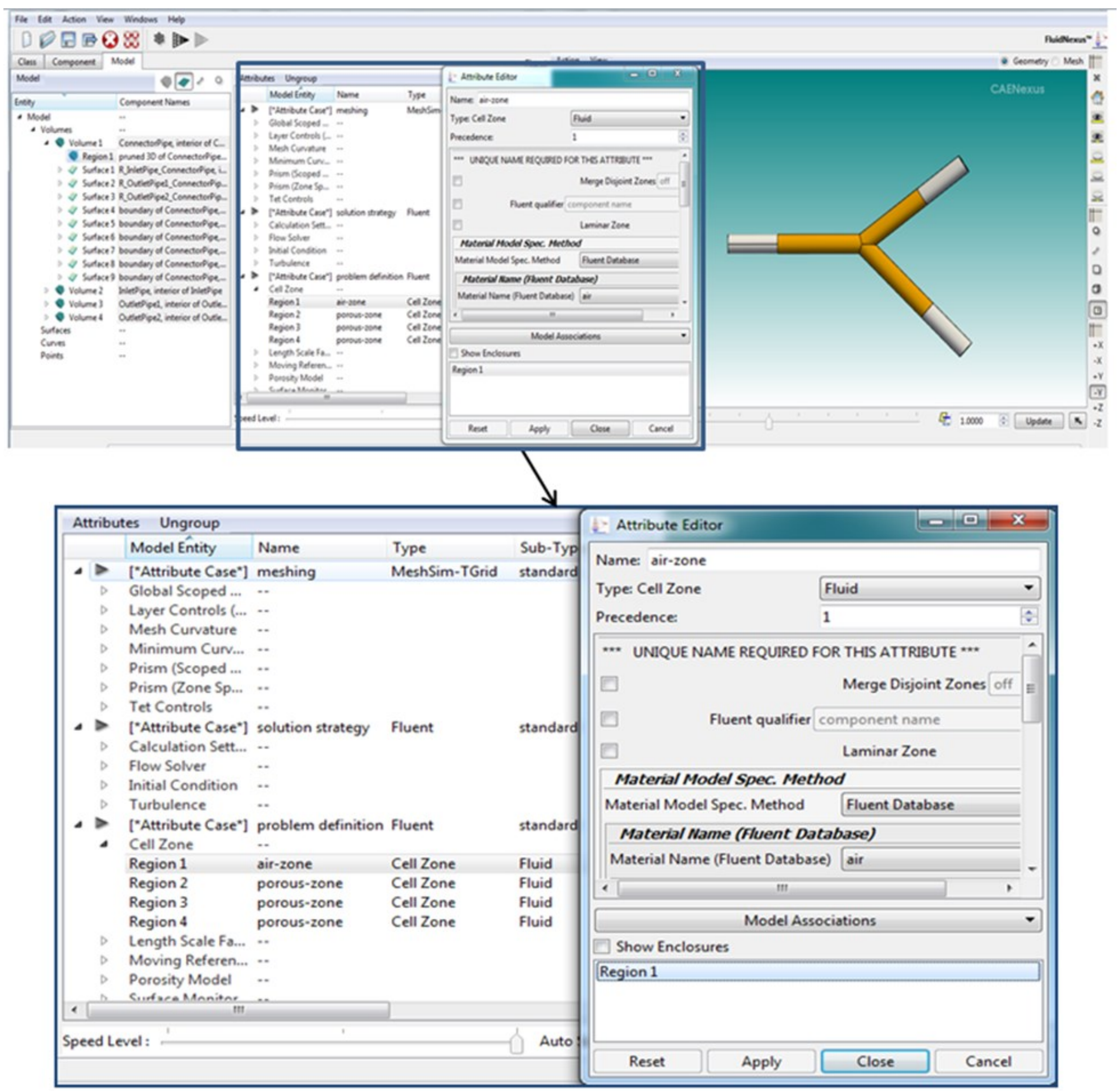


*Figure 5: Simulation Model Geometry 1*

The simulation model shown in Figure 5 highlights geometry with the “air” class and the cell zone fluid attribute transferred from the “air” class to region 1 (real geometry). 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 different 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 6 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.

