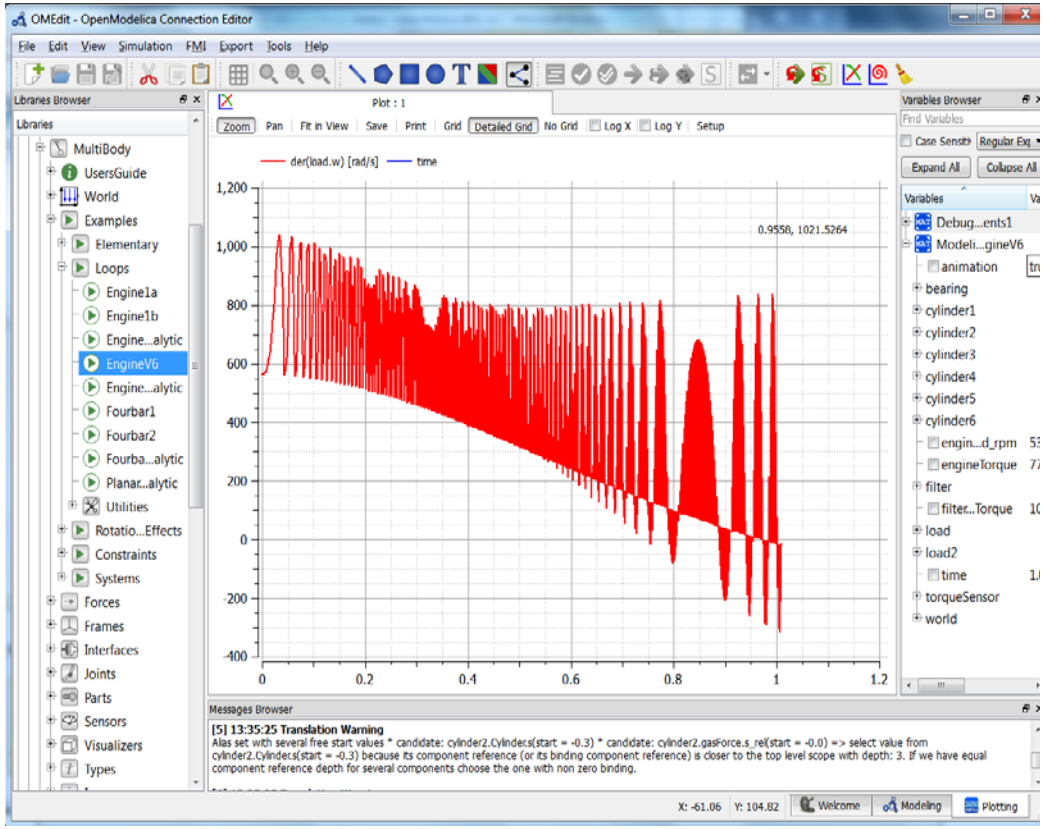
Modeling, Simulation, and Development of Cyber-Physical Systems with OpenModelica and FMI



Presentation at the USA Modelica Conference

October 10, 2018

Adrian Pop

adrian.pop@liu.se

Technical Coordinator at the Open Source Modelica Consortium

Adjunct Associate Professor at Linköping University

LINKÖPING UNIVERSITY

MODELICA

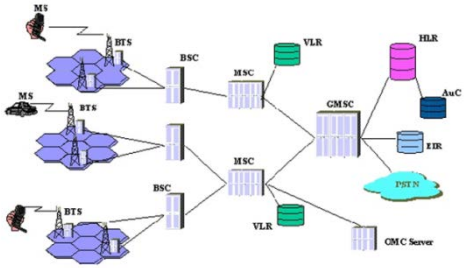
Main Goals for the OpenModelica Effort

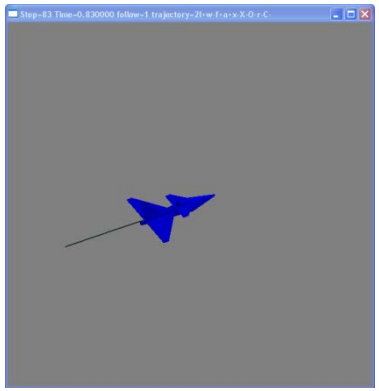
- A comprehensive Open Source Modelica and FMI modeling, compilation, simulation and optimization environment based on free software distributed in binary and source code form for research, teaching, and industrial usage

- Support model-based development of cyber-physical systems, from requirements, to models, to simulation and production code

Industrial Challenges for Complex Cyber-Physical System Products of both Software and Hardware

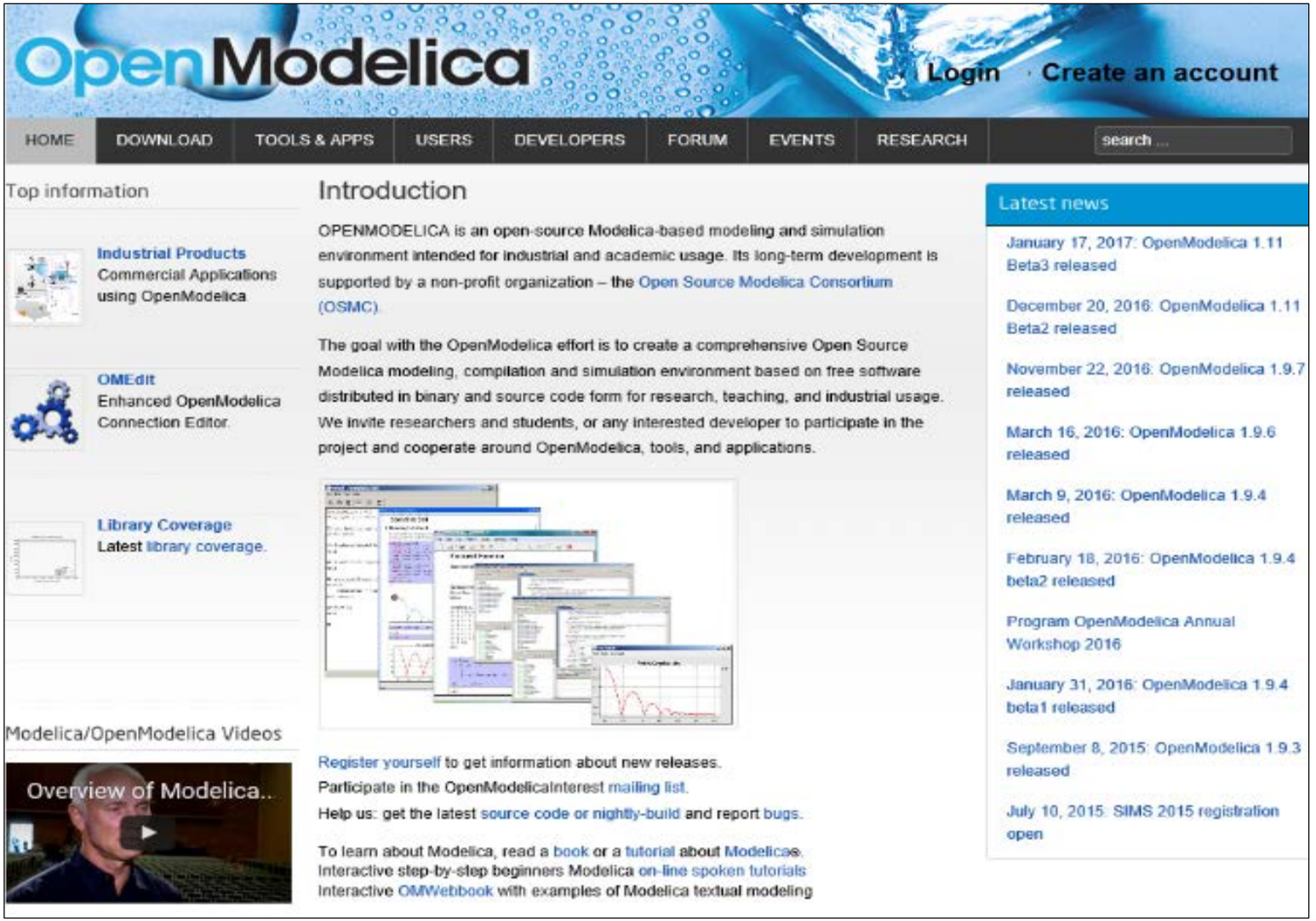
- Increased **Software** Fraction
- **Shorter** Time-to-Market
- Higher demands on effective strategic **decision** making
- **Cyber-Physical** (CPS) – Cyber (software) Physical (hardware) products





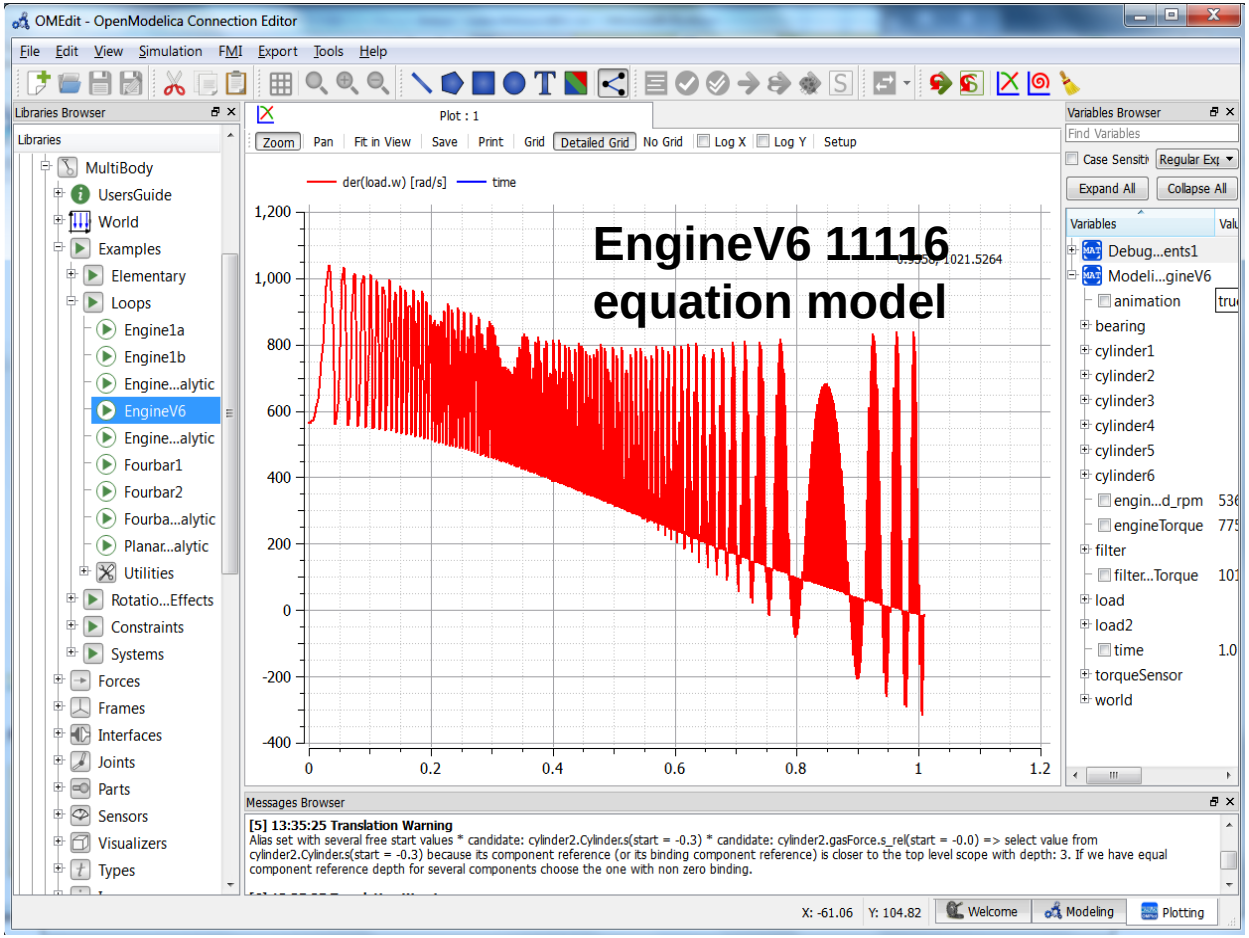
The OpenModelica Environment

www.openmodelica.org



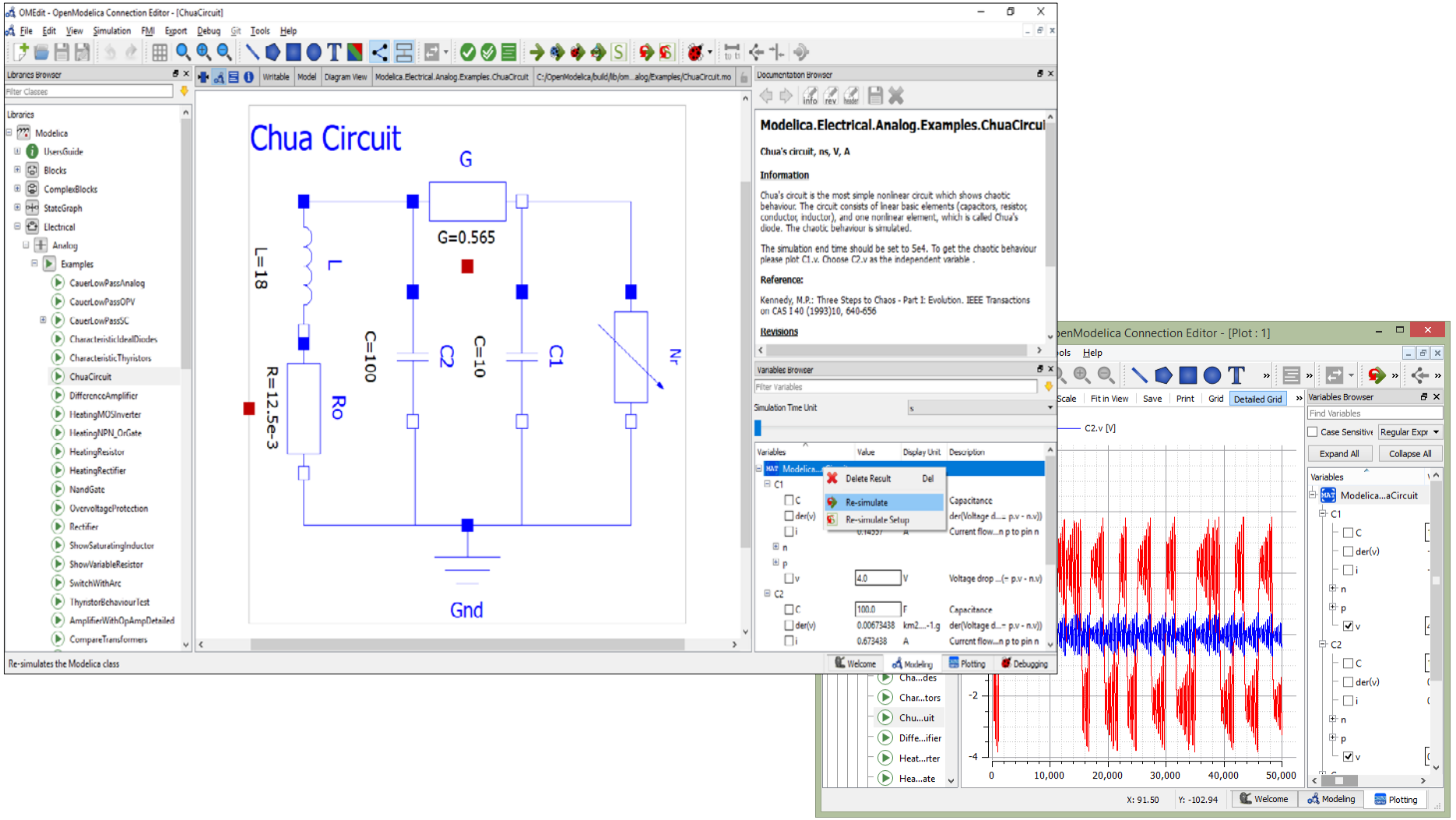
OpenModelica – Free Open Source Tool
Developed by the Open Source Modelica Consortium (OSMC)

