

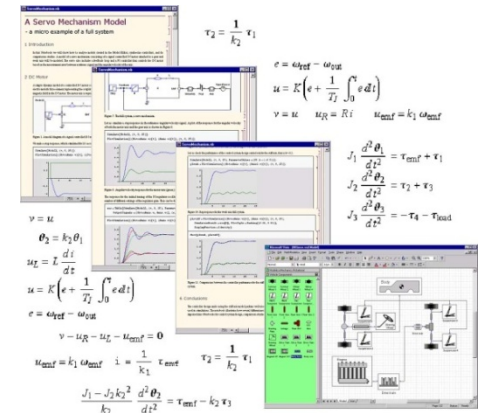
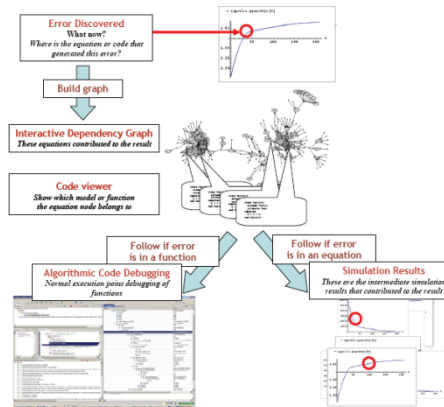
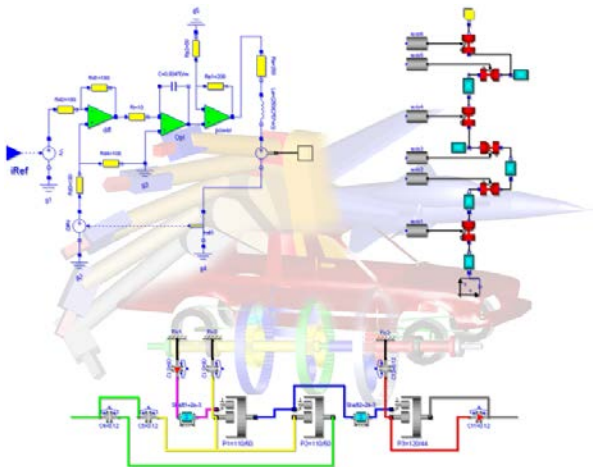
OpenModelica.org

Presentation, Status and Future Developments

Adrian.Pop@liu.se

2021-02-02

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 (2020-2021)

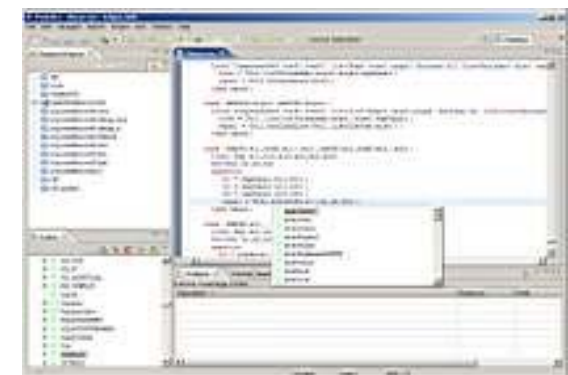
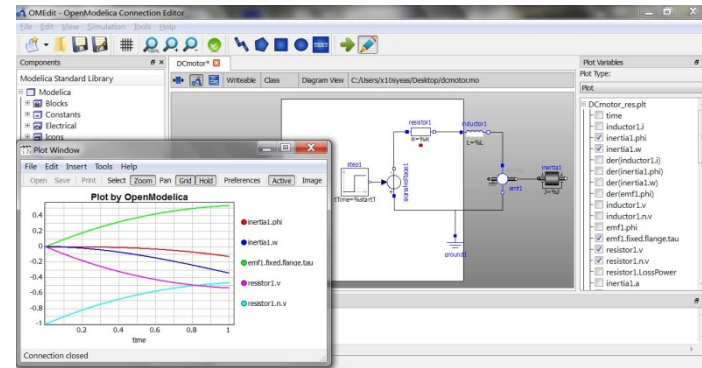
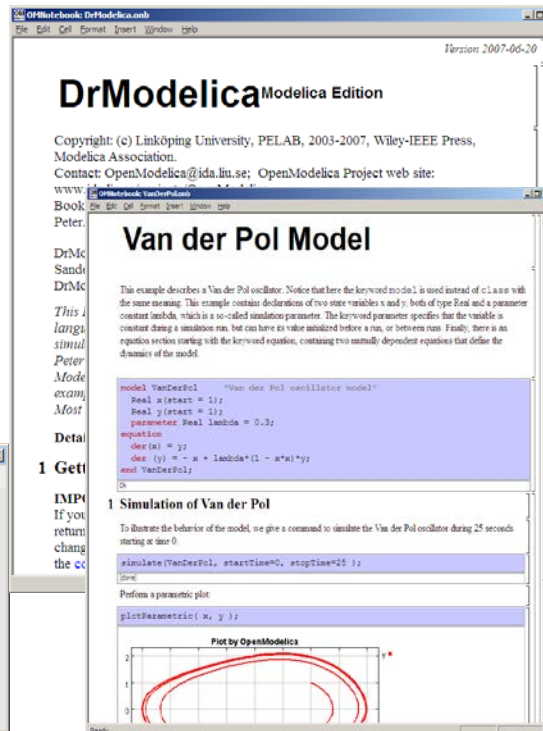
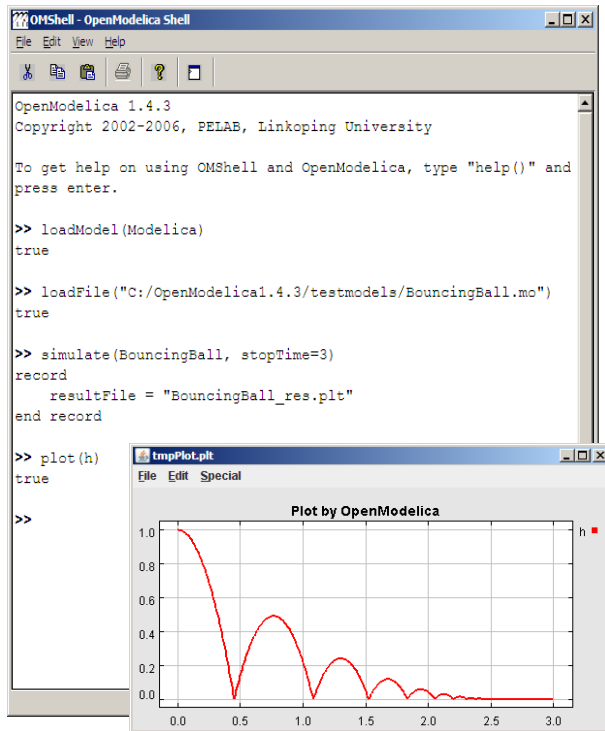
OpenModelica is ... its developers,
testers, bug reporters, contributors
and OSMC members

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.

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.

OMLibraries

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

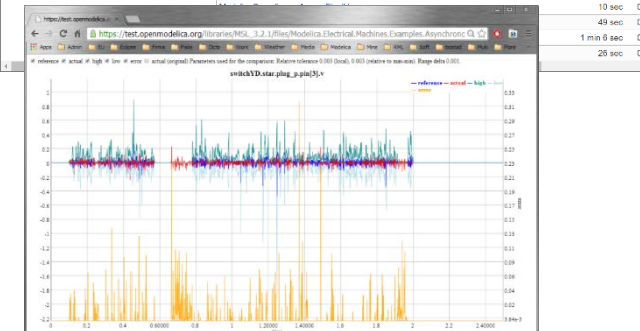
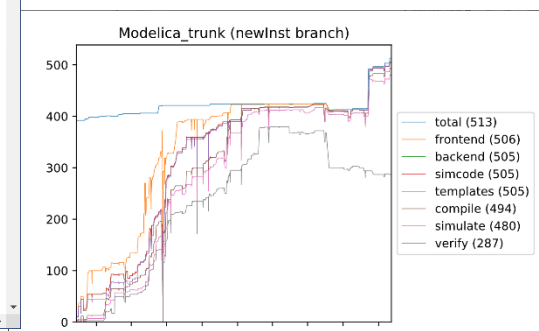
What is OpenModelica? (IV)

- Open-source community services
 - Extensive testing (unit & library coverage: 77 libraries, 17001 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) - (deprecated after 1.16)
 - Platforms: x86, x86_64, ARM
 - 3 runtimes: FMI, C runtime, C++ runtime
 - ~10,449 tests ran on each pull request via Hudson
 - Automatic nightly builds for Window & Linux & Mac OS (deprecated after 1.16)