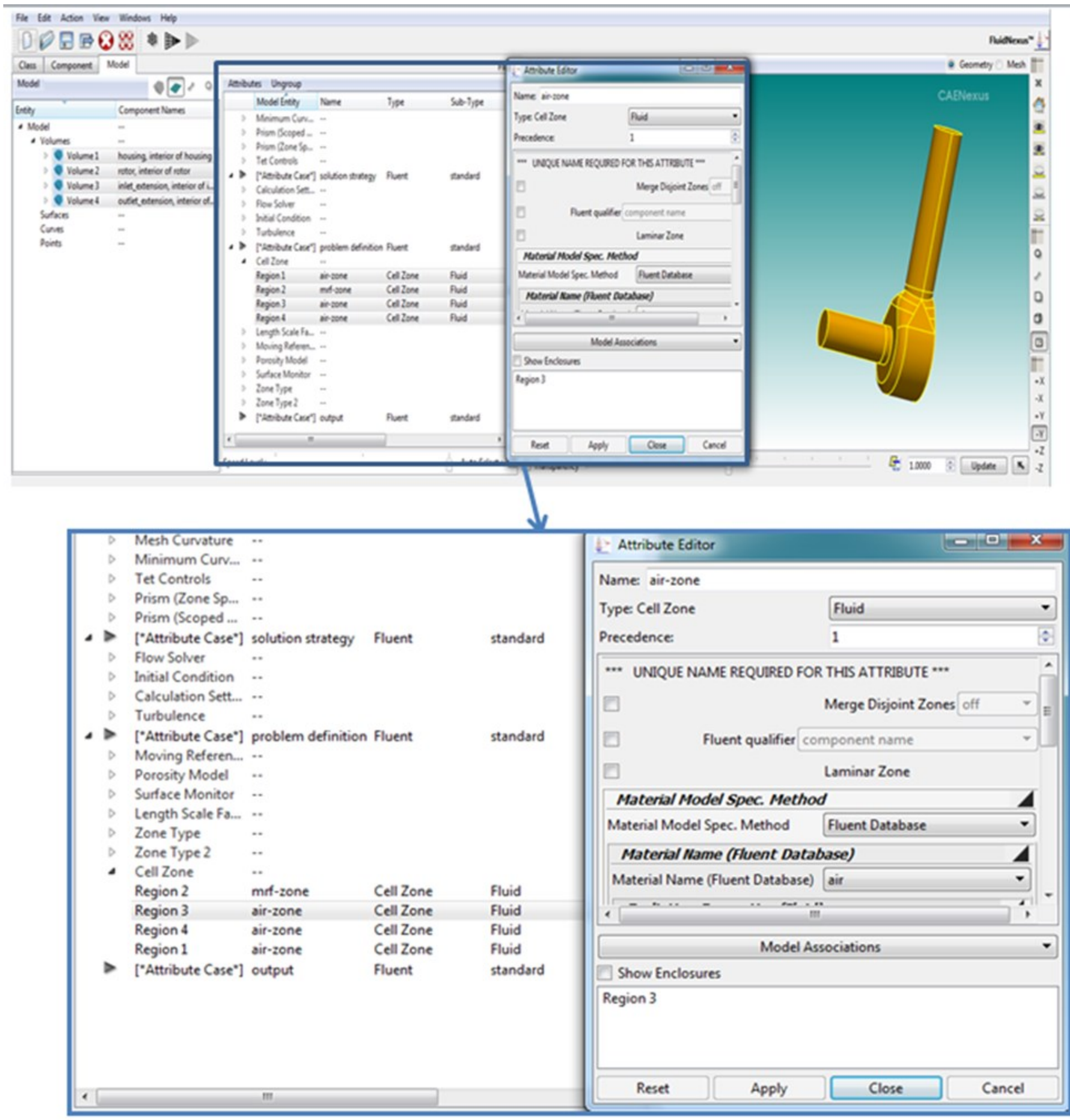


*Figure 6: Simulation Model Geometry 2*

*One abstract model can be applied automatically to many geometry variations*

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

- **a.** The abstract model contains a class "air", independent of a specific geometry.
- **b.** The abstract model further has a "cell zone-fluid" attribute which holds material and other parameters as specified by the analyst for the class "air".
- **c.** 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 geometries. The attributes on an "air" class get transferred to real geometry as well.
- **d.** The explanation above for one class and one attribute is representative of CAENexus' ability 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.

## Accessing Simulation Parameters from CAD

As shown above, an abstract model can work with 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, e.g. different angular velocities or mesh sizes? An obvious possibility is to change the attribute value in the abstract model, but there is also a more flexible way that does not require editing the abstract model. Users can define simulation parameters on the CAE ready CAD model and 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”.

*Parameters specified on CAD entities can overwrite equivalent parameters from the abstract model when desired or needed*

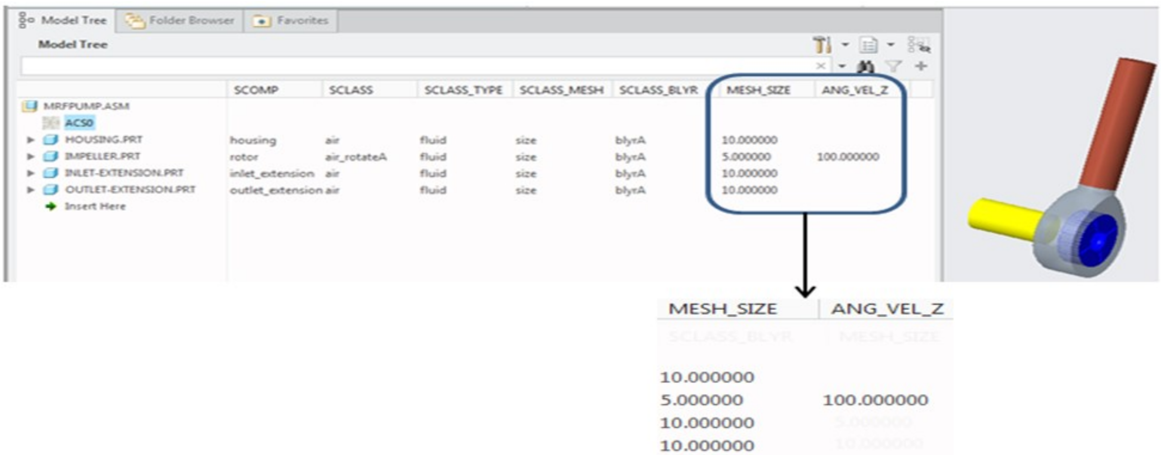


*Figure 7: Attributes Defined on CAD*

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.