- Graphical editor
- Model compiler and simulator
- Debugger
- Performance analyzer
- Dynamic optimizer
- Symbolic modeling
- Parallelization
- Electronic Notebook and OMWebbook for teaching
- Spokentutorial for teaching

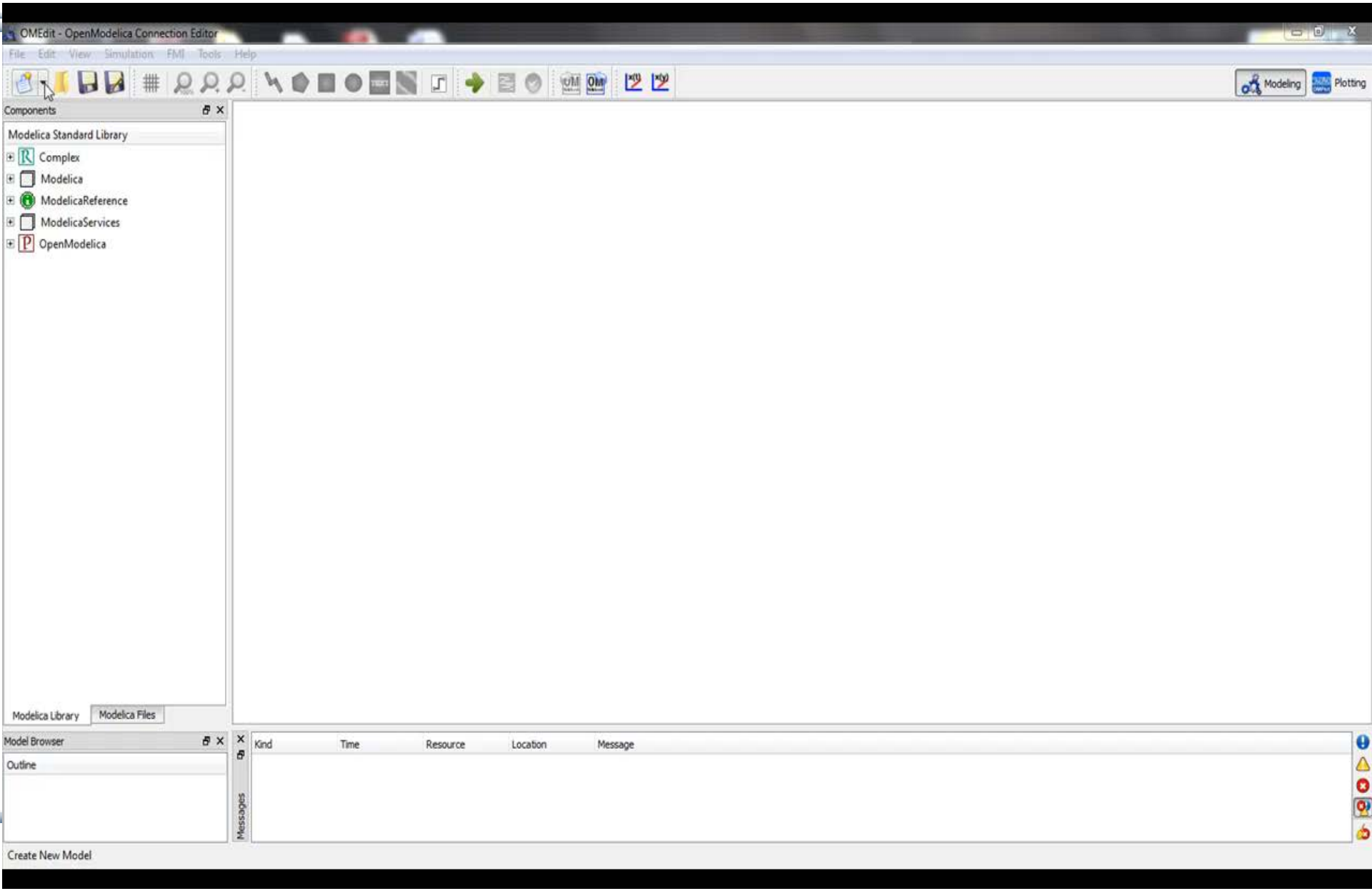


MODELICA

OpenModelica Graphical Editor and Plotting



Graphical Modeling with OpenModelica Environment



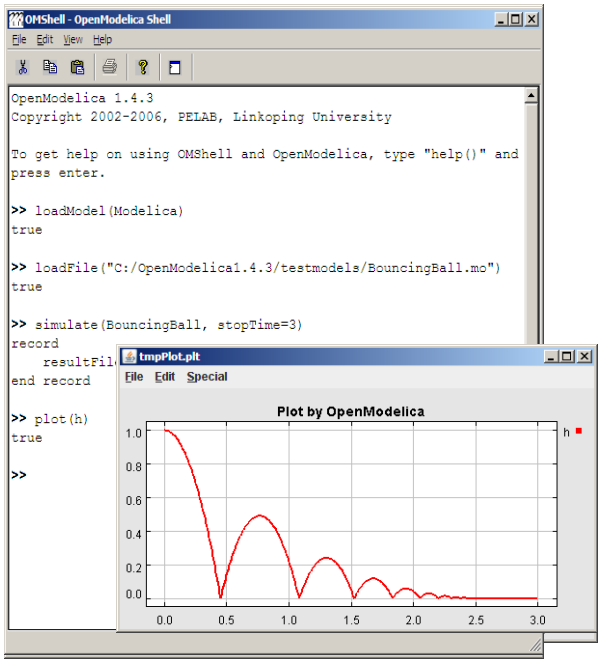
The OpenModelica Open Source Environment

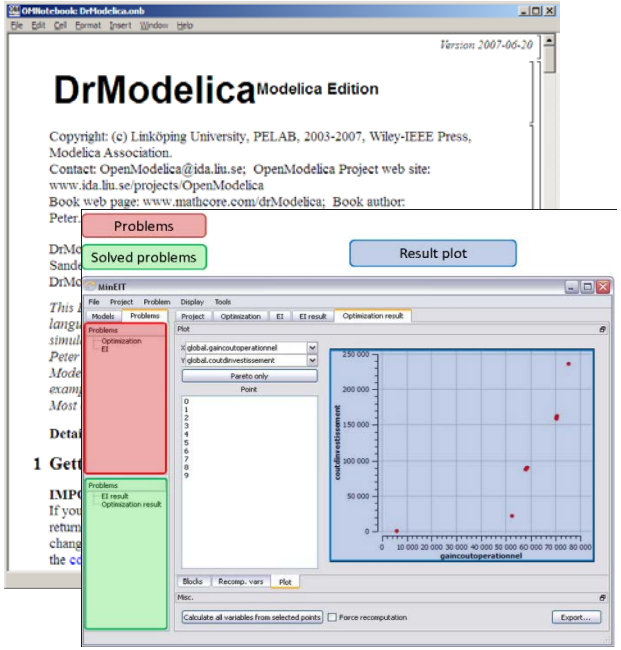
www.openmodelica.org

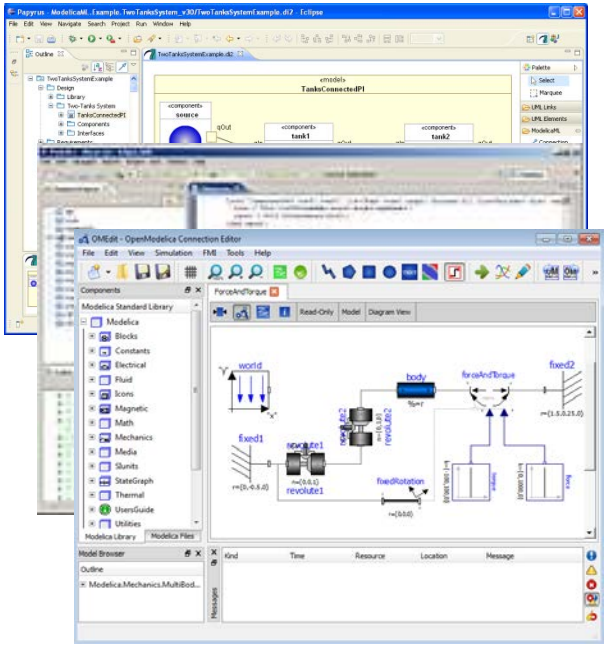
- Advanced Interactive Modelica compiler (OMC)
  - Supports most of the Modelica Language
  - **Modelica** and **Python scripting**
- Basic environment for creating models
  - **OMShell** – an interactive command handler
  - **OMNotebook** – a literate programming notebook
  - **MDT** – an advanced textual environment in Eclipse

new

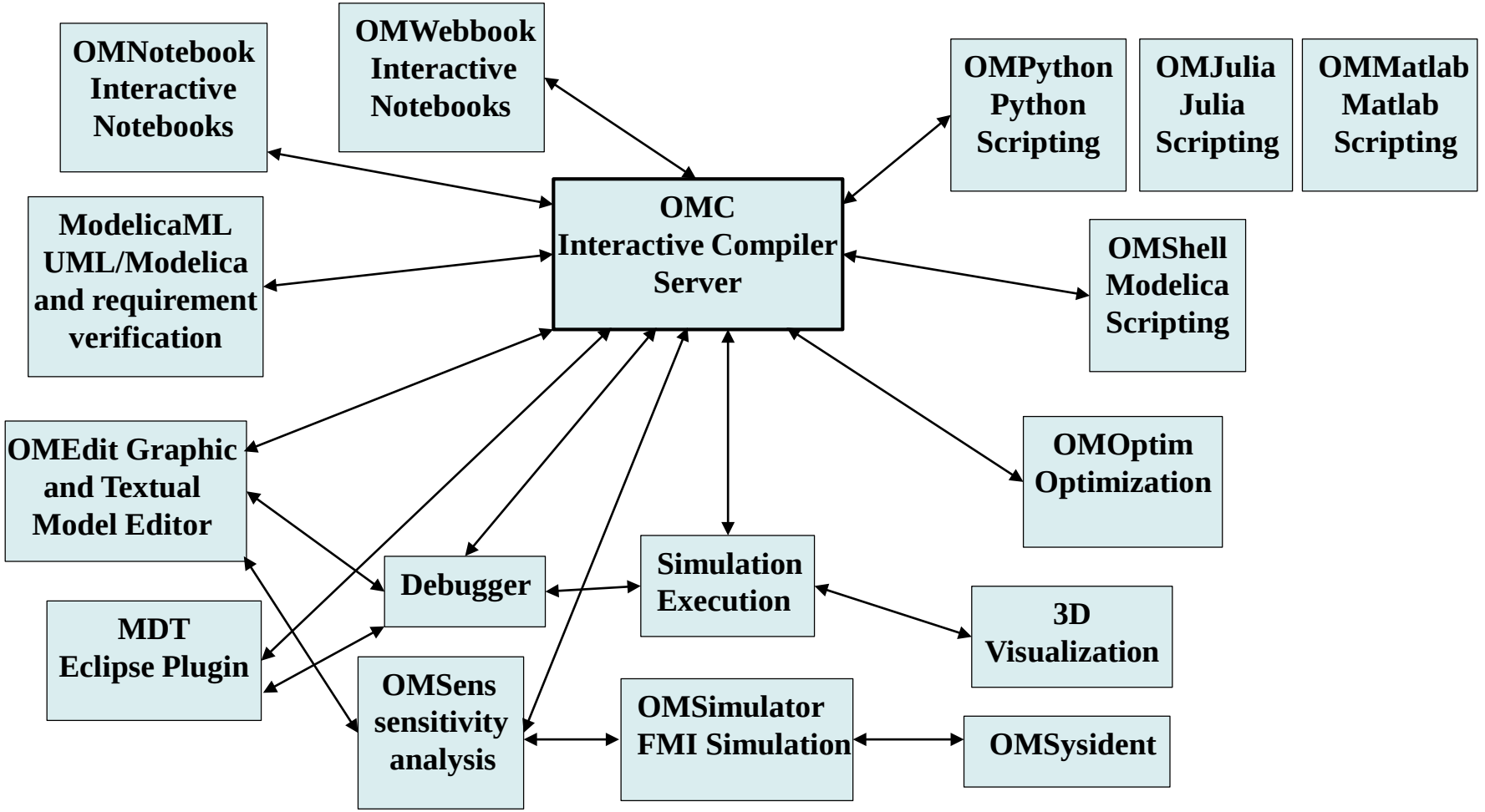
- **OMEdit** graphic Editor
- **OMDebugger** for equations
- **OMOptim** optimization tool
- **OM Dynamic optimizer** collocation
- **ModelicaML** UML Profile
- **MetaModelica** extension
- **ParModelica** extension
- **OMSimulator** – FMI/TLM simulator







The OpenModelica Tool Architecture



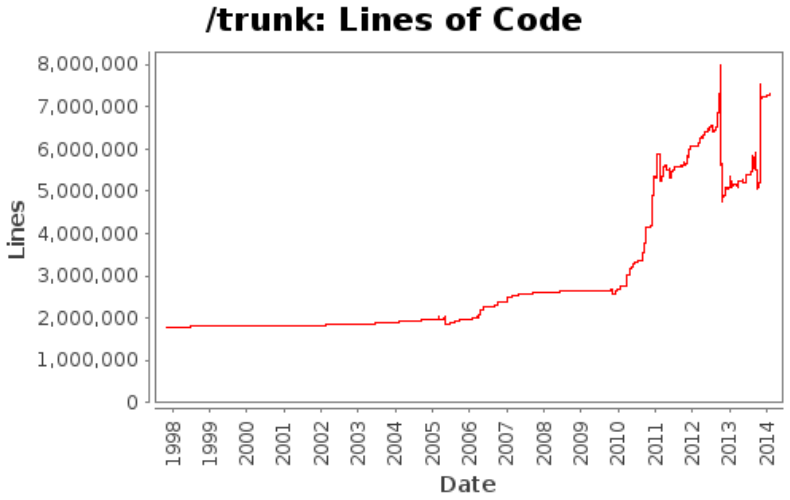
OSMC – International Consortium for Open Source Model-based Development Tools, 53 members Febr 2018

Founded Dec 4, 2007

Open-source community services

- **Website and Support Forum**
- **Version-controlled source base**
- **Bug database**
- **Development courses**
- **www.openmodelica.org**