Package	Duration	Fail
(root)	0 hr 26 min 0 sec	0
ModelicaCompliance	3.9 sec 0.00	0
ModelicaCompliance.Algorithms.Assert	1 min 48 sec 0.00	0
ModelicaCompliance.Algorithms.Assignment	55 sec 0.00	0
ModelicaCompliance.Algorithms.Brakes	52 sec 0.00	0
ModelicaCompliance.Algorithms.For	3 min 5 sec 0.00	0
ModelicaCompliance.Algorithms.If	1 min 43 sec 0.00	0
ModelicaCompliance.Algorithms.Return	11 sec 0.00	0
ModelicaCompliance.Algorithms.Terminate	9.8 sec 0.00	0
ModelicaCompliance.Algorithms.While	49 sec 0.00	0
ModelicaCompliance.Algorithms.White	12 sec 0.00	0
ModelicaCompliance.Arrays.Declarations	2 min 42 sec 0.00	0

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, 7:02:18	1461	12322	12322
v1.9.0-rml	1.9.0 (r17627)	2020-11-15 08:03:08 3 days, 19:57:56	6259	16395	16395
v1.9.1	1.9.1 (r22929) (Bootstrapping version)	2021-01-23 04:24:29 16:19:29	742	16441	16441
v1.9.2	1.9.2 (r25115 C++)	2021-01-23 04:37:39 2 days, 1:48:43	4291	16441	16441
v1.9.3	OpenModelica 1.9.3	2018-06-09 07:30:59 2 days, 1:36:35	5762	10832	10832
v1.9	v1.9.7-v1.9.7.3-g634761f81	2021-01-23 04:51:36 1 day, 15:49:42	5267	16441	16441
v1.11	v1.11.0-v1.11.0.8-g8d6991e5b	2021-01-23 05:05:06 1 day, 11:43:55	5395	16441	16441
v1.12	OMCCompiler v1.12.0-v1.12.0.7 ga21325026	2021-01-23 05:18:16 2 days, 1:51:22	5598	16441	16441
v1.13	OMCCompiler v1.13.2	2021-01-23 05:31:29 2 days, 20:44:05	12372	15079	15079
v1.14	OMCCompiler v1.14.2-v1.14.2.6-g52d52477	2021-01-22 10:41:08 3 days, 0:34:32	12575	15079	15079
v1.16	OMCCompiler v1.16.1-v1.16.1.15-gfd2a6c15b	2021-01-22 12:40:03 3 days, 2:17:46	12408	15079	15079
master	OMCCompiler v1.18.0-dev.4-g263a0c58e8	2021-01-29 21:59:28 2 days, 15:46:18	13692	16441	16441

Branch	Total Parsing	Frontend	Backend	SimCode	Templates	Compilation	Simulation	Verification
v1.8.1-rml	15	15	0	0	0	0	0	0
v1.9.0-rml	15	15	0	0	0	0	0	0
v1.9.1	15	15	0	0	0	0	0	0
v1.9.2	15	15	0	0	0	0	0	0
v1.9.3	15	15	0	1	1	0	0	0
v1.9	15	15	0	0	0	0	0	0
v1.11	15	15	0	0	0	0	0	0
v1.12	15	15	0	0	0	0	0	0
v1.13	15	15	2	2	2	2	1	0
v1.14	15	15	2	2	2	2	1	0
v1.16	15	15	2	2	2	2	2	0
master	15	15	2	2	2	2	4	0



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

OpenModelica Roadmap - Past

- 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

- 2019 - 2020 - 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
- General
 - From Feb 2019 - Feb 2020
 - 30+ contributors
 - 929 commits (OpenModelica/OMCompiler/OMCedit)
 - 100 commits (OMSimulator)
 - Releases 1.13.2, 1.14.1

■ 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
- 77 libraries, 17001 models with interactive result comparison.
 - 9 dedicated test servers
 - Linux (GCC & CLANG), Windows (MinGW GCC), Mac OS (GCC) (to be deprecated)
 - Platforms: x86, x86_64, ARM
 - 5 runtimes: FMI, C runtime, C++ runtime, newInst, daeMode

Statistics

Number of libraries 77

Number of models 17001

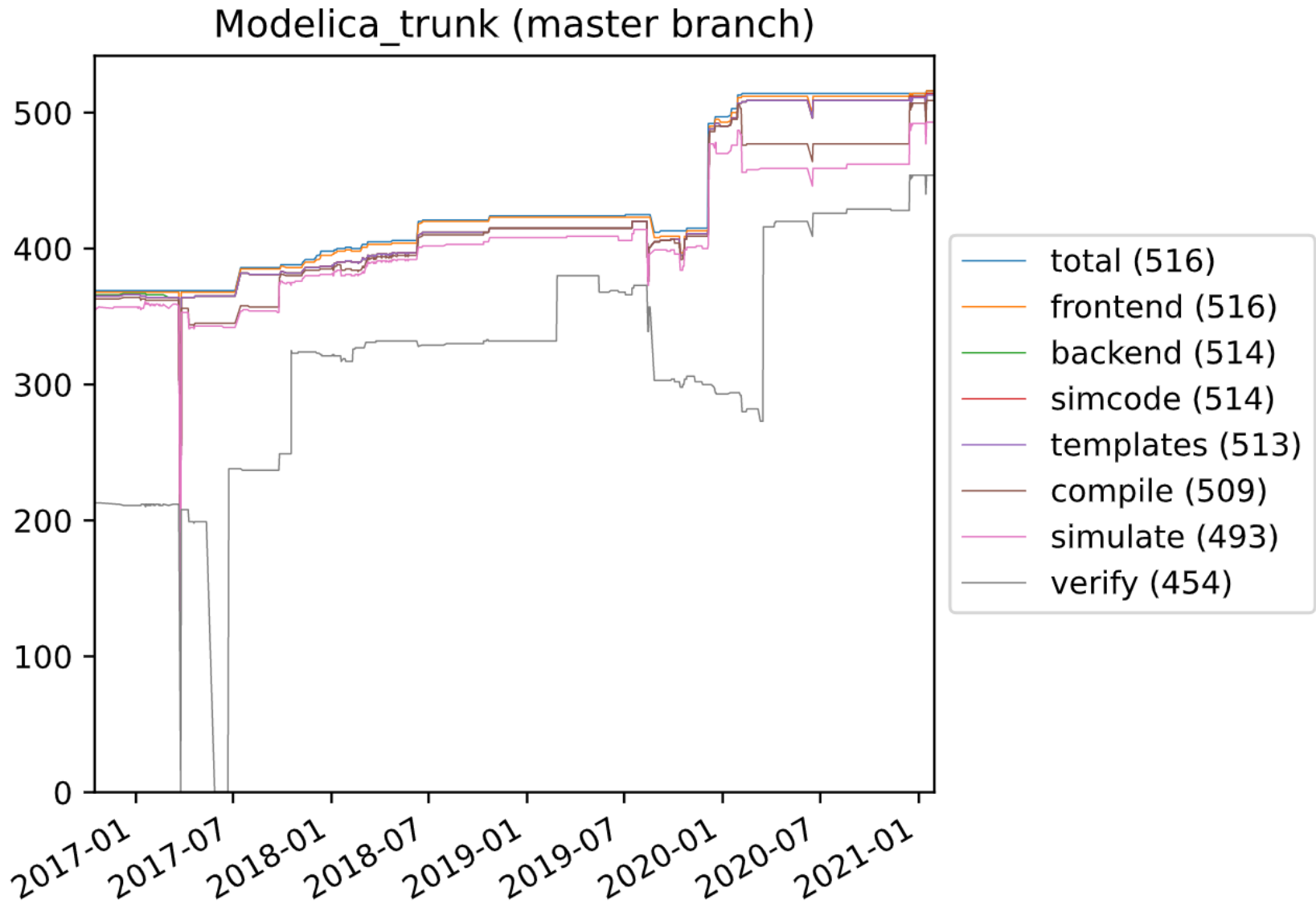
Tested branches

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, 7:02:18	1461	12322
v1.9.0-rml	1.9.0 (r17627)	2020-11-15 08:03:08	3 days, 19:57:56	6259	16395
v1.9.1	1.9.1 (r22929) (Bootstrapping version)	2021-01-23 04:24:29	16:19:29	742	16441
v1.9.2	1.9.2 (r25115 C++)	2021-01-23 04:37:39	2 days, 1:48:43	4291	16441
v1.9.3	OpenModelica 1.9.3	2018-06-09 07:30:59	2 days, 1:36:35	5762	10832
v1.9	v1.9.7-v1.9.7.3+g6347e1f61	2021-01-23 04:51:36	1 day, 15:49:42	5267	16441
v1.11	v1.11.0-v1.11.0.8+gbda991e5b	2021-01-23 05:05:00	1 day, 11:43:55	5395	16441
v1.12	OMCompiler v1.12.0-v1.12.0.7+ga21325026	2021-01-23 05:18:16	2 days, 1:51:22	5598	16441
v1.13	OMCompiler v1.13.2	2021-01-23 05:31:29	2 days, 20:44:05	12372	15079
v1.14	OMCompiler v1.14.2-v1.14.2.6+g5c52d52477	2021-01-22 10:41:08	3 days, 0:34:32	12575	15079
v1.16	OMCompiler v1.16.1-v1.16.1.15+gfd2a6cf15b	2021-01-22 12:40:03	3 days, 2:17:46	12408	15079
master	OMCompiler v1.18.0-dev.5+g263a0e58e8	2021-01-29 21:59:28	2 days, 15:46:18	13692	16441