## Abstract Classes and Relations

The following examples of abstract classes and relations give a brief window into the powerful configuration possibilities of CAENexus’ Abstract Modelling Technology.

### Dimensions of Abstract Classes

Though an abstract class does not have geometry associated with it, every class has a dimension. CAENexus supports different classes which relate to corresponding geometry entities: **Volume (3D)**, **Surface (2D)**, **Curve (1D)** and **Point (0D)** classes.

*Abstract classes can represent 3D, 2D, 1D, and 0D entities*

### Abstract Class Relations

A relation between abstract classes is the common entity between them, e.g. the relation between two volume classes is the common (inter-)face between them. The relation and resulting interface represent a common entity, but relations allow users to be more precise, enabling them to specify the sides of a face explicitly. When using CAENexus, a relation is useful to attach contact attributes to a specific side of a face.

*Relations specify common entities between classes*

Listed below are the main types of relations used in CAENexus. All relations define the common geometric entity between components of the classes involved. Relations usually have child entities like sides towards each class.

a. Relations between two different volume classes (R_fluid_solid)
b. Relation between same volume class (R_fluid_fluid)
c. Relation between volume and surface class (R_load_solid)
d. Interface relation between volume and surface class (RI_fluid_inlet_2d)
e. Relation between surface class and void (R_shell_2d_Void)

### Child Entities of Abstract Classes and Relations

Child items help to access geometric entities in very effective ways, enabling to specify CAE attributes on these child entities. The following are examples of child items of an abstract class.

1. **Interface** : It is the result of a relation and defines the common entities of a class with other classes.

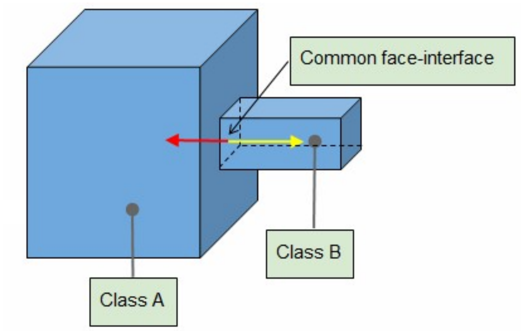


*Figure 8: Class Child Entity Interface*

As shown in Figure 8, the interface of volume class A is its common face with class B. This interface has two child items.

   a. **Inside**: It represents the interface side pointing towards class A for which an interface is calculated as indicated by the red arrow.

   b. **Outside**: It represents the interface side pointing away from the class A as indicated by the yellow arrow.

2. **Boundary**: It defines all the entities which form a class minus its interfaces with other classes.

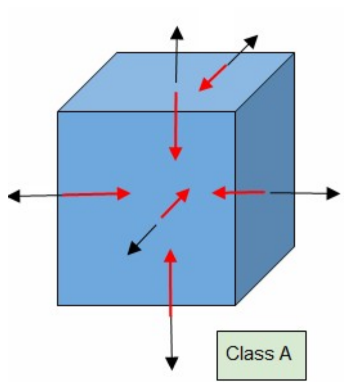


*Figure 9: Class Child Entity Boundary*

   a. **Inside:** It represents the side of the boundary surfaces pointing towards class A as indicated by the red arrows.

   b. **Outside:** It represents the side of boundary surfaces pointing away from the class as indicated by the black arrows.

3. **Pruned entities:** An entity can be pruned to a specific dimension, 3D, 2D, 1D and 0D. For example, a volume is made up of its interior region (3D), outer faces (2D), curves/face edges (1D) and vertices where curves intersect (0D).

   Pruned entities allow users to effectively access the child geometric items of an abstract entity to specify attributes for them. For example, if we need to apply a load on four corner points of a rectangular plate with a class "shell_2d", the load attribute can be applied on the pruned 0D entities of the "shell_2d" class, which consist of the four corner points of the rectangular plate.

## Abstract Modeling Automation

To facilitate abstract modeling automation, there are two distinctive roles involved.

a. 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.

b. CAE Engineer: CAE engineers author the abstract model itself by creating all necessary abstract classes, relations, child entities, and applying CAE physics attributes. Abstract models define the functions to be executed in a CAENexus based automation process. They are much easier and faster to create than writing scripts for automation—days instead of months.

Once these initial roles are implemented and fulfilled, a vast number of simulations can be run seamlessly with minimal effort through CAENexus' automated three-step process: generate simulation model, generate mesh model, and export solver deck.

