Dymola 2019
Overview of new features

1 June 2018

Executive Summary

Model creation
- Easily capture parameters of existing components to create new models.
- Improved use of display units in the diagram layer and in plots reduces model ambiguity.

Simulation
- More robust and efficient handling of models with nonlinear equation systems through improved handling of guess values and choice of extrapolation methods.
- Improved diagnostics at compilation and runtime helps find difficult numeric problems, and intermediate values can be checked during debugging.
- More efficient code generation for parameters, and eliminated code-bloat for large tables of data.

Modelica libraries
- New Pneumatic Systems library.
Model editing

Capture parameters of a component

- You have a component with parameters or other modifiers such as re-declarations
- Create a new model or a favorite capturing those settings

```plaintext
  J=1,
  pos(fixed=true, start=0),
  w(start=0, fixed=true)
}
```
Display units

- Improved presentation of display units
  - Diagram layer (can be disabled)
  - Parameter dialog
  - Plot window

- Possible to select display unit in the variable browser

Customizing the Libraries menu

- Select which libraries will be shown
  - Rather than all installed libraries
- Reorder libraries as you want them
Other features in the graphical editor

- Filtering of inherited components in the diagram layer of editable models
- Improved plotting of Boolean and Integer arrays in the matrix editor
- Possibility to add variables to bus declarations when connecting expandable connectors

Simulation
Initial guesses for nonlinear equations

- Initial guesses are used for nonlinear systems of equations
  - To enable fast and robust solution
  - Improved in Dymola 2019
- Inter-/extrapolation method used to compute initial guesses can be controlled
  - Chose between linear or constant extrapolation
  - Advanced.Define.InitialGuessInterpolation=n
- Can force the integrator to solve nonlinear equations each integrator step
  - Updates the initial guesses more often
  - Advanced.MoveEquationsToDynamics=true

Improved nonlinear solver diagnostics

- Help to debug nonlinear equations
  - Equation systems appropriately named
  - Dependencies logged during translation
  - Information about common causes for failures
  - List of settings to get more information
- Nonlinear solver summary for performance estimation

```
Nonlinear solver summary, accumulated amounts:
  Tag       : Calls, Residues, Iterations, Jacobians
  initialization nonlinear[2]: 1, 0, 0, 0
  simulation nonlinear[2]: 13956, 339516, 389826, 439513
  simulation nonlinear[0]: 159196, 159310, 199100, 993256
To also log used CPU-times enable Advanced.GenerateBlockTimers.
```
Values after failed initialization or simulation

- Plot variables after failed initialization or simulation
  - Intermediate values available even though a solution has not been found
  - Helps find what variables caused the problem
  - May be inconsistent, depending on how far the calculation got before aborting
- Intended as a debugging tool
  - Not enabled by default

DAE mode for nonlinear equation systems

- May give more efficient simulation if there are many or large nonlinear equation systems
  - Fewer nonlinear equation systems are solved
  - Not a universal solution
- Must be explicitly enabled
  - Advanced.Define.DAEsolver = true
- Only available for certain numerical integrators
  - Dassl, Radau Ila, Esdirk23a, Esdirk34a, Esdirk45a, and Sdirk34hw
More efficient C code generation

- Alias elimination for parameters reduces code size
  - Advanced.AllowParameterAlias = true
- More efficient handling of constant table data
  - Table is not allocated in result file and generated C code is reduced

```c
model HideArray
    parameter Real myTable[;2]=[linspace(0, 90, 60), sort(linspace(0, 0, 100, 60))]
    annotation (Evaluate=true, HideArray=true);
    Modelica.Blocks.Tables.CombiTable1D combiTable1D(table=myTable) a;
    Modelica.Blocks.Sources.Clock clock b;
    equation
        connect(combiTable1D.u[1], clock.y) c;
        annotation (_Uimols_experimentFlags (Advanced.AllowParameterAlias=true)) d;
    end HideArray;
```

Functional Mockup Interface

- Reorganization of simulation setup dialog for FMI
- FMU Export
  - Defining user-defined model identifiers for FMUs in GUI
  - Sparse Jacobian handling when including source code supported
  - Sparse solver support
  - Export of models in DAE mode
- FMI Kit for Simulink
  - Support for MATLAB R2017a and R2017b
Other new features

- Plot improvements: Signal operators for curves plotted against independent variables other than time
  - For example: fuel consumption vs. distance travelled
- Taskbar icon shows approximate progress
  - Relative the simulation stop time
- Support for Visual Studio 2017 compiler

Modelica libraries
Pneumatic Systems Library

- Architecture design, component sizing, modelling and simulation of pneumatic systems
- Physical effects
  - Capacitance, resistance, heat transfer, power transformation
- Components
  - Valves, reservoirs, actuators, piping, sources

Major updates in several libraries

- Battery
- ClaRa DCS
- ClaRa Plus
- Cooling
- Electrified Powertrains
- Fluid Power
- Human Comfort
- Hydrogen
- Testing
- Thermal Systems
- VeSyMA
- VeSyMA Engines
- VeSyMA Powertrain
- VeSyMA Suspensions