← 10 hours faster 20

OpenModelica Testing (II)

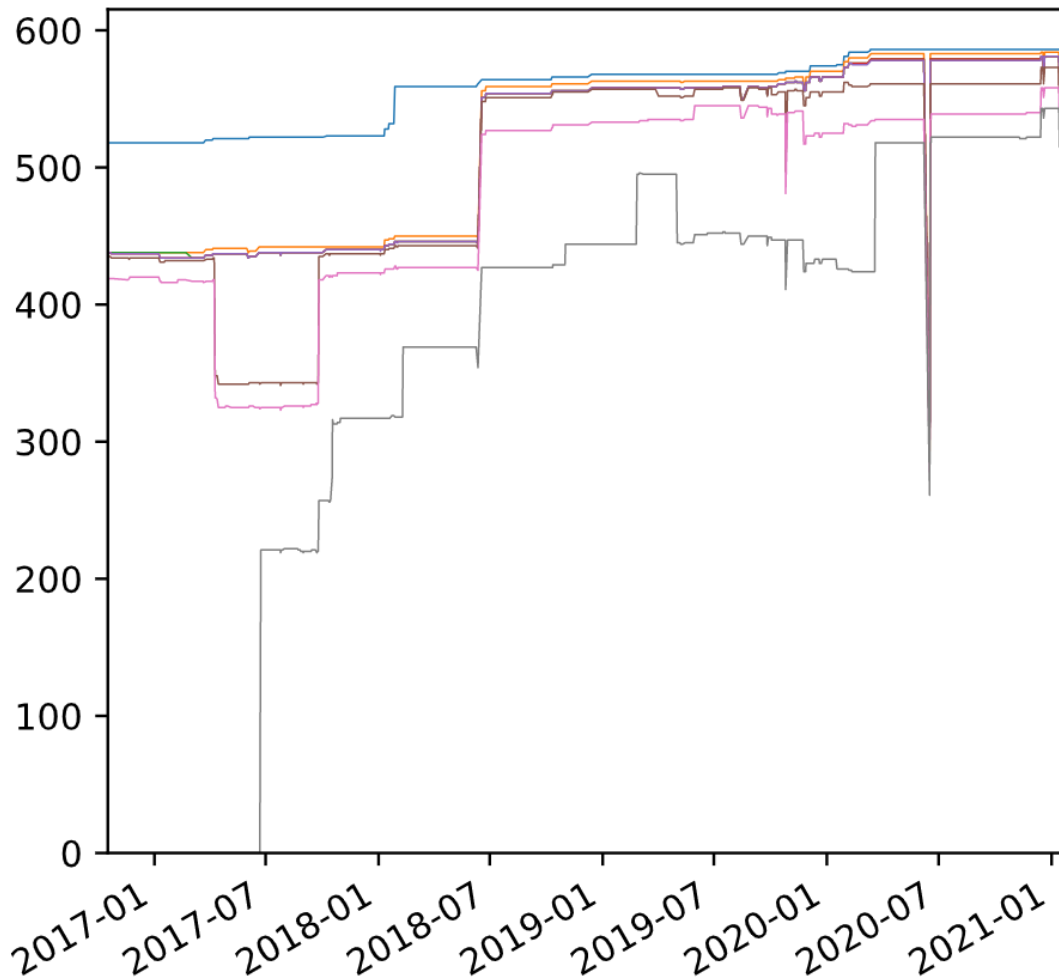
- 2021-02-01 v1.18-dev - total 516 - build 509 (98%) - sim 493 (95%)
- Up 2% since last year



OpenModelica Testing (III)

- 2021-02-01 v1.16-dev - total 586 - build 573 (98%) - sim 560 (95%)
- Up 5% / 8% since last year

ModelicaTest_trunk (master branch)



2021

2020

total (586)	total (581)
frontend (584)	frontend (579)
backend (581)	backend (576)
simcode (581)	simcode (576)
templates (581)	templates (575)
compile (573)	compile (538)
simulate (560)	simulate (516)
verify (546)	verify (415)

- 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)

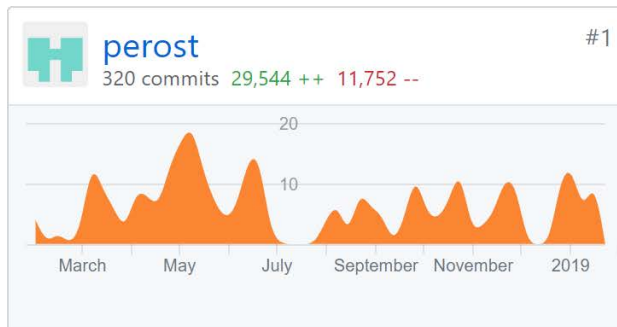
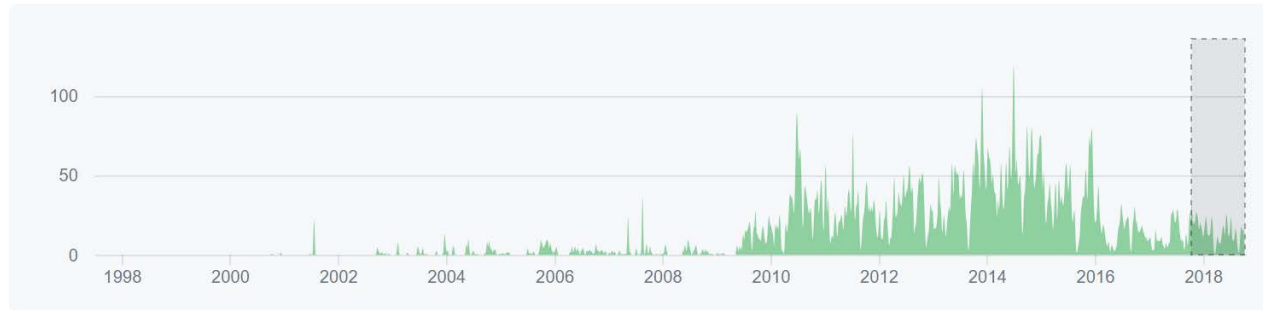
- From Feb 2019 - Feb 2020
 - 30+ contributors
 - 800 commits (OMCompiler)
 - 459 commits (OMSimulator)
 - 213 commits (OMEdit)

OpenModelica Statistics (II)

Feb 5, 2018 – Feb 3, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits

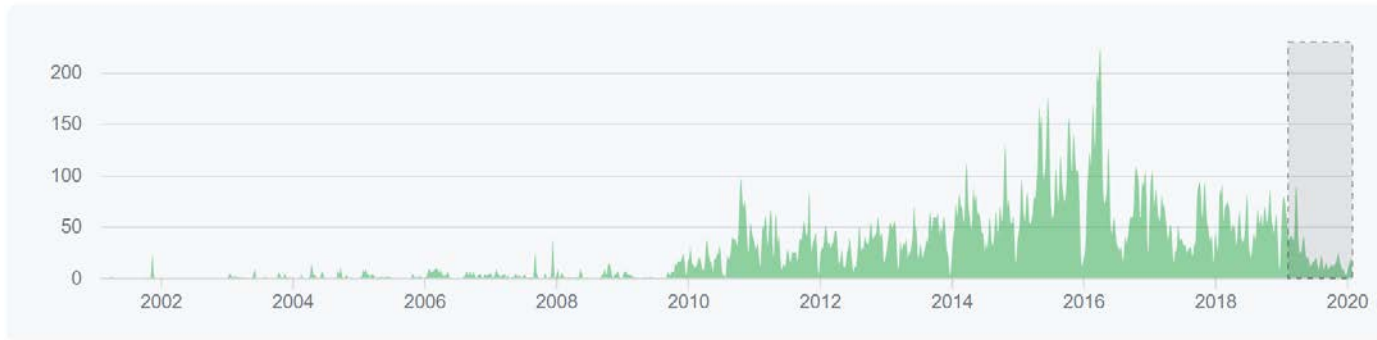


OpenModelica Statistics (III)

