

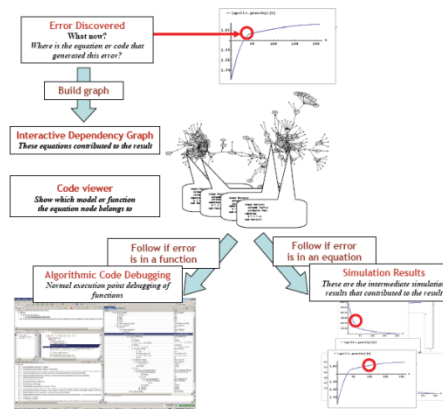
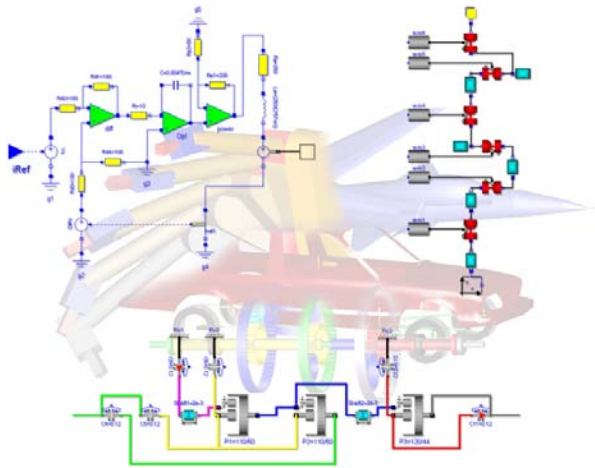
# Technical Overview of OpenModelica and its Development Environment

Adrian Pop

2011-02-07

Open Source Modelica Consortium  
 Programming Environment Laboratory  
 Department of Computer and Information Science  
 Linköping University

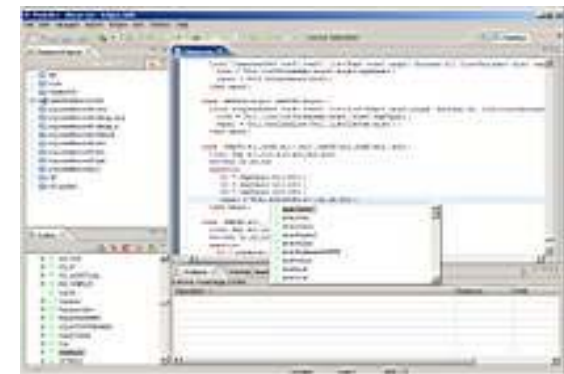
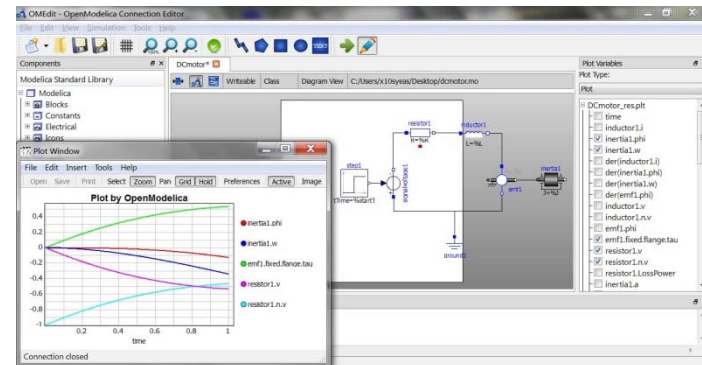
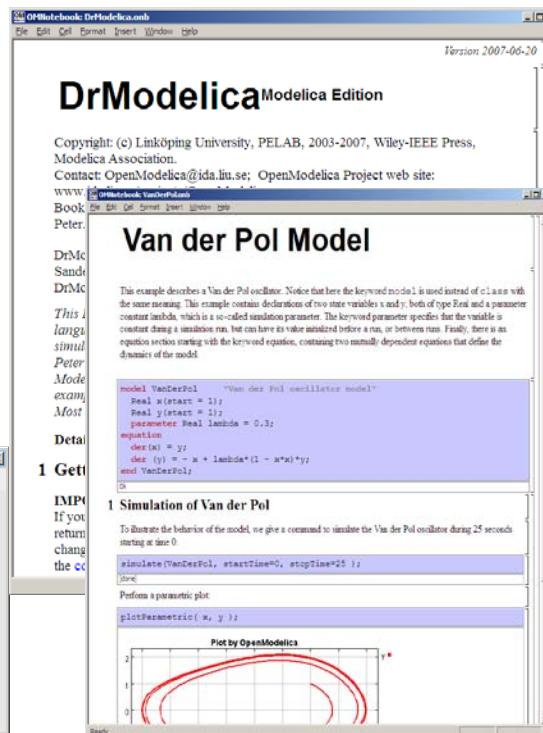
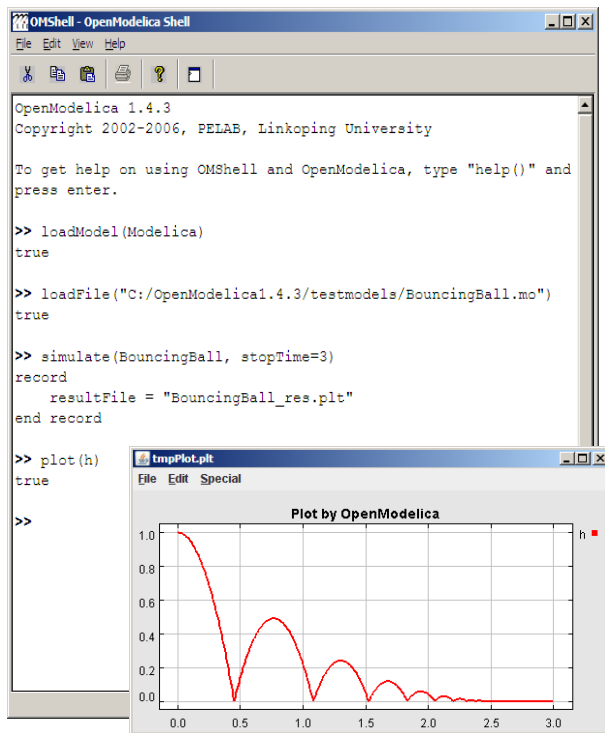
[www.OpenModelica.org](http://www.OpenModelica.org)



- **OpenModelica**
  - What is OpenModelica?
  - The past and present
- **OpenModelica Technical Overview**
  - OMC, OMShell, OMNotebook,
  - OMEdit, ModelicaML, SimForge
- **OpenModelica Development Environment**
  - MetaModelica (RML/OMC)
  - The Eclipse Environment (MDT)
- **OpenModelica Latest Developments (2010-2011)**

# What is OpenModelica? (I)

- Advanced Interactive Modelica compiler (OMC)
  - Supports MLS v. 3.1 (without Media & Fluid)
- Basic environments for creating models
  - OMShell - an interactive command handler
  - OMNotebook - a literate programming notebook
  - OMEdit - Open Modelica Connection Editor (New)
  - MDT - an advanced textual environment in Eclipse



# 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 8 OpenModelica Development Courses (INRIA, PELAB)
- **ModelicaML UML/SysML integration (New)**

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
- Version-controlled source base
- Bug database (unfortunately)
- Development courses

The screenshot shows the OpenModelica website homepage. The header features the OpenModelica logo and navigation links: HOME, DEVELOPER, FORUM, DOWNLOAD, CONTACT US, WORKSHOP, RESEARCH. A search bar is also present. The main content area is divided into sections: 'Top information' with a 'New OpenModelica website is up' announcement, 'Registration' with a call to action, and 'Introduction' with a paragraph about the project's goals. A 'Latest news' sidebar lists recent updates like 'Feb 5: OpenModelica Release 1.5.0 RC2'. An 'Upcoming Events' section mentions 'OpenModelica Workshop 2010'. A 'Note' at the bottom left states that registration is required to receive mailing lists. A small image of a 'Rectifier' circuit simulation is also visible.

The screenshot shows a 'Log Messages' window from a version control system. It displays a list of revisions with columns for Revision, Author, Date, and Message. The messages describe updates to the OpenModelica project, including updates to the OMScheme project, Linux test suite files, and various bug fixes and minor changes. A table below the log shows the file paths affected by the selected revision.

Action	Path
Added	/trunk/Compiler/VC7/mil/runtime/milruntime_vobj.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/arr-add.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/arr-create.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/arr-length.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/arr-elt.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/arr-nth.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/arr-senth.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/arr-update.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/arr-vec.c
Added	/trunk/Compiler/VC7/mil/runtime/runtime/common/array.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/bool-and.c
Deleted	/trunk/Compiler/VC7/mil/runtime/runtime/common/bool-or.c

The screenshot shows the OpenModelica bug tracker interface. The header includes the 'pelab' logo and navigation links: My Start, Projects, Wiki, Documents, Trainers, Reports, Forums, Chats, Builds, Source Code, Members. The main content area displays a list of bugs with columns for ID, Tracker, P, Summary, Status, Resolution, and Submit. The bugs listed include issues related to parameter fixed values, variable declarations, integer arithmetic, component annotations, simulation execution, return values, model validation, examples, and error variables.