Code Statistics



Industrial members

- **ABB AB, Sweden**
- **Berger IT-Cosmos, Germany**
- **Bosch Rexroth AG, Germany**
- **Brainheart Energy AB, Sweden**
- **CDAC Centre, Kerala, India**
- **Creative Connections, Prague**
- **DHI, Aarhus, Denmark**
- **Dynamica s.r.l., Cremona, Italy**
- **EDF, Paris, France**
- **Equa Simulation AB, Sweden**
- **Fraunhofer IWES, Bremerhaven**
- **INRIA, Rennes, France**
- **ISID Dentsu, Tokyo, Japan**
- **Maplesoft, Canada**
- **RTE France, Paris, France**
- **Saab AB, Linköping, Sweden**
- **Scilab Enterprises, France**
- **SKF, Göteborg, Sweden**
- **TLK Thermo, Germany**
- **Siemens Turbo, Sweden**
- **Sozhou Tongyuan, China**
- **Talent Swarm, Spain**
- **VTI, Linköping, Sweden**
- **VTT, Finland**
- **Wolfram MathCore, Sweden**

University members

- **FH Bielefeld, Bielefeld, Germany**
- **University of Bolivar, Colombia**
- **TU Braunschweig, Germany**
- **University of Calabria, Italy**
- **Univ California, Berkeley, USA**
- **Chalmers Univ, Control,Sweden**
- **Chalmers Univ, Machine, Sweden**
- **TU Darmstadt, Germany**
- **TU Delft, The Netherlands**
- **TU Dresden, Germany**
- **Université Laval, Canada**
- **Georgia Inst of Technology, USA**
- **Ghent University, Belgium**
- **Halmstad University, Sweden**
- **Heidelberg University, Germany**
- **TU Hamburg/Harburg Germany**
- **IIT Bombay, Mumbai, India**
- **KTH, Stockholm, Sweden**
- **Linköping University, Sweden**
- **Univ of Maryland, Syst Eng USA**
- **Univ of Maryland, CEEE, USA**
- **Politecnico di Milano, Italy**
- **Ecoles des Mines, CEP, France**
- **Mälardalen University, Sweden**
- **Univ Pisa, Italy**
- **Univ College SouthEast Norway**
- **Tsinghua Univ, Beijing, China**
- **Vanderbilt Univ, USA**

MODELICA

Spoken-Tutorial step-by-step OpenModelica and Modelica Tutorial Using OMEdit. Link from www.openmodelica.org

OpenModelica

Login · Create an account

HOME | DOWNLOAD | TOOLS & APPS | USE

Spoken Tutorial | Software Training | Creation | News | Forums | About | Statistics

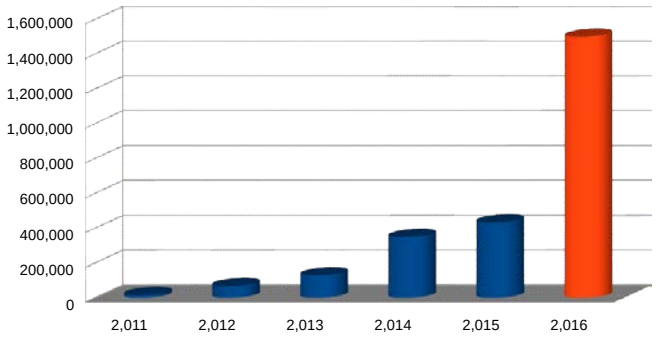
Search Tutorials

To learn about Modelica, read a book or a tutorial about Modelica®.
Interactive step-by-step beginners Modelica on-line spoken tutorials
Interactive OMWebbook with examples of Modelica textual modeling

nglish (804) | Submit | Reset dropdowns

OpenModelica is an open source modelling and simulation environment intended for industrial and academic usage.It is an object oriented declarative multi domain modelling language for complex systems. This environment can be used to work for both steady state as well as dynamic systems. Attractive strategy when dealing with design and optimization problems. As all the equations are solved simultaneously it doesn't matter whether the unknown variable in an input or output variable. Read more

Number of students/teachers trained in their colleges/schools



About **12** results found. Instruction Sheet

1. Introduction to OMEdit — Basic
Foss : *OpenModelica - English*
Outline: Introduction to OpenModelica Introduction to OMEdit Perspectives in OMEdit Browsers in OMEdit View icons in OMEdit Open a Class from Libraries Browser Checking for correctnes..

2. Examples through OMEdit — Basic
Foss : *OpenModelica - English*
Outline: Expand Modelica library Expand Electrical library Expand Analog library Open Rectifier Class Compare the values of IDC & Losses time vs Losses plot Expand Mechanics library ..

3. Developing an equation-based model — Basic
Foss : *OpenModelica - English*
Outline: Introduction to OMEdit Declaration of variables and equations Simulation of a model in

MODELICA

OMNotebook Interactive Electronic Notebook Here Used for Teaching Control Theory

OMNotebook: Kalman.onb

File Edit Cell Format Insert Window Help

1 Kalman Filter

Often we don't have access to the internal states of
have to reconstruct the state of the system based o
The idea with an observer is that we feedback the
the estiamtion is correct then the difference should

Another difficulty is that the measured quantities of

$$\begin{cases} \dot{\hat{x}} = \\ \end{cases}$$

Here are e denoting a disturbance in the input signa
be evaluated by the difference

$$K(y(t)$$

By using this quantity as feedback we obtain the ob

$$\dot{\hat{x}} = A\hat{x}(t) + Bu($$

Now form the error as

The differential error is

Ready

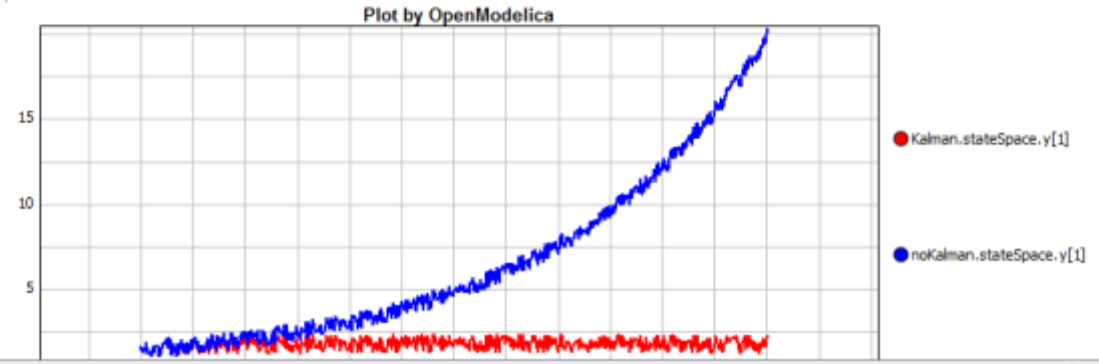
OMNotebook: Kalman.onb*

File Edit Cell Format Insert Window Help

```
model KalmanFeedback
  parameter Real A[:,size(A, 1)] = {{0,1},{1,0}} ;
  parameter Real B[size(A, 1),:] = {{0},{1}};
  parameter Real C[:,size(A, 1)] = {{1,0}};
  parameter Real[2,1] K = [2.4;3.4];
  parameter Real[1,2] L = [2.4,3.4];
  parameter Real[:,:] ABL = A-B*L;
  parameter Real[:,:] BL = B*L;
  parameter Real[:,:] Z = zeros(size(ABL,2),size(AKC,1));
  parameter Real[:,:] AKC = A-K*C;
  parameter Real[:,:] Anew = [0,1,0,0 ; -1.4, -3.4, 2.4,3.4; 0,0,-2.4,1;0,0,-2.4,0];
  parameter Real[:,:] Bnew = [0;1;0;0];
  parameter Real[:,:] Fnew = [1;0;0;0];
  stateSpaceNoise Kalman(stateSpace.A=Anew,stateSpace.B=Bnew,  stateSpace.C=[1,0,0,0],
stateSpace.F = Fnew);
  stateSpaceNoise noKalman;
end KalmanFeedback;
```

```
simulate(KalmanFeedback,stopTime=3)
plot({Kalman.stateSpace.y[1],noKalman.stateSpace.y[1]})
```

true

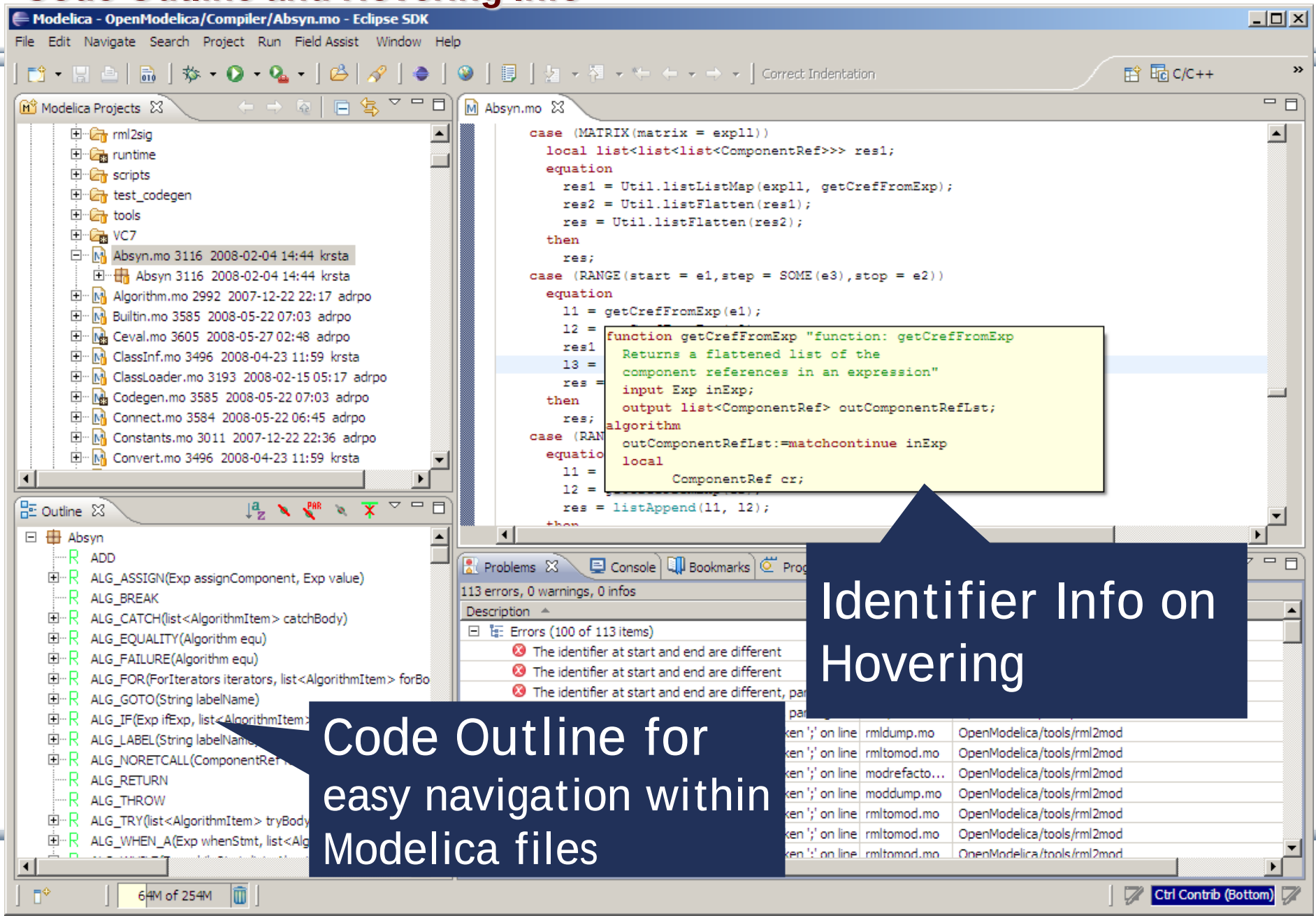


Ready Ln 12, Col 39

MODELICA

OpenModelica MDT Eclipse Plug-in:

Code Outline and Hovering Info



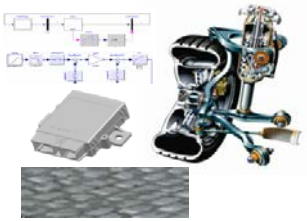
General Tool Interoperability & Model Exchange Functional Mock-up Interface (FMI)

Engine with ECU

Gearbox with ECU

Thermal systems

Automated cargo door



Chassis components, roadway, ECU (e.g. ESP)

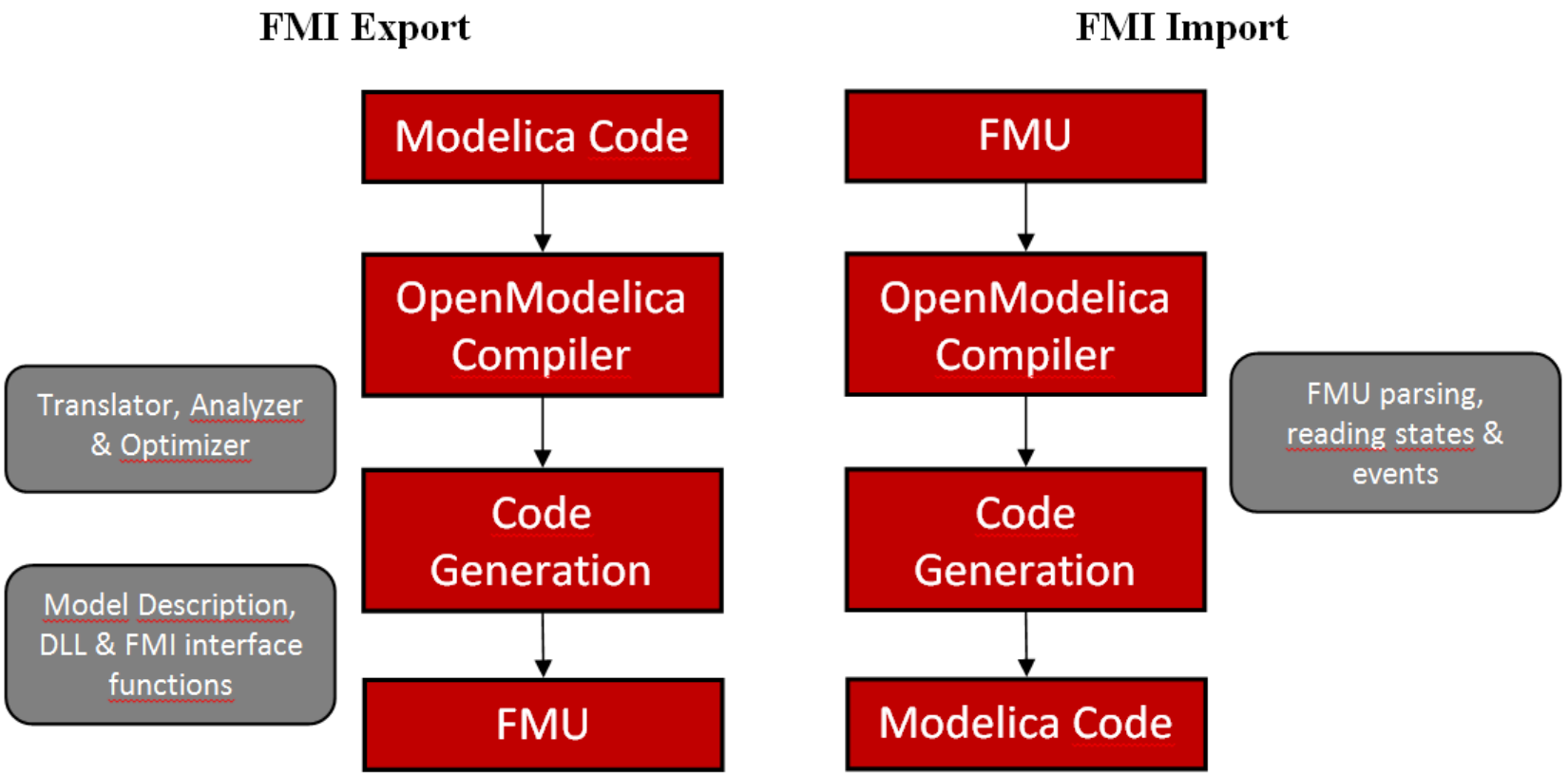
etc.