Feb 12, 2019 – Feb 3, 2020

Contributions: Commits ▾

Contributions to master, excluding merge commits

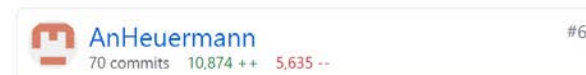
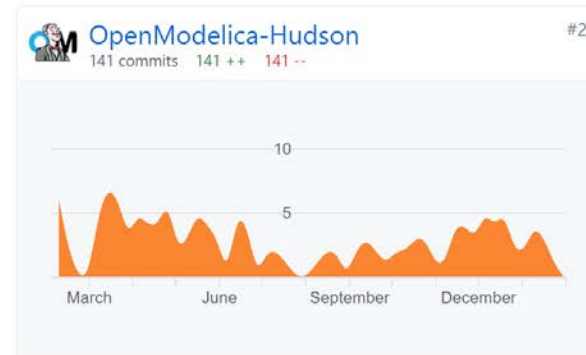
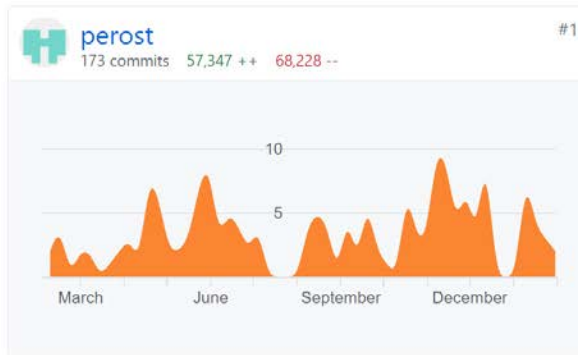
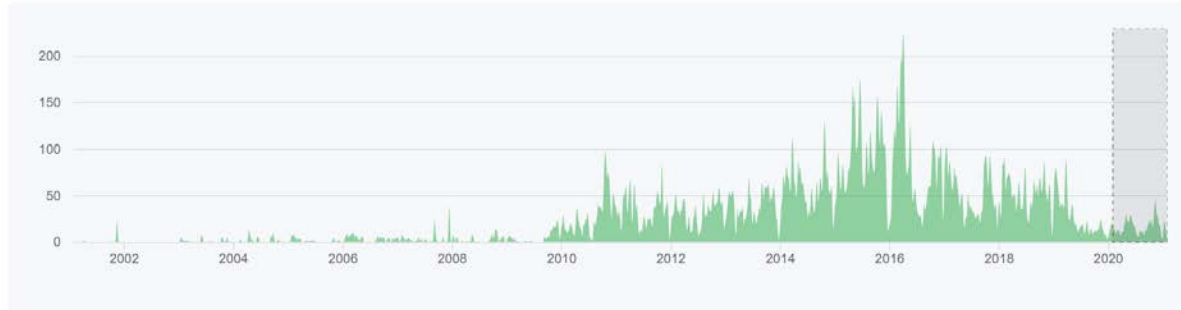


OpenModelica Statistics (IV)

Feb 3, 2020 – Feb 1, 2021

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 (2020-2021)

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

h

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

y

OMEdit- OpenModelica Connection Editor

The screenshot displays the OMEdit - OpenModelica Connection Editor window. The title bar shows the application name and standard window controls. The menu bar includes File, Edit, View, Simulation, FMI, Export, Debug, QMSimulator, Git, Tools, and Help. The Libraries Browser on the left lists various libraries, with 'Elementary' expanded to show 'DoublePendulum'. The main workspace contains a mechanical diagram of a double pendulum system. It features a 'world' frame with a coordinate system (x, y) and gravity vectors. A 'damper' block with a damping coefficient $d=0.1$ is connected to the first mass. The system consists of two masses, 'boxBody1' and 'boxBody2', connected by springs and dampers. The first mass has a position vector $n=\{0, 0, 1\}$ and a center of mass $r=\{0.5, 0, 0\}$. The second mass also has a position vector $n=\{0, 0, 1\}$ and a center of mass $r=\{0.5, 0, 0\}$. The Messages Browser at the bottom is currently empty.

- 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 the Eclipse IDE interface with the 'File' menu open and 'Project...' selected. A 'New Project' wizard dialog is displayed, showing a tree of project types with 'Modelica Project' selected. A 'New Modelica Project' dialog is also visible, with 'demo' entered in the 'Project name' field. Red arrows indicate the flow from the menu to the wizard, then to the selected wizard, and finally to the 'Next >' button.

Modelica - Eclipse SDK

File Edit Refactor Navigate Search Project Run Window Help

New Alt+Shift+N ▶ Project...

Open File... Ctrl+F4

Close Ctrl+Shift+F4

Save Ctrl+S

Save As... Ctrl+Shift+S

Save All Ctrl+P

Revert

Move... F2

Rename... F5

Convert Line Delimiters To

Print... Ctrl+P

Switch Workspace...

Import

New Project

Select a wizard

Create a new Modelica project.

Wizards:

- Plug-in Project
- C
- C++
- CVS
- Eclipse Modeling Framework
- EJB
- Functional Programming
- J2EE
- Java
- Modelica
- Modelica Project
- Plug-in Development
- Simple
- Web
- Examples

New Modelica Project

Create a Modelica project

Create a Modelica project in the workspace.

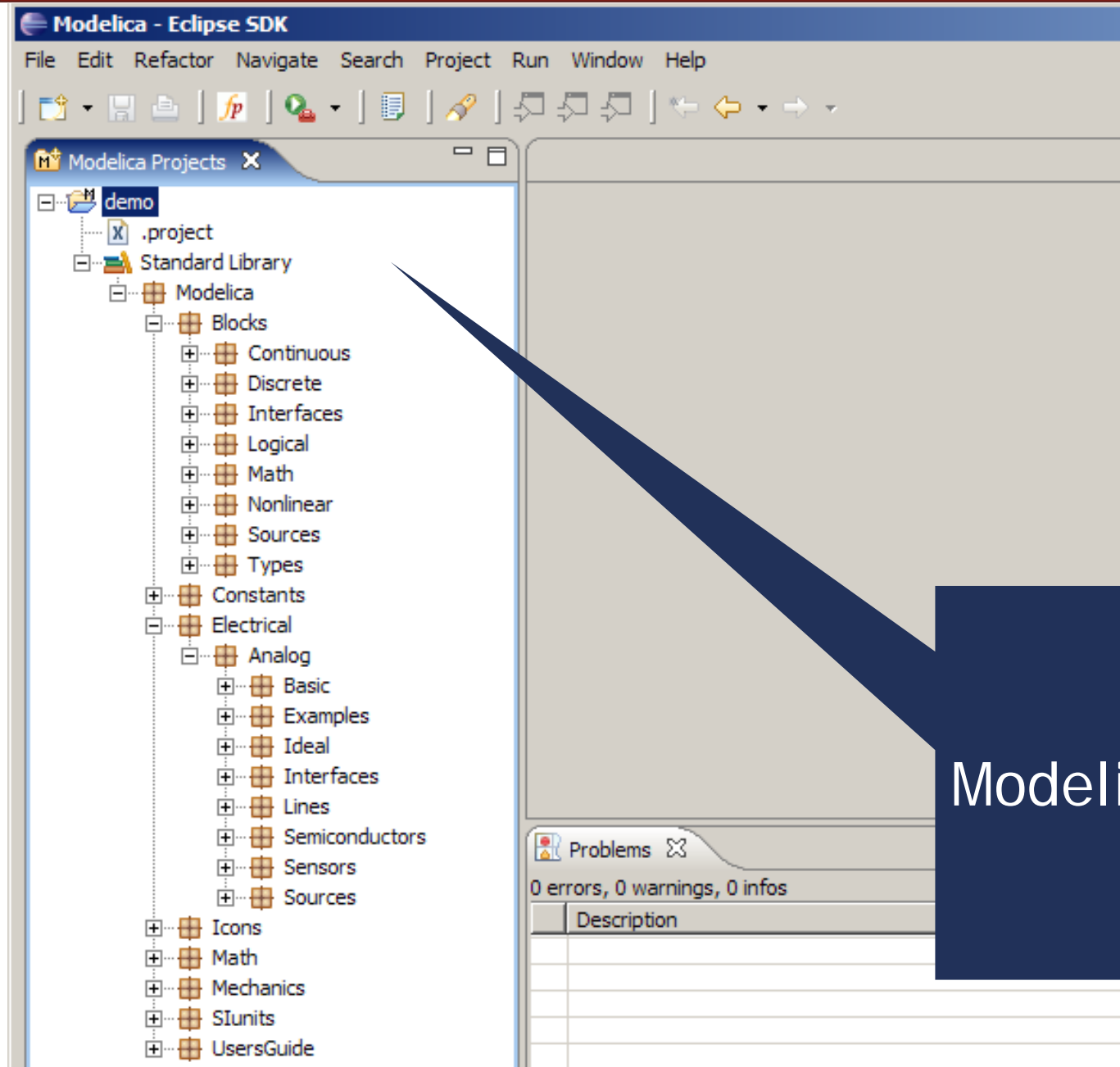
Project name: demo

< Back Next >

< Back Next > Finish Cancel

Creation of Modelica projects using wizards

Creating Modelica projects (II)



