

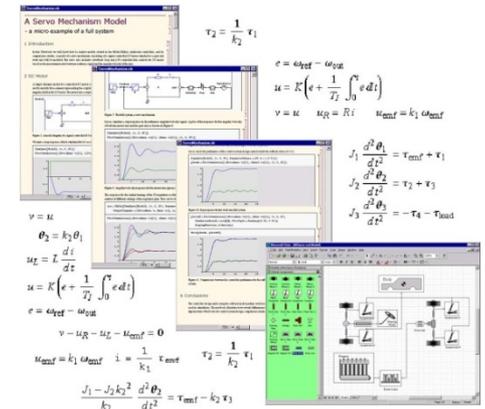
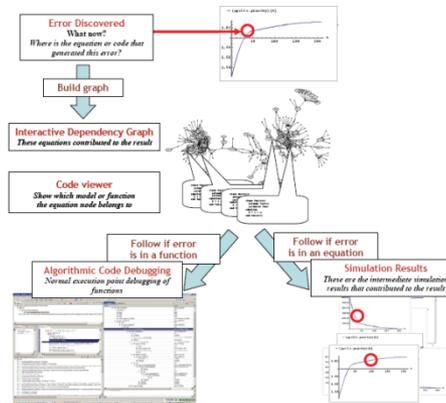
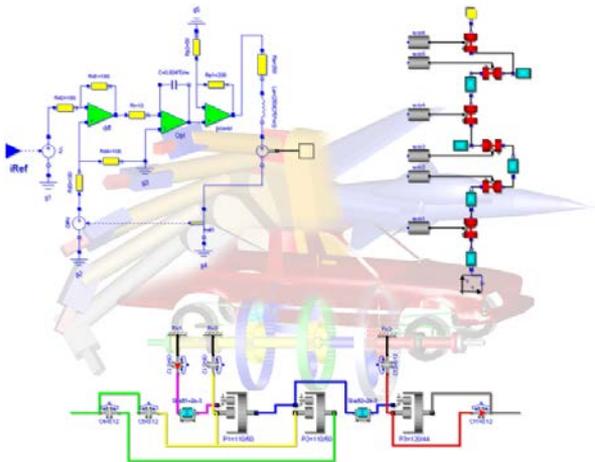
OpenModelica.org

Presentation, Status and Future Developments

Adrian.Pop@liu.se

2020-02-03

Open Source Modelica Consortium
PELAB, Linköping University
RISE, Research Institutes of Sweden



- OpenModelica
 - What is OpenModelica?
 - The past
- OpenModelica Technical Overview
 - OMC, OMShell, OMNotebook, OMEdit, ModelicaML, OMSimulator, OMPython, OMJulia, OMMatlab
- OpenModelica Development Environment
 - MetaModelica
 - The Eclipse Environment (MDT)
- OpenModelica Latest Developments (2019-2020)

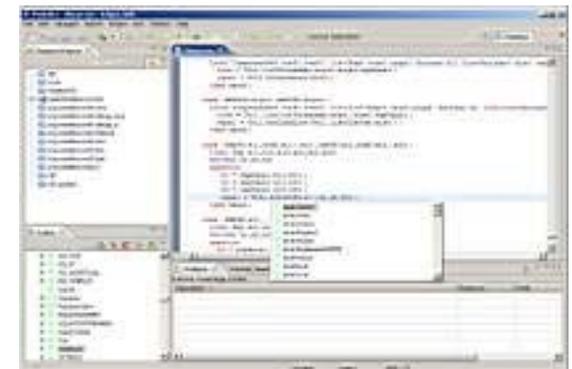
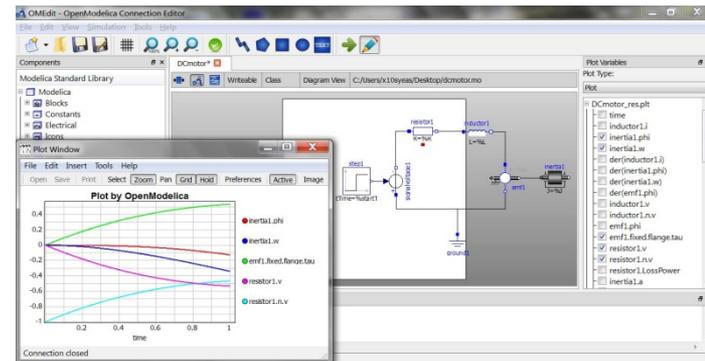
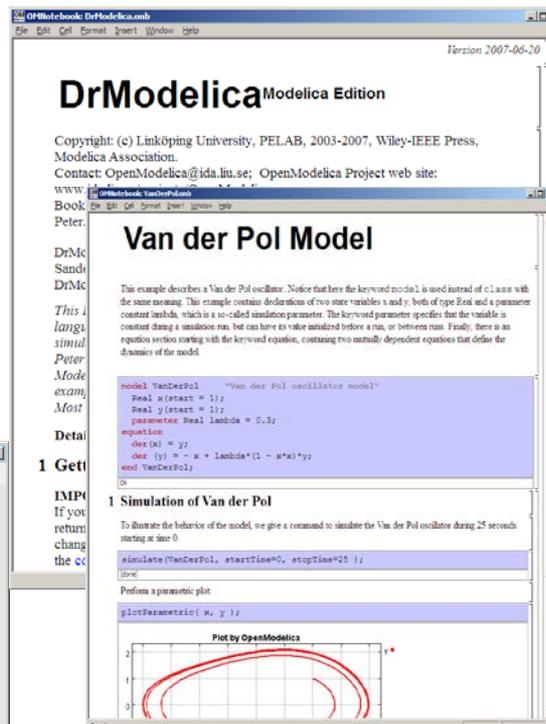
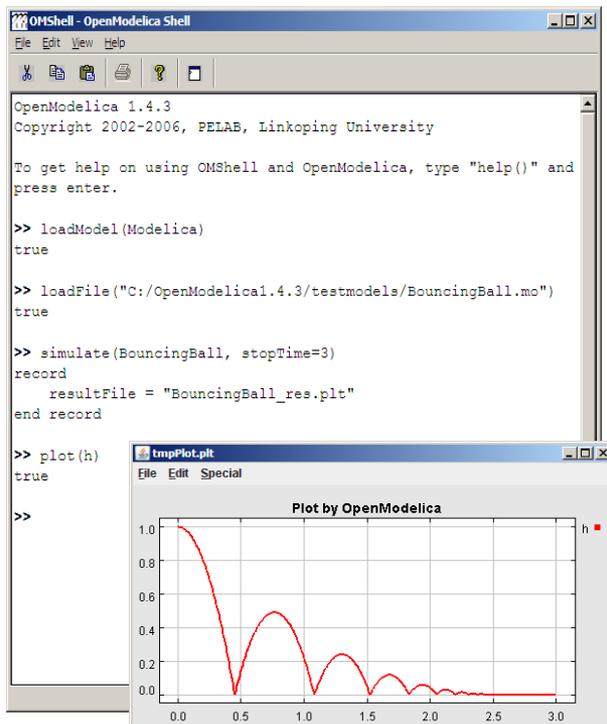
OpenModelica is ... its developers,
testers, bug reporters, contributors
and OSMC members

Thank you!

*asodja, sjoelund.se, sebco011, lochel, wbraun, niklwors, hubert.thieriot,
petar, perost, Frenkel TUD, Unknown, syeas460, adeas31, ppriv, ricli576,
haklu, dietmarw, levsu, mahge930, x05andfe, mohsen, nutaro, x02lucpo,
florosx, x06hener, x07simbj, stebr461, x08joekl, x08kimja, Dongliang Li,
jhare950, x97davka, krsta, edgarlopez, hanke, henjo, wuzhu.chen, fbergero,
harka011, tmtuomas, bjozac, AlexeyLebedev, x06klasj, ankar, kajny,
vasaie_p, niemisto, donida, hkiel, davbr, otto@mathcore.com, Kaie Kubjas,
x06krino, afshe, x06mikbl, leonardo.laguna, petfr, dhedberg, g-karbe,
x06henma, abhinck, azazi, x02danhe, rruusu, x98petro, mater, g-bjoza,
x02kajny, g-pavgr, x05andre, vaden, jansilar, ericmeyers, x05simel, andsa,
leist, choeger, Ariel.Liebman, frisk, vaurich, mwalther, mtiller, ptauber,
casella, vitalij, hkiel, jank, rfranke, mflehmg, crupp2, kbalzereit,
marchartung, Andreas, Karim, adrpo*

What is OpenModelica? (I)

- Advanced Interactive Modelica compiler (OMC)
 - Supports MSL v. 3.2.1/3.2.2/MSL trunk
- Basic and advanced environments for creating models
 - OMShell - an interactive command handler
 - OMNotebook - a literate programming notebook
 - OMEdit - Connection Editor, *Transformational and Algorithmic Debugger*, 3D Viewer
 - OMPlot - OpenModelica Plotting
 - OMOptim - OpenModelica Optimization Editor
 - OMPython/OMJulia/OMMatlab - OpenModelica Python/Julia/Matlab Environment
 - MDT - an advanced textual environment in Eclipse
 - OMSimulator - co-simulation of composite models using FMUs or via TLM



What Is OpenModelica? (II)

- Advanced Eclipse-based Development Environment
- Modelica Development Tooling (MDT) - started in 2005
 - Code Assistance, Debugging, Outline & a lot more
 - *Used heavily for OpenModelica development*
 - Used in many OpenModelica Development Courses
 - *Should be replaced by OMEdit*
- ModelicaML UML/SysML integration

① System Modeling with ModelicaML

② Modelica Code Generation

③ System Simulation with Modelica Tools

What is OpenModelica? (III)

- Open-source community services
 - Website and Support Forum
 - Source versioning (github.com)
 - Trac with bug database
 - Development courses
 - Mailing lists

Welcome to OpenModelica

https://openmodelica.org

OpenModelica

HOME DOWNLOAD TOOLS & APPS USERS DEVELOPERS FORUM EVENTS RESEARCH

Top information

- OMEdit: Enhanced OpenModelica Connection Editor.
- OMPYthon: The new OpenModelica Python Interface.

Modelica/OpenModelica Videos

- Overview of Modelica, an...
- Modelica Cyber Physical...

Registration

Here is an overview presentation about Modelica and OpenModelica.

Introduction

OPENMODELICA is an open-source Modelica-based modeling and simulation environment intended for industrial and academic usage. Its long-term development is supported by a non-profit organization – the Open Source Modelica Consortium (OSMC).

The goal with the OpenModelica effort is to create a comprehensive Open Source Modelica modeling, compilation and simulation environment based on free software distributed in binary and source code form for research, teaching, and industrial usage. We invite researchers and students, or any interested developer to participate in the project and cooperate around OpenModelica, tools, and applications.

Donate

Please consider supporting our efforts.

Amount: SEK

Donate

Latest news

- CFP OpenModelica Workshop February 2014
- October 09: OpenModelica 1.9.0 released
- September 27: OpenModelica 1.9.0 RC1 released
- February 1: OpenModelica 1.9.0 Beta4 released
- October 19: OpenModelica 1.9.0 Beta2 released
- Oct 16 : CFP OpenModelica/MODPROD Workshops February 2013
- August 31: OpenModelica 1.9.0 Beta released
- April 4: OpenModelica 1.8.1 released

OpenModelica

https://github.com/OpenModelica

Search GitHub

Repositories

- OpenModelica: OpenModelica is an open-source Modelica-based modeling and simulation environment intended for industrial and academic usage. Updated an hour ago. 40 stars, 37 forks.
- OMCompiler: The OpenModelica Compiler is the core of the OpenModelica project, which is an open-source Modelica-based modeling and simulation environment intended for industrial and academic usage. Updated an hour ago. 12 stars, 35 forks.
- OMLibraries: Updated 14 hours ago. 1 star, 2 forks.

People

Invite someone

OpenModelica

https://trac.openmodelica.org/OpenModelica/wiki

OpenModelica Project

OpenModelica is an open-source Modelica-based modeling and simulation environment intended for industrial and academic usage. Its long-term development is supported by a non-profit organization – the Open Source Modelica Consortium (OSMC).

