Apollo

Project Explorer

Feature list

**confidential**

Revision number: 1

Date:

Name: P. van der Velde

# Goal

The goal for the Apollo Project Explorer is to provide the user with a stand-alone UI for the creation and exploration of Apollo projects.

# Features

For the Apollo Project Explorer a series of user features are suggested. These user features are described below. A few abbreviations are explained below.

**Notation**

TBD To Be Determined

The feature list, sorted by area:

1. **Users**
   1. The Project Explorer is aimed at two different groups of users. These are:
      1. Designers who will use Apollo to design new products and improve existing ones.
      2. Researchers who will use Apollo to investigate the behaviour of different objects in different physical conditions.
2. **Operations**
   1. Global
      1. The user can easily create a new project. The Project Explorer will create the new project and display the data for the newly created project. Note that no geometry can be created in the project explorer.
      2. The user can open an existing project. Upon opening a project the current project will be saved, should the user desire so, and the selected project will be loaded.
      3. The user can save the current project, either with an already provided file name or with a new one. The Project Explorer will ensure that all data is written in a redundant manner so as to always be able to reopen the users file, even if the file is damaged.
      4. The user will be able to close a project without closing the Project Explorer application. When closing a project the user is given the opportunity to save the project if any changes have been applied since the last save action.
      5. The user will be able to import project data from other systems (TBD).
      6. The user will be able to export project data to other systems.
   2. On project
      1. New data set
      2. Open project in CAD
      3. Properties
   3. On data set
      1. Copy data set
      2. Open data set in CAD
      3. Lock
      4. Unlock
      5. Properties
      6. Activate (run / process)
3. **Workflow**
   1. Create new project
   2. Open project
   3. Save project
      1. Prevent data corruption
   4. Add new data set
   5. Generate script
   6. Show data in data set
   7. Install new components
   8. Add data / visualizers / tools to a data set / project
   9. Set properties
   10. Set project goals / tolerances / error levels
   11. Setting / changing parameters
   12. Data generation / processing
   13. Batch running of experiments
   14. Reset data set
   15. Delete data set
   16. Export data
   17. Revert to save point
   18. Scheduling on local machine / remote machine(s)
4. User interface
   1. Undo / redo
   2. Help system
   3. Visual help
      1. Indicate which data set(s) are auto-created
      2. Indicate which data set(s) is/are being processed
      3. Indicate which data set is out-of-date
      4. Indicate which data set is selected + parents of selected item
      5. Run mode (batch, directly, postponed etc.)
      6. State of data set (running, paused, stopped, not-run-yet, loaded, not loaded etc.)
      7. Copy on write state (copied but not written, partial write, complete write)
      8. Data set type (simulation, experiment ...)
   4. Data set information
      1. Processing information (time, memory, last write date/time, components etc.)
   5. Processing information
      1. Progress
      2. Current state

Ideas

* Configuration files
* Auto update
* Checks on file integrity?
* Ability to check a file for loadability (i.e. if the current version of the explorer / CAD can actually load the file. Loading can fail because of version or missing components etc.)
* Auto-add missing components
* Components per project / data set

UI Screenshots

* Project base (geometry etc.) has:
  + Base data (Geometry, Physics models, Boundary conditions, Initial conditions etc.)
  + Visualizations (way data should be shown)
  + Solvers / Equations necessary to translate known data into other known data (e.g. calculation of Mach number from speed and temperature etc.)
* Individual data blocks have:
  + Visualizers necessary to translate between local data format and project data format
  + Additional tools necessary for generation / translation of data
* Expecting to show the user a graph of the project with:
  + The project initial data (geometry, physics models etc.) as the main node.
  + Each child node is a single experiment. Experiments that are directly connected to the parent are user created (?)
  + System created experiments (child experiments) are shown in a different colour (lighter) and cannot be edited (but can be selected)
  + User can drag nodes around (connection stays)
  + Connections cannot be changed. This would imply changing the parent of an experiment which leads to mayhem.
  + User can drag-and-drop nodes onto another node. This changes the links. The dropped node becomes the child of the drop-target node. This only works for user created nodes. Bonds between the child nodes and the dragged node are maintained (important for system generated experiments).
* The graph control should allow
  + Selecting a node. All parent nodes should also be highlighted (but not selected) in weaker colours as they are further away from the selected node. This allows the user to quickly see where the data comes from. The highlighted node will be surrounded by a glow in the system highlight colour.
  + Allow auto filtering of nodes through the search control. ‘disable’ nodes that do not match with the search criteria. In this case nodes aren’t really disabled but more faded out a bit. Selecting a ‘faded’ node should still be possible
* When a node is selected the properties for that node are displayed in the properties dialog (use data binding to link).
* When processing an experiment a set of progress controls should be shown. These are:
  + A progress bar
  + A estimated time for the processing + error indication on the time
  + A text block indicating the current (global) action
  + Buttons that allow the user to pause and stop the processing
  + An identifier which identifies the experiment
  + An expander button which can be used to hide / show a block with more advanced details (e.g. which exact action is being processed etc.)
* When the user clicks on a progress block the relevant experiment should be selected. Ditto if the user selects an experiment that is being processed then the progress block should be selected.
* If an experiment has child experiments which have to be processed when processing the parent then the progress block should show the progress of the parent experiment. The progress of the child experiments should be shown in progress blocks which are children of the parent progress block. It should be possible to hide these.
* Parent progress blocks should also indicate if the parent experiment is in a waiting state or not.

Questions:

* What should nodes display? Do we allow a node to expand or do we show stuff in the properties section of the application?
* How do we show tools / visualizers / data?

Non-features:

* No displaying of geometry
* No setting variables / properties on geometry