Modelica project

Creating Modelica packages

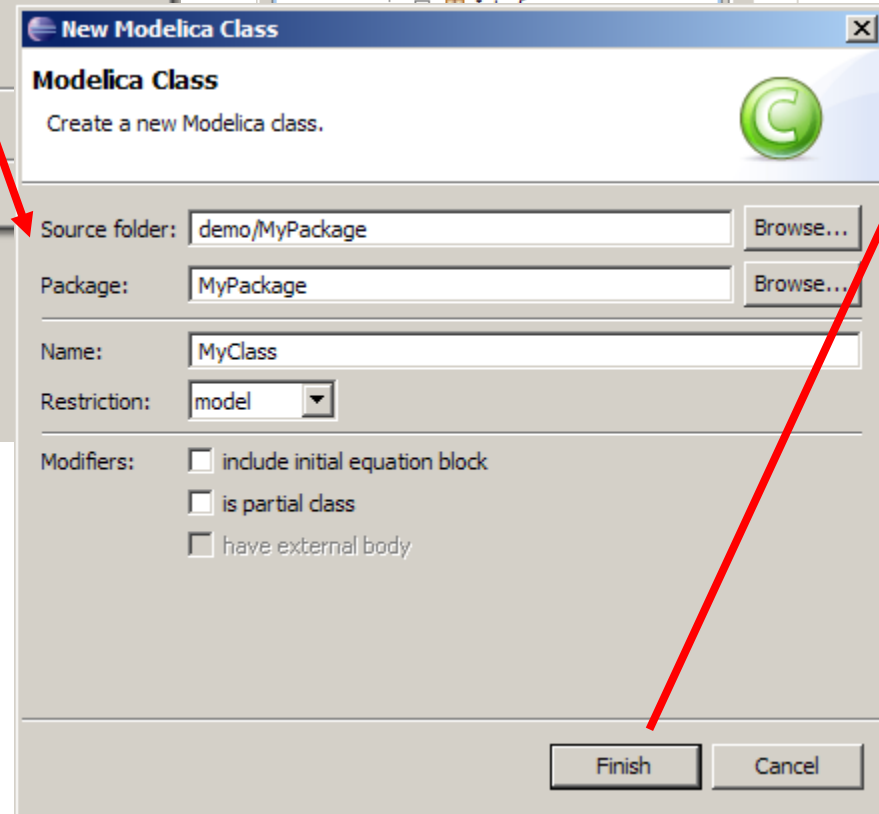
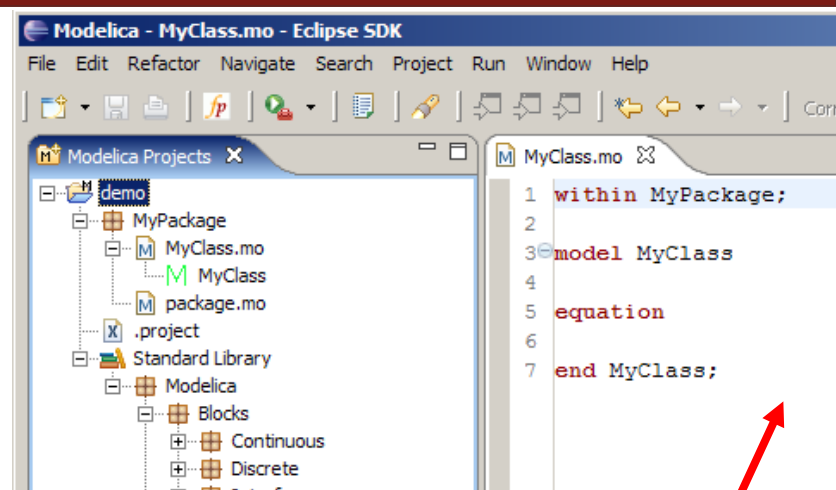
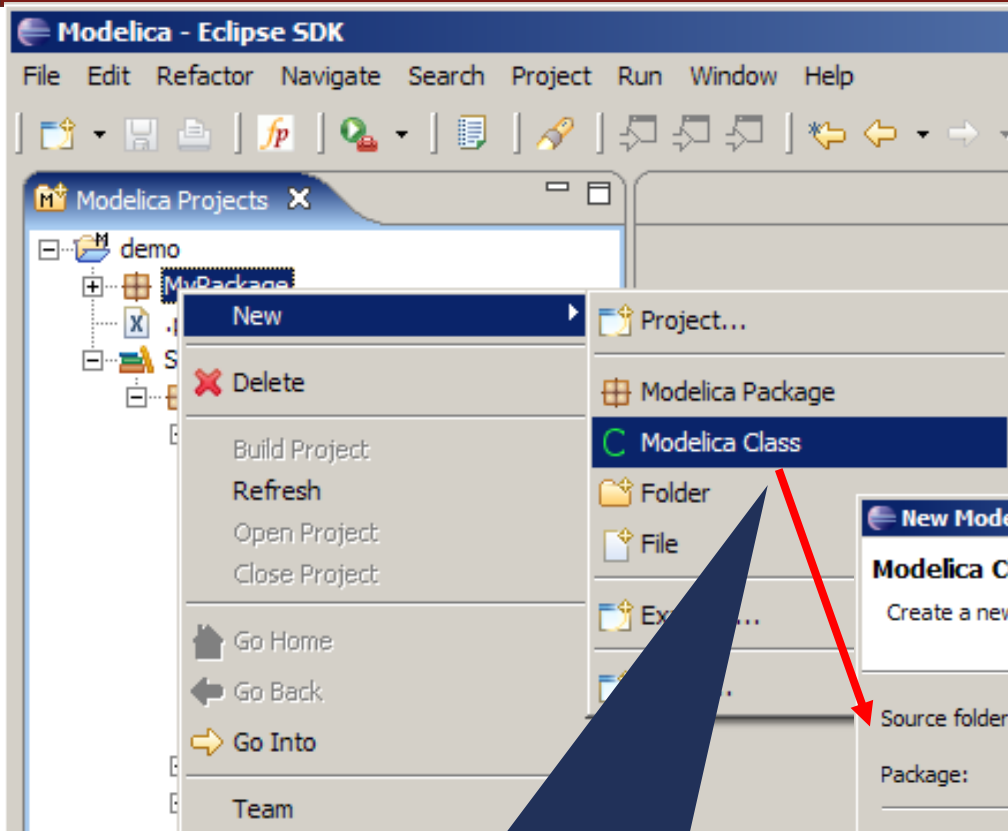
The image shows the Eclipse IDE interface for creating a Modelica package. The 'New' menu is open, and the 'Modelica Package' option is selected. A 'New Modelica Package' wizard dialog is displayed, with the following fields:

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

The 'Finish' button is highlighted with a red arrow. The background shows the 'Modelica Projects' view with a tree structure including 'demo', 'MyPackage', 'package.mo', '.project', 'Standard Library', 'Modelica', 'Blocks', and 'Continuous'.