ID	Tracker	P	Summary	Status	Resolution	Submit
1163	[1] Bug		parameter(fixed = true) should be treated as a constant during runtime	--	--	
1162	[1] Bug		Backend: Fixed attribute not working properly for variables	New	--	
1161	[1] Bug		Weird integer arithmetics	Resolved	FIXED	
1160	[1] Bug		GetComponentAnnotations() and getWithComponentAnnotation() API are not working with Modelica standard library 3.x	--	--	
1159	[1] Bug		simulation runs old executable when compilation of model fails	Unconfirmed	WORKSFORME	
1158	[1] Bug		mismatch of return values (return + pointer) of external function and result structure	Resolved	FIXED	
1157	[1] Bug		MultiBody Validation a model with Cylindrical joint returns errors but is built successfully (from MathCore)	--	--	
1156	[1] Bug		The examples in Machines and Multiphase fails to check (by MathCore)	--	--	
1155	[1] Bug		Wrong Error Variable eAxis is trying to override final variable in class	Resolved	FIXED	
1164	[1] Bug		Loading of report constructor in modification fails (from MathCore)	Resolved	--	

# What is OpenModelica? (IV)

- **An incubator platform for research**
  - 4 PhDs since 2004 (Debugging, Parallelization, PDEs Extensions)
  - 18 Master's theses since 2004
  - Both the students and the project benefit
- **Master theses at PELAB 2006-2011**
  - 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 Cuda parallel simulation
  - OMEdit - Modelica Connection Editor
- **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, InterCAX (MagicDraw SysML), VTT

# 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
- Much more 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 Booststrapping (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 Booststrapping (Martin Sjölund)
  - Additional Solvers + Events (Willi Braun, FH-Bielefeld)
- **General**
  - New MDT based on Xtext (Antanas Pavlov, SysMO and BMW)
  - 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**

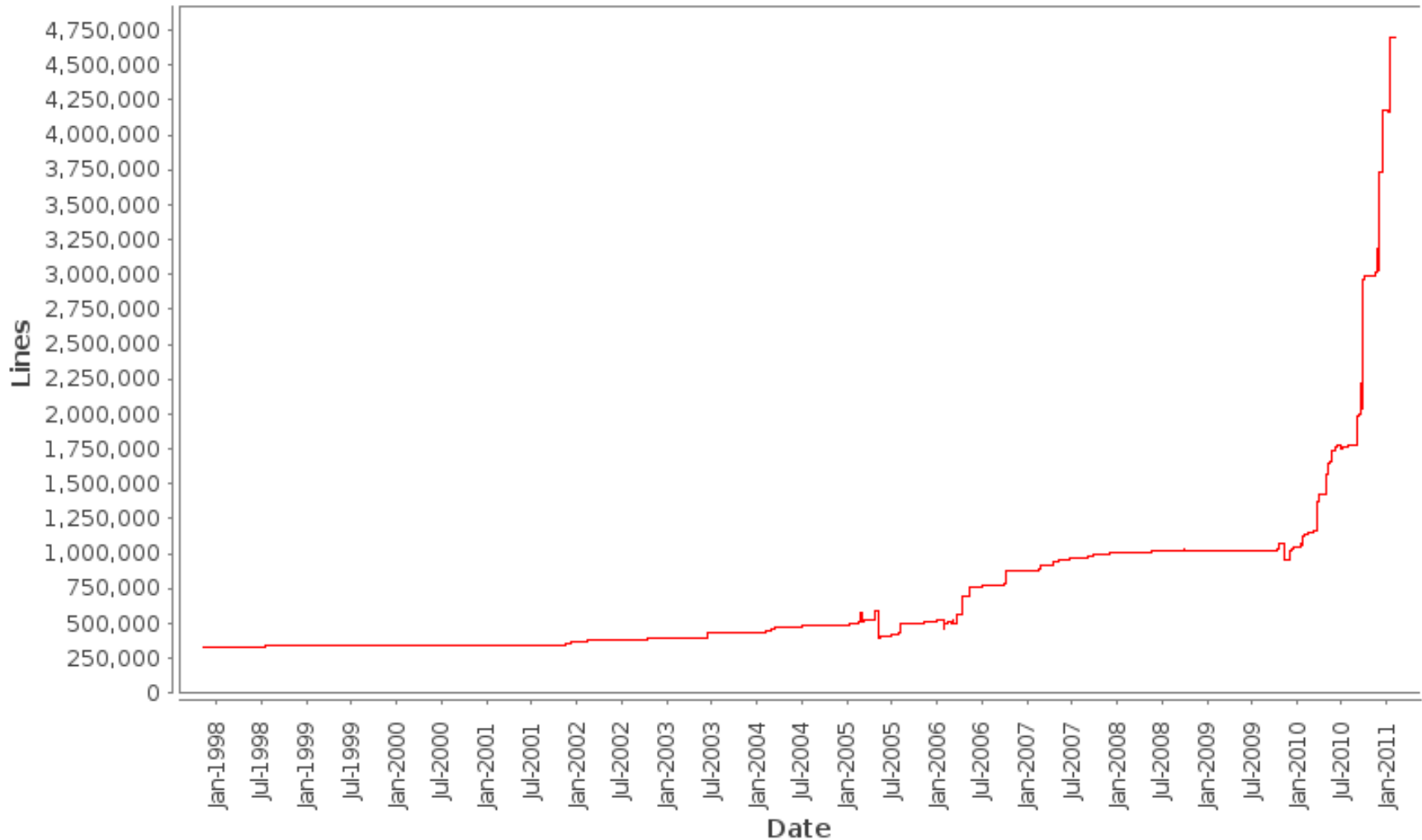
# OpenModelica Roadmap – Past & Present

## 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)
  - 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**

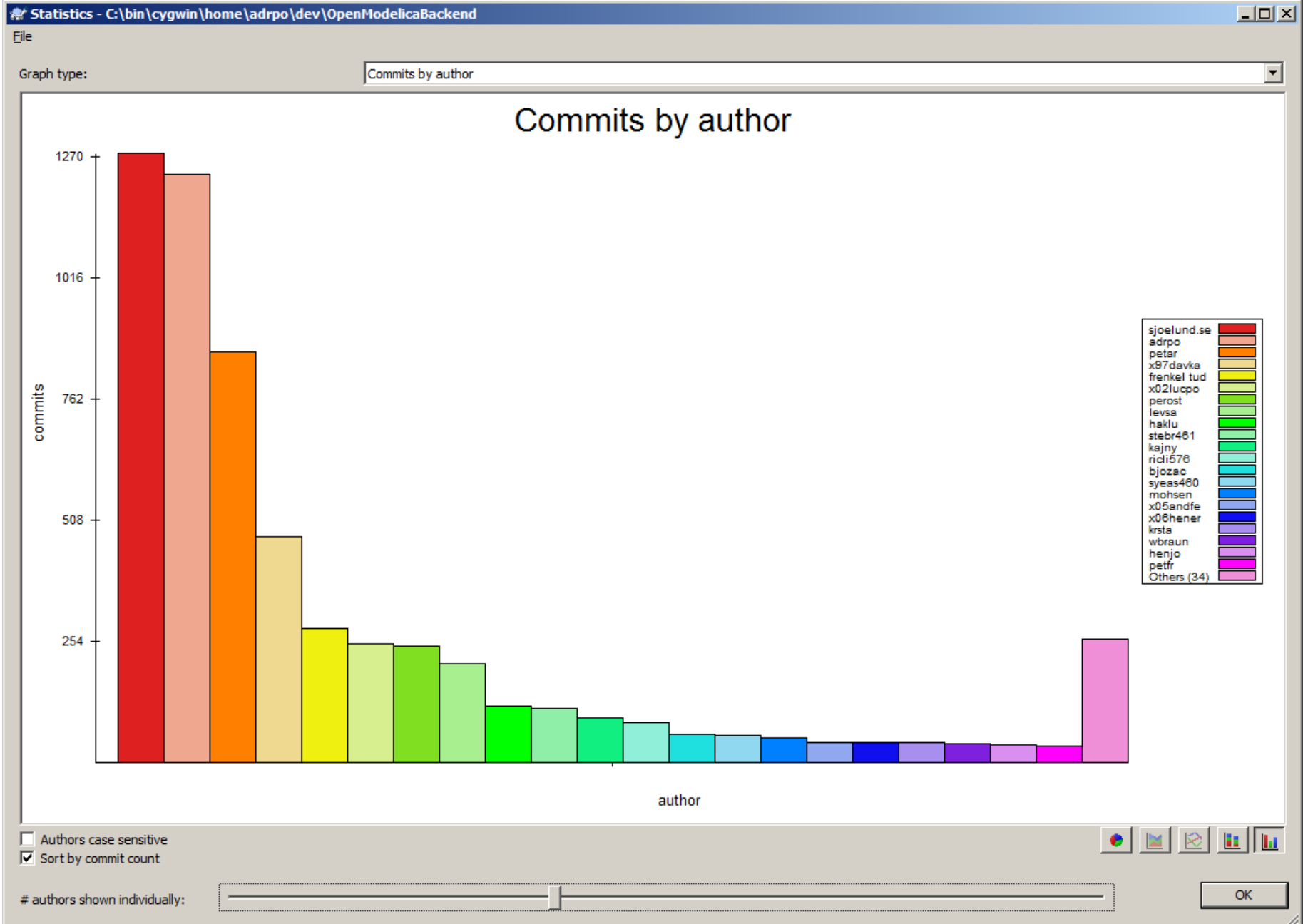
# OpenModelica Statistics (I)

## /trunk: Lines of Code



- Mature code base ([http://build.openmodelica.org/omc/statsvn\\_trunk/](http://build.openmodelica.org/omc/statsvn_trunk/))
- ~ 4000K lines of code and tests, steady increase

# OpenModelica Statistics (II)





- OpenModelica
  - What is OpenModelica?
  - The past and present
- OpenModelica Technical Overview
  - OMC, OMShell, OMNotebook,
  - OMEdit, ModelicaML, SimForge
- OpenModelica Development Environment
  - MetaModelica (RML/OMC)
  - The Eclipse Environment
- OpenModelica Latest Developments (2010-2011)