*Child items help to access geometric entities in very effective ways*

***Abstract models are much easier and faster to create than writing scripts for automation — days instead of months***

*Abstract modeling automation uses a three-step process: generate simulation model, generate mesh model, and export solver deck*

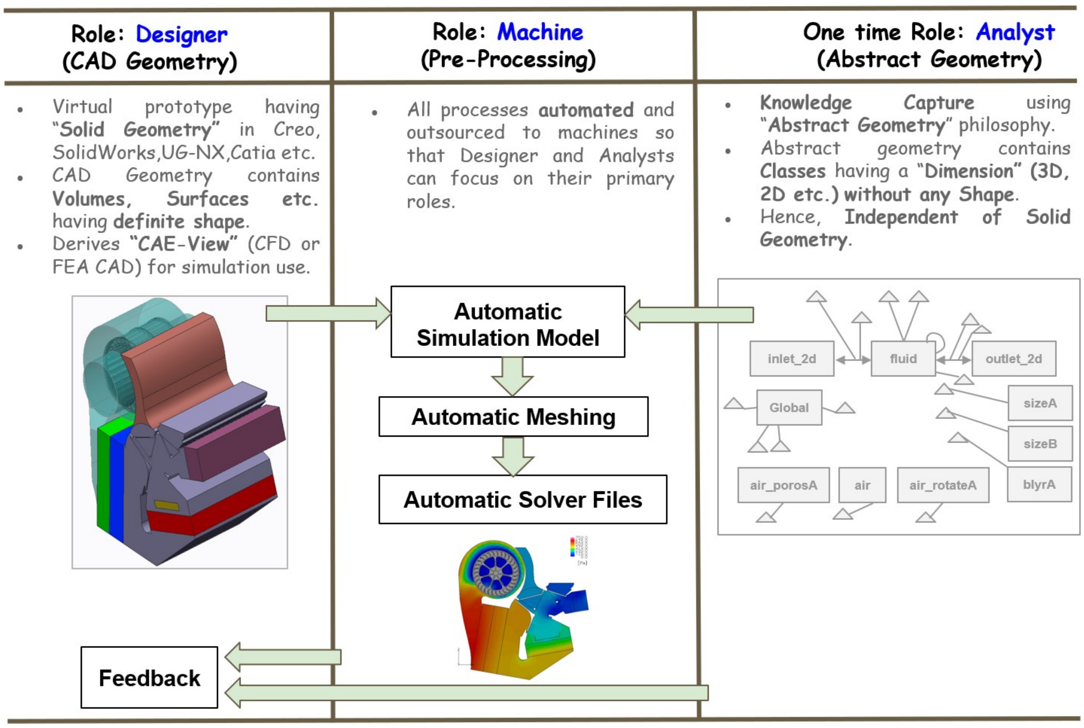


*Figure 10: Automatic Three-Step Process*

## Process Improvement by Automation

Best practice CAENexus automation involves more than just creating a reusable abstract model and then executing the three step automatic process. A more holistic approach can boost the efficiency of the whole process further.

a. **Simulation Based Design CAD (SBD)**: Instead of converting a specific CAD model instance and cleaning it in a pre-processor for often just one simulation, SBD CAD models for CAENexus are efficiently created within the same CAD system used for the manufacturing representation and without the need for geometry translations. An SBD CAD model is CAE-ready and can be considered a "CAE View" or version of the manufacturing CAD model. SBD CAD does not contain features or details unnecessary for the simulation, adds parts or geometries needed (e.g. to define dummy parts to apply forces or the fluid space for CFD), and contains the text strings required for the connection to an abstract model. SBD CAD may also contain simulation parameters to replace attribute values from the abstract model (for example, velocity, mesh size etc.).

   A smart set-up for generating SBD CAD models using macros for automatic defeaturing and/or adding geometries can make the SBD CAD generation very straight forward. Also, using parametric capabilities in connection with part libraries that contain models with both manufacturing and CAE views has proven to be much more efficient than the traditional converter approach.

   The experience of our customers suggests that SBD CAD creation improves their process efficiency **2-3 times** as users can quickly build final CAE assemblies for simulation from library, parametric, and manual feature content.

b. **CAENexus Automation**: This aspect relates to the three-step process explained before. Simulation knowledge capture and reusability, automatic meshing, plus input deck generation improve the CAD to CAE process efficiency **3-4 times,** offering the possibility to maximize machine use for reliable simulations 24 hours a day, 7 days a week, without manual intervention.

*c.* **Simulation Data Management**: It is possible to connect the whole process to data management tools already installed at the customer site. The SDM system can call CAENexus via Python scripts to generate and submit the mesh and input deck for solving the job. It can also queue up jobs per CAD/ CAE solver license availability, HPC management etc. This can improve the efficiency another **2-3 times**.

*CAENexus methodology and automation improves process efficiency by 5-7 times*

Based on customer experience, CAENexus methodology and automation itself improves process efficiency by **5-7** times, and together with simulation data management efficiency can increase overall by **7-10** times.

# Use Cases

In this section, two use cases are taken as examples to demonstrate the reusability of one abstract model for two geometry models, and includes the three-step automated generation of the simulation model, mesh, and solver input deck.

The two use cases selected are first a pressure drop calculation in a HVAC model and second, a temperature and pressure distribution in a burner model.

These models are geometrically highly different as shown with their physics below.