functional mockup interface for model exchange and tool coupling

courtesy Daimler

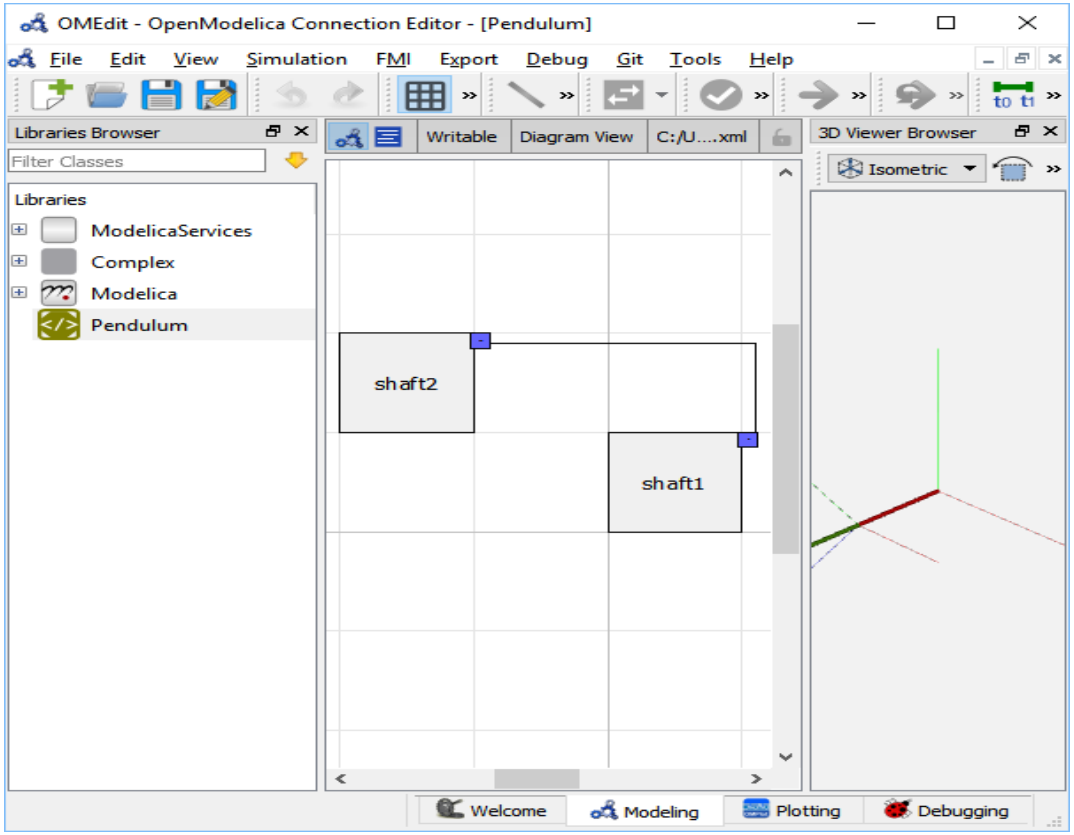
- FMI development was started by ITEA2 MODELISAR project. FMI is a Modelica Association Project now
- **Version 1.0**
- FMI for Model Exchange (released Jan 26,2010)
- FMI for Co-Simulation (released Oct 12,2010)
- **Version 2.0**
- FMI for Model Exchange and Co-Simulation (released July 25,2014)
- **> 100 tools** supporting it (https://www.fmi-standard.org/tools)

OpenModelica Functional Mockup Interface (FMI)



OMSimulator Composite Model Editor with 3D Viewer

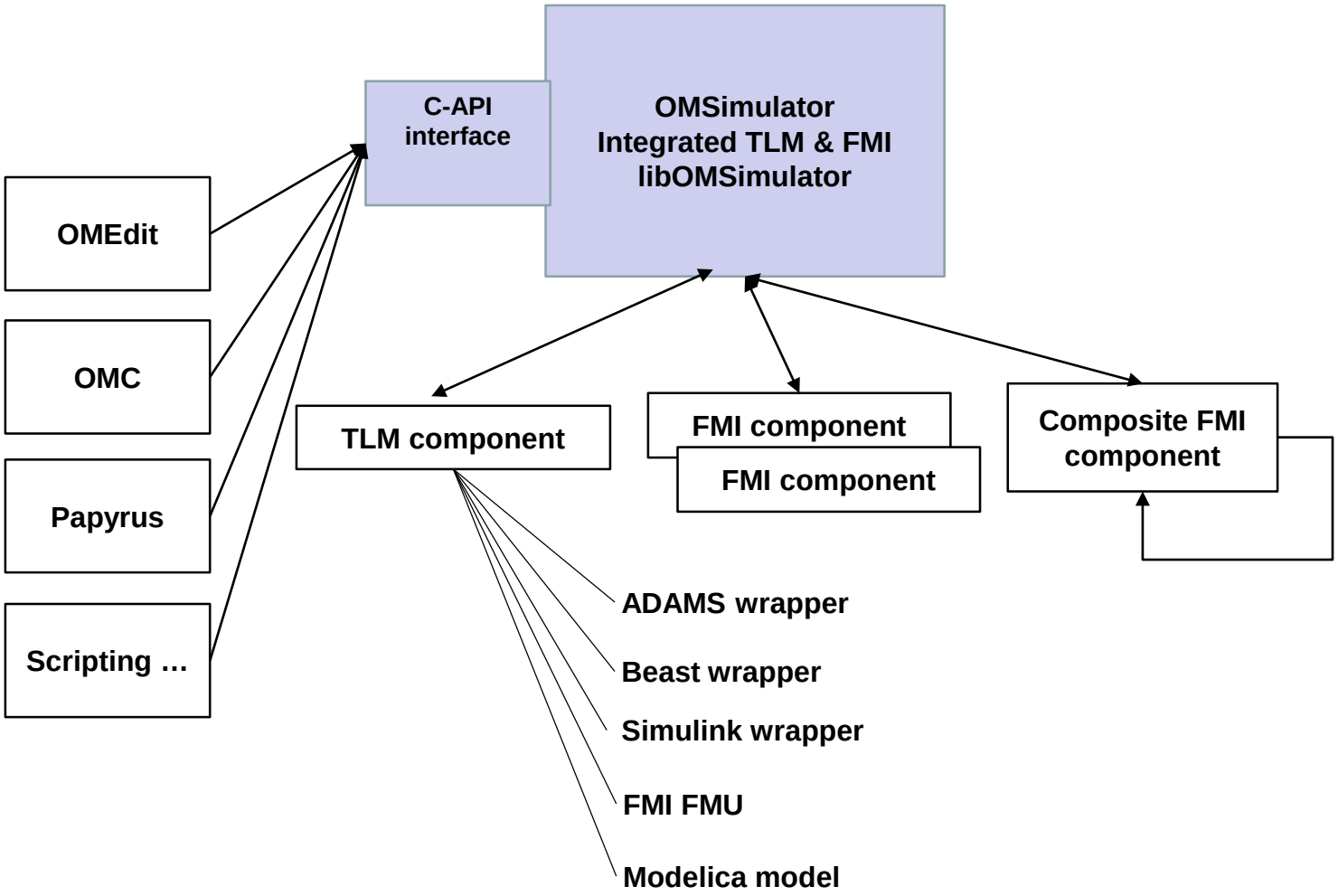
Combine External (FMI) Models into New Models



- **Composite model editor** with 3D visualization of connected mechanical model components which can be FMUs, Modelica models, etc., or co-simulated components
- **3D animation** possible
- Composite model saved as **XML**-file

MODELICA

OMSimulator – Integrated FMI and TLM-based Cosimulator/Simulator in OpenModelica

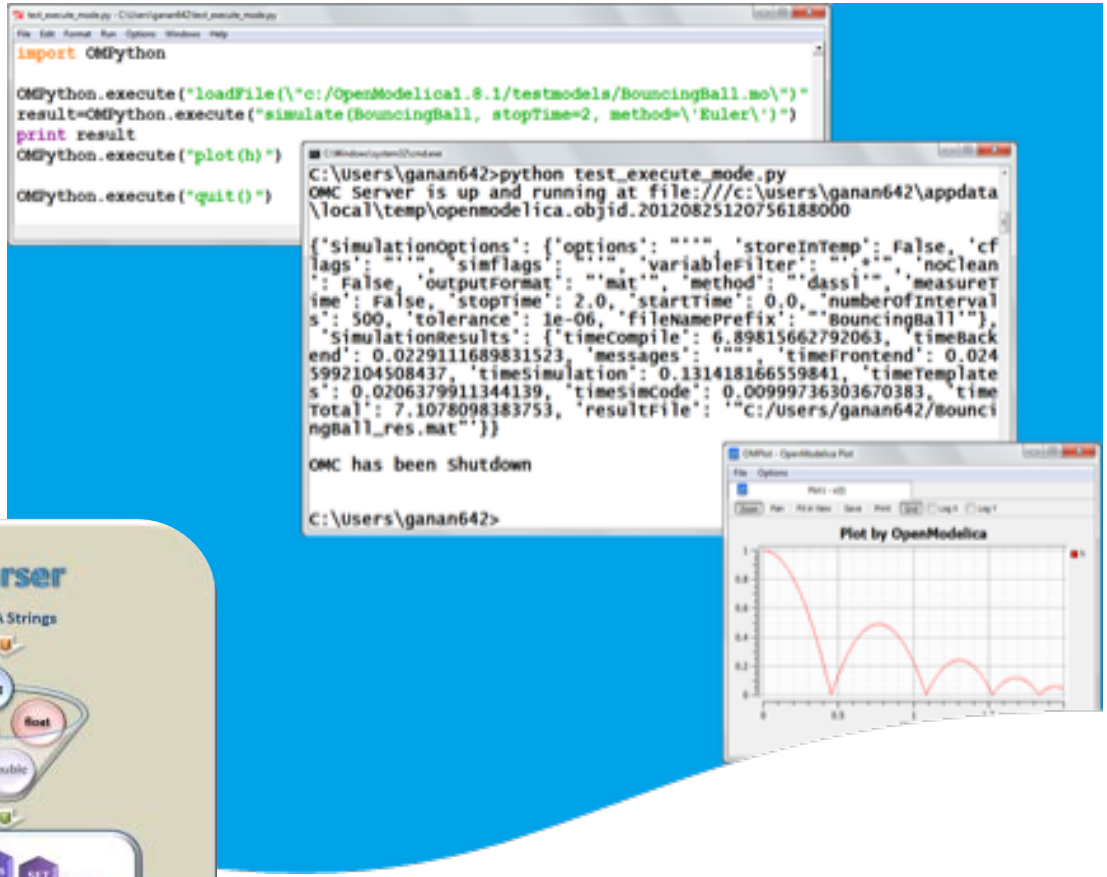


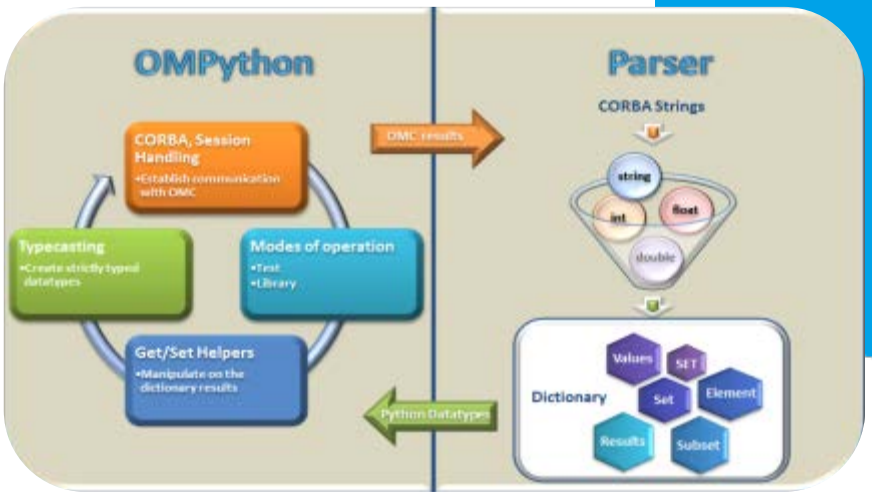
OMSens – Sensitivity Analysis Subsystem

- ??? Fill in
- Under development, prototype available

OMPython – Python Scripting with OpenModelica

- Interpretation of Modelica commands and expressions
- Interactive Session handling
- Library / Tool
- Optimized Parser results
- Helper functions
- Deployable, Extensible and Distributable





MODELICA

OMJulia – Julia Scripting with OpenModelica

- Interpretation of Modelica commands and expressions from Julia, transfer of data
- Control design using Julia control package together with OpenModelica
- Interactive Session handling
- Library / Tool
- Separately downloadable. be run with OpenModelica 1.13.0 nightly build
- Works with Jupyter notebooks
- See separate presentation

Control example with OMJulia in Jupyter notebooks

Use of Modelica + Julia in Process Systems Engineering Education

Complex models of "Seborg reactor"

Bernt Lie*, Arunkumar Palanisamy, Peter Fritzson****

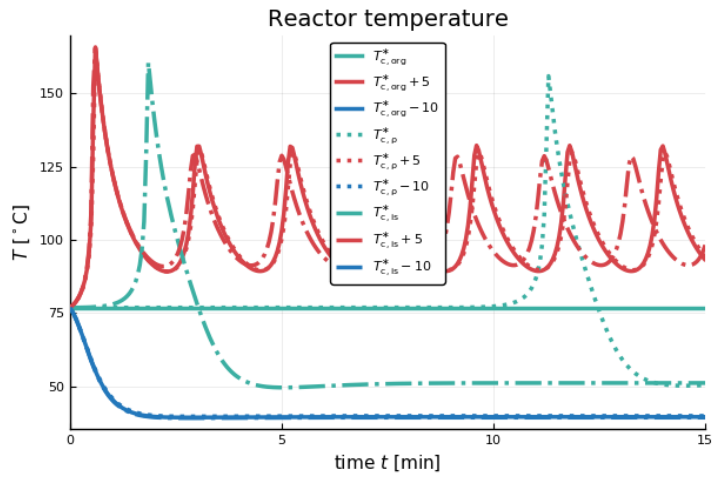
***University of South-Eastern Norway, Norway**

****University of Linköping, Sweden**

Introducing packages

In [1]:
```
# Pkg.add("Plots") -- we assume that this step already has been carried out
using Plots; pyplot()
using LaTeXStrings
using DataFrames
using OMJulia
#using DifferentialEquations
```

