ProMo Equation Editor

Purpose:

This editor defines the multi-way bipartite graph of variables defined by expressions(variables). ProMo allows for more than one definition for a variable, thus generating a multi-way bipartite graph.

Description:

The equation editor builds on the ProMo ontology generated with the ontology editor. The construction of the bipartite graph is initialised by defining a set of port variables. They include the constants and the fundamental state variables. For physics, the state variables are the conserved and balanced quantities, thus energy, mass in as species mass, and momentum linear and rotational. Each variable and expression is associated with a node in the domain tree, which allows for multiple definitions of a variable.

Rules:

- The equation system is built on a state-space background.
- Tokens are the basic conserved quantities and species mass.
- Tokens in a capacity is defined as a **state.**
- **Equations** are, in fact, definitions of the form *variable := expression(already defined variables).*
- Equation definition starts with states and constants and network – whilst not fixed at this point, it is envisioned that this will become the standard.
- Variables are either **port variables** or defined via equations.
- **Port variables** are "starting points" – states, constants, network descriptors, etc.
- Equations can only be established with already existing variables.
- Units and indexing are computed. That is, they are derived from the right-hand-side expression – removes possible errors.
- Already defined equations can be edited to a limited extent.
- A variable may be described by more than one equation to cover different use.

Limitations:

The whole system is currently configured for experimentation with ProMo ontologies. Thus, it has a couple of elements that are to be done by hand, but if found useful, they will be implemented in the background.

Usage:

1) Initial dialogue asks for an ontology name

- rejecting opens a dialogue asking for a new ontology name
  - Rejecting exits.

        - Providing a name generates the directories and the infrastructure for the ontology and models being build using this ontology.

2) Main window opens, providing four choices

- variables --> opens a tree view of the ontology-defined structure in the first tab
- selection of an entry triggers the editing mode for the tree components
    - graph
    - node
    - arc
- The selection of variable class shown below opens a new window with a table of the defined variables for this node in the definition tree and the selected variable class. See point 3) to continue.
- Variable aliases --> opens the same tree view selection of an entry in the tree opens a new window with the alias table for the selection. The columns are labelled with the available compilation languages.
- Compile --> compiles all variables & equations into the target languages and generates LaTex documentation that is displayed.
- Index aliases --> opens a window to change the notation of the aliases. These are usually not to be modified. The default notation served us so far very well indeed.

3) Entering or modifying an equation

- On selecting a variable class to edit, a table for the variable class on the selected tree node is opened. It shows the currently defined variable. New variables are defined by clicking on them anywhere in the type column. A new variable may be:
    - a port variable that is not a function of other variables or
    - a variable defined by an expression.
- Variables are defined for a specific tree node. The context extends all the way down the branch.
- Once a variable is defined, equations can be added to its definition. Those new/additional definitions may be on a node in the context branch.
- Labels and descriptions can be changed.
- Units and indices cannot be changed once the variable is used in an expression because the information defines the new variables.
- Equations can be edited by selecting the respective field in the "eqs" column.

– end –