*Two use cases are taken as examples to demonstrate the reusability of one abstract model*

*Components from each model are different*

*CFD physics for both models are the same*

### HVAC Model

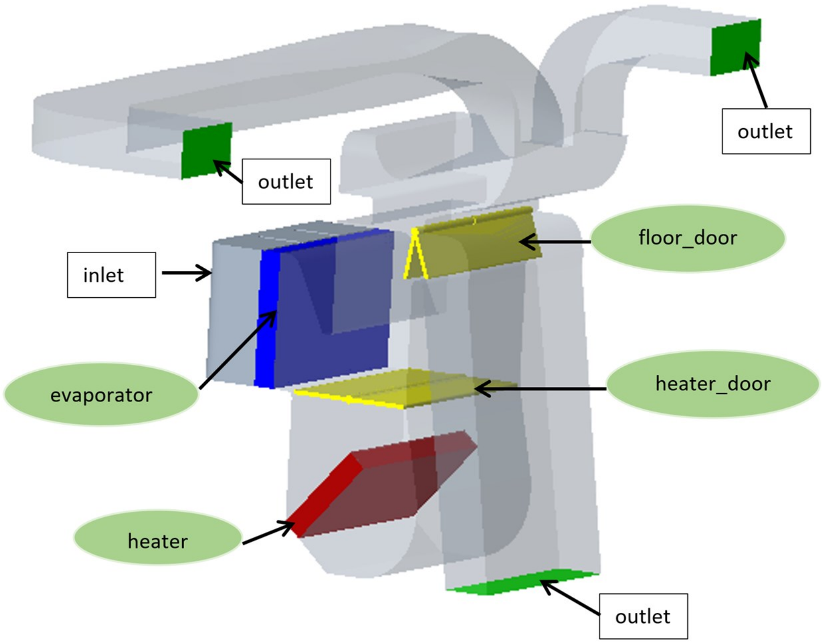


*Figure 11: HVAC Model Use Case*

NOTE: All the parts/bodies have SCLASS* **"air"**.

### Burner Model

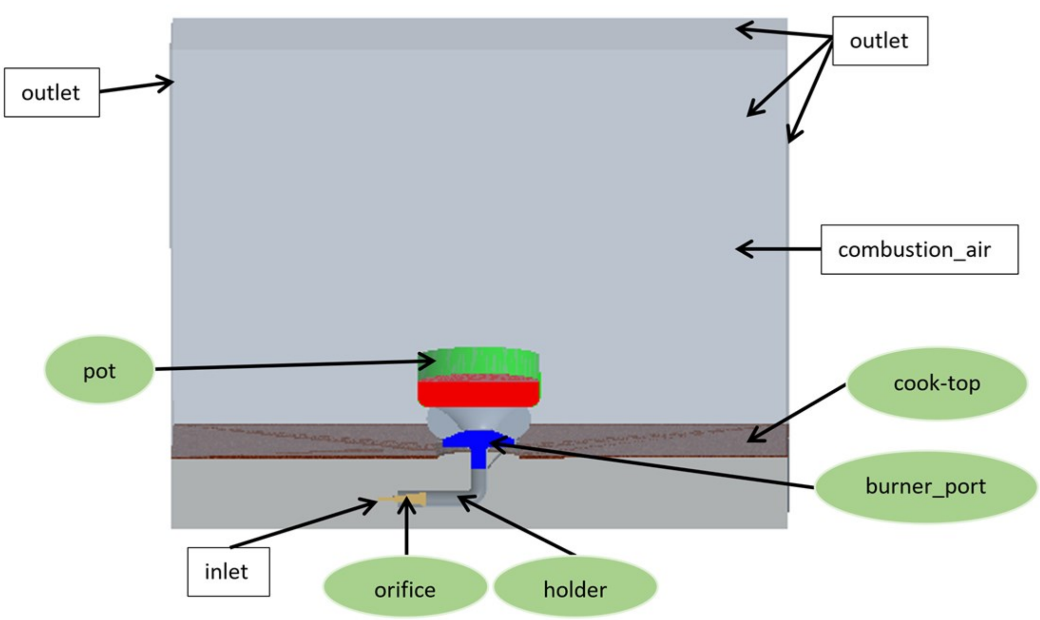


*Figure 12: Burner Model Use Case*

As seen in Figures 11 and 12, the components from each model are different. The HVAC model has components like evaporator, heater, floor_door etc. while the burner model has components like orifice, holder, cook top etc.

However, the CFD physics for both models are the same. In both models, the user needs to apply an inflow, outflow, and wall, as well as define fluid cell sets with material and related properties. The species mixture needs to be considered for the burner model to enable combustion but not for the HVAC model. Despite these differences, a user can create the required simulation classes in one abstract model to work with both geometries.

## Abstract Model Schematic

Figure 13 below shows a schematic representation of the abstract classes used for both cases and the attributes applied to them.