## ■ Demo?

**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

**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 Association, Peter Fritzson

Contact: OpenModelica  
www.ida.liu.se/~ida  
Book web page  
Peter.Fritzson@ida.liu.se

### 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;
```

Detailed Copy

### 1 Getting Started

**IMPORTANT**  
If you end a command returned in an open notebook, you can change the directory by using the cd() command.

### 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]

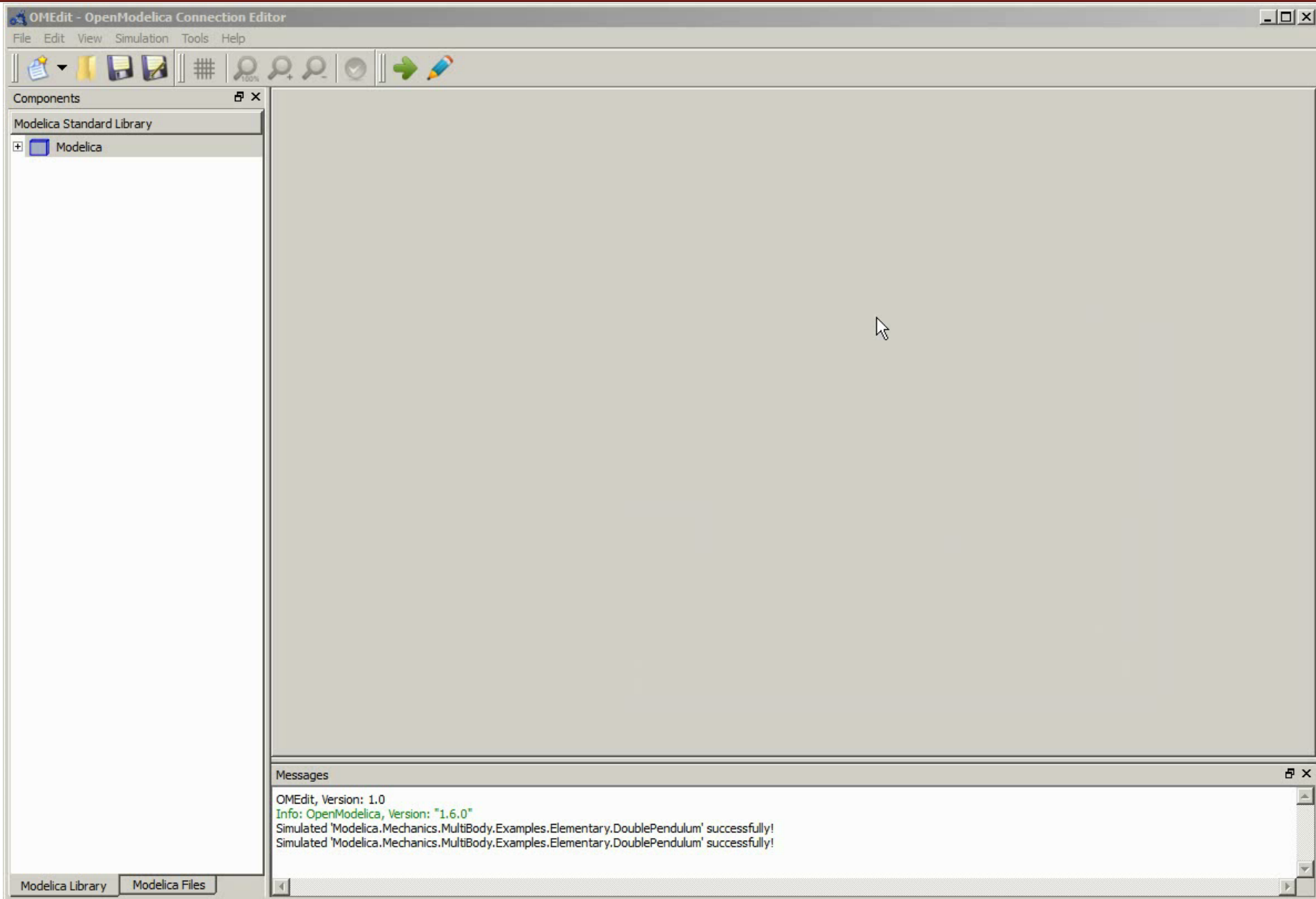
Perform a parametric plot:

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

Plot by OpenModelica

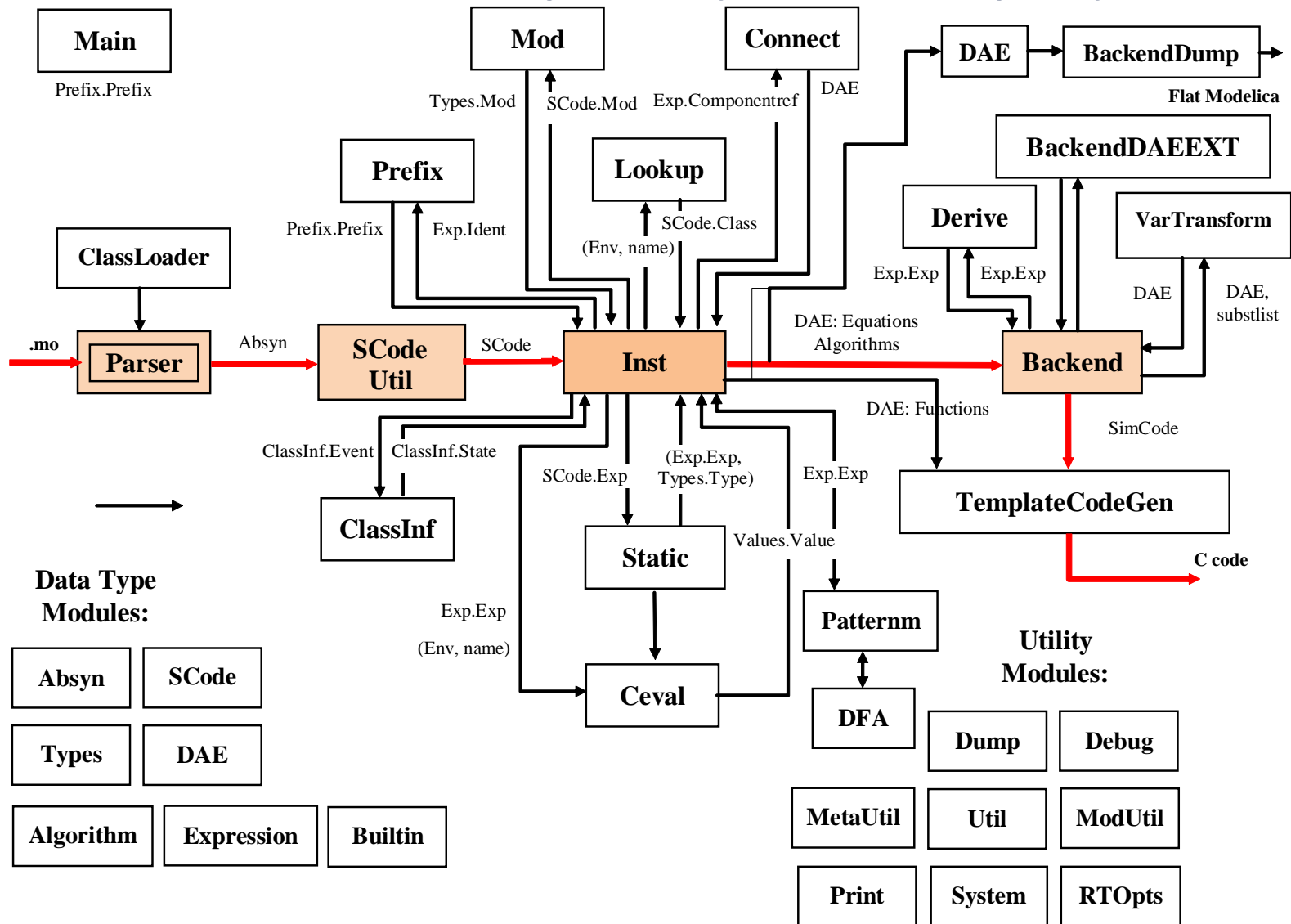


# OMEdit - Demo? Maybe a movie!



# The OMC Compiler

- Implemented mainly in MetaModelica and C/C++
- The compiler has 114 packages (in my local working copy)



# Modelica->AST->SCode->DAE->C Code

```
// Parse the file and get an AST back
```

```
ast = Parse.parse(modelicaFile);
```

```
// Elaborate the file
```

```
scode = SCode.elaborate(ast);
```

```
// flatten the simplified code
```

```
(cache, dae1) = Inst.instantiate(Env.emptyCache, scode);
```

```
// Call the function that optimizes the DAE
```

```
optimizeDae(scode, ast, dae, dae, lastClassName);
```

## Two libraries:

- **libc\_runtime.a**
  - Runtime used by the generated functions in the model
  - Linked with the model
- **libsim.a**
  - Runtime used for simulations, it contains solver implementations and a main function for the simulation

## Executable Model

### OMC Simulation Runtime Library

```
DATA *globalData: simulation_runtime.h
simParams:start,stop,stepSize,
           outputSteps,tolerance, method
```

```
main: simulation_runtime.cpp
```

```
globalData = initializeDataStruc(FLAGS);
setLocalData(globalData);
read_input(globalData, simParams);
switch (method)
  "dassl": dassl_main(simParams);
  "euler": euler_main(simParams);
deInitializeDataStruct(DATA, FLAGS);
```