This Trac installation is intended to help with the OpenModelica Project management, development, bug fixing, etc.

Documentation

- Automatically generated documentation of OpenModelica and Modelica.
- Writing efficient MetaModelica code: Details on how to write efficient MetaModelica code for the bootstrapped compiler is here.
- Modelica Compliant Libraries: We have made a list with compatibility issues between tools. To write compliant library follow this document.

Contribute

You can report a bug by adding a new ticket. Please have a look at all the open tickets first.

Testing

We run builds and tests using Hudson. Check the latest build and test status. Check the status of the (in development) compliance suite of the Modelica specification. Check the latest MSL 3.2.1 coverage. Check the latest ModelicaTest 3.2.1 coverage. Check the historical MSL coverage or trend of all tested libraries. Check the directory of all tested libraries.

MSL 3.2.1 Coverage

ModelicaTest 3.2.1 Coverage

Legend

- Target: 274
- Compile: 269
- Simulate: 248

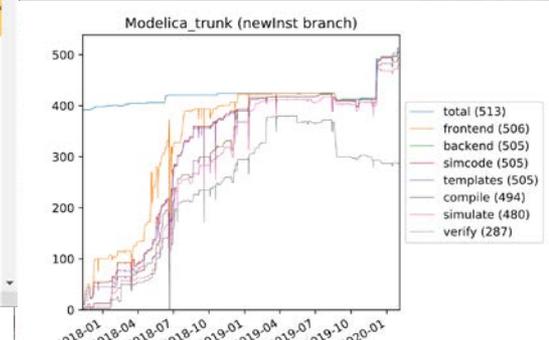
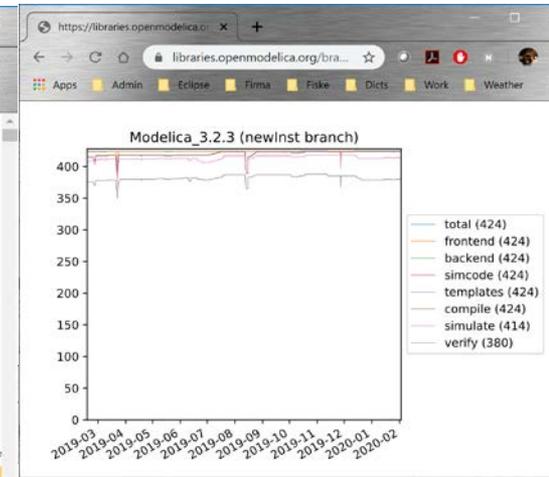
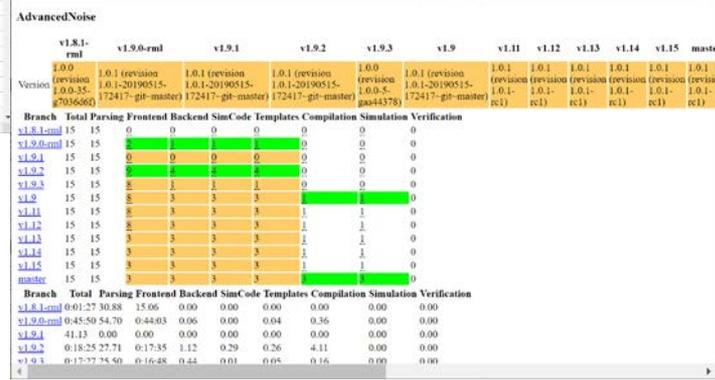
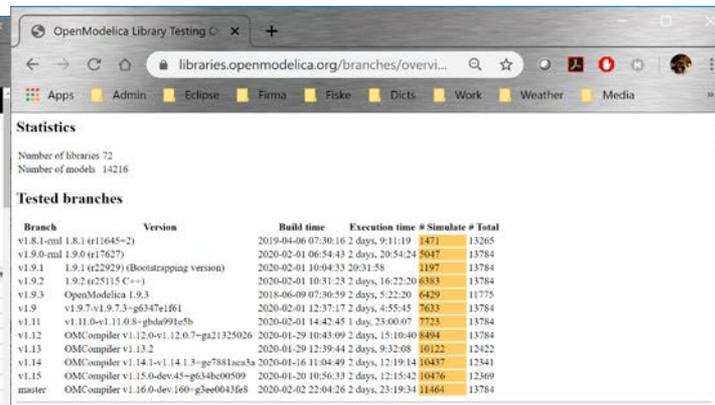
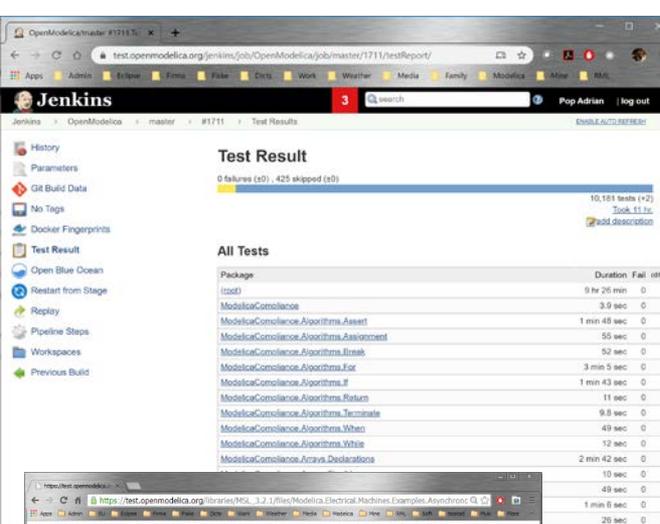
Legend

- Target: 431
- Compile: 414
- Simulate: 305

What is OpenModelica? (IV)

Open-source community services

- Extensive testing (unit & library coverage: 72 libraries, 14216 models) with interactive result comparison. 9 test servers currently
 - <https://libraries.openmodelica.org/branches/overview-combined.html>
 - Linux (GCC & CLANG), Windows (MinGW GCC), Mac OS (GCC)
 - Platforms: x86, x86_64, ARM
 - 3 runtimes: FMI, C runtime, C++ runtime
- ~10,181 tests ran on each pull request via Hudson
- Automatic nightly builds for Window & Linux & Mac OS



What is OpenModelica? (V)

- **An incubator platform for research**
 - 9 PhDs since 2004 (Debugging, Parallelization, PDEs Extensions)
 - 36 Master's theses since 2004
 - Both the students and the project benefit
- **Master theses at PELAB 2006-2018**
 - Refactoring/Parsing and Language extensions
 - UML/SysML view of Modelica code
 - 2D and 3D visualization tools
 - Static and runtime debugging tools
 - Advanced code generation and parallelization of simulation code
 - Bootstrapping and Java Interface
 - Function pointers
 - NVIDIA for Cuda and OpenCL parallel simulation
 - OMEdit - Modelica Connection Editor
 - OMWeb - server based Modelica simulation for teaching
 - OMCcc parser
 - PDE-solver using ParModelica
- **External Master theses**
 - Model based diagnostics at ISY (Dep. Of Electrical Engineering)
 - Monte-Carlo simulation of Satellite Separation Systems at SAAB
 - Interactive Simulations (EADS)
 - Additional Solvers + Event handling (FH-Bielefeld)
 - EADS - ModelicaML
- **A Base for commercial and open source products**
 - MathCore AB, Bosch Rexroth, VTT, Equa, Evonik, ABB

OpenModelica Roadmap - Past

1997 - started as a master thesis

2003 - first usable internal version

2004 - first external version: OpenModelica 1.1

2005 - more development: OpenModelica 1.3.1

2006 - major milestone

- Translated the whole compiler to MetaModelica
- Integrated Development Environment for the compiler
- OpenModelica website started
- Moved the code repository to Subversion management
- Extended the OpenModelica environment with new tools
- 4 versions released during the year
- External people start using OpenModelica
 - ~ 200 downloads/month
 - first development course at INRIA

OpenModelica Roadmap - Past

2007 - continued development and community involvement

- Improvement in website, support and documentation
- Answered ~1000 questions on the forum
- Portability is highly improved, ported to 4 platforms
 - Linux, Mac, Solaris, Windows (version 1.4.3)
- Improvement of the compiler development tools in Eclipse
- OpenModelica Community starts to react
 - contribute code & report bugs & request enhancements & participate in answering questions in the OpenModelica forum
 - participate at courses and workshops
- New server acquired for better community services
- Increased usage: ~600 downloads/month
- Open Modelica Consortium created in December 4
 - 4 months of work
 - 9 organizations as members already (3 Universities, 6 Companies)
 - discussions are ongoing with other 6 companies

2008 - Further work on the compiler

- Release 1.4.4 and 1.4.5
 - Linux, Mac, Solaris, Windows
- New Solver Interface
- Refactoring
- Dynamic loading of functions
- Merging of MathCore front-end code
- 744 commits in Subversion
- Other things I don't remember

2009

- Work mainly happened in OSMC (partially on a non-public branch)
- **Front-end**
 - Refactoring (OSMC)
 - Enumerations (OSMC)
 - Java Interface and Bootstrapping (Martin Sjölund)
 - MultiBody flattening (OSMC)
 - Constraint connection graph breaking (VTT + OSMC)
 - Support for Modelica 3.x and 3.x annotations (OSMC)
- **Back-end**
 - Tearing in the back-end (Jens Frenkel)
 - Template Code Generation and CSharp backend (Pavol Privitzer, Charles University Prague)
 - Interactive Simulations (EADS)
 - C++ Code generation (Bosch Rexroth)
 - Java Interface and Bootstrapping (Martin Sjölund)
 - Additional Solvers + Events (Willi Braun, FH-Bielefeld)
- **General**
 - New ModelicaML + SysML prototype (EADS)
 - 1144 commits in subversion (Since 2009 to February 8, 2010)
 - Bug fixes (OSMC)
 - Release 1.5.0 and 1.5.0-RC_X (Linux, Mac, Solaris, Windows)
- **More things I don't remember**

OpenModelica Roadmap - Past

2010 - 2011

- Support for Modelica Standard Library 3.1 (Media & Fluid in works)
- **Front-end**
 - MultiBody flattening (OSMC)
 - Support for Modelica 3.x and 3.x annotations (OSMC)
 - Performance Enhancements
 - Stream connectors
 - Media & Fluid work is on the way
- **Back-end**
 - Back-end redesign (Jens, Willi, Martin, Per, Adrian, Kristian, Filippo)
 - Tearing in the back-end (Jens Frenkel)
 - Template Code Generation and CSharp backend (Pavol Privitzer, Charles University Prague)
 - Interactive Simulations (EADS)
 - C++ Code generation (Bosch Rexroth)
 - Additional Solvers + Events + Linearization (Willi Braun, FH-Bielefeld)
- **General**
 - OMEdit - new connection editor
 - Bootstrapping OMC (90% finished)
 - 2550 commits in subversion from 2010 to Feb. 7, 2011 (double than 2009-2010)
 - Bug fixes ~300+ (OSMC)
 - Release 1.6.0 (Linux, Mac, Windows)
 - Downloads Windows (~16434) , Linux (~8301), Mac (~2816)
- **More things I don't remember**

OpenModelica Roadmap - Past

2012 - 2013

- Support for Modelica Standard Library 3.2.1 including Media & Fluid
- **Front-end**
 - Performance Enhancements
 - Media & Fluid work
 - Operator overloading
 - New instantiation module started
- **Back-end**
 - Modular back-end with more optimization modules (Jens, Willi, Martin)
 - New simulation runtime redesign (Willi, Lennart, Jens, Martin, Adrian)
 - C++ Code generation (Bosch Rexroth)
 - FMI export & import
 - Initialization, Jacobians (Lennart Lochel, Willi Braun, FH-Bielefeld)
 - Support for parallelization (Martin)
 - Parallel extensions in functions
- **General**
 - Uncertainties support (OpenTURNS connection & Data reconciliation)
 - MDT GDB debugging based on GDB and the bootstrapped compiler
 - OMEdit - improvements
 - Bootstrapping OMC (100% finished) using Boehm GC
 - 3909 commits in subversion from 2012 to Feb. 4, 2013
 - 2000 forum posts (questions and answers)
 - Bug fixes ~247+ (OSMC)
 - Release 1.9.0 (Linux, Mac, Windows)
 - Downloads Windows (~45307) , Linux (~15543), Mac (~5367)
- **More things I don't remember**