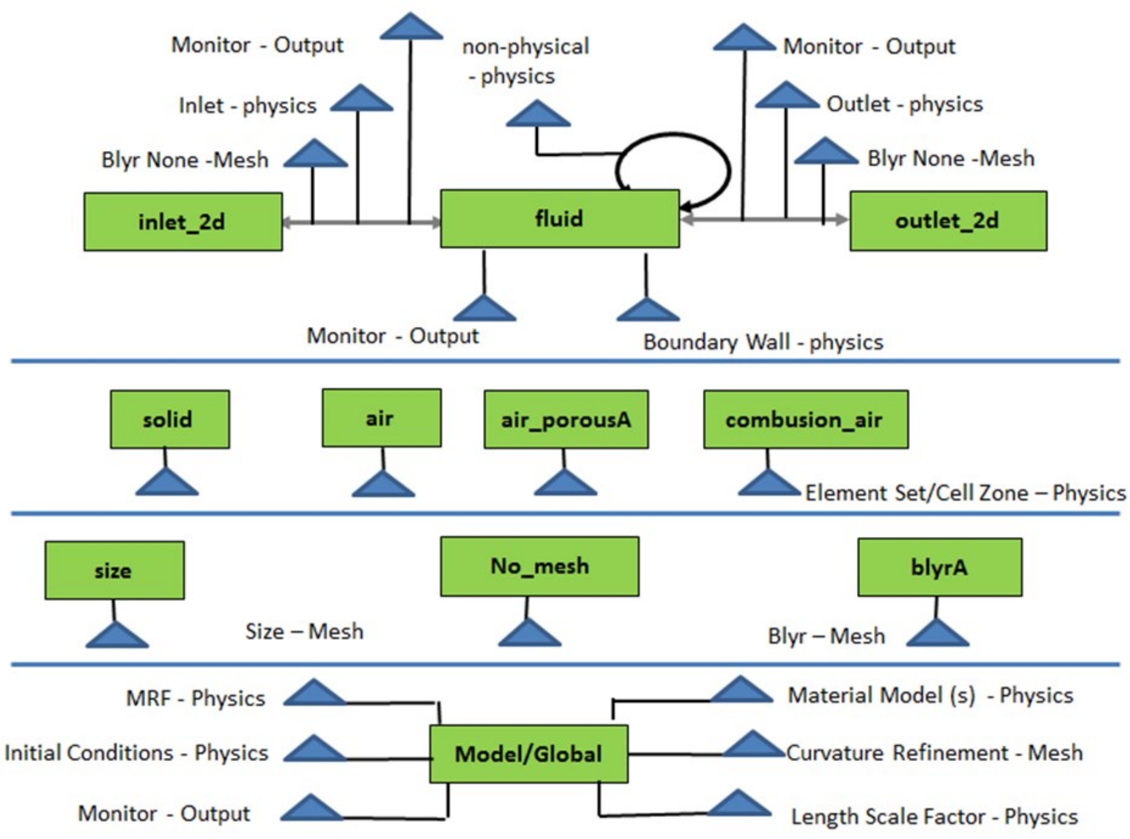


*Figure 13: Abstract Model Schematic Representation*

Using these classes and attributes, we can see the versatility of a single abstract model used for two different applications. A CAD designer trained in this methodology, or a CAE engineer with CAD experience, can attach above class names as simulation class (SCLASS*) parameters on appropriate geometry of the CAD model and add component names (SCOMP) for easier exploration of the simulation model.

*Schematic shows the versatility of a single abstract model used for different applications*

## CAENexus Roles

As mentioned in earlier sections, there are two important roles to automate workflows with CAENexus.

1. **CFD analyst**: The CFD analyst will author all attributes defining CFD behavior and specify them on applicable classes (e.g. as shown in Figure 13 - schematic), class relations and/or their child entities.