```
dassl_main: solver_dasrt.cpp
```

```
euler_main: solver_euler.cpp
```

```
read_input: simulation_input.cpp
```

### OMC Generated Code

```
DATA *localData
```

```
initializeDataStruc
```

```
setLocalData
```

```
deInitializeDataStruc
```

# Simulation Runtime Solver

## OMC Simulation Runtime Library

```
DATA *globalData: simulation_runtime.h
simParams: start, stop, stepSize, outputSteps, tolerance, method
    dassl_main: solver_dasrt.cpp
    // set the solver parameters and calculate step from
    simParams
    initializeEventData(); initializeResult(numpoints,
    globalData);
    bound_parameters(); initial_function();
    storeExtrapolationData();
    initialize(init_method);
    function_updateDependents();
    CheckForInitialEvents(globalData->timeValue);
    StartEventIteration(globalData->timeValue);
    // calculate initial derivatives
    functionODE();
    // calculate initial output values
    functionDAE_output(); functionDAE_output2();
    // take a tiny step
    tout = globalData->timeValue + epsilon;
    function_updateDependents(); saveall(); emit();
    calcEnabledZeroCrossings();
    // call the solver for that tiny step
    DDASTR(functionDAE_res, function_zeroCrossing, jroot);
    checkForInitialZeroCrossings(jroot);
    // check if we can continue the simulation
    functionDAE_res(globalData); functionDAE_output();
    // calculate the next step
    tout = newTime(tout, step, stop);
    // enter solver loop

storeExtrapolationData: simulation_runtime.cpp
initializeResult: simulation_result.cpp
emit: simulation_result.cpp
initializeEventData: simulation_events.cpp
CheckForInitialEvents: simulation_events.cpp
StartEventIteration: simulation_events.cpp
saveall: simulation_events.cpp
initialize: simulation_init.cpp
```

## OMC Generated Code