OpenModelica Roadmap - Past

- 2014 - 2017 - Most focus on libraries support & performance
 - MSL 3.2.1 (100% build/98% simulate), ModelicaTest 3.2.1, PetriNet, Buildings, PowerSystems, OpenHydraulics, ThermoPower, and ThermoSysPro
 - Switch to bootstrapped compiler
- Front-end, Back-end, Simulation Runtime, Graphical Clients
 - Development switched to bootstrapped compiler since November 2014
 - Partially new graph-based front-end with better support for libraries
 - Improved back-end: initialization, system solving, parallelization, cse optimization, dynamic optimization
 - Faster and much more user friendly OpenModelica Connection editor
- General
 - ~9000 commits in subversion from Feb. 2014 to Feb., 2016
 - Bug fixes
 - Release 1.9.2 (Linux, Mac, Windows)

OpenModelica Roadmap - Past

- 2018 - 2019 - focus on performance, scalability, bug fixes
- OMC & Clients
 - Performance & scalability improvements
 - Bug fixes to OMC, OMCedit, FMI
- OMSimulator
 - Combined FMI & TLM support, SSP support
 - OMCedit GUI support
- OMJulia
 - API to access OpenModelica from Julia
- General
 - From Feb 2018 - Feb 2019
 - 30+ contributors
 - 800 commits (OMCompiler)
 - 969 commits (OMSimulator)
 - 213 commits (OMCedit)
 - Releases 1.13.0, 1.13.1, 1.13.2

- 2018 - 2019 - focus on performance, scalability, bug fixes
- **New Front-End - status**
 - The new front-end ~90% complete, (see #4138 on Trac)
 - 100+ times faster, 5+ times less memory consumption (no array expansions, no expansion of for loops in equations)
 - The new front-end also brings better support for libraries
 - Developed in line with MCP-0019: Flattening
 - Currently 423/424 models from MSL 3.2.3 pass the new front-end
 - Last year 107/387 models from MSL 3.2.3 passed the new front-end
- **New Front-End - remaining work**
 - Expandable connectors (add virtual nodes)
 - Making the backend cooperate with the new way the DAE is produced
 - Support for state machines
 - (Support for MetaModelica)

OpenModelica Roadmap - Past

- 2018 - 2019 - focus on performance, scalability, bug fixes
- OMEdit - better Modelica support
 - Much more stable OMEdit, a lot of bug fixes and new usability features
 - Auto completion support
 - Support for OMSimulator
- Redeclare and Replaceable Support
 - Waiting for the new front-end to become mature enough so we don't frustrate users

■ Testing procedure

- <https://libraries.openmodelica.org/branches/overview-combined.html>
 - Run tests on previous OpenModelica version until 1.8.1
 - Detect both model regression and performance regression, all information saved in a database
 - 72 libraries, 14216 models with interactive result comparison.
 - 3 dedicated test servers
 - Linux (GCC & CLANG), Windows (MinGW GCC), Mac OS (GCC)
 - Platforms: x86, x86_64, ARM
- Statistics**
- 5 runtimes: FMI, C runtime, C++ runtime, newInst, daeMode

Number of libraries 72
Number of models 14216

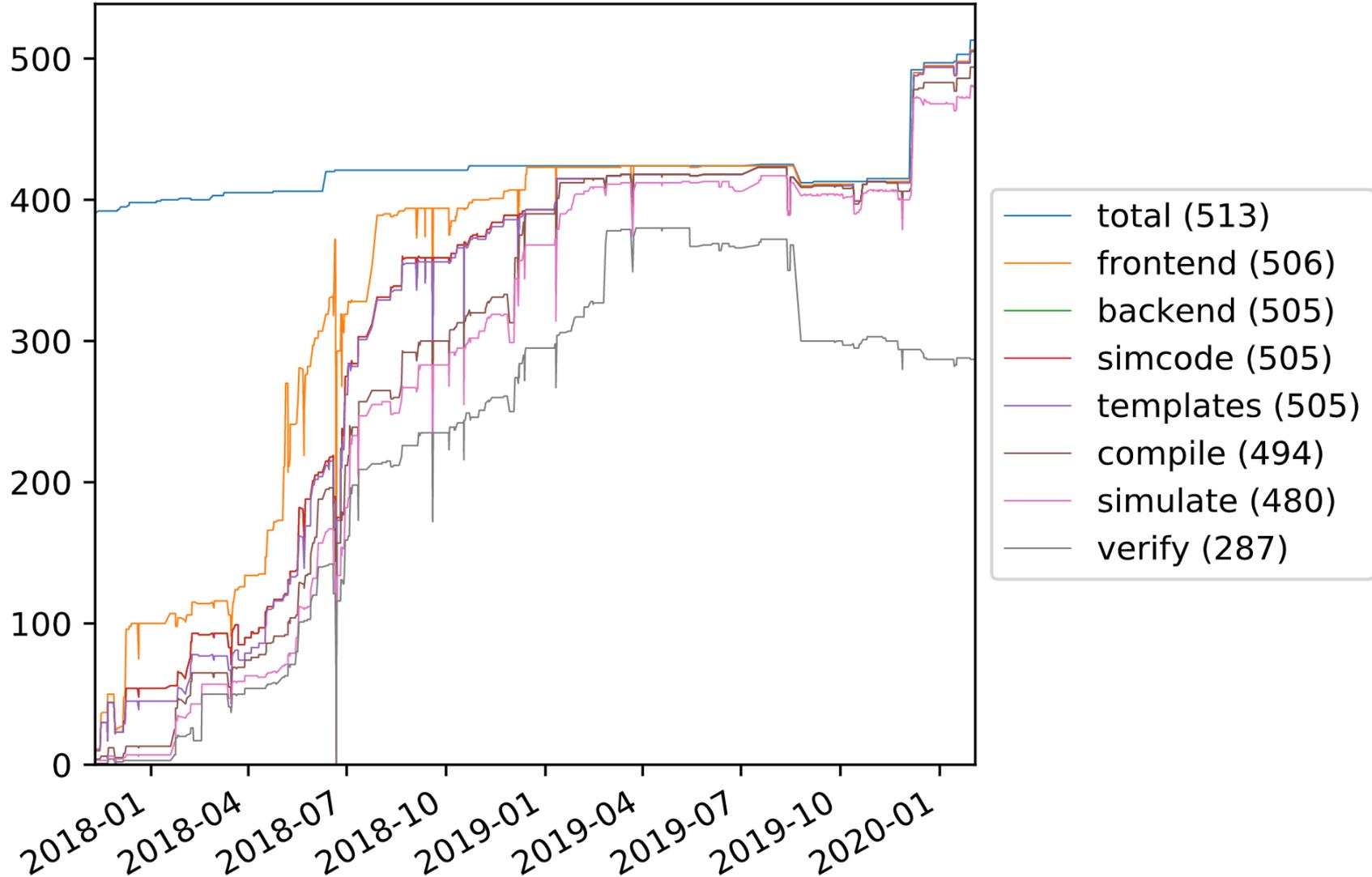
Tested branches

Branch	Version	Build time	Execution time	# Simulate	# Total
v1.8.1-rml	1.8.1 (r11645+2)	2019-04-06 07:30:16	2 days, 9:11:19	1471	13265
v1.9.0-rml	1.9.0 (r17627)	2020-02-01 06:54:43	2 days, 20:54:24	5047	13784
v1.9.1	1.9.1 (r22929) (Bootstrapping version)	2020-02-01 10:04:33	20:31:58	1197	13784
v1.9.2	1.9.2 (r25115 C++)	2020-02-01 10:31:23	2 days, 16:22:20	6383	13784
v1.9.3	OpenModelica 1.9.3	2018-06-09 07:30:59	2 days, 5:22:20	6429	11775
v1.9	v1.9.7-v1.9.7.3+g6347e1f61	2020-02-01 12:37:17	2 days, 4:55:45	7633	13784
v1.11	v1.11.0-v1.11.0.8+gbda991e5b	2020-02-01 14:42:45	1 day, 23:00:07	7723	13784
v1.12	OMCompiler v1.12.0-v1.12.0.7+ga21325026	2020-01-29 10:43:09	2 days, 15:10:40	8494	13784
v1.13	OMCompiler v1.13.2	2020-01-29 12:39:44	2 days, 9:32:08	10122	12422
v1.14	OMCompiler v1.14.1-v1.14.1.3+ge7881aca3a	2020-01-16 11:04:49	2 days, 12:19:14	10437	12341
v1.15	OMCompiler v1.15.0-dev.45+g634bc00509	2020-01-20 10:56:33	2 days, 12:15:42	10476	12369
master	OMCompiler v1.16.0-dev.160+g3ee0043fe8	2020-02-02 22:04:26	2 days, 23:19:34	11464	13784

OpenModelica Testing (II)

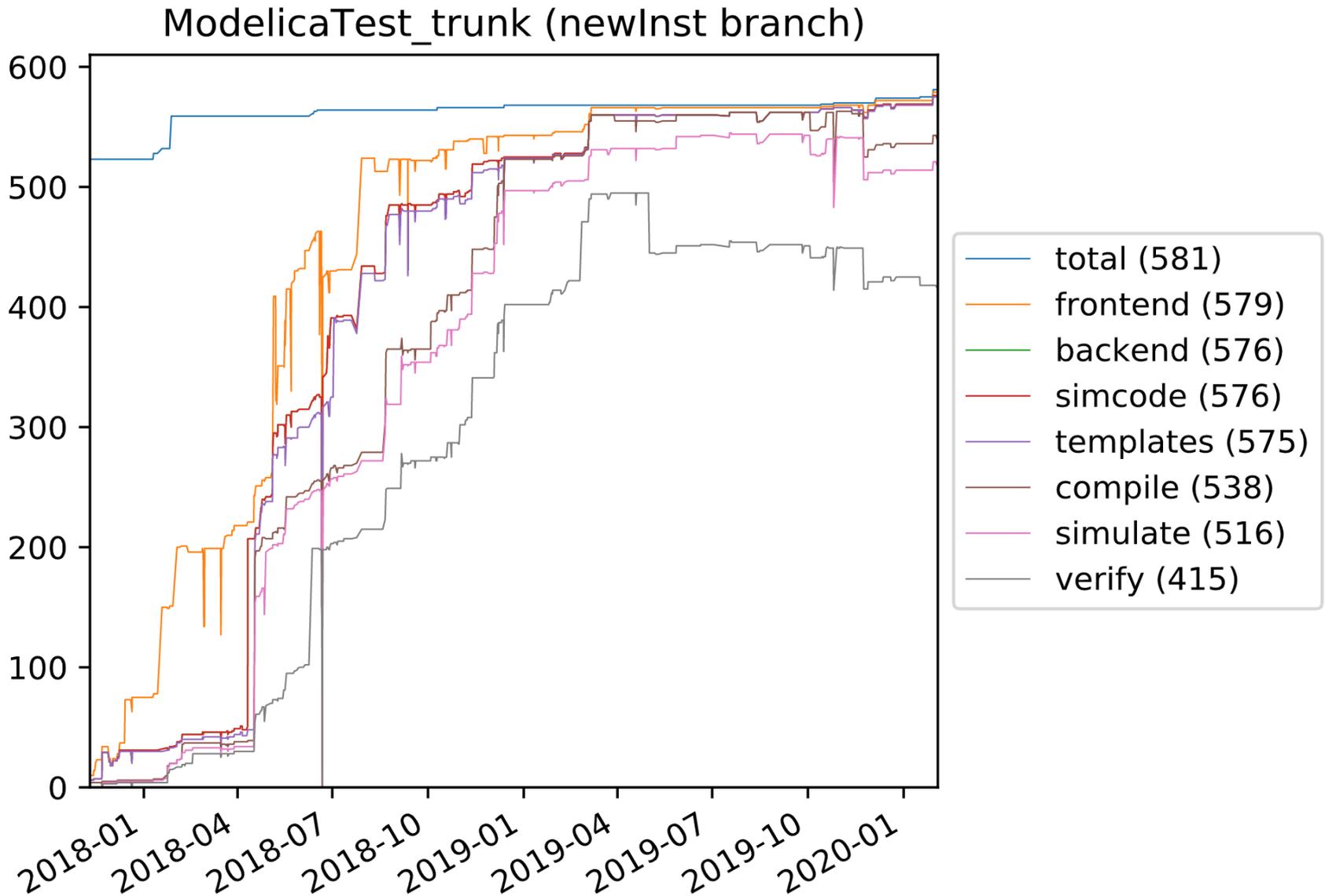
- 2020-02-03 v1.16-dev - total 513 - build 494 (96%) - sim 480 (93%)