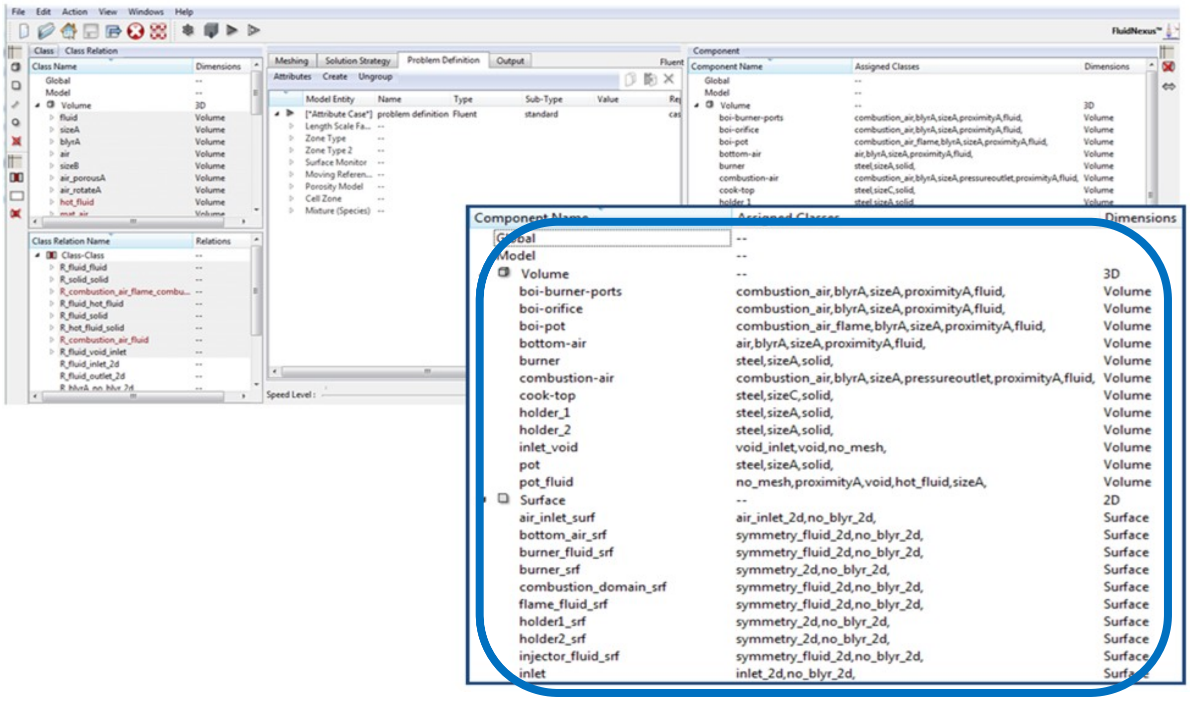


*Figure 14: Abstract Model Used for Both Use Cases*

Note: The “Component Tree” with assigned classes and dimensions (right panel of the abstract model GUI) is updated according to the most recent CAD model imported.

As an example for the abstract model authoring, Figure 15 shows the inflow attribute panel used by the analyst for our use case abstract model.

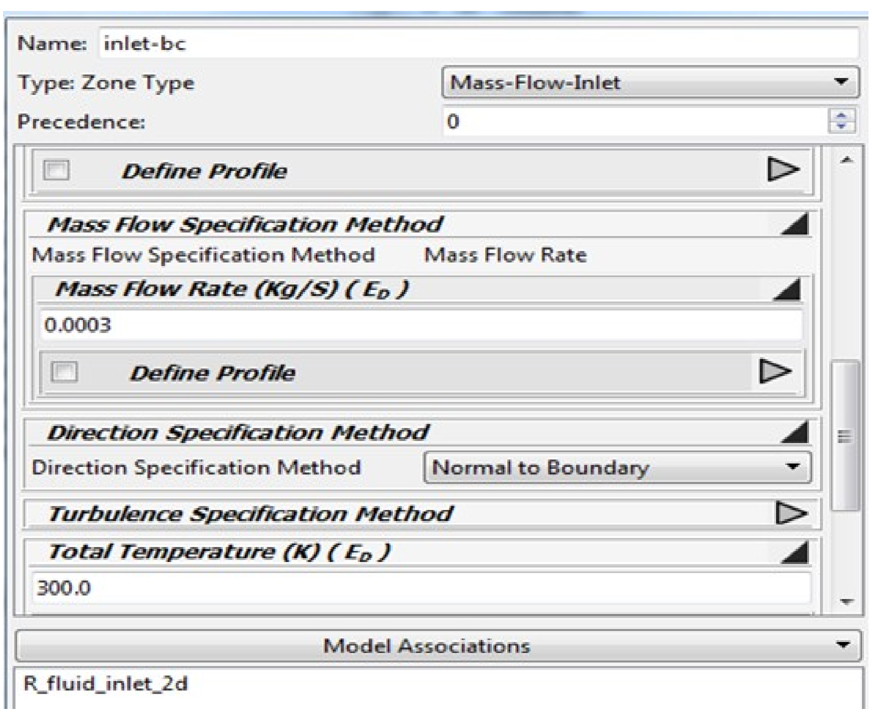


*Figure 15: Inflow Parameter Panel*

*CFD analysts will author all attributes defining FEA behavior and specify them on applicable classes*

2. **CAD designer**: The CAD designer is responsible for applying text strings referring to class names to the CAD models s/he creates. Figure 16 shows the CAD model with SCLASS* and SCOMP parameter values used by the CFD analyst in the abstract model (relates to area inside blue border in Figure 14).

*The CAD designer is responsible for applying text strings referring to class names to the CAD models*

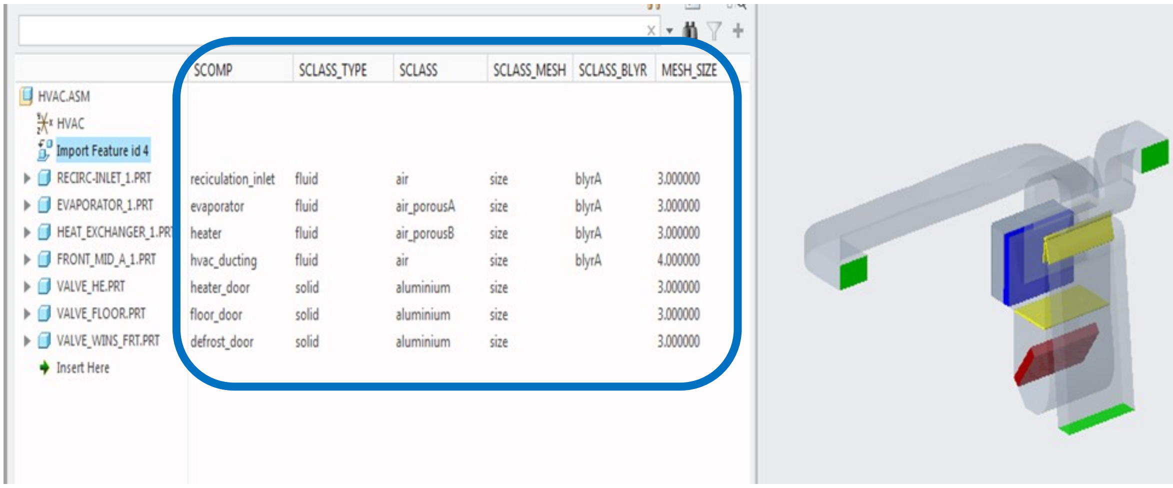


*Figure 16: CAE CAD HVAC Model*

## Abstract Model Reusability

Figure 17 shows the reusability of our abstract model for creating meshes and solver input decks in three automatic steps: 1st create simulation model, 2nd create mesh model, 3rd create solver input deck. The abstract model is reusable due to the attributes applied on simulation classes (CAE behavior like flow velocity, material, etc.) which are common across different CAD models.