```
DATA *localData

    initializeDataStruc
    setLocalData
    deInitializeDataStruc

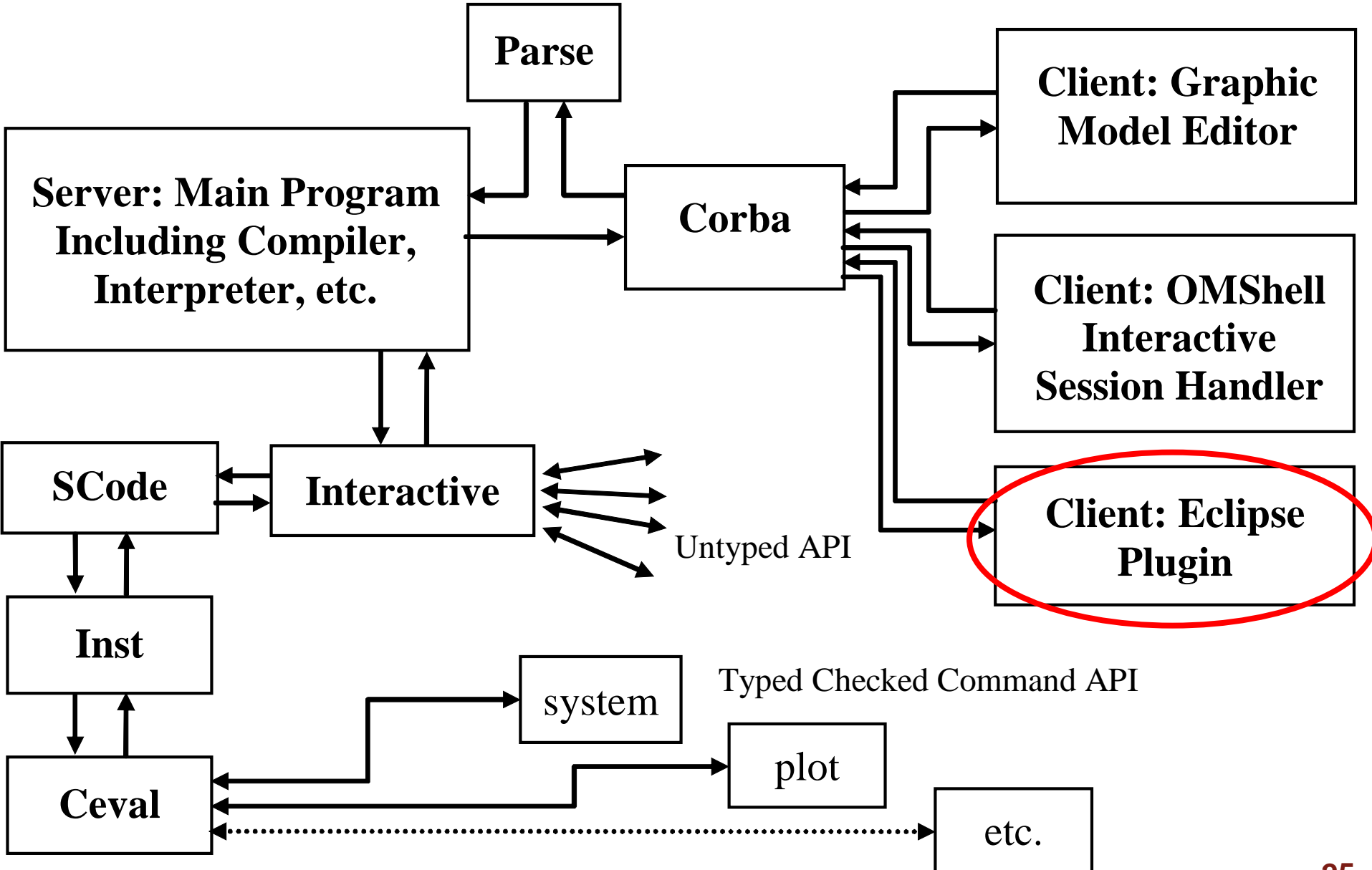
    bound_parameters
    initial_function
    functionODE
    functionDAE_output
    functionDAE_output2
    function_updateDependent
    s
    functionDAE_res
    function_zeroCrossing
```

- OpenModelica
  - What is OpenModelica?
  - The past and present
- OpenModelica Technical Overview
  - OMC, OMShell, OMNotebook
  - OMEdit, ModelicaML, SimForge
- **OpenModelica Development Environment**
  - MetaModelica
  - The Eclipse Environment
- **OpenModelica Latest Developments (2009-2010)**

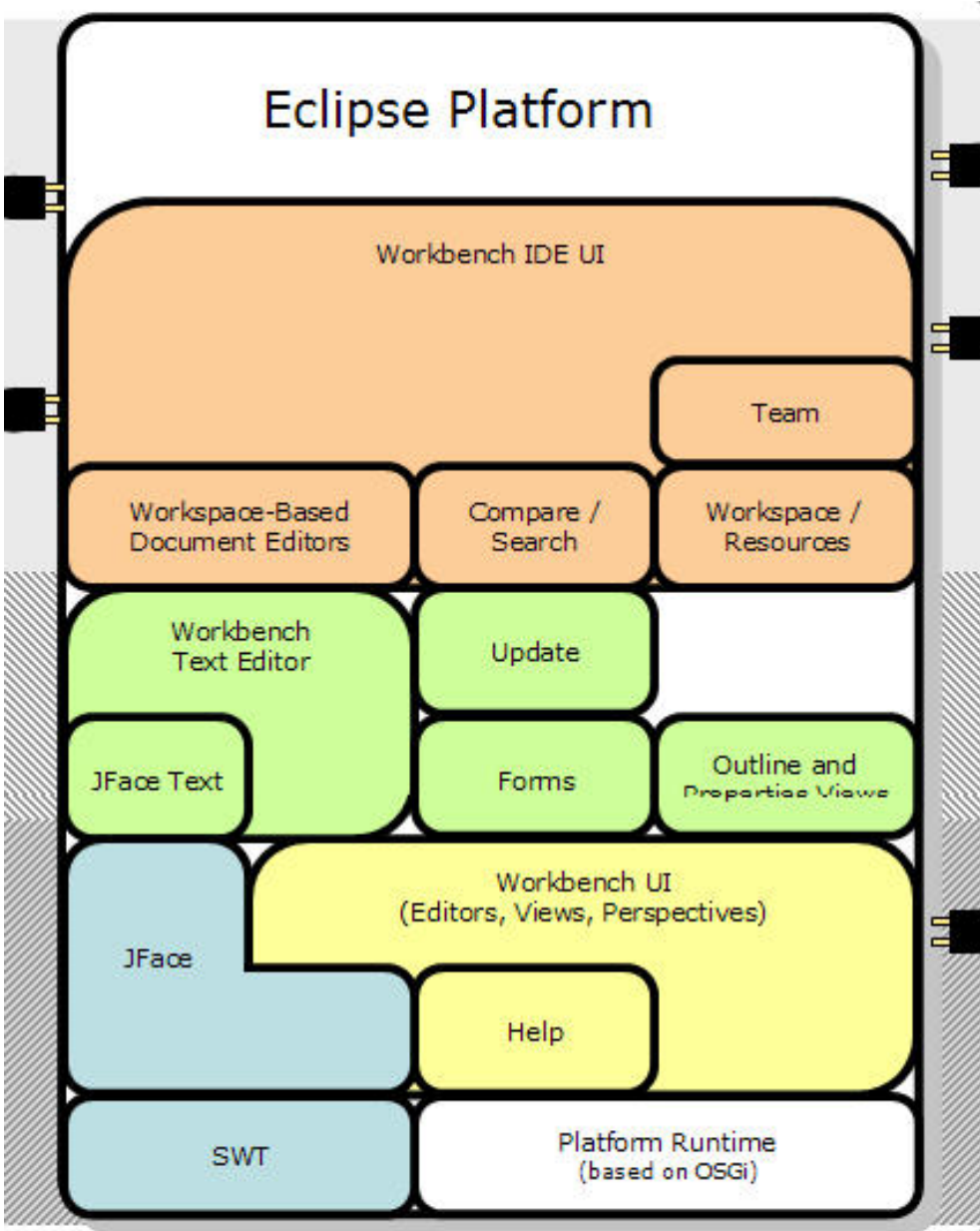
- **OMC**
  - Implemented mainly in MetaModelica and C/C++
- **Modelica**
  - classes, models, records, functions, packages
  - behavior is defined by equations or/and functions
  - equations
    - differential algebraic equations and conditional equations
- **MetaModelica extensions**
  - local equations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



# OpenModelica Context



# The MDT Eclipse Environment (I)



**Modelica Browser**

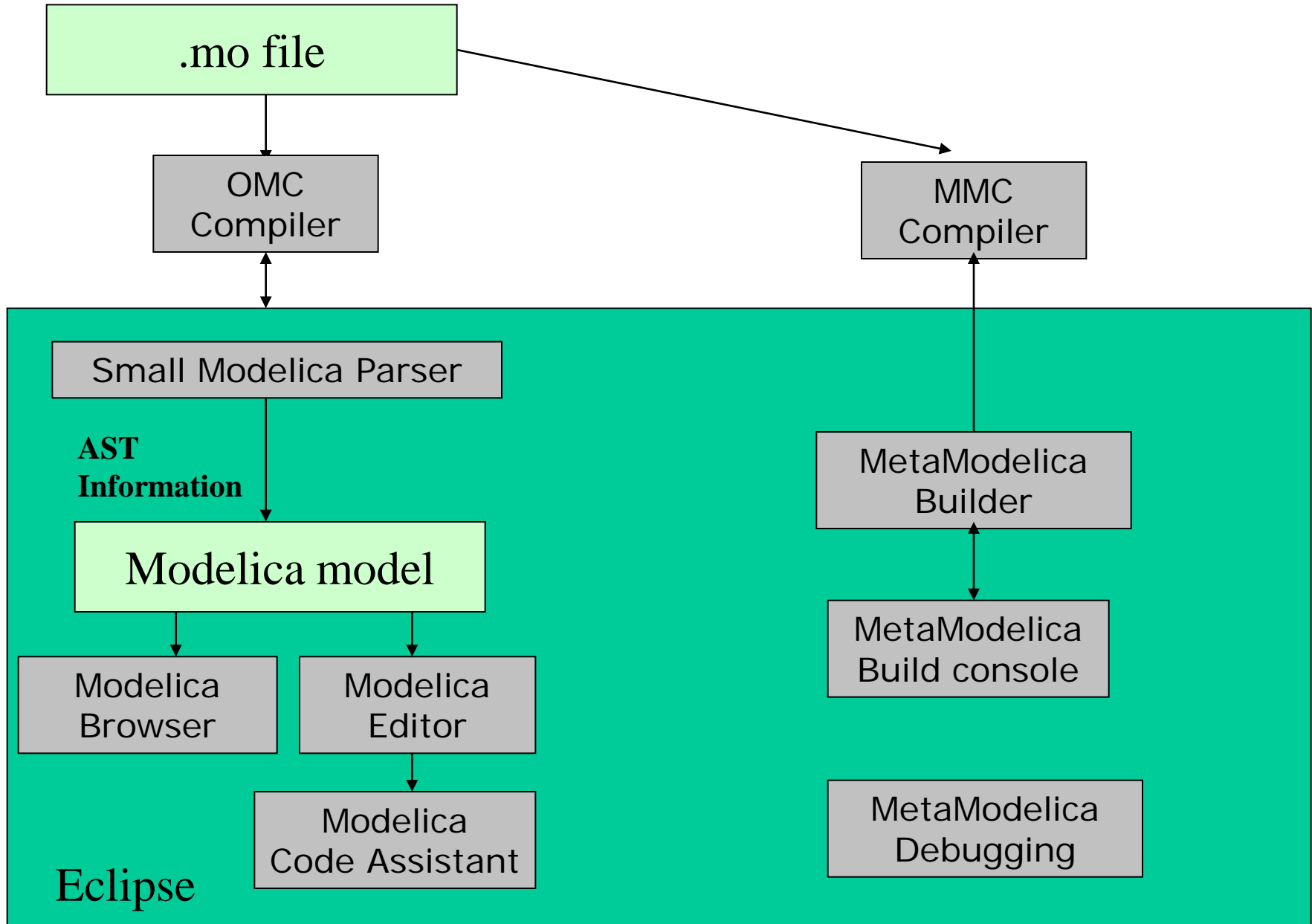
**Modelica Editor**

**Modelica Code Assistant**

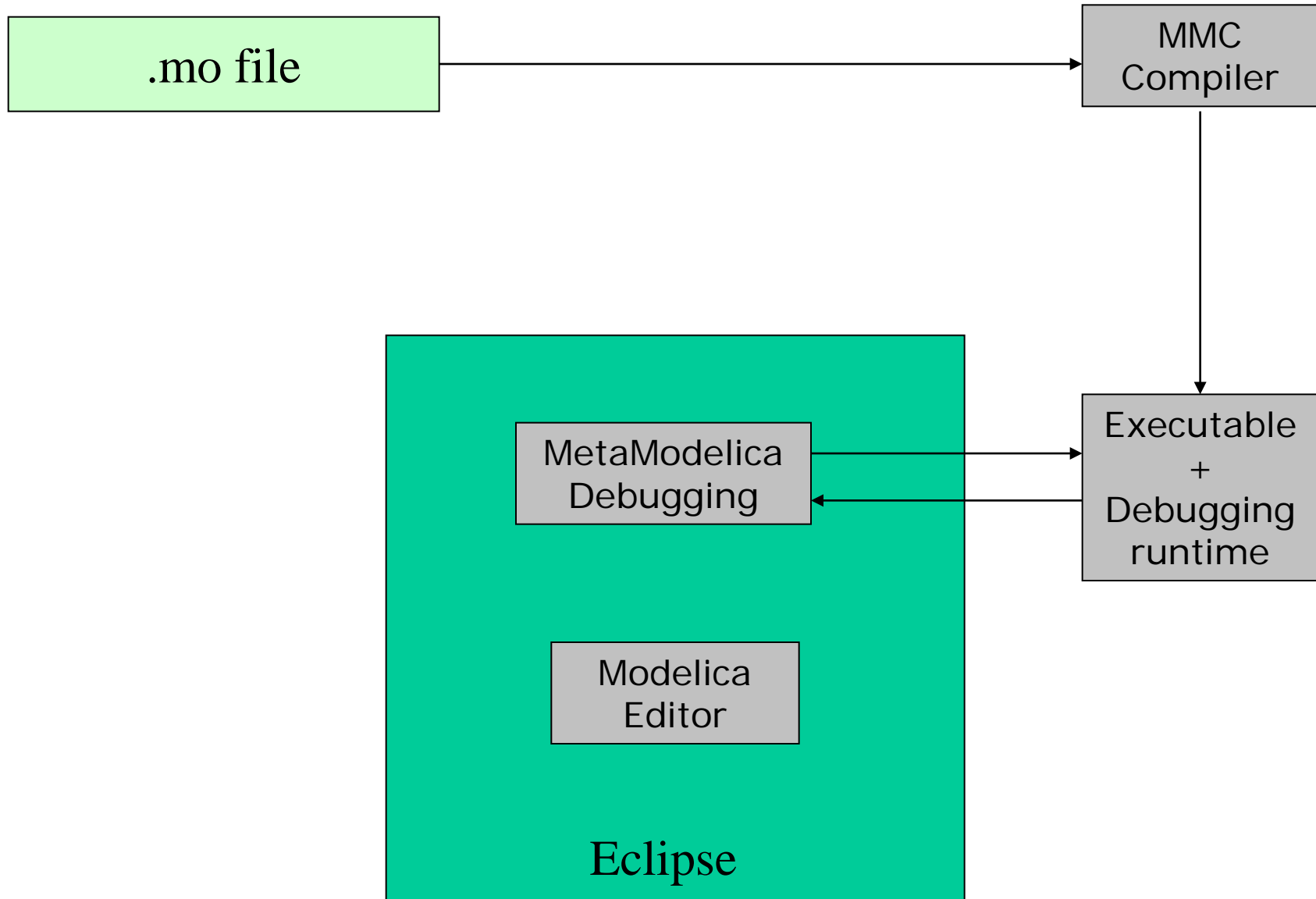
**MetaModelica Debugging**

**Modelica Perspective**

# The MDT Eclipse Environment (II)



# The MDT Eclipse Environment (III)



# 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 open, showing a list of wizards. The 'Modelica Project' wizard is selected. A 'New Modelica Project' dialog is also open, showing the project name 'demo' and 'Next >' buttons. Red arrows indicate the flow from the menu to the wizard list, then to the 'Modelica Project' wizard, and finally to the 'Next >' button in the 'New Modelica Project' dialog.

**File** Edit Refactor Navigate Search Project Run Window Help

New Alt+Shift+N ▶ Project...  