Modelica_trunk (newInst branch)



OpenModelica Testing (III)

- 2020-02-03 v1.16-dev - total 581 - build 538 (93%) - sim 516 (88%)



- Moved the source code to github May 2015
- Mature code base: <https://github.com/OpenModelica>
- ~9000K lines of code and tests

- From Feb 2017 - Feb 2018
 - 20 contributors
 - 794 commits (OMCompiler)

- From Feb 2018 - Feb 2019
 - 30+ contributors
 - 800 commits (OMCompiler)
 - 969 commits (OMSimulator)
 - 213 commits (OMEdit)

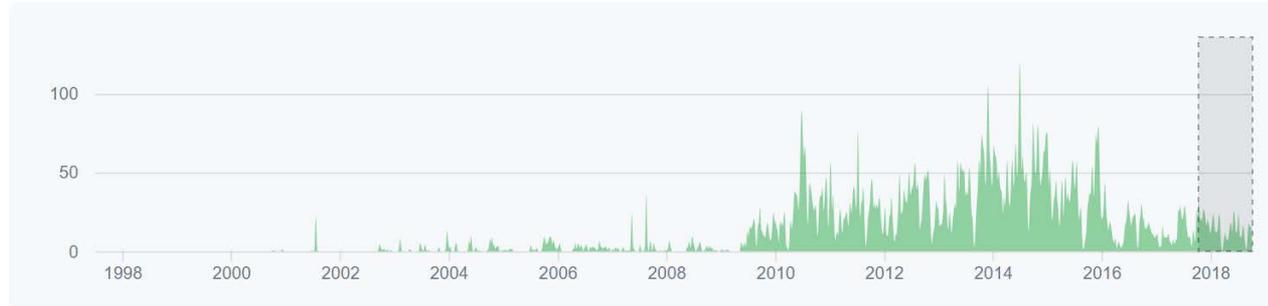
- From Feb 2019 - Feb 2020
 - 30+ contributors
 - 800 commits (OMCompiler)
 - 969 commits (OMSimulator)
 - 213 commits (OMEdit)

OpenModelica Statistics (II)

Feb 5, 2018 – Feb 3, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits

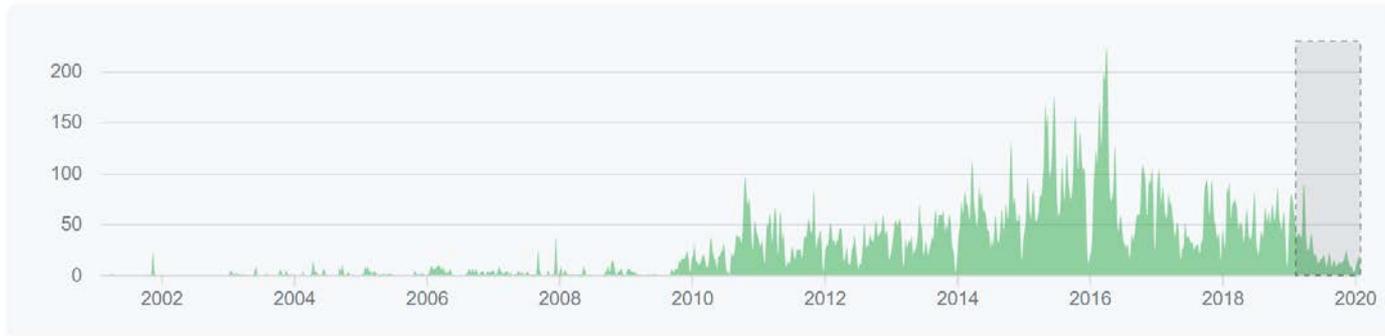


OpenModelica Statistics (III)

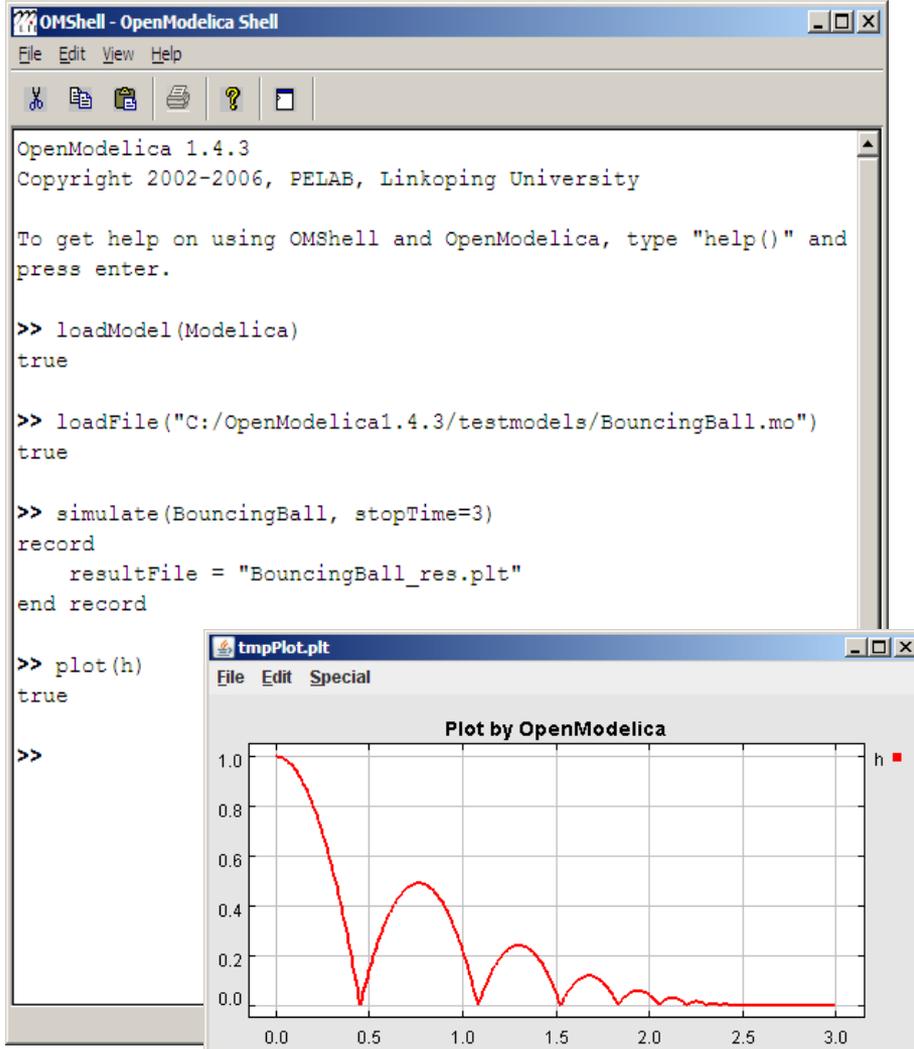
Feb 12, 2019 – Feb 3, 2020

Contributions: Commits ▾

Contributions to master, excluding merge commits



- OpenModelica
 - What is OpenModelica?
 - The past
- OpenModelica Technical Overview
 - OMC, OMShell, OMNotebook, OMEdit, ModelicaML, OMSimulator, OMPython, OMJulia, OMMatlab
- OpenModelica Development Environment
 - MetaModelica
 - The Eclipse Environment (MDT)
- OpenModelica Latest Developments (2019-2020)



OMShell - OpenModelica Shell

File Edit View Help

OpenModelica 1.4.3
Copyright 2002-2006, PELAB, Linköping University

To get help on using OMShell and OpenModelica, type "help()" and press enter.

```
>> loadModel(Modelica)
true

>> loadFile("C:/OpenModelica1.4.3/testmodels/BouncingBall.mo")
true

>> simulate(BouncingBall, stopTime=3)
record
  resultFile = "BouncingBall_res.plt"
end record

>> plot(h)
true

>>
```

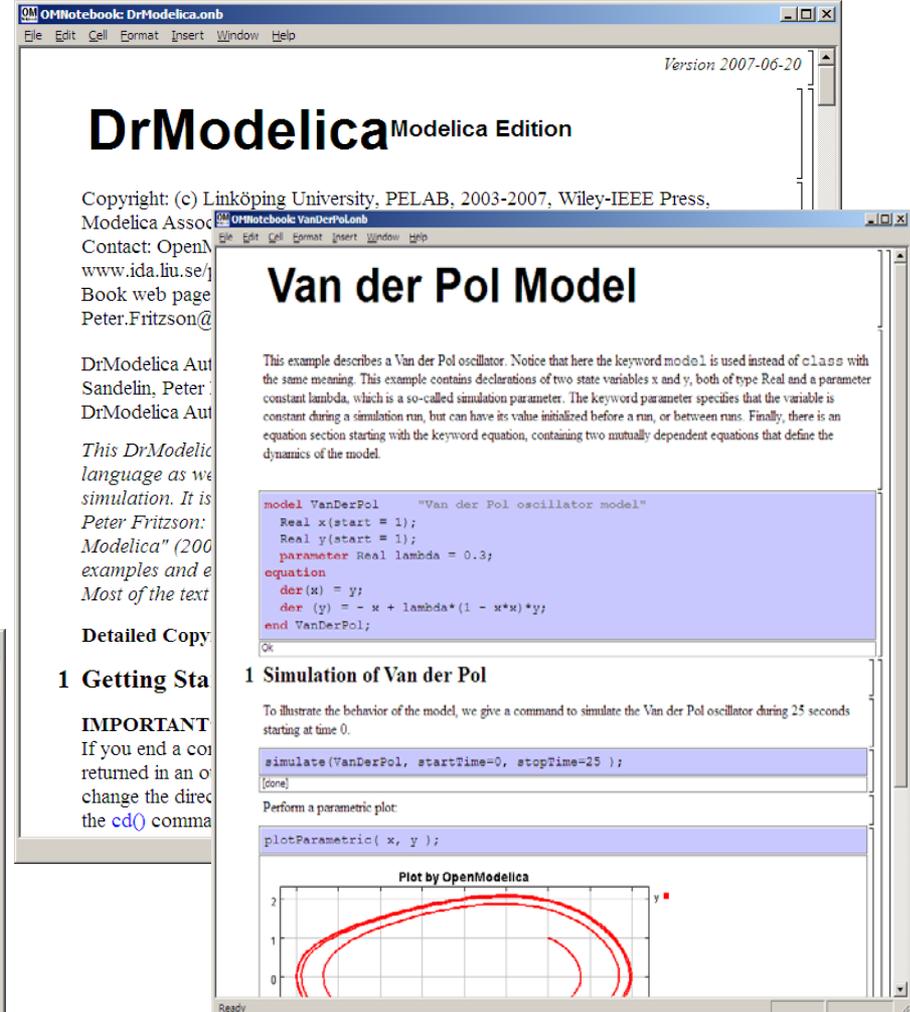
tmpPlot.plt

File Edit Special

Plot by OpenModelica



Time (s)	Height (h)
0.0	1.00
0.5	0.00
1.0	0.50
1.5	0.25
2.0	0.12
2.5	0.06
3.0	0.03



OMNotebook: DrModelica.onb

File Edit Cell Format Insert Window Help

Version 2007-06-20

DrModelica Modelica Edition

Copyright: (c) Linköping University, PELAB, 2003-2007, Wiley-IEEE Press,
Modelica Assoc. www.ida.liu.se/
Book web page
Peter.Fritzson@

