

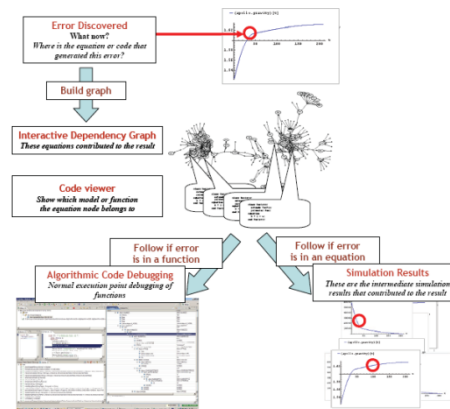
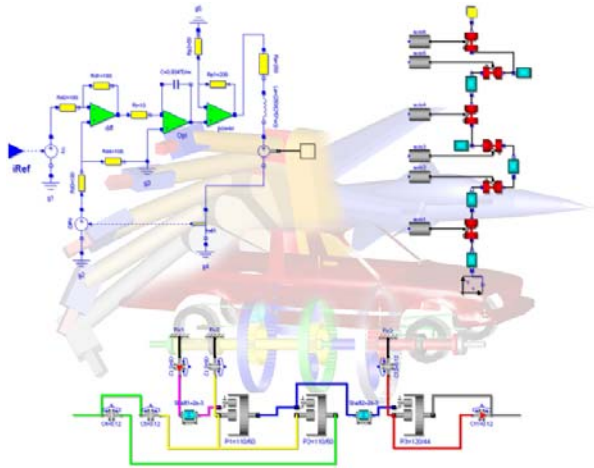
# Technical Overview of OpenModelica and its Development Environment

Adrian Pop

2010-02-08

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

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



**A Servo Mechanism Model**  
 A nice example of a full system

$$\tau_2 = \frac{1}{k_2} \tau_1$$

$$e = \omega_{ref} - \omega_{out}$$

$$u = K \left( e + \frac{1}{T_I} \int_0^t e dt \right)$$

$$v = u \quad \omega_R = R \cdot i \quad M_{emf} = k_1 \omega_{emf}$$

$$J_1 \frac{d^2 \theta_1}{dt^2} = \tau_{emf} + \tau_1$$

$$J_2 \frac{d^2 \theta_2}{dt^2} = \tau_2 + \tau_3$$

$$J_3 \frac{d^2 \theta_3}{dt^2} = -\tau_4 - \tau_{load}$$

$$v - u_R - u_L - u_{emf} = 0$$

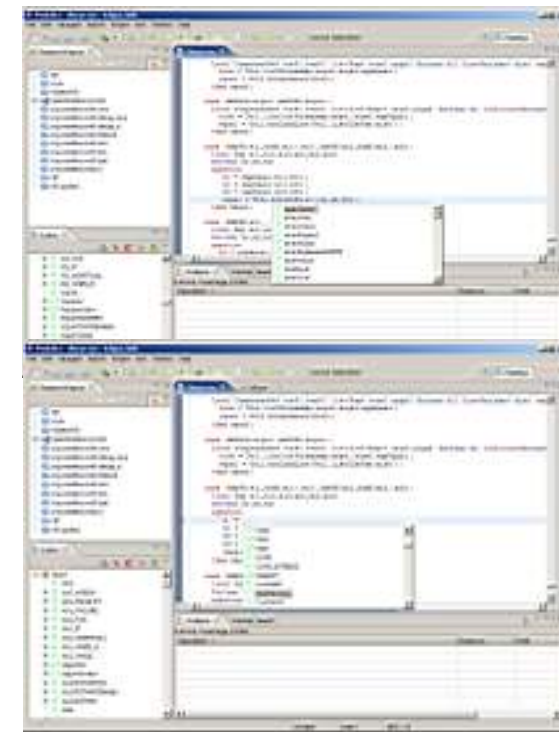
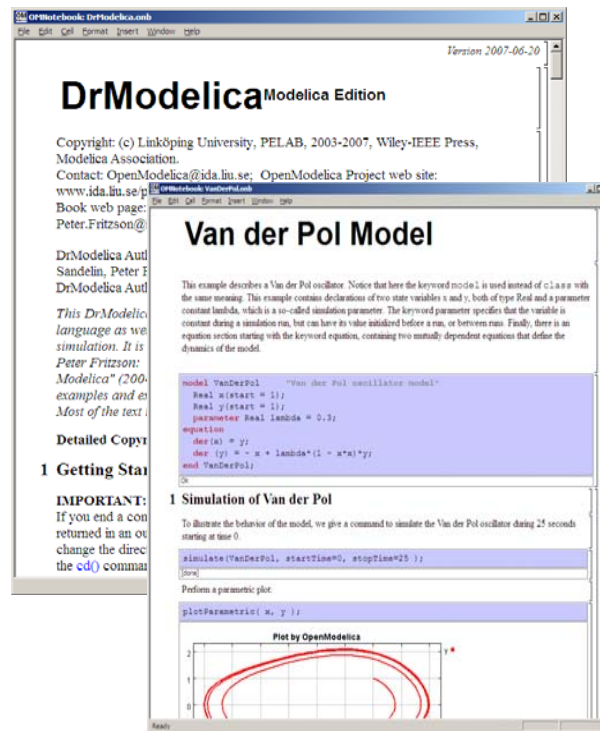
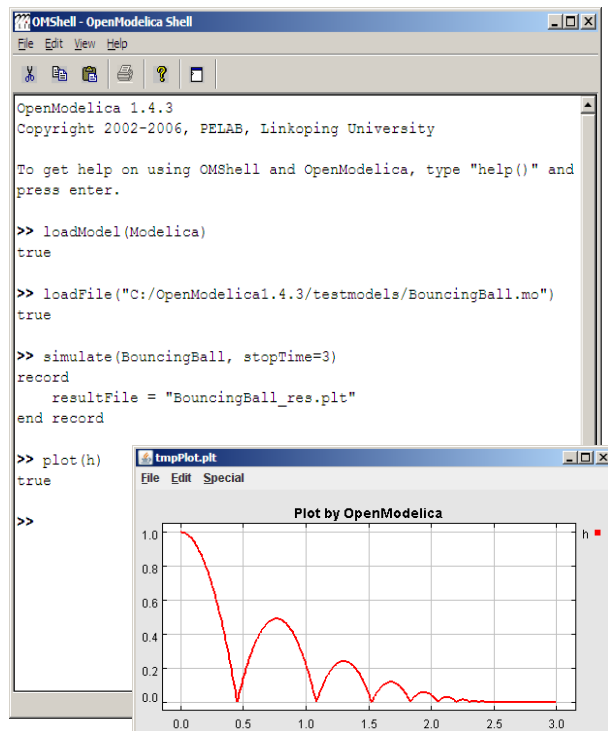
$$M_{emf} = k_1 \omega_{emf} \quad i = \frac{1}{k_1} \tau_{emf} \quad \tau_2 = \frac{1}{k_2} \tau_1$$