Open File...  
Close Ctrl+F4  
Close All Ctrl+Shift+F4  
Save Ctrl+S  
Save As...  
Save All Ctrl+Shift+S  
Revert  
Move...  
Rename... F2  
Refresh F5  
Convert Line Delimiters To  
Print... Ctrl+P  
Switch Workspace...  
Import

Modelica Package  
Modelica Class  
Folder  
File  
Example...  
Other...

**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

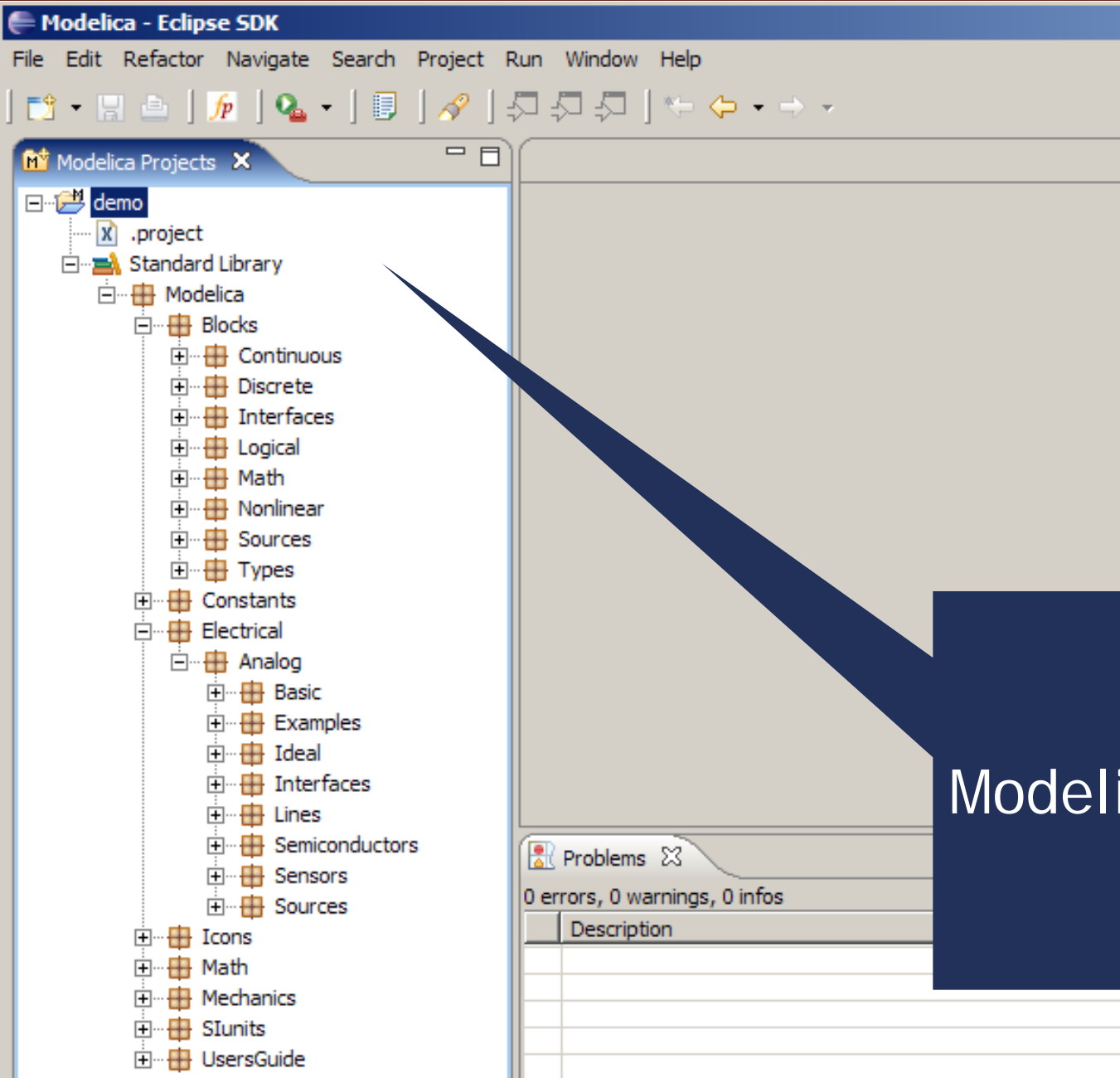
**New Modelica Project**  
Create a Modelica project  
Create a Modelica project in the workspace.

Project name: demo

< Back Next >  
< Back Next > Finish Cancel

Creation of Modelica projects using wizards

# Creating Modelica projects (II)



Modelica project

# Creating Modelica packages

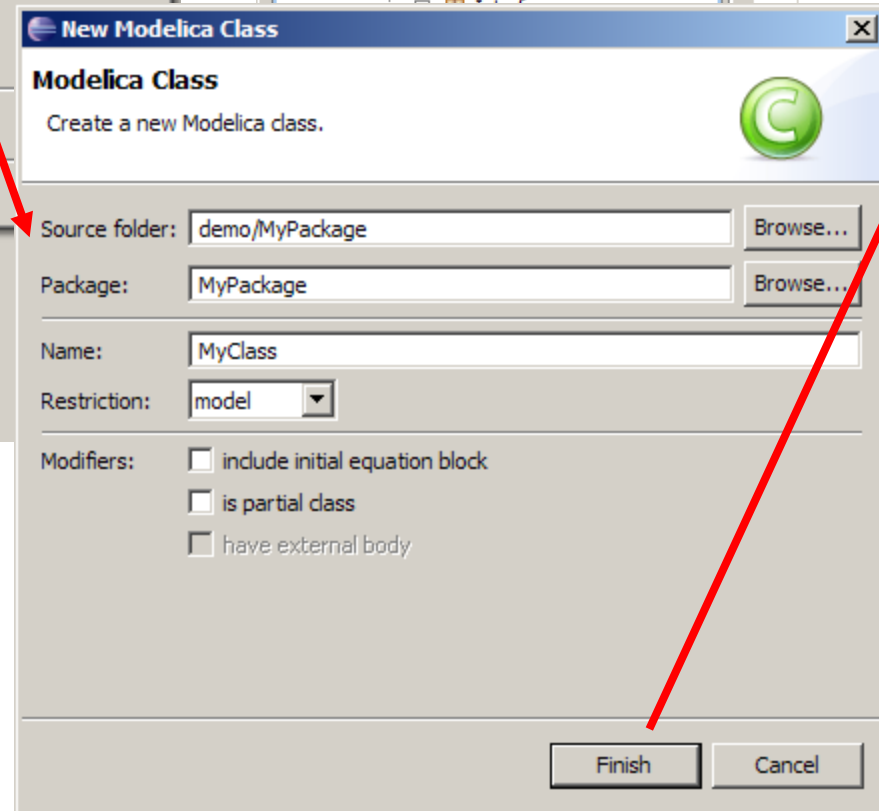
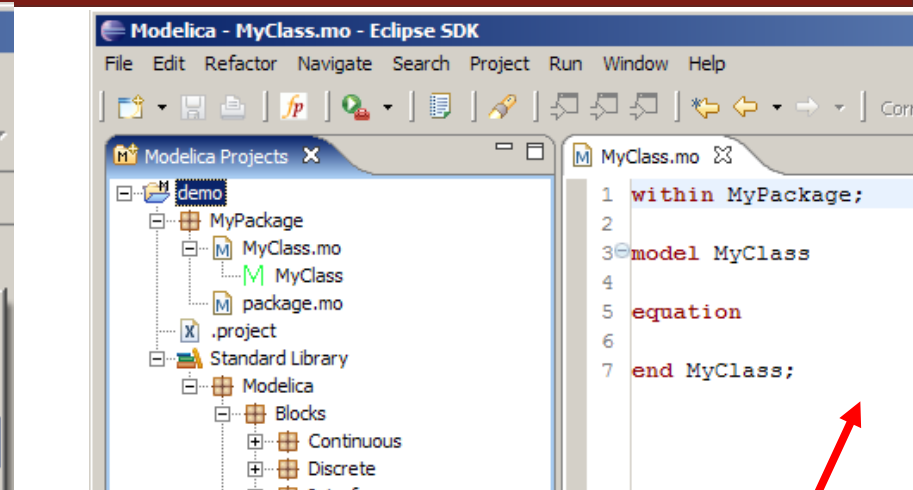
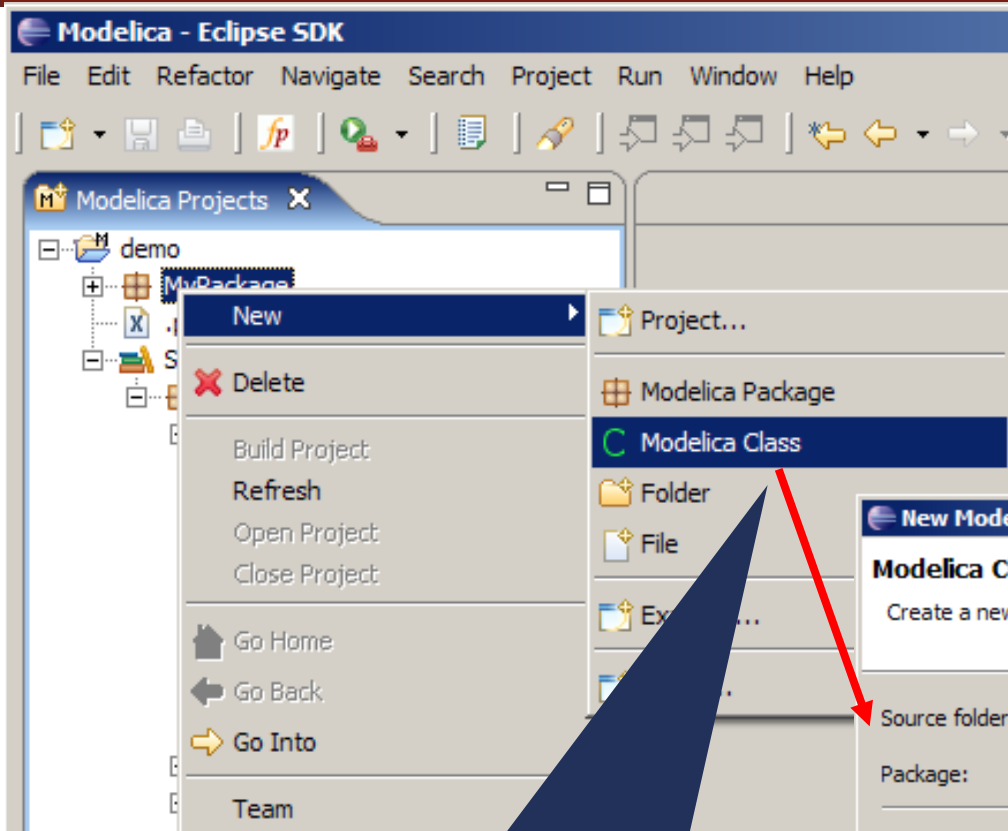
The image shows the Eclipse IDE interface for creating a Modelica package. The 'New Modelica Package' wizard is open, displaying the following fields and options:

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

The 'Finish' button is highlighted with a red arrow, indicating the next step in the process.

Creation of Modelica packages using wizards

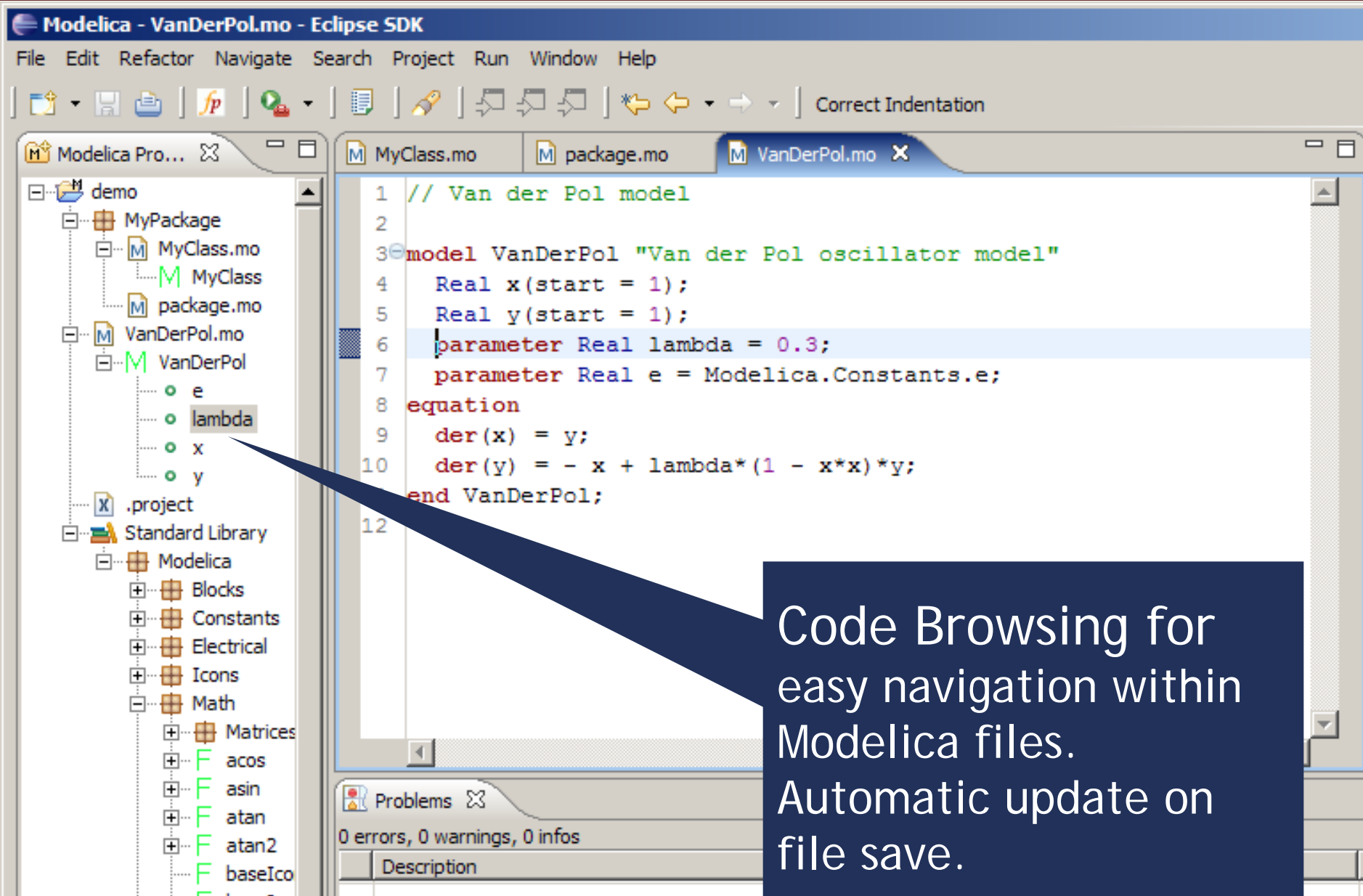
# Creating Modelica classes



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 tree with the following structure:

- demo
  - 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

The main editor window shows 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   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
```