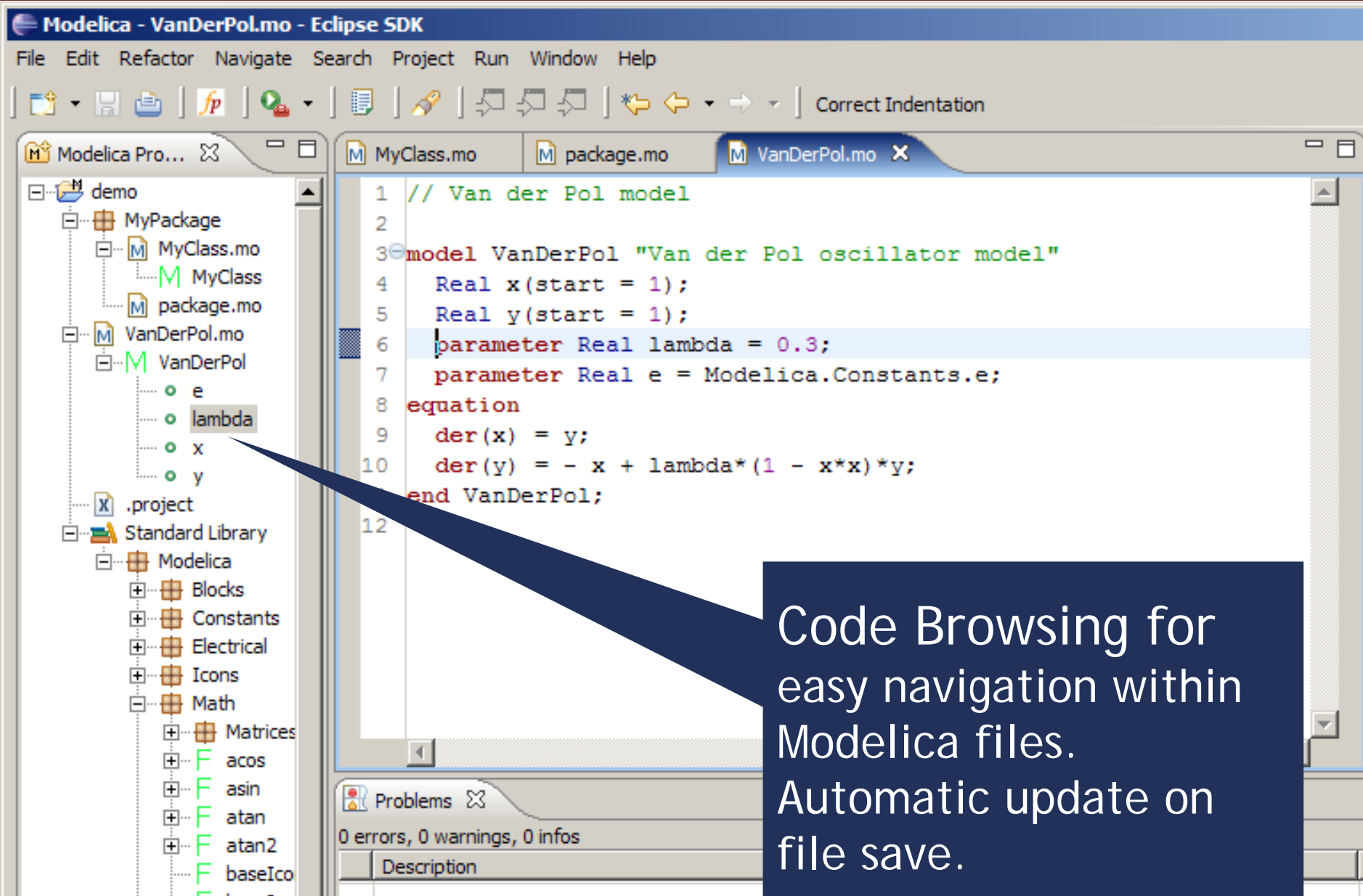
Creation of Modelica packages using wizards

Creating Modelica classes



Creation of Modelica classes, models, etc, using wizards

Code browsing



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

- Menu Bar:** File, Edit, Refactor, Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Includes icons for file operations and a 'Correct Indentation' button.
- Project Explorer (Left):** Shows a project named 'demo' with a tree structure:
 - MyPackage
 - MyClass.mo
 - MyClass
 - package.mo
 - VanDerPol.mo
 - VanDerPol
 - e
 - lambda
 - x
 - y
 - .project
 - Standard Library
 - Modelica
 - Blocks
 - Constants
 - Electrical
 - Icons
 - Math
 - Matrices
 - acos
 - asin
 - atan
 - atan2
 - baseIco

- Editor (Center):** Displays the code in 'VanDerPol.mo'. The code is:

```
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. A blue callout arrow points from the 'lambda' entry in the Project Explorer to this line.
- Problems View (Bottom):** Shows '0 errors, 0 warnings, 0 infos' and a 'Description' tab.

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

Error detection (I)

The screenshot shows the Eclipse IDE interface 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 to show parameters '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:** Located at the bottom, it displays the error message: "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. On the left, the 'Modelica Projects' view displays a tree structure of project folders, with 'Absyn.mo' selected. The main editor window shows the source code of 'Absyn.mo'. The code defines a 'Program' uniontype and a 'PROGRAM' record. Line 77 contains the line: `Withi within_ "within ; Within statement" ;`. A red 'X' icon is visible to the left of this line, indicating an error. Below the editor, the 'Problems' view shows the error message: `Absyn.mo:77.5-77.9 Error: unbound type constructor Withi`. The 'Console' view shows the compilation process, including the command `make` and the error messages: `make[2]: *** [Absyn.h] Error 1`, `make[1]: *** [omc_release] Error 2`, and `make: *** [omc] Error 2`. A blue callout box with a white arrow points to the error message in the Problems view.

```
69 public
70 uniontype Program "- Programs, the top level construct
71 A program is simply a list of class definitions declared at top
72 level in the source file, combined with a within statement that
73 indicates the hieractical position of the program.
74 "
75 record PROGRAM
76 list<Class> classes "classes ; List of classes" ;
77 Withi within_ "within ; Within statement" ;
78 end PROGRAM;
79
```

`<terminated> OMDev-MINGW-OpenModelicaBuilder [Program] c:\OMDev\tools\msys\bin\make.exe`

```
cp -p ../Static.mo Static.mo
cp -p ../SimCodegen.mo SimCodegen.mo
cp -p ../Values.mo Values.mo
cp -p ../System.mo System.mo
/c/OMDev//tools/rml/bin/rmlc -v -Wc,-O3 -c Absyn.mo
"/c/OMDev//tools/rml//bin/rml" -Eplain Absyn.mo
Absyn.mo:77.5-77.9 Error: unbound type constructor Withi
Error: StaticElaborationError
make[2]: Leaving directory `c:/bin/mingw/home/...
make[1]: Leaving directory `c:/bin/cy...
make[2]: *** [Absyn.h] Error 1
make[1]: *** [omc_release] Error 2
make: *** [omc] Error 2
```

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 model 'VanDerPol' with parameters 'e', 'lambda', 'x', and 'y'.
- Editor (Center):** Displays the 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.
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
```
- Code Assistance (Right):** A dropdown menu is visible over the 'import Modelica.' line, listing various categories: Blocks, Constants, Electrical, Icons, Math, Mechanics, SIunits, and UsersGuide.
- Problems (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:

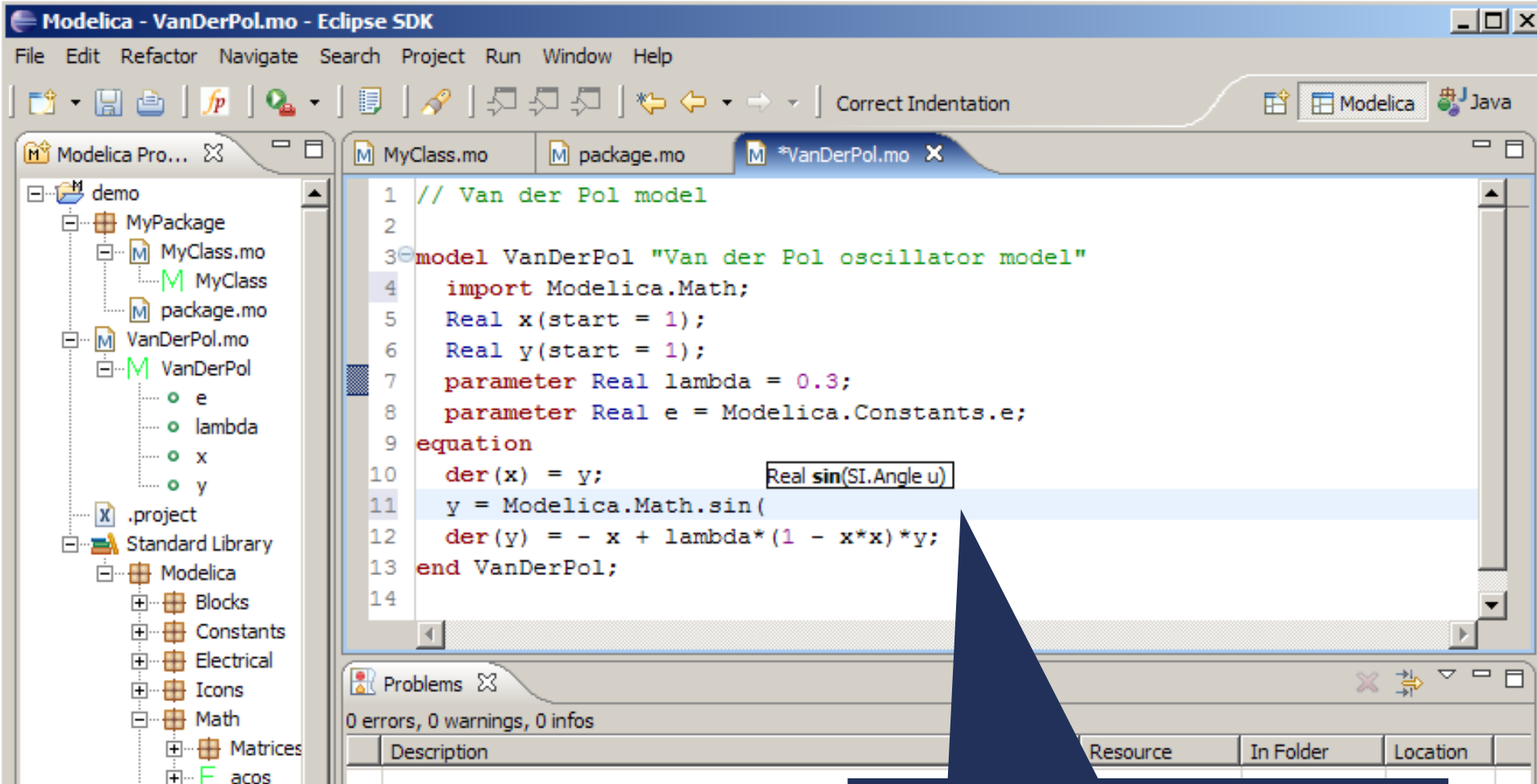
- Project Explorer:** Shows a project named 'demo' containing a package 'MyPackage' with files 'MyClass.mo', 'MyClass', and 'package.mo'. It also contains a file 'VanDerPol.mo' with a sub-package 'VanDerPol' containing parameters 'e', 'lambda', 'x', and 'y'. A 'Standard Library' is also visible with sub-packages like 'Modelica', 'Blocks', 'Constants', 'Electrical', 'Icons', 'Math', 'Matrices', and functions like 'acos' and 'asin'.
- Editor:** Displays the 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 list of completion suggestions is shown on the right, including 'c', 'D2R', 'e', 'eps', 'epsilon_0', 'G', 'g_n', 'h', and 'inf'. The suggestion 'e' 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.