Van der Pol Model

This example describes a Van der Pol oscillator. Notice that here the keyword `model` is used instead of `class` with the same meaning. This example contains declarations of two state variables `x` and `y`, both of type `Real` and a parameter constant `lambda`, which is a so-called simulation parameter. The keyword parameter specifies that the variable is constant during a simulation run, but can have its value initialized before a run, or between runs. Finally, there is an equation section starting with the keyword `equation`, containing two mutually dependent equations that define the dynamics of the model.

```
model VanDerPol "Van der Pol oscillator model"
  Real x(start = 1);
  Real y(start = 1);
  parameter Real lambda = 0.3;
equation
  der(x) = y;
  der(y) = -x + lambda*(1 - x*x)*y;
end VanDerPol;
```

Ok

1 Simulation of Van der Pol

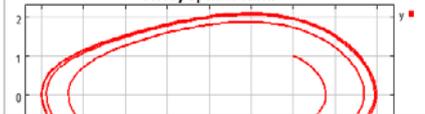
To illustrate the behavior of the model, we give a command to simulate the Van der Pol oscillator during 25 seconds starting at time 0.

```
simulate(VanDerPol, startTime=0, stopTime=25 );
```

[done]

Perform a parametric plot:

```
plotParametric(x, y);
```



Plot by OpenModelica

x	y
1.0	1.0
2.0	1.5
3.0	1.8
4.0	1.5
5.0	1.0
6.0	0.5
7.0	0.0
8.0	0.5
9.0	1.0
10.0	1.5

- Implemented mainly in MetaModelica (401 packages) and a C/C++ runtime
- Is available as a dynamic library (faster than CORBA/ZMQ)
- Used from OMEdit, OMNotebook, OMShell, OMOptim, OMPython, MDT
- Automatically generated API that can be used from QT

- OpenModelica
 - What is OpenModelica?
 - The past and present
- OpenModelica Technical Overview
 - OMC, OMShell, OMNotebook, OMEdit, ModelicaML, OMSimulator, OMPython, OMJulia, OMMatlab
- OpenModelica Development Environment
 - MetaModelica
 - The Eclipse Environment
- OpenModelica Latest Developments (2019-2020)

- **OMC**
 - Implemented mainly in MetaModelica and C/C++
- **Modelica**
 - classes, models, records, functions, packages
 - behavior is defined by equations or/and functions
 - equations
 - differential algebraic equations and conditional equations
- **MetaModelica extensions**
 - local equations
 - pattern equations
 - match expressions
 - high-level data structures: lists, tuples, option and uniontypes

MDT - Creating Modelica projects (I)

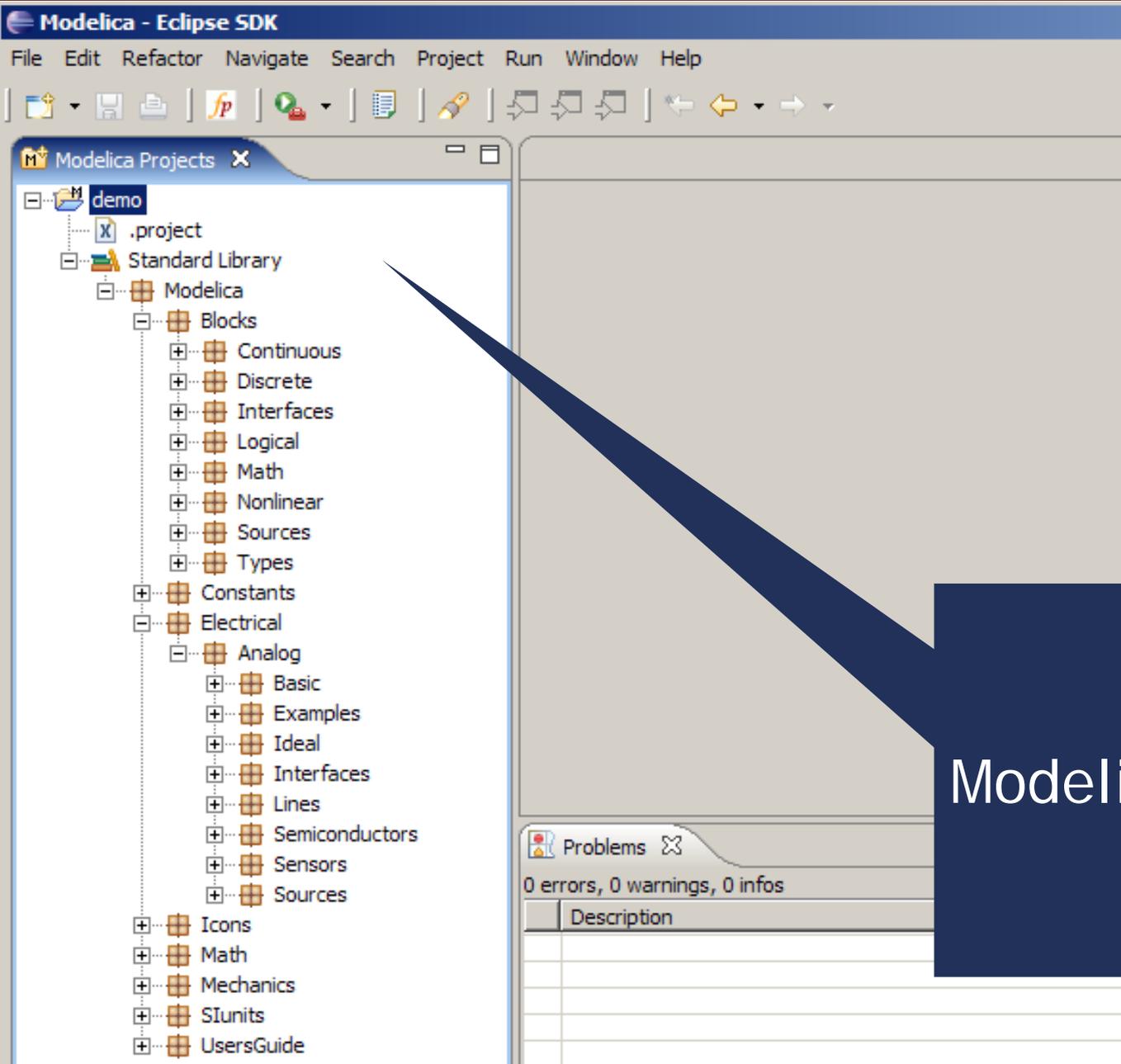
The image shows a sequence of three overlapping windows in the Eclipse IDE:

- Modelica - Eclipse SDK:** The main application window with the 'File' menu open and 'Project...' selected.
- New Project:** A dialog box titled 'Select a wizard' with a tree view of wizard categories. 'Modelica' is expanded, and 'Modelica Project' is selected.
- New Modelica Project:** A wizard step titled 'Create a Modelica project' with a text field for 'Project name:' containing the text 'demo'.

Red arrows indicate the flow: from 'Project...' in the main menu to 'Modelica Project' in the wizard list, and from the 'Next >' button in the wizard to the 'Next >' button in the 'New Modelica Project' dialog.

Creation of Modelica projects using wizards

Creating Modelica projects (II)



Modelica project

Creating Modelica packages

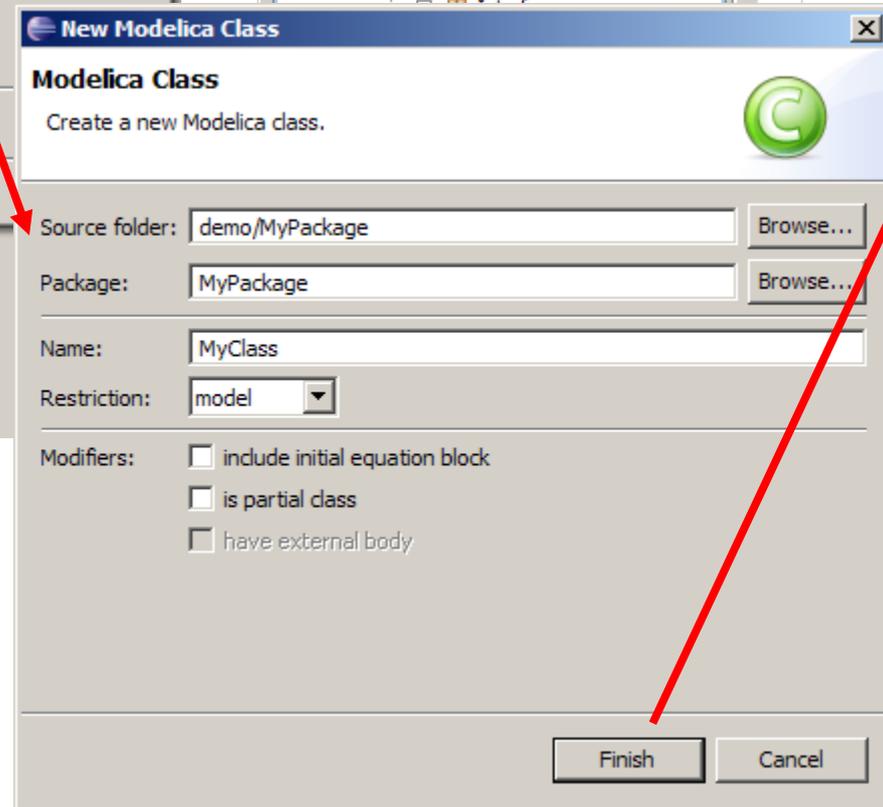
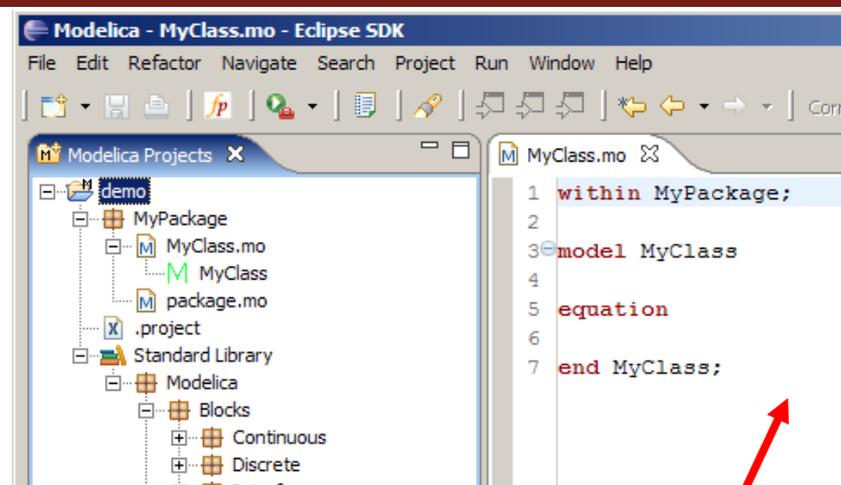
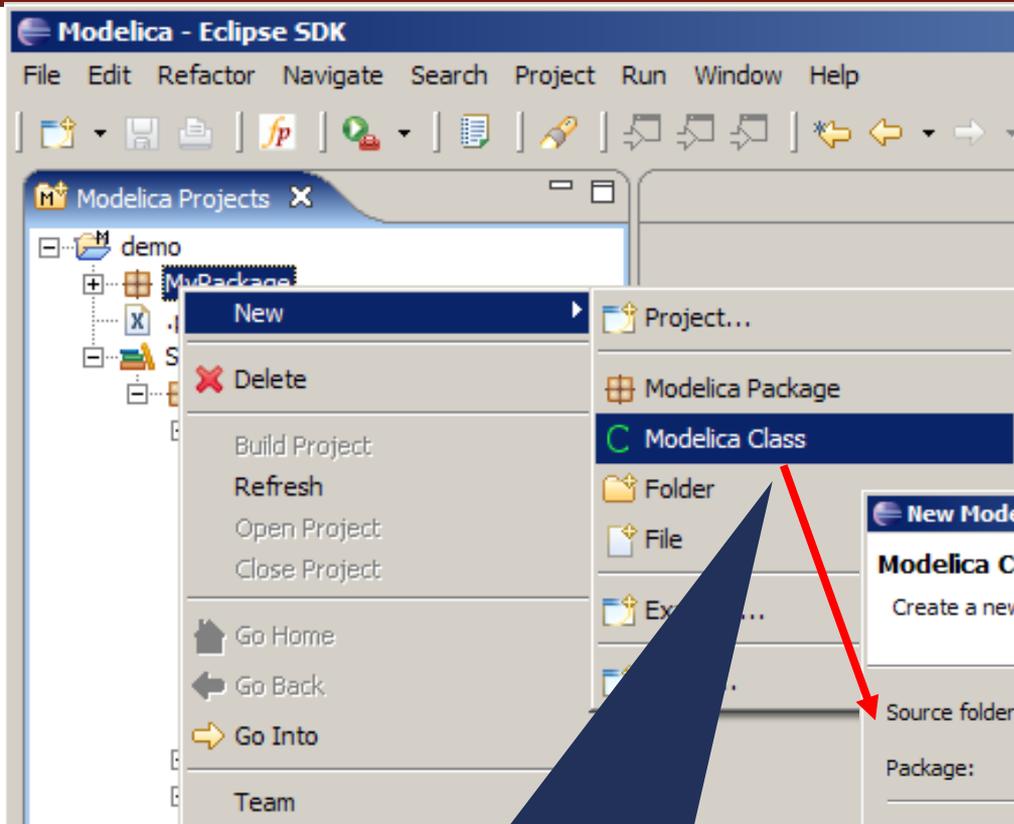
The image shows the Eclipse IDE interface for creating a new Modelica package. The 'New' menu is open, and the 'Modelica Package' option is selected. The 'New Modelica Package' wizard dialog is displayed, showing the following fields:

- Source folder: demo
- Package: (empty)
- Name: MyPackage
- Description: A Modelica Package
- is encapsulated package

The 'Finish' button is highlighted with a red arrow.