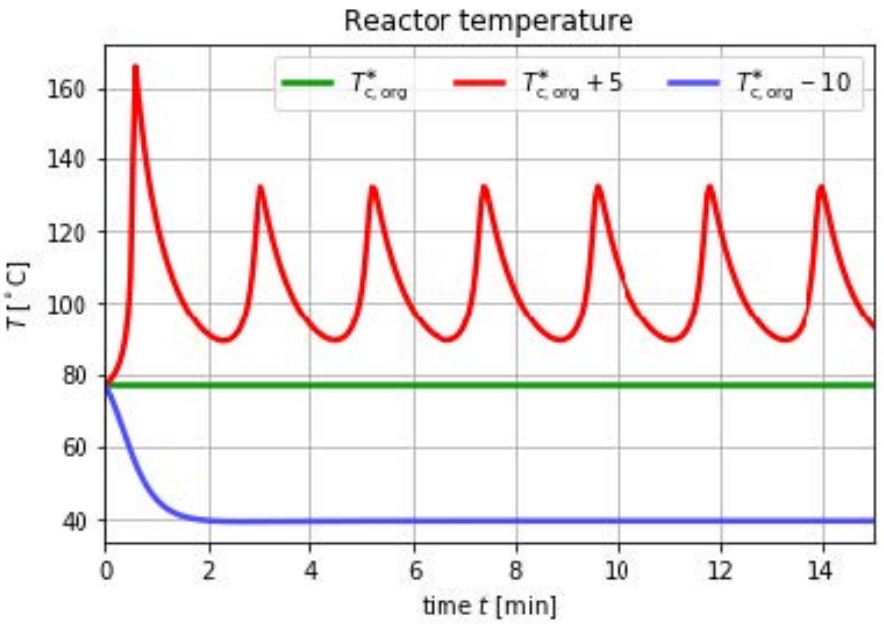
Out[9]:



In [10]: figfile = "tempSeborgCSTR.pdf"

OMMatlab – Matlab Scripting with OpenModelica

- Interpretation of Modelica commands and expressions from Matlab, transfer of data
- Interactive Session handling
- Library / Tool
- Separately downloadable. be run with OpenModelica 1.13.0 nightly build
- Now (October 2018) basic version supporting basic simulation and plotting



Embedded System Support in OpenModelica

- Code generation of real-time Controllers from Modelica models for small foot-print platforms

Use Case: SBHS (Single Board Heating System)

Single board heating system (IIT Bombay)

- Use for teaching basic control theory
- Usually controlled by serial port (set fan value, read temperature, etc)
- OpenModelica can generate code targeting the ATmega16 on the board (AVR-ISP programmer in the lower left).

  Program size is 4090 bytes including LCD driver and PID-controller (out of 16 kB flash memory available).

Movie Demo!

Example – Code Generation to SHBS

Code Generator Comparison, Full vs Simple

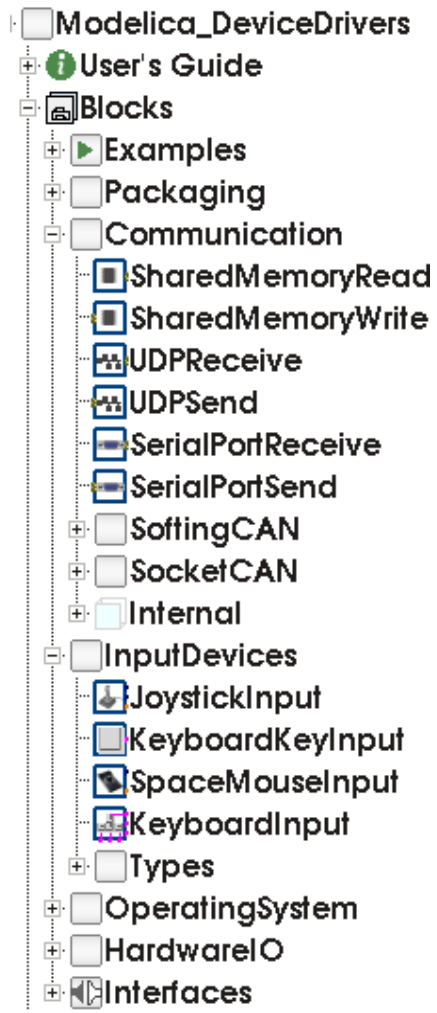
	Full Source-code FMU targeting 8-bit AVR proc	**Simple code generator targeting 8-bit AVR proc**
Hello World (0 equations)	43 kB flash memory 23 kB variables (RAM)	130 B flash memory 0 B variables (RAM)
SBHS Board (real-time PID controller, LCD, etc)	**68 kB** flash memory **25 kB** variables (RAM)	**4090 B** flash memory **151 B** variables (RAM)

The largest 8-bit AVR processor MCUs (Micro Controller Units) have 16 kB SRAM.

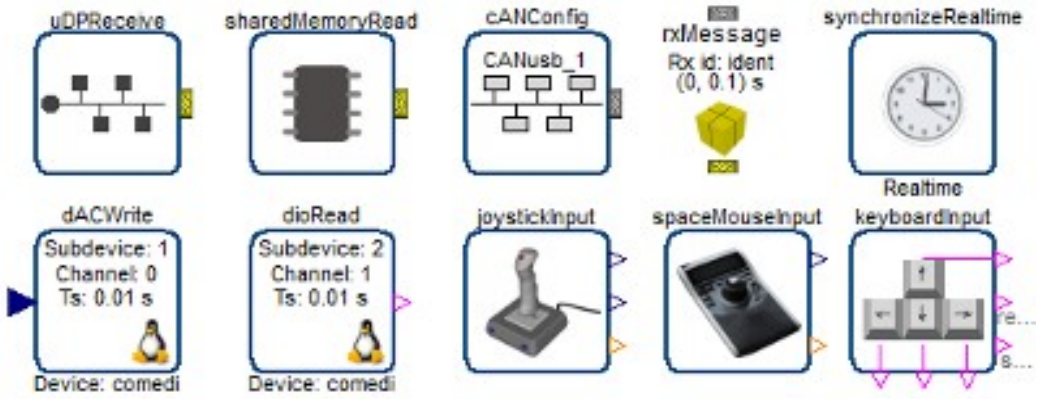
One of the more (ATmega328p; Arduino Uno) has 2 kB SRAM.

The ATmega16 we target has **1 kB SRAM available** (stack, heap, and global variables)

Communication & I/O Devices:
Modelica_DeviceDrivers Library



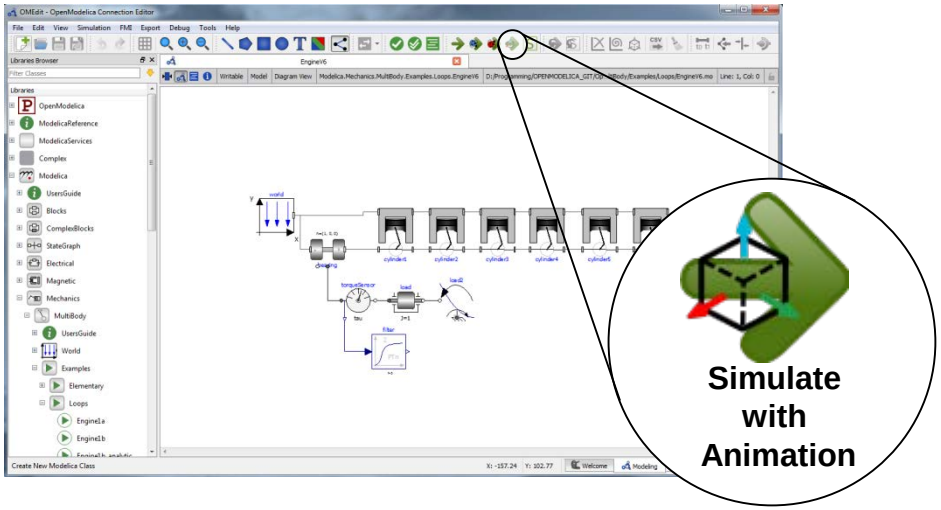
- **Free library** for interfacing hardware drivers
- **Cross-platform** (Windows and Linux)
- UDP, SharedMemory, CAN, Keyboard, Joystick/Gamepad
- DAQ cards for digital and analog IO (only Linux)
- Developed for **interactive real-time** simulations

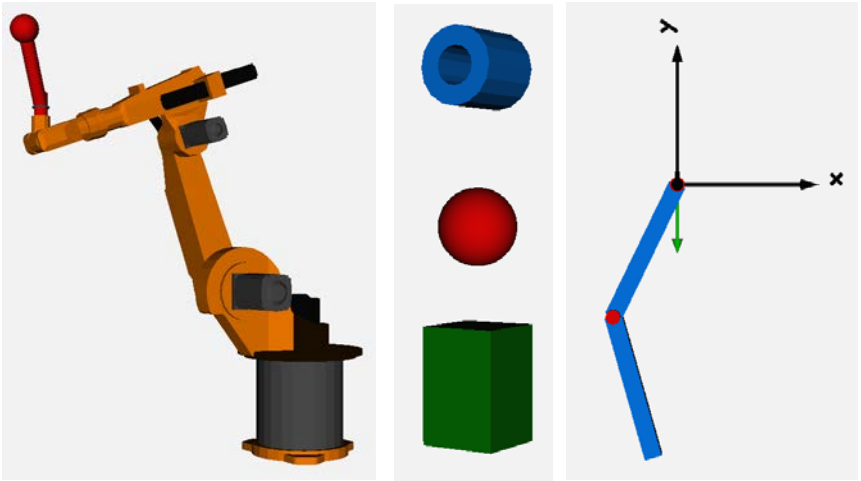


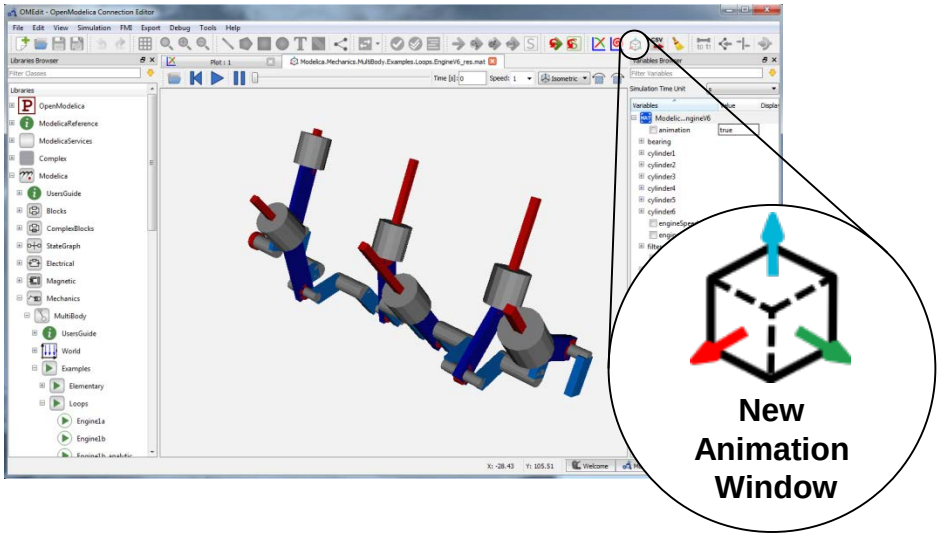
MODELICA

OMEdit 3D Visualization of Multi-Body Systems

- Built-in feature of OMEdit to animate MSL-Multi-Body shapes
- Visualization of simulation results
- Animation of geometric primitives and CAD-Files