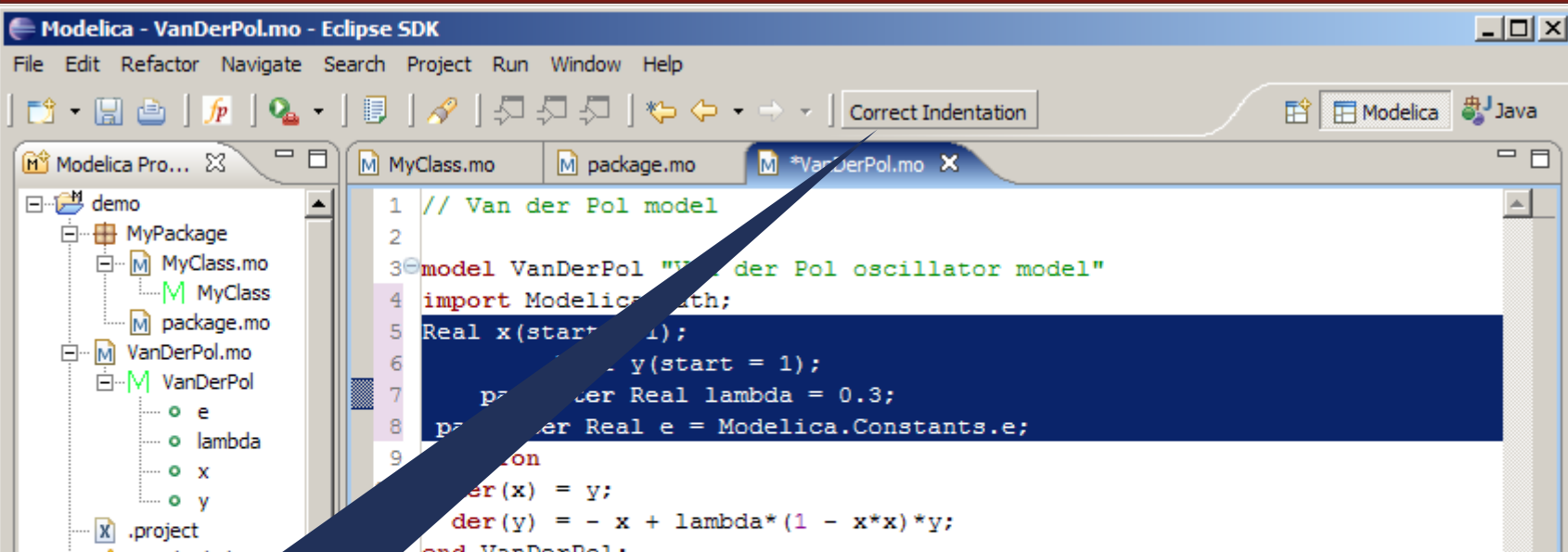
Code Assistance on assignments

Code assistance (III)



Code Assistance on
function calls

Code indentation

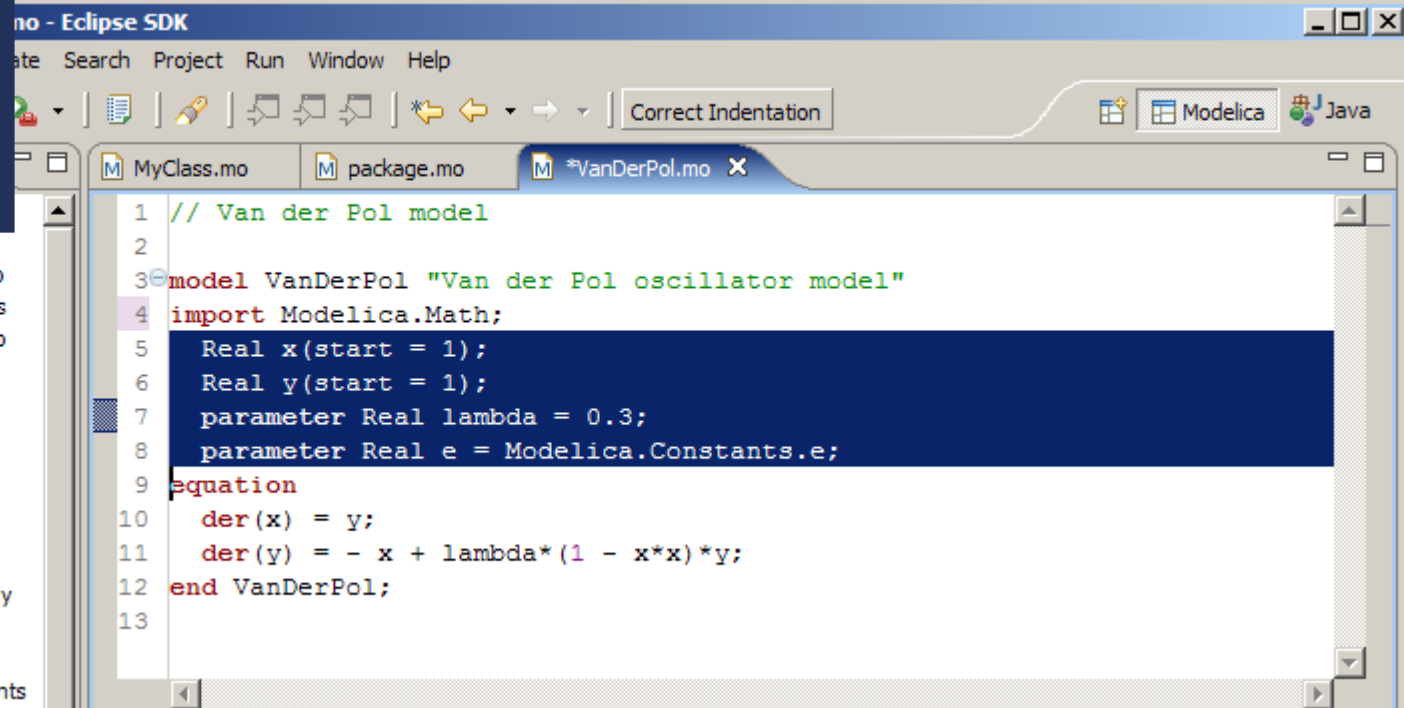


The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu is a toolbar with various icons, including a 'Correct Indentation' button. The main editor window displays the following code:

```
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 code is now properly indented, with the 'equation' block and its contents indented relative to the 'model' block.

Code Outline and Hovering Info

Modelica - OpenModelica/Compiler/Absyn.mo - Eclipse SDK

File Edit Navigate Search Project Run Field Assist Window Help

Modelica Projects

- rml2sig
- runtime
- scripts
- test_codegen
- tools
- VC7
- Absyn.mo 3116 2008-02-04 14:44 krsta
- Absyn 3116 2008-02-04 14:44 krsta
- Algorithm.mo 2992 2007-12-22 22:17 adrpo
- Builtin.mo 3585 2008-05-22 07:03 adrpo
- Ceval.mo 3605 2008-05-27 02:48 adrpo
- ClassInf.mo 3496 2008-04-23 11:59 krsta
- ClassLoader.mo 3193 2008-02-15 05:17 adrpo
- Codegen.mo 3585 2008-05-22 07:03 adrpo
- Connect.mo 3584 2008-05-22 06:45 adrpo
- Constants.mo 3011 2007-12-22 22:36 adrpo
- Convert.mo 3496 2008-04-23 11:59 krsta

