Apollo

User features

**confidential**

Revision number: 1

Date:

Name: P. van der Velde

# Goal

The goal for Apollo is to provide engineers and designers with the capability to evaluate and change the behaviour of geometry based designs / objects in different physical situations. Apollo achieves this goal by allowing users to perform different experiments on a design. These experiments of one or more physical models can be virtual (i.e. simulations) experiments which Apollo executes or real experiments for which Apollo only stores and processes the data.

# Features

For Apollo a series of user features are suggested. These user features are described below. Each feature belongs to a certain area.

1. **User interface**
   1. The user interface should provide the user with easy access to the most important operations. Which operations are most important depends on the current context that the application is in. The user interface controls will be designed to provide a look and feel that is consistent with the hosting application. Depending on the hosting application, menu's toolbars, commands and context menus should be provided.
   2. Apollo will provide two different User Interface types, one UI which will integrate with CAD software and one stand-alone UI for project management.
   3. When Apollo is integrated into a CAD program then it should be easy for the user to mark geometry elements and regions with the desired characteristics, whether they are physical or numerical.
   4. The stand-alone UI will provide the user with the ability to browse existing projects and create new ones. Unlike the CAD plug-in the stand-alone UI will not be able to visualize any geometric or experiment data. It will however be able to provide the user access to the project (meta-)data
   5. Both UIs will be able to generate scripts which can be used to create new projects and/or make changes to existing projects.
   6. Both UIs will make it easy for the user to install or upgrade additional capabilities when the user needs them.
2. **Projects**
   1. The user will be able to define a project which is the unit that maintains information about the relations between one or more data sets.
   2. In a project the user can define one or more top level experiments, each of which is independent from each other, although they are usually related in some way.
   3. Each experiment in a project can have multiple sub-experiments which are based on the parent project.
3. **Experiments**
   1. The user will be able to define an experiment which is a unit which describes a real or virtual simulation. The experiment unit will store the setup and properties for the simulation as well as any data that is gathered.
   2. Experiments will be able to store both calculated data (e.g. from virtual simulations), gathered data (e.g. from real simulations) and theoretical data (e.g. from theoretical approximations). Apollo will be able to differentiate between these three data types and will be able to treat them accordingly.
   3. The user only needs to specify the physical and geometric properties for an experiment. Based on the physical and geometric properties Apollo can select or suggest the appropriate processing data / algorithms / methods.
   4. The user only needs to specify the geometry of the object under study. Apollo should be able to determine if additional boundaries need to be created and will automatically do so if required.
   5. Based on the users input Apollo will be able to suggest one or more approaches for the processing of the specified data.
4. **Workflow**
   1. **General**
      1. The user can easily specify which 'variables' are of interest
      2. The user can easily specify why an project is being run. Possible suggestions for running an project are:
         1. What-if; also known as concept exploration
         2. Optimization or robust design
         3. Scenario research
      3. The user can define the properties of a project in any order. Properties that the user can define are:
         1. Goals for the project
         2. Physics models for each domain and expected input and output values for the boundaries
         3. Geometry and the meaning of different geometry elements (e.g. wall, symmetry line etc.)
         4. Accuracy and output requirements
   2. **CAD environment**
      1. The user can easily indicate what the properties, both physical and chemical, are for each domain.
      2. The user can easily specify what the desired error bounds are on the final results. Apollo will ensure that all calculations fall within these error bounds.
      3. Data visualization
         1. Apollo will make it easy for users to switch between data sets in a project
         2. Apollo will allow users to compare two or more data sets for differences or similarities
         3. Apollo will assist the user in finding areas where 'interesting' phenomena are occurring
         4. Apollo will allow multiple visualizations to be active at the same time
         5. Apollo will allow users to view 2D data on 3D geometry
   3. **Project explorer**
5. **Input**
   1. Apollo should always display the units for a specific value, however the user should not have to specify the value in the default units. Apollo will automatically convert the value if needed.
   2. If a user enters a value with a non-standard unit then Apollo will always show this value with the user-specified unit.
   3. Users should be able to specify tolerances and error bounds on all values.
   4. Apollo will offer the user the opportunity to reset a value back to the default value.