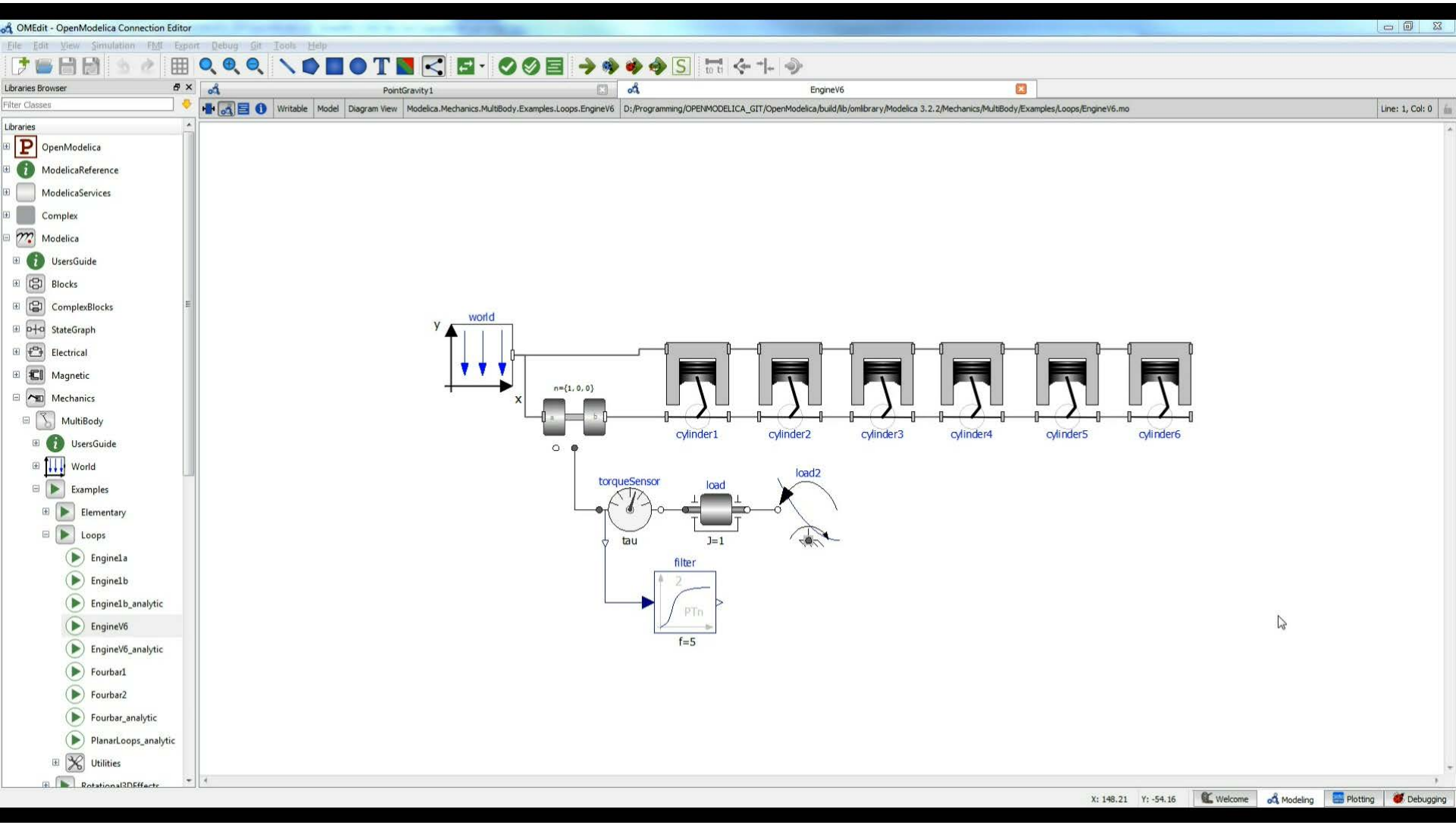






MODELICA

OpenModelica 3D Animation Demo



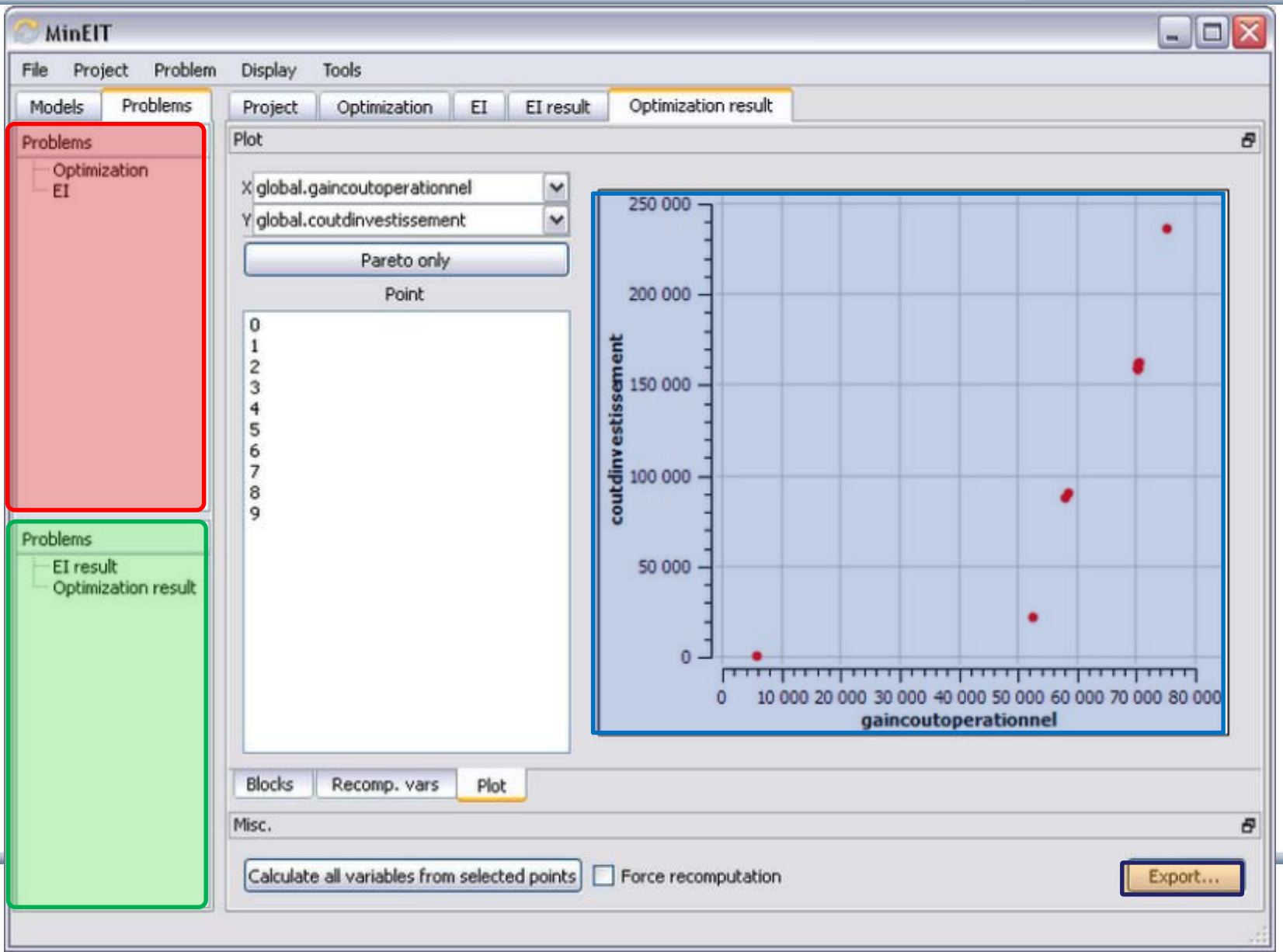
OMOptim – Parameter Sweep Design Optimization

Problems

Solved problems

Result plot

Export result data .csv



Here Pareto front optimiza-tion

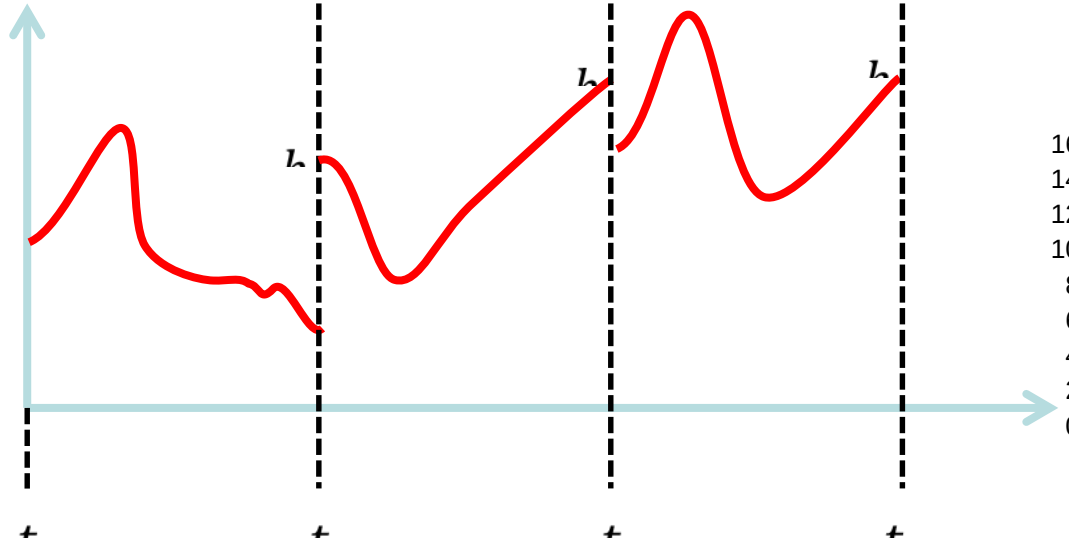
MODELICA

Optimization of Dynamic Trajectories Using Multiple-Shooting and Collocation

- Minimize a goal function subject to model equation constraints, useful e.g. for NMPC
- Multiple Shooting/Collocation
  - Solve sub-problem in each sub-interval

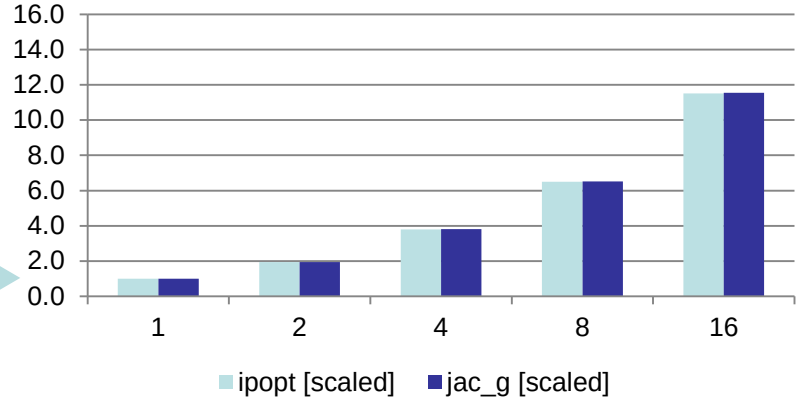
In OpenModelica 1.9.1 beta release Jan 2014.

$$x_i(t_{i+1}) = h_i + \int_{t_i}^{t_{i+1}} f(x_i(t), u(t), t)\, dt \approx F(t_i, t_{i+1}, h_i, u_i), \qquad x_i(t_i) = h_i$$



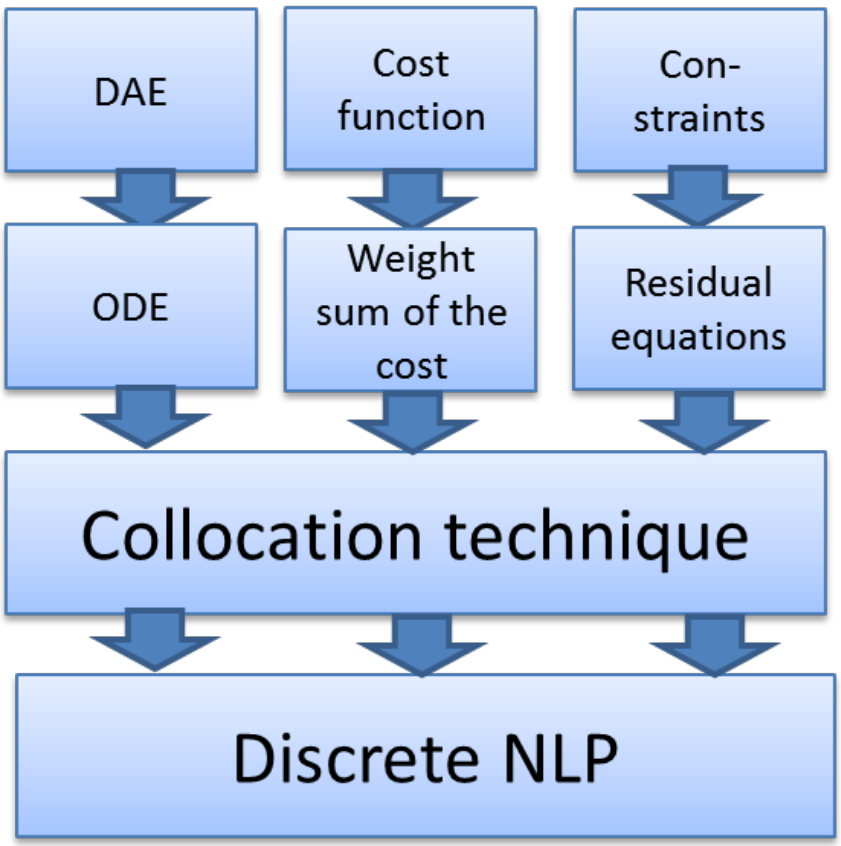
Example speedup, 16 cores:

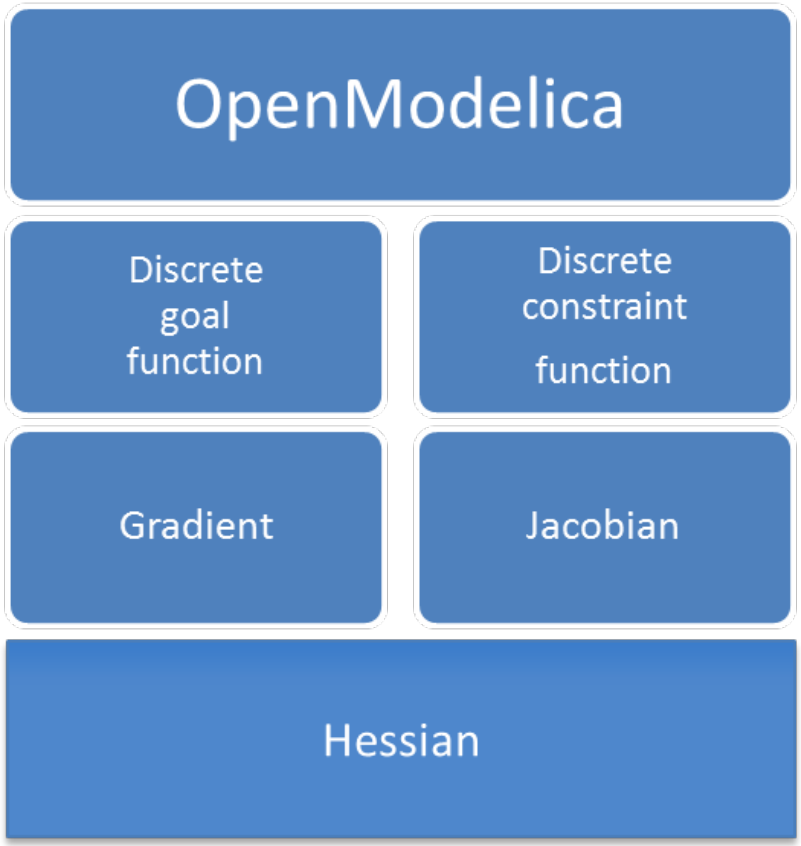
MULTIPLE_COLLOCATION



MODELICA

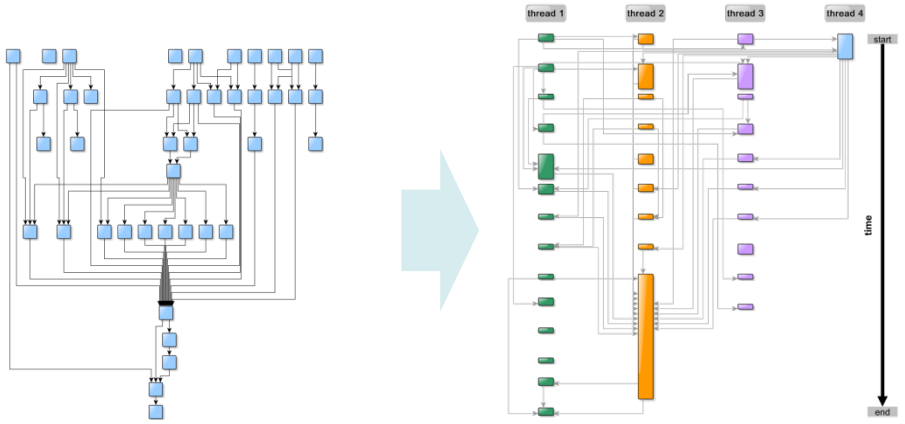
OpenModelica Dynamic Optimization Collocation

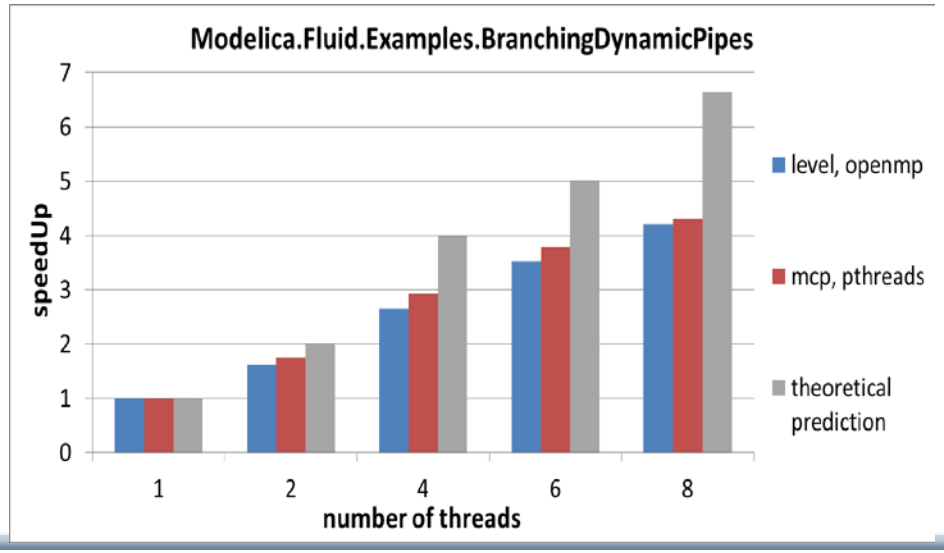




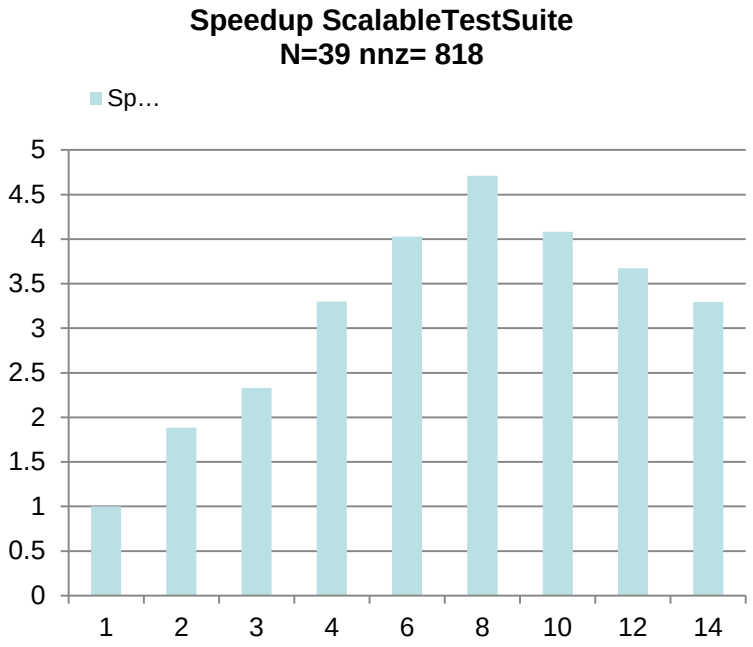
OpenModelica Model Parallelization
Faster Simulation on Multi-Core

