Executive Summary

Model editing
- Fine-grained control of what signals are exported in an FMU

Simulation
- Possible to terminate simulation after max run-time, even if stuck in external code
- Run simulation until all transients damp out
- Plot bar charts and area charts, or externally generated contents including HTML

Environment
- Support for Windows Subsystem for Linux (compilation, FMU cross-compilation)
- Follows the latest Modelica language specification and MSL 4.0.0
Model Editing

Filter signals when exporting FMU

- Select individual signals to expose in exported FMU
  - Allows fine-grained control of the FMU interface
  - By default follows selection of model description filters (e.g. “black box”)
  - Inputs cannot be hidden
  - Same for `translateModelFMU()`

- Complements signal filtering for import of FMUs
Library startup script

► Run a script when opening a package
  ▶ Any setup needed for the library
  ▶ Applies to top-level packages only
► Default requires no change in library
  Resources/Scripts/startup.mos
► Change default with Attributes>Version

Miscellaneous improvements

► Support for Modelica Language Specification 3.5
► Improved model editing API in ModelManagement.Structure.AST
  ▶ Restructured and extended
► Shortcut for File>Search
  ▶ Ctrl-Shift-F
Maximum runtime of simulation

- Terminates simulation after a maximum run time
  - Wall-clock time, not CPU time
  - Run time unit is independent of simulation time unit
  - Can be stored in model
- Applied to both single simulations and batch runs
  - Total run time for batch simulation
- Checked by Dymola so it works even if the simulation is stuck in
  - Imported FMUs
  - External C code
Transient mode to reach steady-state

- Runs simulation until transients damp out (or until stop time)
  - Checks the derivatives of all states
  - Tolerance can be set, 0.02 by default
  - Set in Simulation>Setup>General
- Does not detect
  - Periodic steady states
  - Steady-state of a subset of the states

Formal definition

Terminate simulation when for each state \( x_i \)

\[ |\text{der}(x_i)| \leq \text{tolerance} \times (|x_i| + |\text{nominal}|) / 2 \]

where tolerance is

Advanced.Simulation.SteadyStateTerminationTolerance

if non-zero, or by default

\[ 0.02 / \min(T_{\text{stop}} - T_{\text{start}}, 500 \times \text{interval}\_\text{length}) \]

Enhanced plotting

- Plotting bar charts and area charts
  - New marker styles
- Support for plotting of external files
  - PNG or SVG format
  - HTML with active content from e.g. “plotly” in Python
Environment

Compilation using Windows Subsystem for Linux

- Windows Subsystem for Linux (WSL)
  - Runs Linux under a micro-kernel in Windows 10
  - Typical distributions: Ubuntu 20.04 LTS and Kali
- Generated C code is compiled using WSL
  - Native gcc/clang compiler on the Linux system
  - Fully integrated in Dymola and transparent
- Cross-compilation for Linux on Windows
  - Generate FMUs with Windows and Linux code directly
Dymola running on WSL

- Dymola for Linux runs under WSL
  - Need to install window manager
  - Remote desktop (xrdp)

Environment

- Improved diagnostics when connecting to a license server
  - Name server lookup
  - Connection to server computer
  - Network latency
  - License server responding

- Intel compiler discontinued