Creation of Modelica packages using wizards

Creating Modelica classes



Creation of Modelica classes, models, etc, using wizards

Code browsing

The screenshot displays the Eclipse IDE interface for a Modelica project. The left-hand side shows a project explorer with a tree view of the project structure. The 'VanDerPol' package is expanded, and the 'lambda' parameter is selected. The main editor window shows the code for 'VanDerPol.mo', with the line 'parameter Real lambda = 0.3;' highlighted. A callout box points to this line, explaining the code browsing feature.

```
1 // Van der Pol model
2
3 model VanDerPol "Van der Pol oscillator model"
4   Real x(start = 1);
5   Real y(start = 1);
6   parameter Real lambda = 0.3;
7   parameter Real e = Modelica.Constants.e;
8 equation
9   der(x) = y;
10  der(y) = - x + lambda*(1 - x*x)*y;
11 end VanDerPol;
12
```

Code Browsing for easy navigation within Modelica files. Automatic update on file save.

Error detection (I)

The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Shows a project named 'demo' containing a package 'MyPackage' with files 'MyClass.mo', 'package.mo', and 'VanDerPol.mo'. The 'VanDerPol' model is expanded, showing variables 'e', 'x', and 'y'.
- Code Editor:** Displays the content of 'VanDerPol.mo'. The code is as follows:

```
1 // Van der Pol model
2
3 model VanDerPol "Van der Pol oscillator model"
4   Real x(start = 1);
5   Real y(start = 1);
6   parameter Real lambda = 0.3;
7   parameter Real e = Modelica.Constants.e;
8 equation
9   der(x) = y;
10  der(y) = - x + lambda*(1 - x*x)*y;
11 end VanDerPol;
12
```

Line 6 is highlighted in blue, and a red 'X' icon is visible in the left margin next to it.
- Problems Window:** Shows '1 error, 0 warnings, 0 infos'. The error table is as follows:

Description	Resource	In Folder	Location
unexpected token: lambda, parsing resumed at token ';' on line 6, column 29	VanDerPol.mo	demo	line 6

Parse error
detection on
file save

Error detection (II)

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Lists various Modelica projects under 'Modelica Projects', including Compiler, absyn_builder, doc, modpar, omc_debug, omc_release, report, rml2mmo, rml2sig, runtime, scripts, test_codegen, tools, VC7, winruntime, and a list of .mo files including Absyn.mo.
- Code Editor:** Displays the source code for Absyn.mo. Lines 69-79 are visible. Line 77, containing `Withi within_ "within ; Within statement" ;`, is highlighted in blue and has a red 'X' icon next to it, indicating an error.
- Problems/Console:** Shows the output of a compilation process. The error message is: `Absyn.mo:77.5-77.9 Error: unbound type constructor Withi`. Other messages include `StaticElaborationError` and `make[2]: Leaving directory`.

Semantic error
detection on
compilation

Code assistance (I)

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer (Left):** Shows a project named 'demo' containing a package 'MyPackage' with files 'MyClass.mo', 'package.mo', and 'VanDerPol.mo'. The 'VanDerPol.mo' file is expanded to show a class 'VanDerPol' with parameters 'e', 'lambda', 'x', and 'y'. Below it is a 'Standard Library' with categories like 'Modelica', 'Blocks', 'Constants', 'Electrical', 'Icons', 'Math', 'Matrices', and 'acos'.
- Editor (Center):** Displays the code for 'VanDerPol.mo'. The code is:

```
1 // Van der Pol model
2
3 model VanDerPol "Van der Pol oscillator model"
4   import Modelica.
5   Real x(start = 1
6   Real y(start = 1
7   parameter Real l
8   parameter Real e
9   equation
10    der(x) = y;
11    der(y) = - x + 1
12 end VanDerPol;
13
```

Line 4 is highlighted, and a code assistance popup is visible.
- Code Assistance Popup (Right):** A list of categories from the Standard Library, including 'Blocks', 'Constants', 'Electrical', 'Icons', 'Math', 'Mechanics', 'SIunits', and 'UsersGuide'.
- Problems View (Bottom):** Shows '0 errors, 0 warnings, 0 infos'.

Code Assistance on imports

Code assistance (II)

The screenshot shows the Eclipse IDE with the following components:

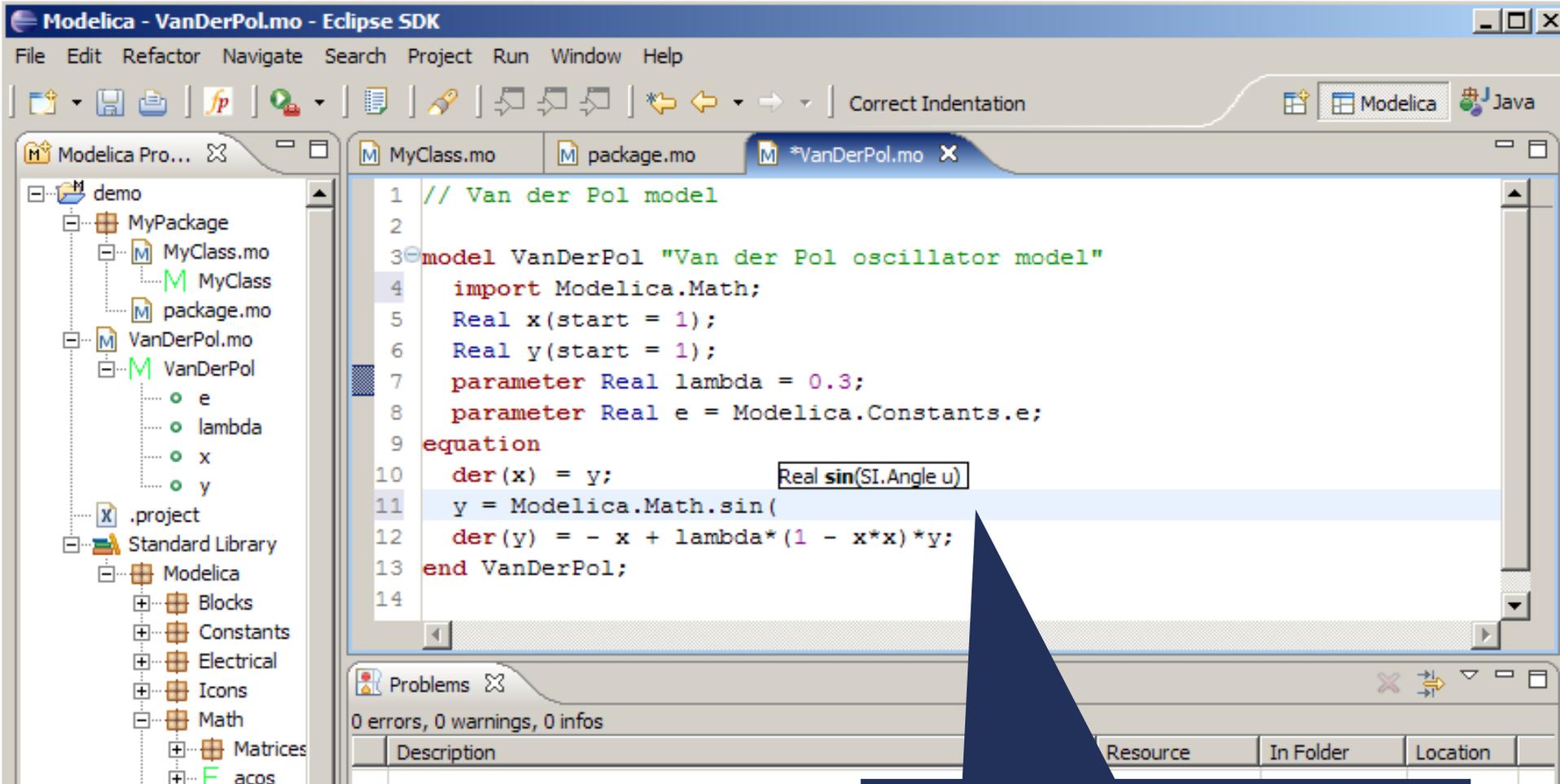
- Project Explorer:** Shows a project named 'demo' with a package 'VanDerPol' containing a class 'VanDerPol' and its attributes 'e', 'lambda', 'x', and 'y'. A 'Standard Library' is also visible with categories like 'Modelica', 'Math', and 'Electrical'.
- Editor:** Displays the source code for 'VanDerPol.mo'. The code is as follows:

```
1 // Van der Pol model
2
3 model VanDerPol "Van der Pol oscillator model"
4   import Modelica.Math;
5   Real x(start = 1);
6   Real y(start = 1);
7   parameter Real lambda = 0.3;
8   parameter Real e = Modelica.Constants.
9 equation
10  der(x) = y;
11  der(y) = - x + lambda*(1 - x*x)*y;
12 end VanDerPol;
13
```

Line 8 is selected, and a completion list is shown on the right.
- Completion List:** A list of constants from the 'Modelica.Constants' package, including 'c', 'D2R', 'e', 'eps', 'epsilon_0', 'G', 'g_n', 'h', and 'inf'. The 'e' entry is highlighted.
- Problems View:** Shows '0 errors, 0 warnings, 0 infos'.
- Table:** A table with columns 'Description', 'Resource', 'In Folder', and 'Location' is partially visible at the bottom.