Automated parallelization of models





Parallelizing numeric Jacobian computations in simulation



Speedup about 4 using 8 threads

MODELICA

Large-scale ABB OpenModelica Application
Generate code for controlling 7.5 to 10% of German Power Production

ABB OPTIMAX PowerFit

- Real-time optimizing control of large-scale virtual power plant for system integration
- **Software including OpenModelica** now used in managing more than 2500 renewable plants, total up to 1.5 GW

High scalability supporting growth

- 2012: initial delivery (for 50 plants)
- 2013: SW extension (500 plants)
- 2014: HW+SW extension (> 2000)
- 2015: HW+SW extension,
  incl. OpenModelica generating optimizing controller code in FMI 2.0 form

Manage 7.5% - 10% of German Power

- 2015, Aug: OpenModelica Exports FMUs for real-time optimizing control (seconds) of about **5.000 MW (7.5%) of power in Germany**

MODELICA

Need for Debugging Tools
Map Low vs High Abstraction Level

- A **major part** of the total **cost** of software projects is due to testing and debugging
- US-Study 2002:
  Software errors cost the US economy **annually~ 60 Billion $**

- **Problem: Large Gap in Abstraction Level**
  from **Equations** to **Executable Code**
- Example error message (hard to understand)

  Error solving nonlinear system 132

  time = 0.002

  residual[0] = 0.288956

  x[0] = 1.105149

  residual[1] = 17.000400

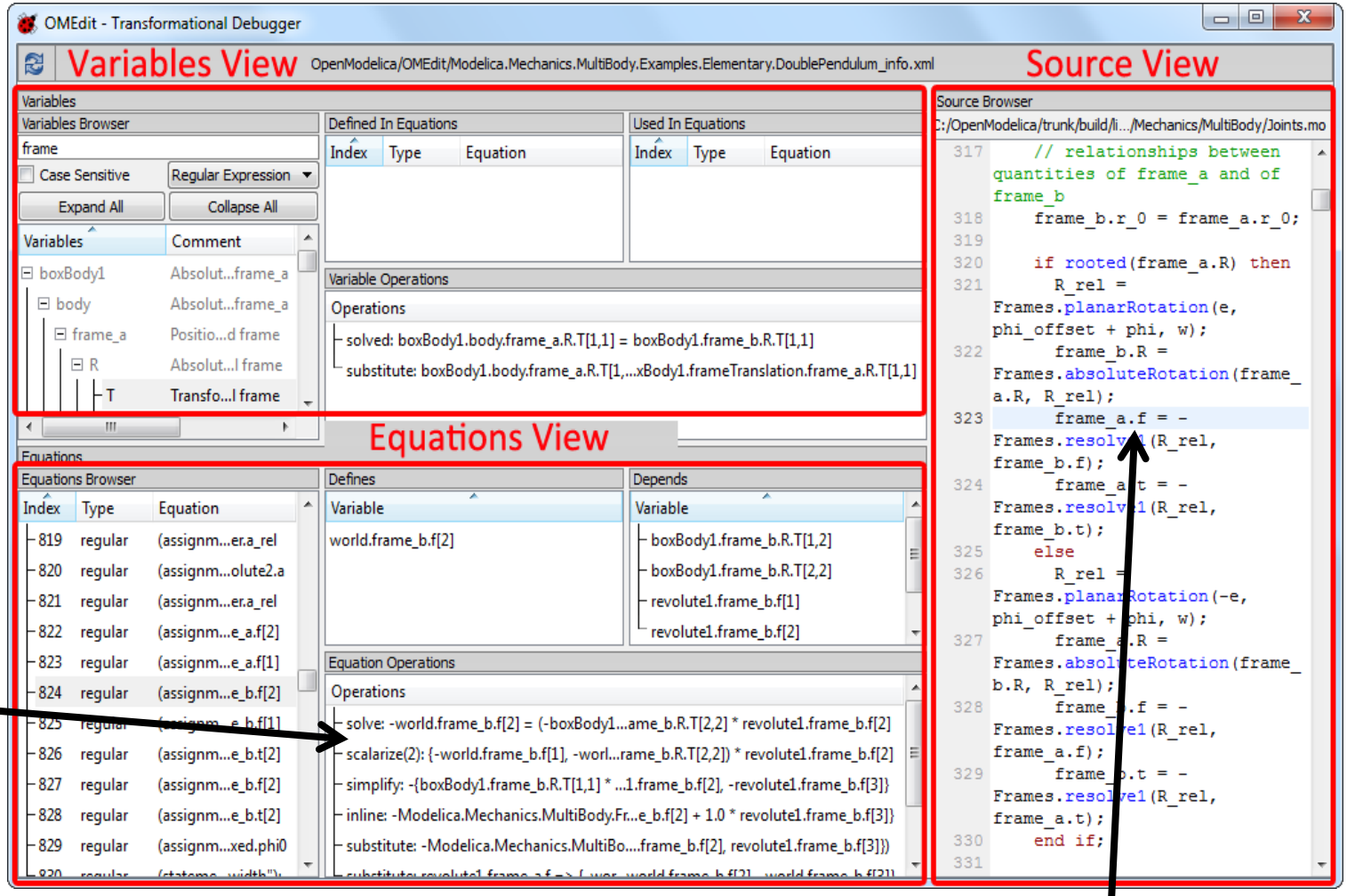
  x[1] = 1.248448

  ...

Integrated Static-Dynamic OpenModelica Equation Model Debugger

Efficient handling of **Large** Equation Systems

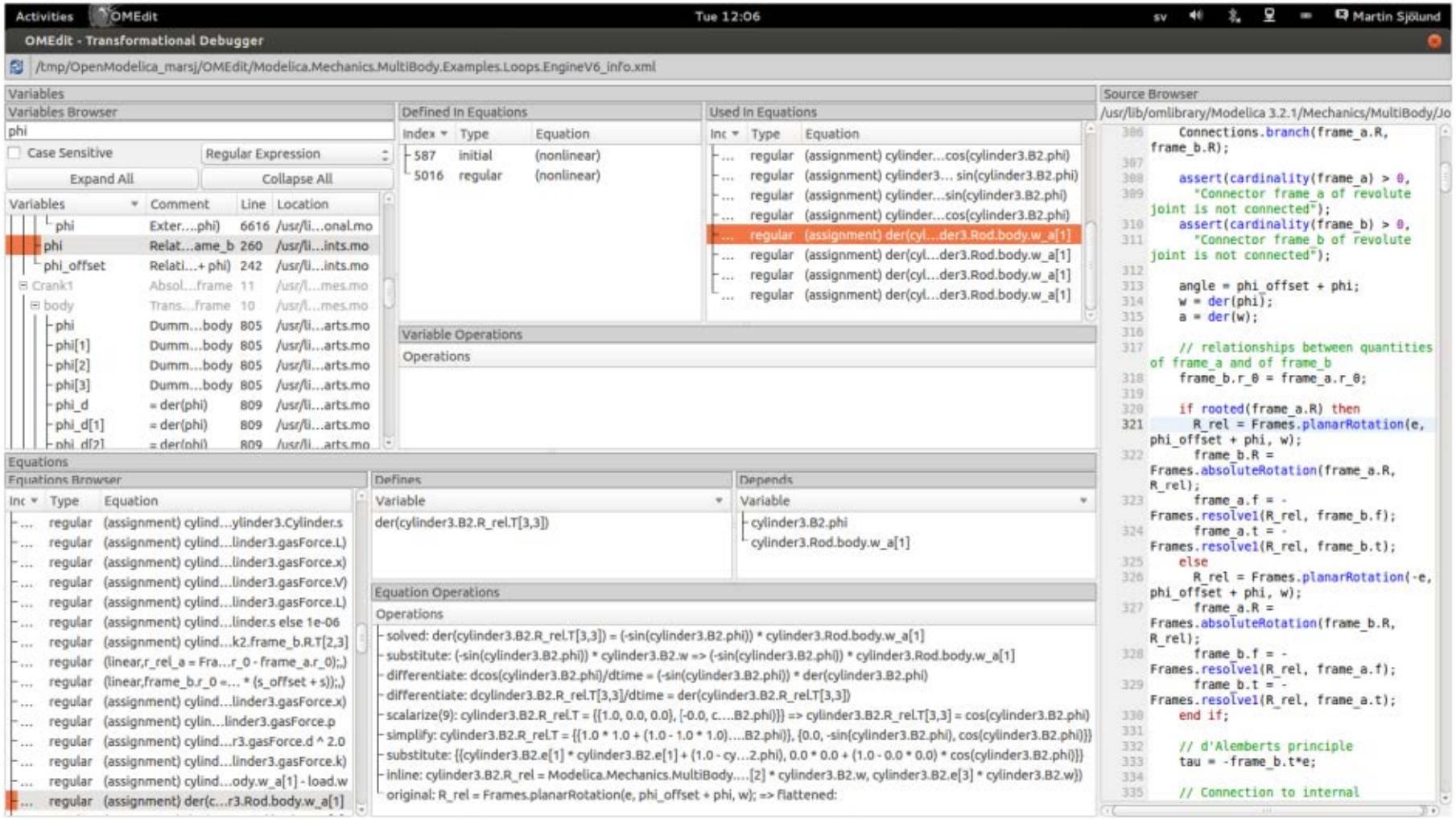
Showing equation transfor mations of a model:



Mapping dynamic run-time error to source model position

MODELICA

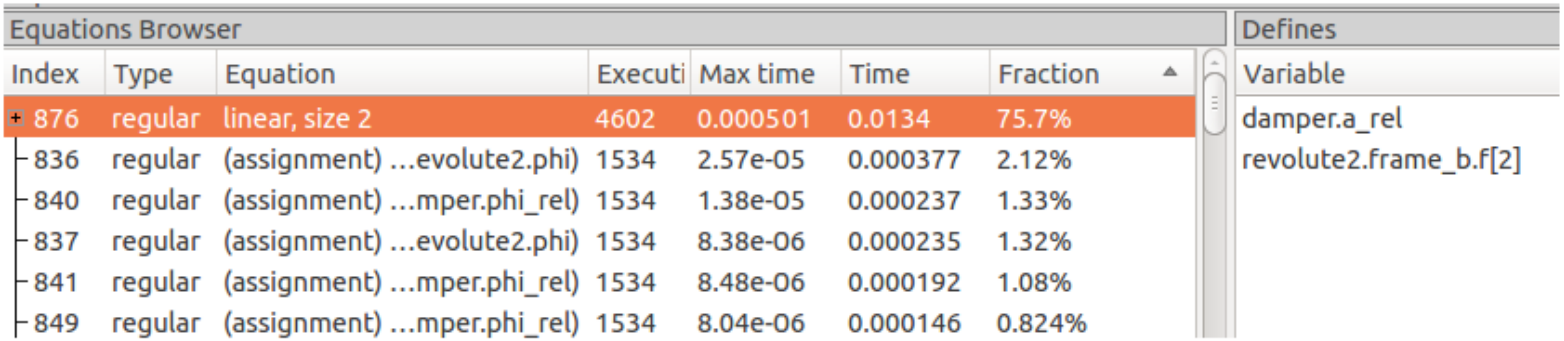
Transformations Browser – EngineV6 Overview (11 116 equations in model)



MODELICA

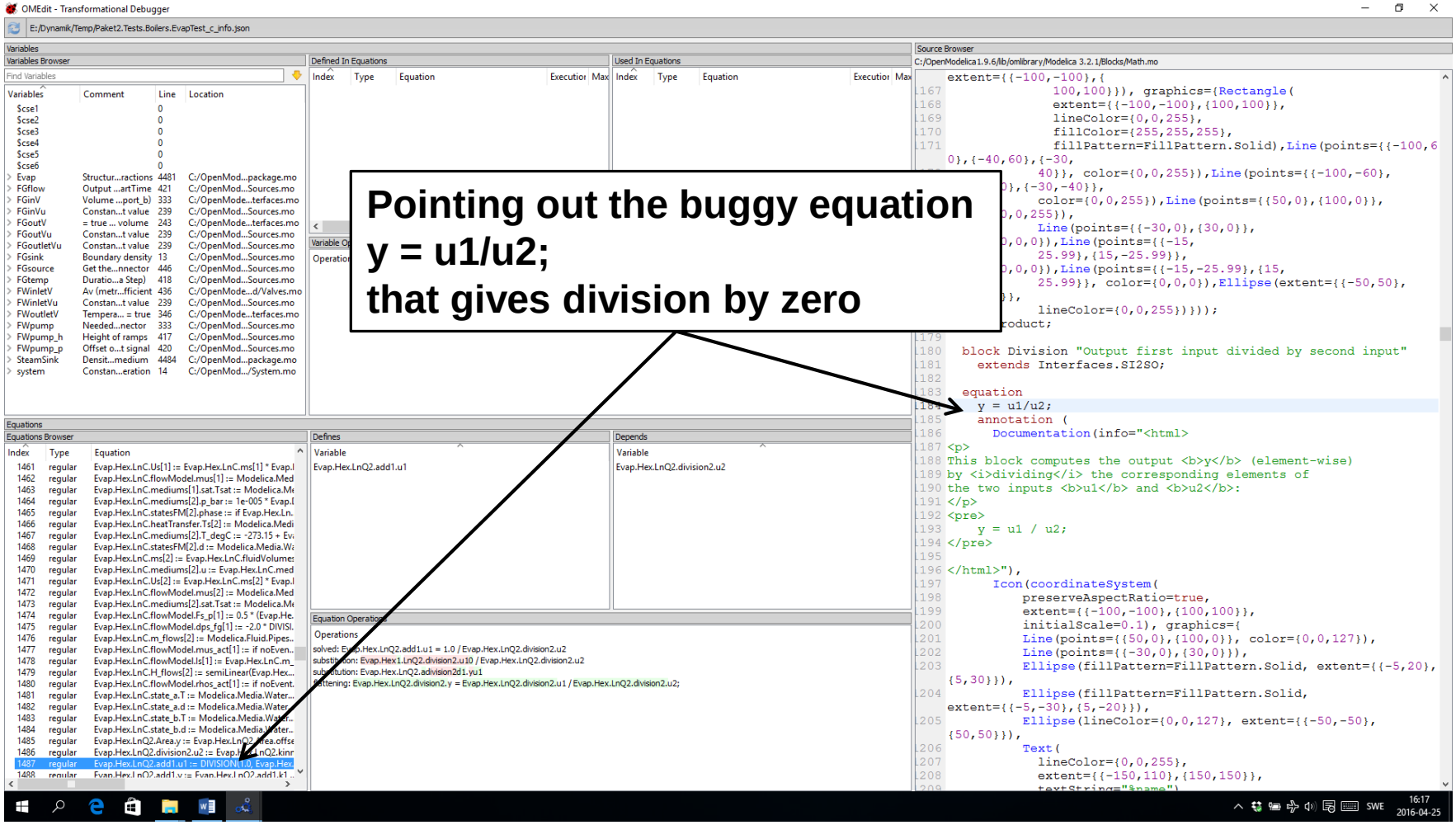
Performance Profiling
(Here: Profiling all equations in MSL 3.2.1 DoublePendulum)