*Abstract Models are highly reusable because they contain CAE behavior, such as flow velocities, wall and material properties, within the attributes of simulation classes, which are common and applicable across many different CAD models*

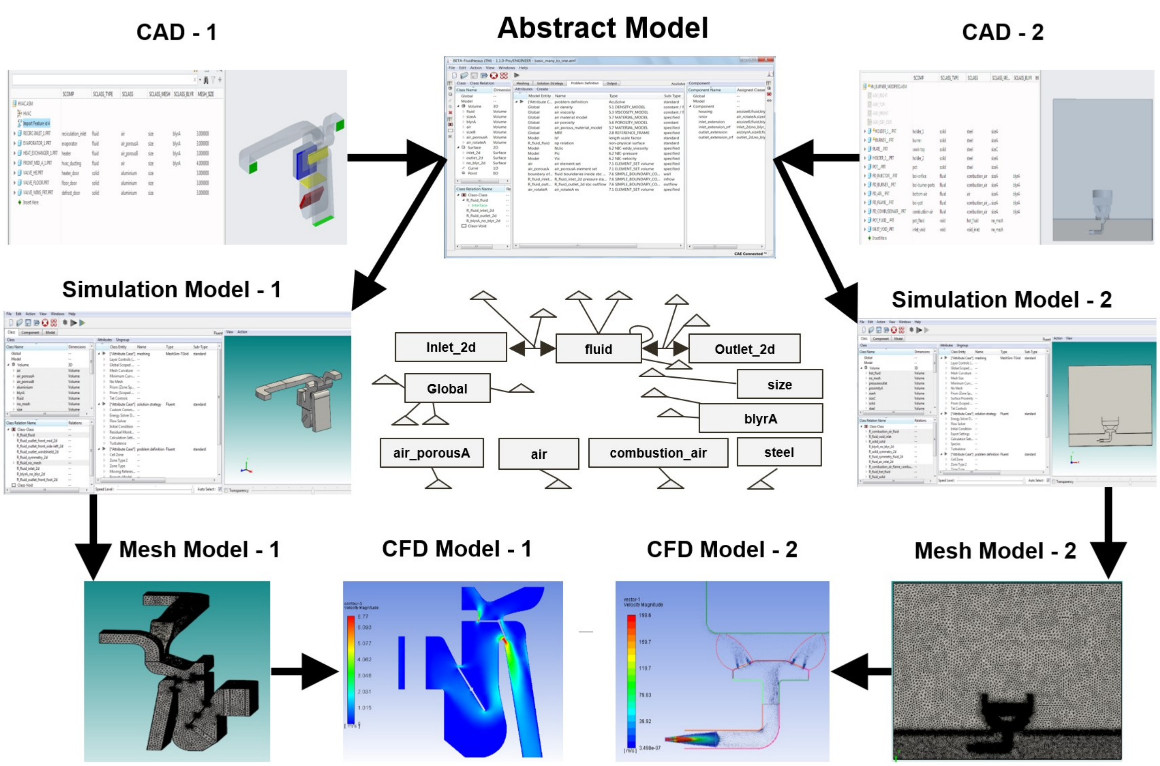


*Figure 17: Abstract Model Reusability*

# Conclusions

As demonstrated in the chapters above, abstract modelling technology offers a unique combination of benefits for product development processes.

⇒ Democratization of simulation enabling CAD designers to start dependable simulations supporting design decisions as models evolve

⇒ Systematic capturing and re-use of simulation know-how and best practices

⇒ Consistent, comparable simulation results independent of where or by whom simulations are performed

⇒ Vastly improved efficiency of CAD to CAE solver input process through robust automation

⇒ **Automation much easier to implement and maintain than with other methods**

At a time of growing international competition and increasing product complexity, CAENexus users, through abstract modeling, save significant time and cost while creating optimized products. They get a better return from their investment in engineering software, gain capacity for more meaningful reports improving design decisions, and overcome crucial losses of knowledge or expertise when experienced simulation specialists leave for another employer or retire.

For more information visit novusnexus.com/ or email to info@novusnexus.com.

**Trademark Notices**

CAENexus, FluidNexus and FEANexus are trademarks of Novus Nexus, Inc.

Creo is a registered trademark of PTC

*CAENexus' Abstract Modeling technology and design methodology provide a unique combination of benefits:*

*Vastly improved efficiency*

*Consistent, comparable simulation results*

*Capturing and re-use of simulation know-how and best practices*

*Democratization of simulation*

*Automation much easier to implement and maintain than with other methods*