```
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 cr;
        l2 =
          ComponentRef cr;
    res = listAppend(l1, l2);
  then
```

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 cr;
l2 =
ComponentRef cr;

Outline

- Absyn
 - ADD
 - ALG_ASSIGN(Exp assignComponent, Exp value)
 - ALG_BREAK
 - ALG_CATCH(list<AlgorithmItem> catchBody)
 - ALG_EQUALITY(Algorithm equ)
 - ALG_FAILURE(Algorithm equ)
 - ALG_FOR(ForIterators iterators, list<AlgorithmItem> forBo
 - ALG_GOTO(String labelName)
 - ALG_IF(Exp ifExp, list<AlgorithmItem>
 - ALG_LABEL(String labelName)
 - ALG_NORETCALL(ComponentRef
 - ALG_RETURN
 - ALG_THROW
 - ALG_TRY(list<AlgorithmItem> tryBody
 - ALG_WHEN_A(Exp whenStmt, list<Alg

Problems

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, pa

64M of 254M

Ctrl Contrib (Bottom)

Identifier Info on Hovering

Code Outline for easy navigation within Modelica files

Eclipse Debugging Environment

The screenshot displays the Eclipse IDE with the following components:

- Breakpoints**: Shows no active breakpoints.
- Variables**: A table showing the state of variables during a debug session.
- Console**: Shows the output of the program, including a message: "Parsed program".
- Outline**: Shows the project structure and a list of functions, with `translateFile(list<String> inStringLst)` selected.
- Code Editor**: Shows the source code of `Bla.mo` and `Main.mo`. The current line in `Main.mo` is: `Debug.fprint("dump", "\n----- Parsed program");`

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

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

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 includes several panels:

- Variables Browser:** Lists variables with their comments, line numbers, and locations. For example, 'A' is 'Storage ... section' at line 120, and 'Tref' is 'Referen...utation' at line 124.
- Equations Browser:** Lists equations with their indices, types, and equations. For example, equation 1 is 'initial (assignment) ...*(T0 - Tref)'. It also shows 'Equation Operations' for equation 8, which are 'solved: h0 = cp * (T0 - Tref)'. The 'Depends' panel shows that equation 8 depends on variables 'cp', 'T0', and 'Tref'.
- Source Browser:** Shows the source code file 'C:/Users/adeas31/Desktop/Debugging.mo' with line numbers and code snippets.

Tutorial 1 - tomorrow at ModProd 2021!

- 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 (2020-2021)

Latest Developments (2020-2021) (I)

- 2020 - 2021 - focus on performance, scalability, bug fixes
- OMC & Clients
 - Performance & scalability improvements
 - Bug fixes to OMC, OMEdit, FMI
 - First replaceable support in OMEdit
 - New Fronted by default in 1.16.x
 - Better FMI export
- OMSimulator
 - Combined FMI & TLM support, SSP support
 - OMEdit GUI support
- General
 - From Feb 2020 - Feb 2021
 - 33+ contributors
 - 878 commits (OpenModelica/OMCompiler/OMEdit)
 - 139 commits (OMSimulator)
 - Releases 1.16.x

Latest Developments (2020-2021) (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
- 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)

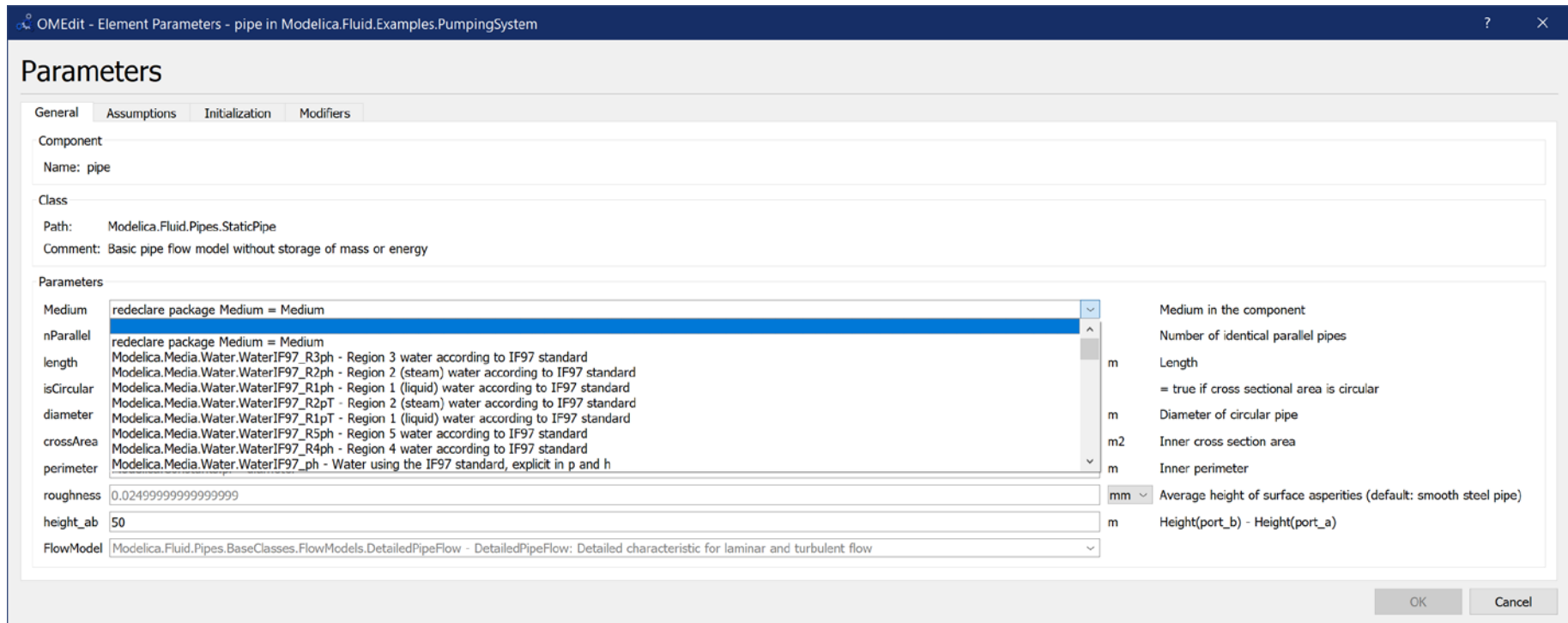
Latest Developments (2020-2021) (III)

- **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
- **OMSimulator**
 - Better OMEdit support
 - Improved SSP support

Latest Developments (2020-2021) (IV)

■ OMEdit - Redeclare and Replaceable Support

- Support for redeclare/replaceable is available since 1.16.x
- The new front-end is mature enough to not frustrate users
- Edit the parameters of replaceable will be available in 1.18.x



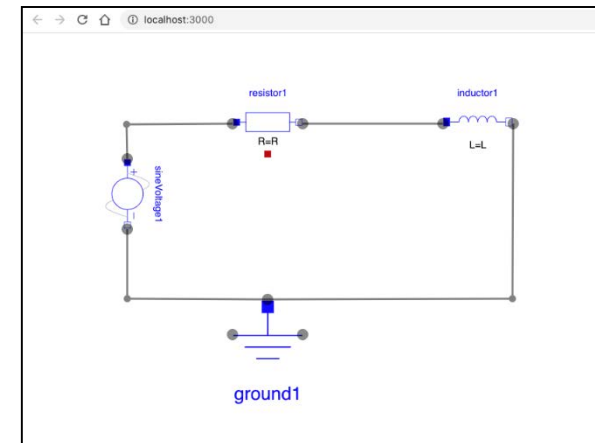
■ OpenModelica on Windows

- Use clang to speed up builds and compile bigger models - available with 1.17.x

- OMC / OMCedit - new API for instance hierarchy editing
 - Faster model display and graphical editing
 - Use the new front-end to instantiate the Model (once!)
 - Give the instance tree (including typed annotations) to OMCedit
 - automatically generated C++ classes for walking the tree
 - Allow OMCedit to edit the instance tree directly
 - Propagate the instance tree edits to the top level class
 - Build a simulation from the changed instance tree

- Web Browser Editor and OMSimulator in the cloud
 - Part of HUBCAP project
 - POC should be available soon

- Julia instead of MetaModelica?
 - OpenModelica front-end translated to Julia
 - Back-end in Julia, support for VSS ongoing
 - Talk by John Tinnerholm (already presented)



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, darbr, 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>