- Measuring performance of equation blocks to find bottlenecks
  - Useful as input before model simplification for real-time platforms
- Integrated with the debugger so it is possible to show what the slow equations compute
- Suitable for real-time profiling (less information), or a complete view of all equation blocks and function calls



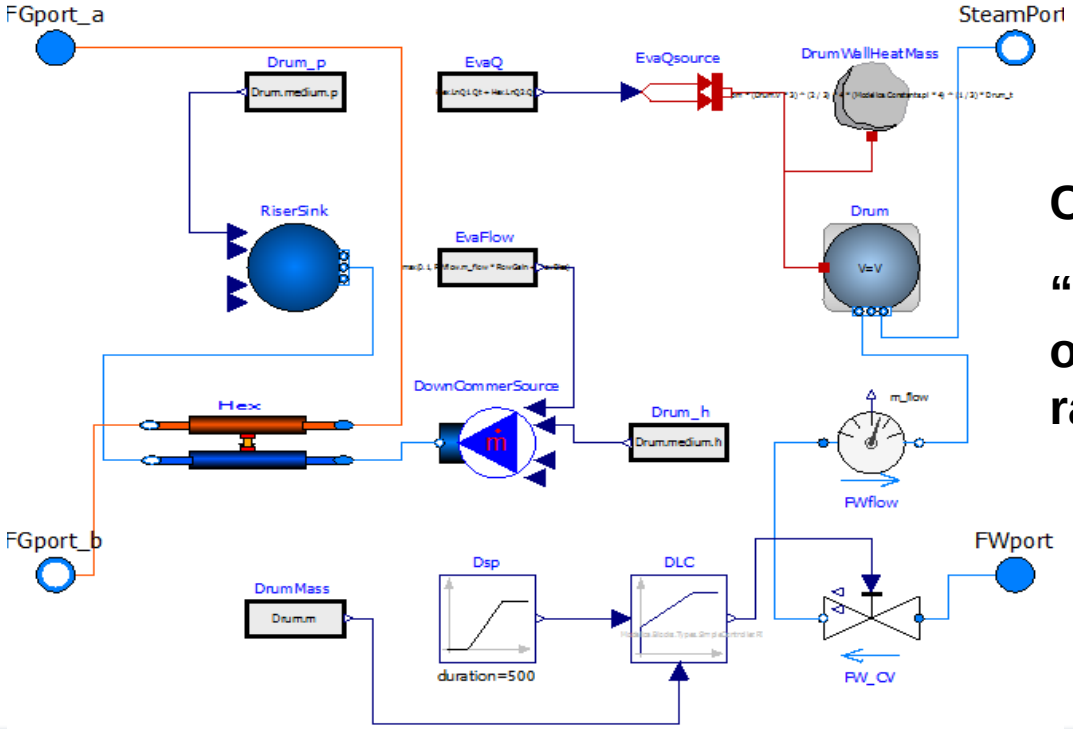
Equation Model Debugger on Siemens Model

(Siemens Evaporator test model, 1100 equations)



Performance Profiling for faster Simulation
(Here: Profiling equations of Siemens Drum boiler model with evaporator

- Measuring **performance** of equation blocks to find bottlenecks
  - Useful as input before model simplification for real-time applications
- Integrated with the debugger to **point out the slow equations**
- Suitable **for real-time profiling** (collect less information), or a complete view of all equation blocks and function calls



Conclusion from the evaluation:

"…the profiler makes the process of performance optimization radically shorter."

MODELICA

OpenModelica – ModelicaML UML Profile
Based on Open-Source Papyrus UML and OpenModelica

- ModelicaML is a UML Profile for SW/HW modeling
  - Applicable to "pure" UML or to other UML profiles, e.g. SysML
- Standardized Mapping UML/SysML to Modelica
  - Defines transformation/mapping for **executable** models
  - Being **standardized** by OMG
- ModelicaML
  - Defines graphical concrete syntax (graphical notation for diagram) for representing Modelica constructs integrated with UML
  - Includes graphical formalisms (e.g. State Machines, Activities, Requirements)
    - Which do not yet exist in Modelica language (extension work ongoing)
    - Which are translated into executable Modelica code
  - Is defined towards generation of executable Modelica code
  - Current implementation based on the Papyrus UML tool + OpenModelica

Example: Simulation and Requirements Evaluation

«model»
(TwoTanksSystemExample::SystemSimulations)
TankSystemSimulation

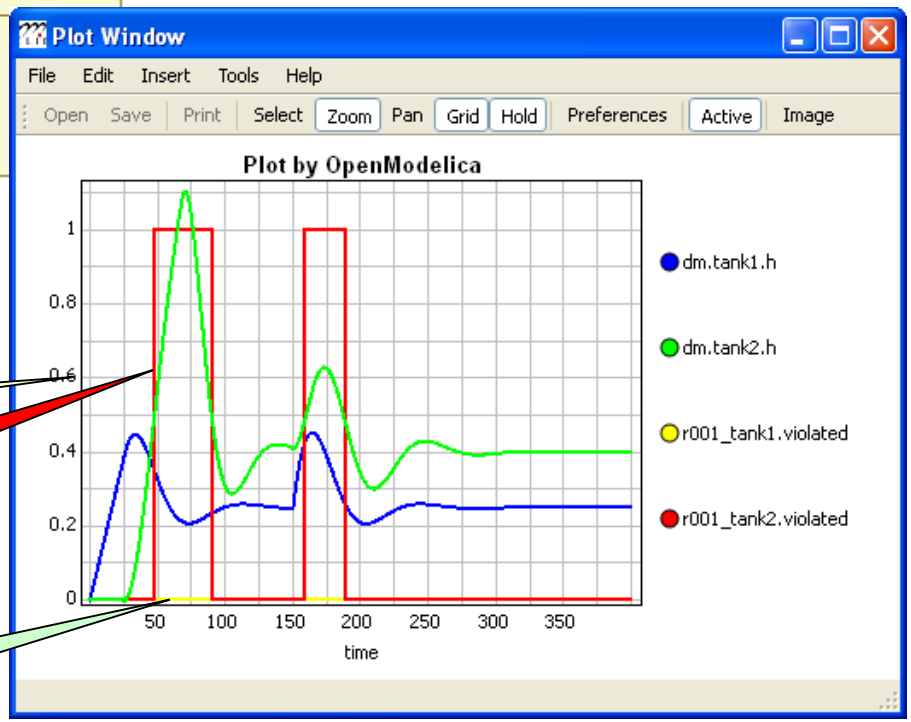
«component» dm: TanksConnectedPI
«requirementInstance» r001_tank1: Max level of liquid in a tank
«requirementInstance» r001_tank2: Max level of liquid in a tank
«requirementInstance» r002_tank1: Volume of the tank1

Req. 001 is instantiated 2 times (there are 2 tanks in the system)

tank-height is 0.6m

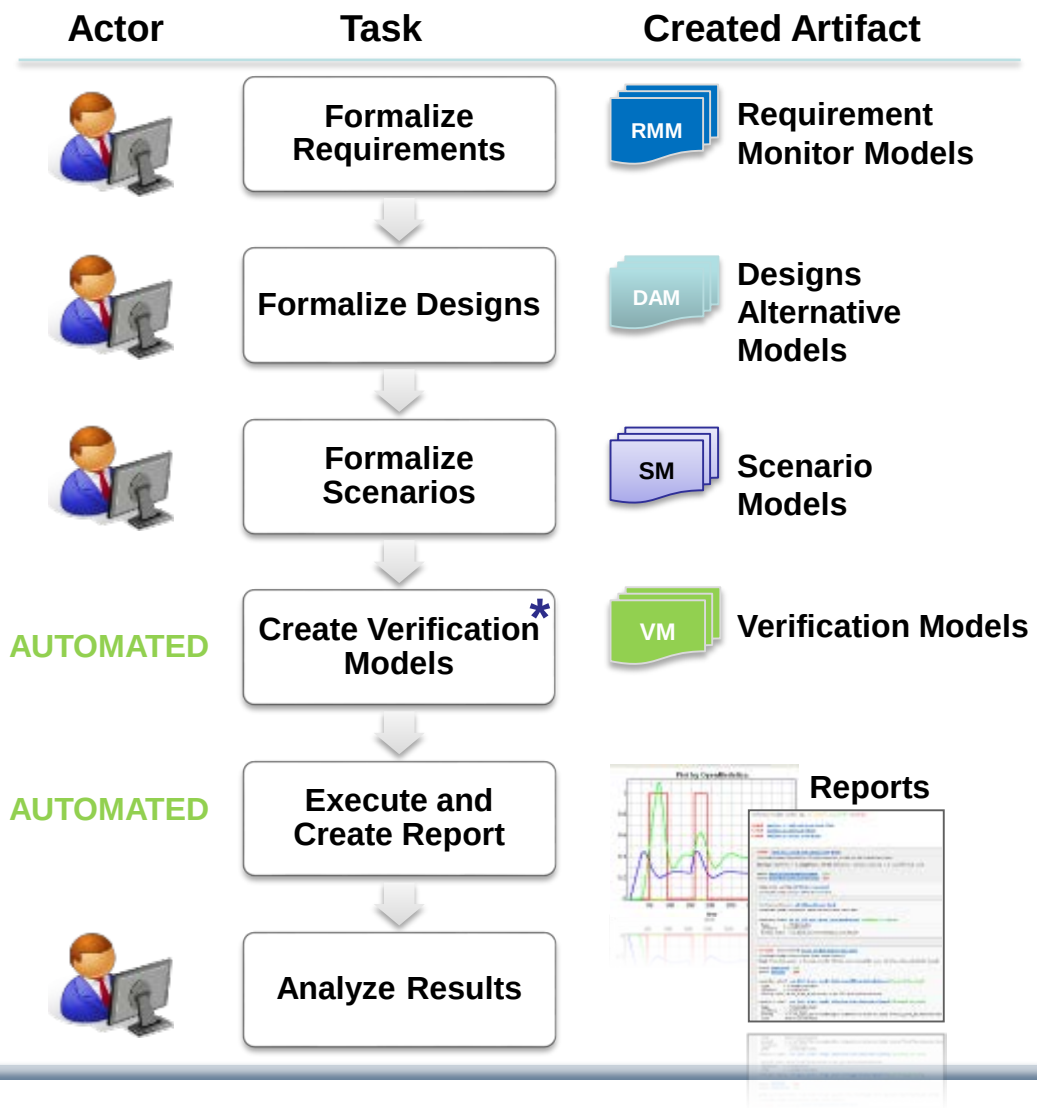
Req. 001 for the tank2 is violated

Req. 001 for the tank1 is not violated



MODELICA

vVDR Method – virtual Verification of Designs vs Requirements

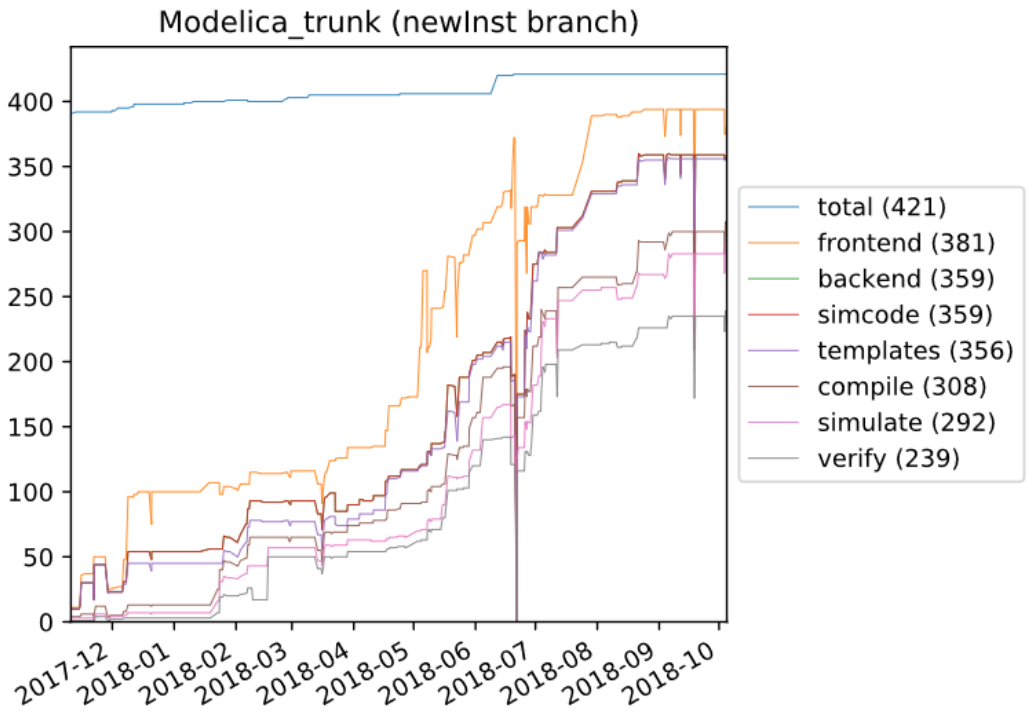


Goal: Enable on-demand verification of designs against requirements using automated model composition at any time during development.

MODELICA

Outlook:
New OpenModelica Frontend for Large-Scale models

- Soon: New OMC Compiler frontend for fast compilation and large-scale models
- Been under development the past 2-3 years
- Now (Oct 05) simulates 69% of MSL models, coverage increases about 6% per month
- About 10-200 times faster than the old frontend, depending on model



OpenModelica DAEMode for Large-Scale models

- Goal – to handle hundreds of thousands to millions of equations
- Introduced sparse solvers in the solution chain:
  - KLU for linear algebraic equations,
  - Kinsol for nonlinear algebraic equations, and
  - IDA for sparse differential-algebraic equations.
- DAEmode: after index reduction, IDA solves the differential equations and the algebraic loops simultaneously
- Largest system so far: electro-mechanical power system model with about 600.000 differential-algebraic equations
- Under development for even larger systems

Summary and Questions

Multi-Domain Modeling

Visual Acausal Component Modeling

MODELICA

www.modelica.org – Language, Standard Library

www.openmodelica.org – Open Source Tool

Typed Declarative Textual Language

Thanks for listening!

Hybrid Modeling