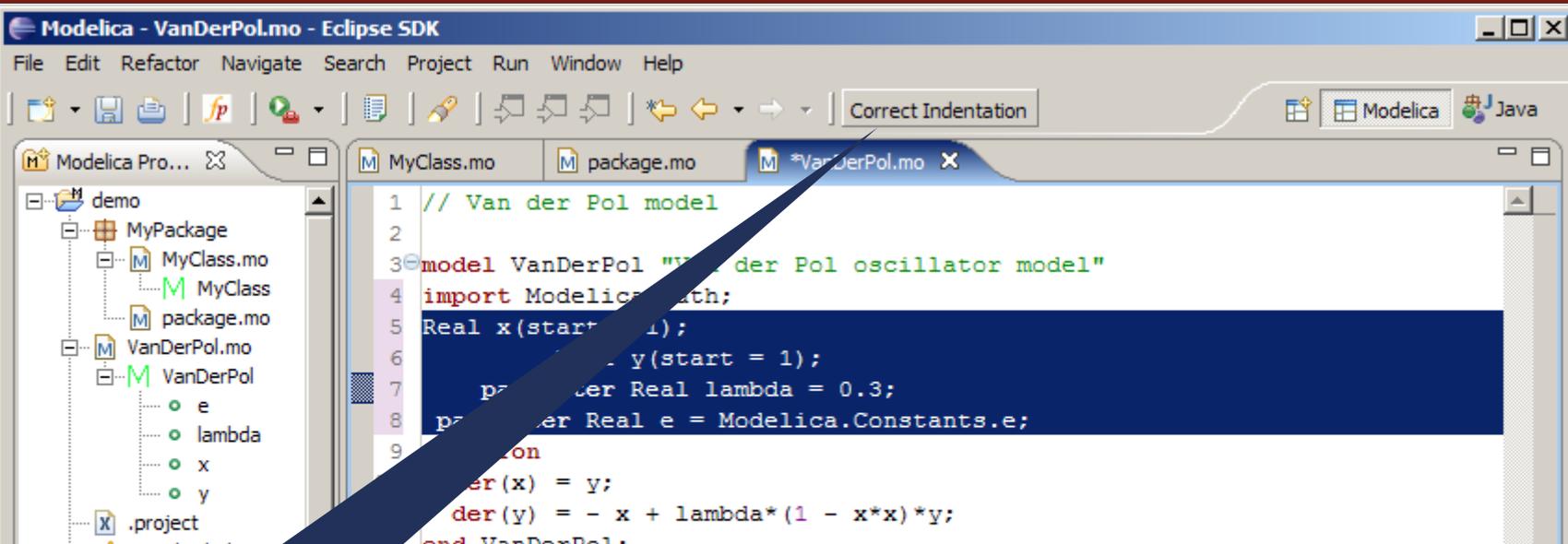
Code Assistance on assignments

Code assistance (III)



Code Assistance on
function calls

Code indentation

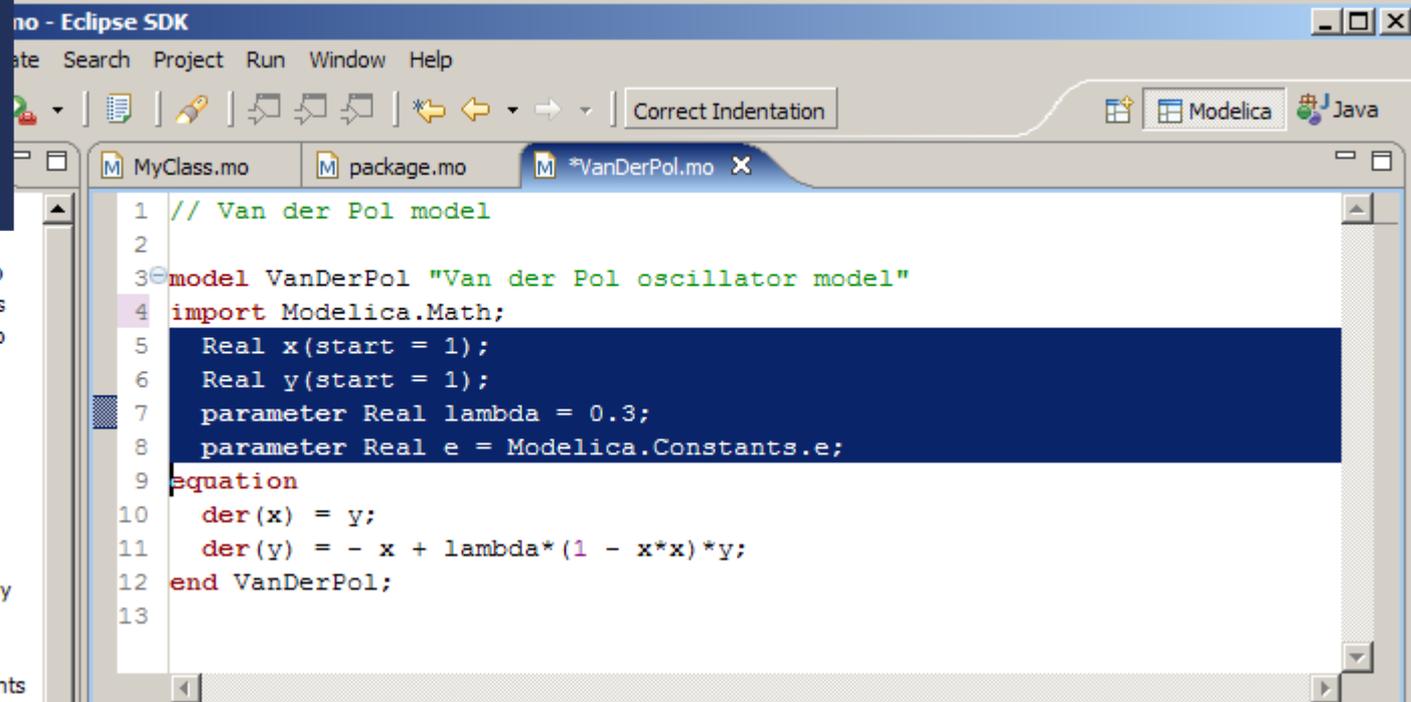


The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu is a toolbar with various icons, including a 'Correct Indentation' button. The main editor window displays the following code:

```
1 // Van der Pol model
2
3 model VanDerPol "Van der Pol oscillator model"
4 import Modelica.Math;
5 Real x(start = 1);
6 Real y(start = 1);
7 parameter Real lambda = 0.3;
8 parameter Real e = Modelica.Constants.e;
9 equation
10   der(x) = y;
11   der(y) = - x + lambda*(1 - x*x)*y;
12 end VanDerPol;
```

A blue arrow points from the 'Correct Indentation' button to the code, highlighting the indentation of lines 5 through 11.

Code
Indentation



The screenshot shows the Eclipse IDE interface with the same code as above, but with the indentation corrected. The code is now formatted as follows:

```
1 // Van der Pol model
2
3 model VanDerPol "Van der Pol oscillator model"
4 import Modelica.Math;
5   Real x(start = 1);
6   Real y(start = 1);
7   parameter Real lambda = 0.3;
8   parameter Real e = Modelica.Constants.e;
9 equation
10   der(x) = y;
11   der(y) = - x + lambda*(1 - x*x)*y;
12 end VanDerPol;
13
```

The indentation of lines 5 through 11 is now consistent, showing the effect of the 'Correct Indentation' button.

Code Outline and Hovering Info

The screenshot displays the Eclipse IDE environment for the Modelica compiler. The top toolbar contains standard IDE actions like File, Edit, and Run. The Project Explorer on the left shows a tree structure of Modelica projects, with 'Absyn.mo' selected. The Outline view below it provides a hierarchical list of code elements. The main editor window shows the source code of 'Absyn.mo', featuring a case statement and a function definition for 'getCrefFromExp'. A tooltip is displayed over the function definition, providing its signature and a description: 'function getCrefFromExp "function: getCrefFromExp Returns a flattened list of the component references in an expression"'. The Problems view at the bottom shows 113 errors, with a list of error messages. The Console view is also visible. A status bar at the bottom indicates '64M of 254M' memory usage and 'Ctrl Contrib (Bottom)'.

```
case (MATRIX(matrix = exp1))
  local list<list<list<ComponentRef>>> res1;
  equation
    res1 = Util.listListMap(exp1, getCrefFromExp);
    res2 = Util.listFlatten(res1);
    res = Util.listFlatten(res2);
  then
    res;
case (RANGE(start = e1, step = SOME(e3), stop = e2))
  equation
    l1 = getCrefFromExp(e1);
    l2 =
      function getCrefFromExp "function: getCrefFromExp
        Returns a flattened list of the
        component references in an expression"
      input Exp inExp;
      then
        output list<ComponentRef> outComponentRefList;
      algorithm
        outComponentRefList:=matchcontinue inExp
      local
        l1 =
          ComponentRef ex;
```

function getCrefFromExp "function: getCrefFromExp
Returns a flattened list of the
component references in an expression"
input Exp inExp;
then
output list<ComponentRef> outComponentRefList;
algorithm
outComponentRefList:=matchcontinue inExp
local
l1 =
ComponentRef ex;

113 errors, 0 warnings, 0 infos
Description
Errors (100 of 113 items)
The identifier at start and end are different
The identifier at start and end are different
The identifier at start and end are different, par

ken ';' on line rml2mod.mo OpenModelica/tools/rml2mod
ken ';' on line rml2mod.mo OpenModelica/tools/rml2mod
ken ';' on line modrefacto... OpenModelica/tools/rml2mod
ken ';' on line moddump.mo OpenModelica/tools/rml2mod
ken ';' on line rml2mod.mo OpenModelica/tools/rml2mod
ken ';' on line rml2mod.mo OpenModelica/tools/rml2mod
ken ';' on line rml2mod.mo OpenModelica/tools/rml2mod

64M of 254M
Ctrl Contrib (Bottom)

Code Outline for easy navigation within Modelica files

Identifier Info on Hovering

Eclipse Debugging Environment

- Type information for all variables
- Browsing of complex data structures
- GDB based

The screenshot displays the Eclipse IDE interface for debugging OpenModelica. The main window is titled "Debug - OpenModelica/Compiler/Main.mo - Eclipse SDK". The interface is divided into several panels:

- Debug Panel:** Shows the execution stack with "Main thread (stepping)" and "Main.main (line: 919, SP: 9, call: extern)".
- Breakpoints Panel:** Shows a list of breakpoints.
- Variables Panel:** Displays a table of variables and their values.
- Console Panel:** Shows the output of the program.
- Outline Panel:** Shows the project structure and the current file being edited.
- Source Editor:** Shows the source code of the file being debugged.

Name	Value	Declared Type
p	Absyn.Program	Absyn.Program
[record]	Absyn.PROGRAM[2]	((Absyn.Class list, Absyn.Within) :
classes	LIST	Absyn.Class list
[0]	Absyn.CLASS[7]	((string, bool, bool, bool, Absyn.R
name	"Bla"	string
partial_	false	bool
final_	false	bool
encapsulated_	false	bool
restriction	1:enum:Absyn.R_MODEL	Absyn.Restriction
body	Absyn.PARTS[2]	((Absyn.ClassPart list, string optio
classParts	LIST	Absyn.ClassPart list
[0]	Absyn.PUBLIC[1]	((Absyn.ElementItem list) => (Abs
contents	LIST	Absyn.ElementItem list
[0]	Absyn.ELEMENTITEM[1]	((Absyn.Element) => (Absyn.Elen
comment	NONE[0]	string option
info	Absyn.INFO[6]	((string, bool, int, int, int, int) =>
within_	Absyn.TOP[0]	Absyn.Within
f	string	string
->	"Bla.mo"	string

```
model Bla
  Integer z[10];
end Bla;
```

```
local String s;
equation
  isModelicaFile(f);
  p = Parser.parse(f);
  Debug.fprint("dump", "\n----- Parsed progr
  Debug.fcall("dumpgraphviz", DumpGraphviz.dump, p);
  Debug.fcall("dump", Dump.dump, p);
```

The Outline panel shows the following functions and classes:

- readSettingsFile(String filePath, Interactive.InteractiveSy
- runBackendQ => Boolean
- runModparQ => Boolean
- serverLoop(Integer inInteger, Interactive.InteractiveSym
- serverLoopCorba(Interactive.InteractiveSymbolTable inIn
- simcodegen(Absyn.Path inPath1, SCode.Program inProgr
- transformFlatProgram(Absyn.Program p, String filename)
- translateFile(list<String> inStringLst)
- versionRequest
- import Absyn;
- import Ceval;
- import Corba;

OMEdit Debugging Environment

The screenshot displays the OMEdit - Transformational Debugger interface. The main window shows the source code for a debugging session, with the following code visible:

```
enthalpy computation";  
parameter  
SI.SpecificHeatCapacity  
cp=4186 "Cp of the fluid";  
127 SI.MassFlowRate w_pump  
"Mass flow rate from the  
pump";  
128 SI.Pressure p1 "Pump  
discharge pressure";  
129 SI.Pressure p2 "Storage  
tank inlet pressure";  
130 SI.Pressure dp_pump  
"Pump dp";  
131 SI.Pressure dp_valve  
"Valve dp";  
132 Real sqrt_dp  
"Regularized sqrt(dp)";  
133 SI.SpecificEnthalpy h0  
"Pump inlet specific  
enthalpy";  
134 SI.SpecificEnthalpy h1  
"Pump discharge specific  
enthalpy";  
135 SI.Power W;  
136 SI.Length y(start=40,  
fixed=true) "Reservoir  
level";  
137 Real eta(final  
unit="1") = (p1 -  
patm)*w_pump/rho/W "Pump  
efficiency";  
138 SI.Temperature T1 "Pump  
discharge temperature";  
139 SI.Time tau=1 "Time  
constant of temperature  
sensor";  
140 equation  
141 dp_pump = p1 - patm  
dp";
```

The interface also includes a Variables Browser, Equations Browser, and Source Browser. The Variables Browser shows a list of variables with their comments, line numbers, and locations. The Equations Browser shows a list of equations with their indices, types, and equations. The Source Browser shows the source code for the debugging session.