The `lambda` parameter is highlighted in blue. A callout box points to this line with the text:

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

The bottom status bar shows "0 errors, 0 warnings, 0 infos" and a "Description" tab.

# 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 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:** 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' tree is visible, with 'Absyn.mo' selected. The main editor window displays the source code for 'Absyn.mo'. The code includes a 'public' declaration, a 'uniontype' for 'Program', and a 'record PROGRAM' containing a 'list<Class>' of 'classes' and a 'Withi' statement. A red 'X' icon is present next to line 77, indicating an error. The 'Problems' view at the bottom 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.exe' and the error message 'make[2]: \*\*\* [Absyn.h] Error 1'. A blue callout box with a white arrow points to the error message in the console, containing the text 'Semantic error detection on compilation'.

```
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.../home...'  
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 with the following components:

- Project Explorer:** Shows a project named 'demo' with a package 'MyPackage' containing 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'.
- 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.
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:** A tooltip is visible over the 'import Modelica.' statement, listing the following categories: Blocks, Constants, Electrical, Icons, Math, Mechanics, SIunits, and UsersGuide.
- Problems View:** Shows '0 errors, 0 warnings, 0 infos'.

Code Assistance on imports

# Code assistance (II)

The screenshot shows the Eclipse IDE with the Modelica SDK. The main editor 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.
9 equation
10  der(x) = y;
11  der(y) = - x + lambda*(1 - x*x)*y;
12 end VanDerPol;
13
```

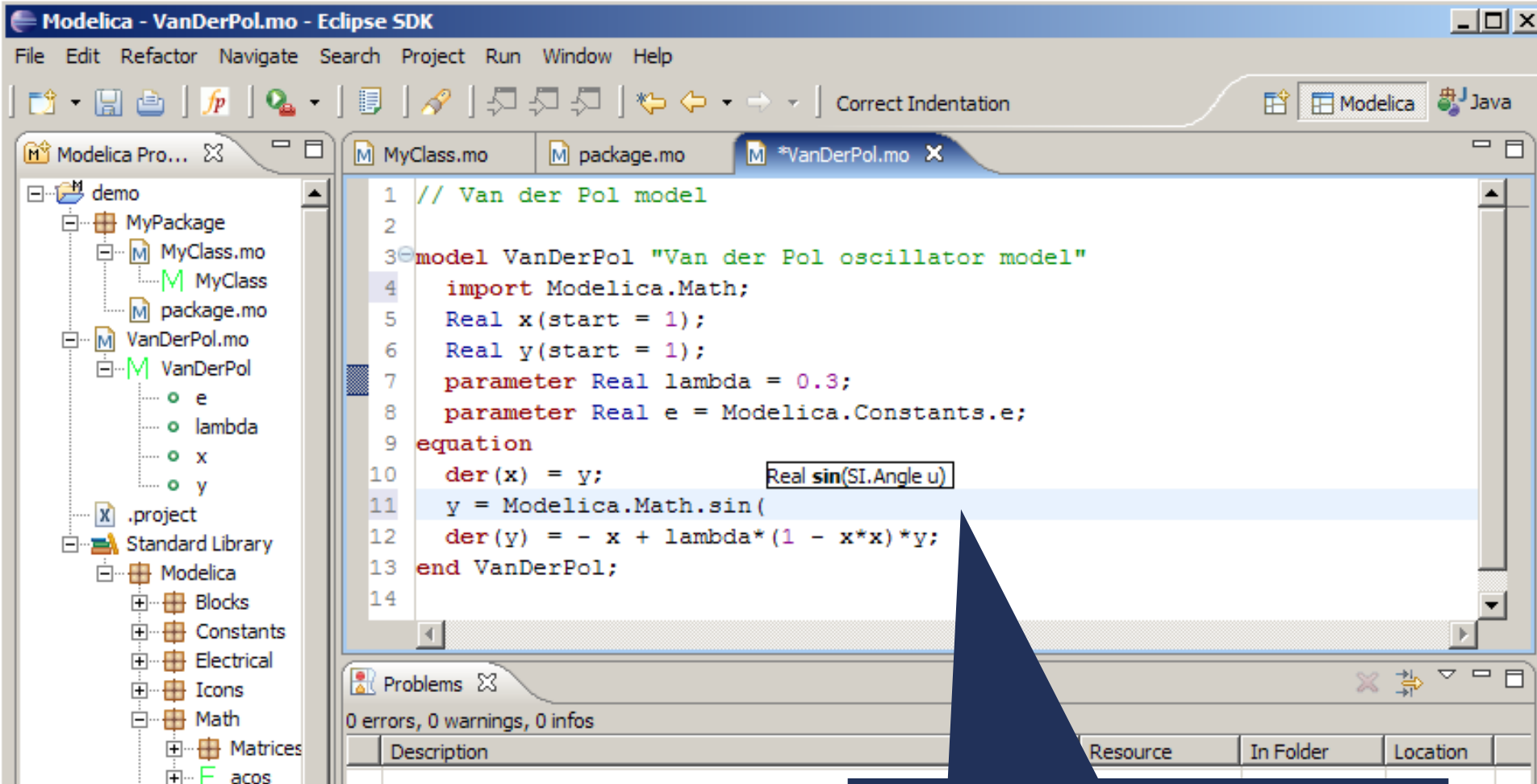
Line 8 is highlighted, and a code completion popup is visible on the right, listing constants from the Modelica.Math package:

- c
- D2R
- e
- eps
- epsilon\_0
- G
- g\_n
- h
- inf

The Problems window at the bottom shows 0 errors, 0 warnings, and 0 infos.

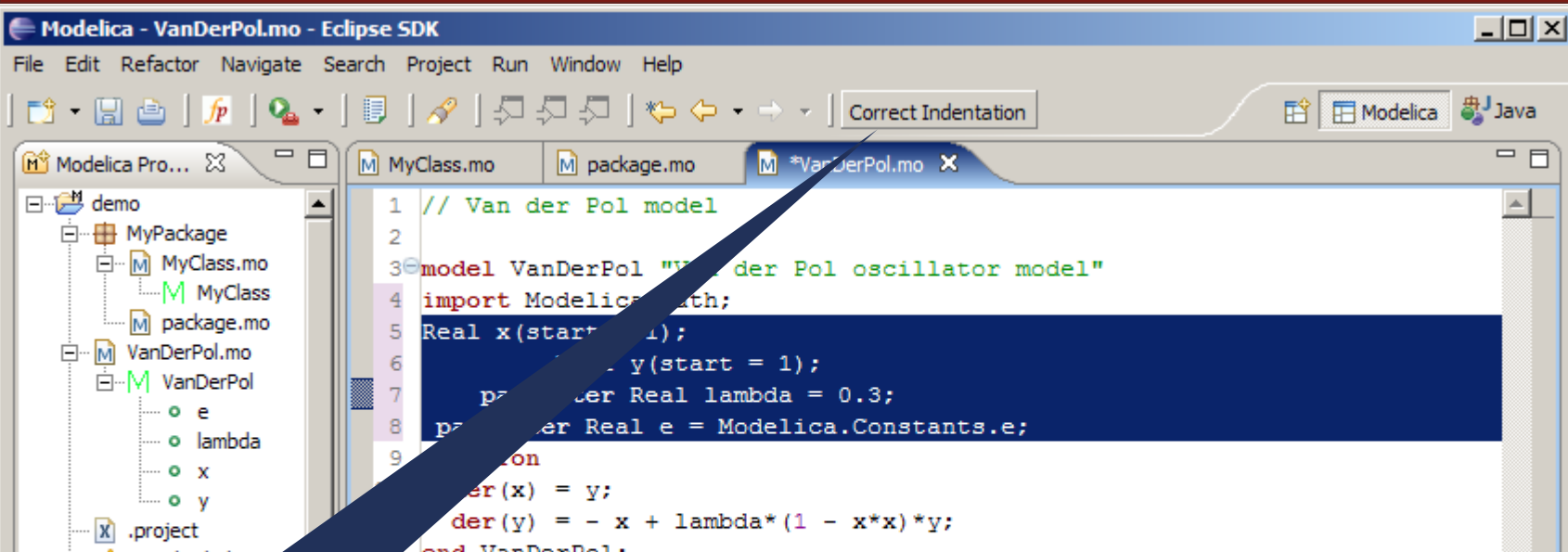
Code Assistance on assignments

# Code assistance (III)



Code Assistance on  
function calls

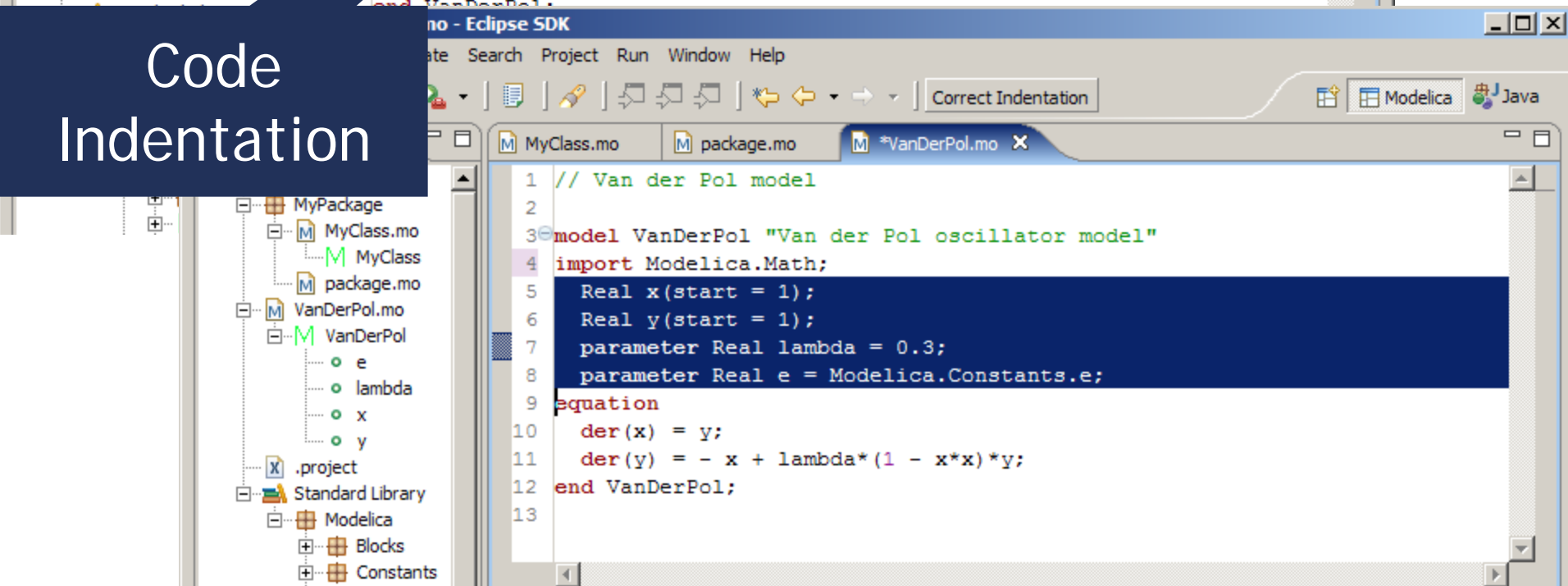
# Code indentation



Modelica - VanDerPol.mo - Eclipse SDK

```
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;
```

Code  
Indentation



Modelica - VanDerPol.mo - Eclipse SDK

```
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
```

# 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 files, with 'Absyn.mo' selected. The Outline view below it provides a hierarchical list of symbols within the selected file. The main editor shows the source code of 'Absyn.mo', featuring a case statement and a function definition for 'getCrefFromExp'. A tooltip is overlaid on 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 indicates 113 errors, with a list of error messages such as 'The identifier at start and end are different'. A status bar at the bottom shows '64M of 254M' and 'Ctrl Contrib (Bottom)'.

```
case (MATRIX(matrix = exp11))
  local list<list<list<ComponentRef>>> res1;
  equation
    res1 = Util.listListMap(exp11, 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;
```

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;

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

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

The screenshot shows the Eclipse IDE with the following components:

- Breakpoints**: No breakpoints are currently set.
- Variables**: A table showing the state of the program. The root variable is `p` of type `Absyn.Program`. It contains a `[record]` of type `Absyn.PROGRAM[2]`, a `classes` list of type `Absyn.CLASS[7]`, and a `f` string variable.
- Console**: Shows the execution log for the `Main` thread, including `Main.translateFile` and `Main.main`.
- Outline**: Displays the project structure, with `translateFile` selected in the `Absyn` package.
- Code Editor**: Shows the source code for `Bla.mo`, with the `translateFile` function 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



- OpenModelica
  - What is OpenModelica?
  - The past and present
- OpenModelica Technical Overview
  - OMC, OMShell, OMNotebook
- OpenModelica Development Environment
  - MetaModelica
  - The Eclipse Environment
- OpenModelica Latest Developments (2010-2011)

# Latest Developments (2010-2011)

2010 - 2011 - Most focus on MSL 3.1 support & some performance

- 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)
  - 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)
- **Much More things I don't remember**

- The most evil Modelica Standard Library: Media
- Everything in the Modelica Language Specification is used
  - partial functions in partial packages
  - full packages in partial packages used via the fully qualified path
  - redeclare replaceable model extends x.
  - functions using redeclare replaceable function extends used to set constants in partial packages.
- ... and then some more unspecified things
  - <https://trac.modelica.org/Modelica/ticket/488>  
Unspecified function in Modelica.Fluid (should pose no issue)
  - <https://trac.modelica.org/Modelica/ticket/482>  
Illegal lookup in Modelica.Media (fixed by Hubertus in a branch)

## Action plan to support Media & Fluid

- Simplify flattening (instantiation) by preprocessing phases
  - Remove imports (100%)
  - Remove extends (95%)
  - Remove redeclare (1%)
  - Perform dependency analysis (0%)
- Handle record constants (10%)
- Any other unknown issues
- *Hopefully Media & Fluid flattening will start working for Modelica Conference 2011 (March 20)*

- *Next presentation by Martin Sjölund*

**Thank You!**  
**Questions?**

**OpenModelica Project**  
**<http://www.OpenModelica.org>**