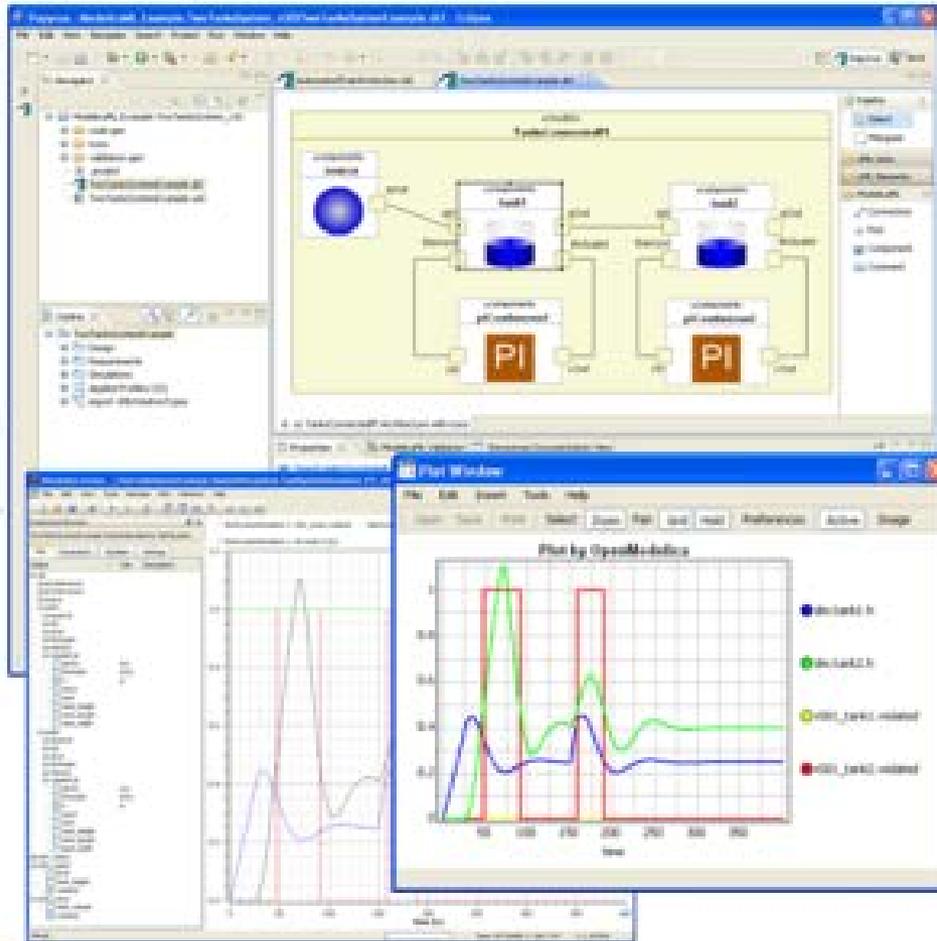
Variables	Comment	Line	Location
A	Storage ... section	120	C:\User
Kv	Valve coefficient	112	C:\User
T0	Tempera...g fluid	118	C:\User
T1	Pump di...erature	138	C:\User
Tref	Referen...utation	124	C:\User

Equations	Type	Equation
1	initial	(assignment) ...*(T0 - Tref)
2	initial	(assignment)...o * y + patm
3	initial	(assignment..._pump ^ 2.0
4	initial	(assignmen...ump + patm
5	initial	(assignment)... Line: 144")
6	initial	(assignment)...ve = p1 - p2
7	initial	(residual,sqr...5 - dp_valve)
8	initial	(nonlinear)
3	initial	(assignment..._pump ^ 2.0
4	initial	(assignmen...ump + patm
5	initial	(assignment)... Line: 144")
6	initial	(assignment)...ve = p1 - p2
7	initial	(residual,sqr...4(String)#
9	initial	(assignment)..._4(String)#
10	initial	(assignment)...a3

Tutorial 1 - tomorrow at ModProd 2020!

Eclipse environment for ModelicaML

① System Modeling with ModelicaML



② Modelica Code Generation

```
Modelica: TestSystem
class TestSystem
public
  Real u[k1];
  Real u[k2];
  Real y[k1];
  Real y[k2];
protected
  // ...
  Real u[k1];
  Real u[k2];
  Real y[k1];
  Real y[k2];
end TestSystem;

Modelica: TestSystem
class TestSystem
  TestSystemModel u[k1] = TestSystemModel(u[k1]);
  TestSystemModel u[k2] = TestSystemModel(u[k2]);
  Real y[k1];
  Real y[k2];
  Real u[k1];
  Real u[k2];
  Real y[k1];
  Real y[k2];
end TestSystem;
```

③ System Simulation with Modelica Tools

- OpenModelica
 - What is OpenModelica?
 - The past
- OpenModelica Technical Overview
 - OMC, OMShell, OMNotebook, OMEdit, ModelicaML, OMSimulator, OMPython, OMJulia, OMMatlab
- OpenModelica Development Environment
 - MetaModelica
 - The Eclipse Environment (MDT)
- OpenModelica Latest Developments (2019-2020)

Latest Developments (2019-2020) (I)

- 2019 - 2020 - focus on performance, scalability, bug fixes
- OMC & Clients
 - Performance & scalability improvements
 - Bug fixes to OMC, OMEdit, FMI
- OMSimulator
 - Combined FMI & TLM support, SSP support
 - OMEdit GUI support
- General
 - From Feb 2019 - Feb 2020
 - 30+ contributors
 - 929 commits (OpenModelica/OMCompiler/OMEdit)
 - 100 commits (OMSimulator)
 - Releases 1.13.2, 1.14.1

Latest Developments (2019-2020) (II)

■ New Front-End - status

- The new front-end ~98% complete, (see #4138 on Trac)
- 100+ times faster, 5+ times less memory consumption (no array expansions, no expansion of for loops in equations)
- The new front-end also brings better support for libraries
- Developed in line with MCP-0019: Flattening
- Currently 424/424 models from MSL 3.2.3 pass the new front-end
- Two years ago 107/387 models from MSL 3.2.3 passed

■ New Front-End - remaining work

- Small issues remaining with array modifiers (new battery models in MSL master)
- Some issues remaining with replaceable support (Buildings library)
- Making the backend cooperate with the new way the DAE is produced
- Support for state machines
- (Support for MetaModelica)

- OMEdit
 - Faster OMEdit using the new frontend
 - A lot of bug fixes and new usability features
 - Auto completion support
 - GUI for OMSimulator, SPP
 - Supports the standard Windows installation
 - Encryption support

Latest Developments (2019-2020) (IV)

- OMEdit - Redeclare and Replaceable Support
 - Support for redeclare/replaceable is implemented
 - The new front-end is now mature enough to not frustrate users

The image shows two overlapping windows of the OMEdit software. The background window is titled "OMEdit - Component Parameters - pipe in PumpingSystem" and displays the "Parameters" tab for a component named "pipe". The "Class" is "StaticPipe" with the comment "Basic pipe flow model without storage of mass or energy". The "Parameters" section lists various attributes:

- Medium: [empty dropdown]
- FlowModel: 'Modelica.Media.Incompressible.TableBased'
- isCircular: 'Modelica.Media.IdealGases.Common.MixtureGasNasa'
- Geometry: 'Modelica.Media.Air.ReferenceAir.Air_Base'
- nParallel: 'Modelica.Media.Air.ReferenceAir.Air_pT'
- length: 'Modelica.Media.Interfaces.PartialRealCondensingGases'
- diameter: 'Modelica.Media.Interfaces.PartialMixtureMedium'
- crossArea: $\text{Modelica.Constants.pi} * \text{diameter} * \text{diameter} / 4$ m² Inner cross section
- perimeter: $\text{Modelica.Constants.pi} * \text{diameter}$ m Inner perimeter

The foreground window is a smaller instance of the same "Parameters" dialog, showing a dropdown menu for the "FlowModel" parameter. The menu is open, displaying a list of available flow models, with "Modelica.Fluid.Pipes.BaseClasses.FlowModels.NominalLaminarFlow" selected. Other visible options include "Modelica.Fluid.Pipes.BaseClasses.FlowModels.DetailedPipeFlow", "Modelica.Fluid.Pipes.BaseClasses.FlowModels.TurbulentPipeFlow", "Modelica.Fluid.Pipes.BaseClasses.FlowModels.NominalTurbulentPipeFlow", "Modelica.Fluid.Pipes.BaseClasses.FlowModels.PartialGenericPipeFlow", "Modelica.Fluid.Pipes.BaseClasses.PartialTwoPortFlow.FlowModel", "Modelica.Fluid.Examples.HeatExchanger.BaseClasses.BasicHX.FlowModel_2", and "Modelica.Fluid.Examples.HeatExchanger.BaseClasses.BasicHX.FlowModel_1".

Latest Developments (2019-2020) (V)

- **OMEdit - Redeclare and Replaceable Support**
 - OMEdit now uses new API called `getElements` instead of `getComponents` which returns also replaceable elements
 - The NF is used to fully qualify classes so that `choicesAllMatching` via `constrainedby` class is queried correctly
`getAllSubtypeOf($TypeName(AllLoaClasses), PartialMedium);`
 - An ANTLR4 based Modelica parser is used to parse the replaceable elements and their modifiers, OMEdit partially understands Modelica now
 - Redeclared classes/components can be modified via the GUI
- **Remaining issues**
 - Discuss OMEdit changes (changed Component to Element)
 - Fix random crashes
 - Integrate the changes in OpenModelica 1.15.0

- **OpenModelica on Windows**
 - Use clang to speed up compilation and be able to compile bigger models
- **OMC / OMEdit - new API for instance hierarchy editing**
 - Concept testing - work in progress
 - Use the new front-end to instantiate the Model
 - Give the instance tree to OMEdit, automatically generated C++ classes for walking the tree
 - Allow OMEdit to edit the instance tree directly
 - Propagate the instance tree edits to the top level class
 - Build a simulation from the changed instance tree
- **Editor in the Web Browser and OMSimulator in the cloud**
 - Will be implemented as part of a new project - HUBCAP
- **Julia instead of MetaModelica?**
 - Concept testing - work in progress
 - Translate the entire compiler from MetaModelica to Julia
 - See talk later

Thank You!

Questions?

*asodja, sjoelund.se, sebco011, lochel, wbraun, niklwors,
 hubert.thieriot, petar, perost, Frenkel TUD, Unknown, syeas460,
 adeas31, ppriv, ricli576, haklu, dietmarw, levs, mahge930,
 x05andfe, mohsen, nutaro, x02lucpo, florox, x06hener, x07simbj,
 stebr461, x08joekl, x08kimja, Dongliang Li, jhare950, x97davka,
 krsta, edgarlopez, hanke, henjo, wuzhu.chen, fbergero, harka011,
 tmtuomas, bjozac, AlexeyLebedev, x06klasj, ankar, kajny, vasaie_p,
 niemisto, donida, hkiel, davbr, otto@mathcore.com, Kaie Kubjas,
 x06krino, afshe, x06mikbl, leonardo.laguna, petfr, dhedberg, g-
 karbe, x06henma, abhinck, azazi, x02danhe, rruusu, x98petro,
 mater, g-bjoza, x02kajny, g-pavgr, x05andre, vaden, jansilar,
 ericmeyers, x05simel, andsa, leist, choeger, Ariel.Liebman, frisk,
 vaurich, mwalther, mtiller, ptauber, casella, vitalij, hkiel, jank,
 rfranke, mflehmg, crupp2, kbalzereit, marchartung, Andreas,
 Karim, adrpo*

OpenModelica Project

<http://www.OpenModelica.org>