$$\frac{J_1 - J_2 k_2^2}{k_2} \frac{d^2 \theta_1}{dt^2} = \tau_{emf} - k_2 \tau_1$$


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

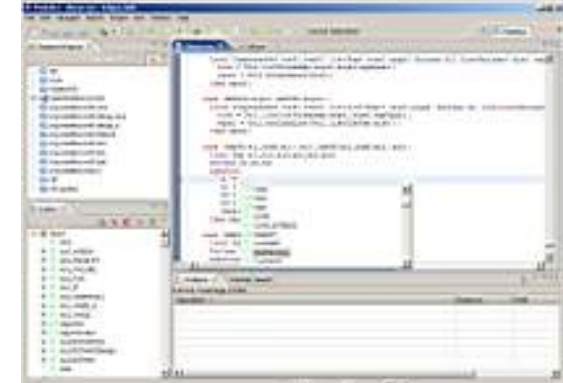
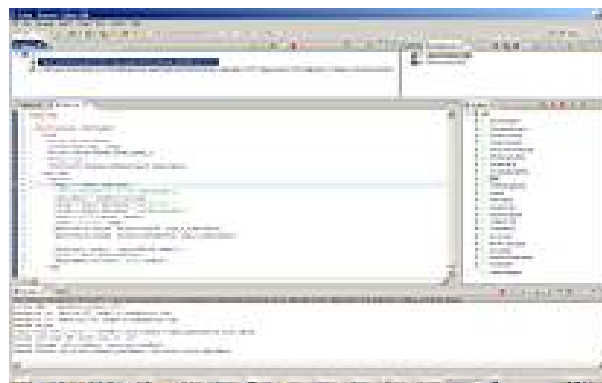
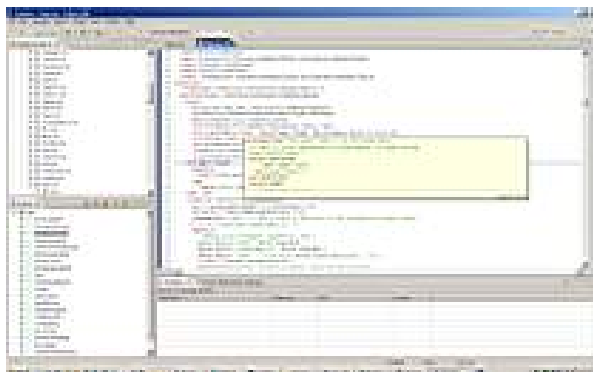
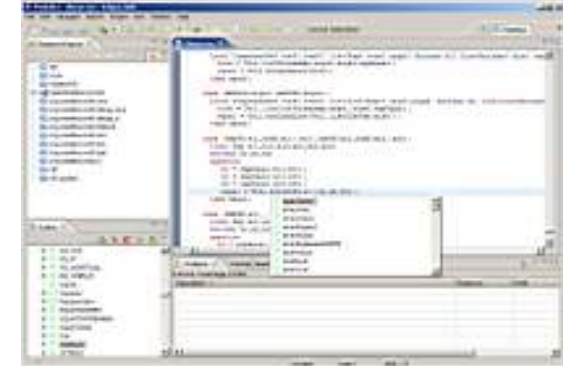
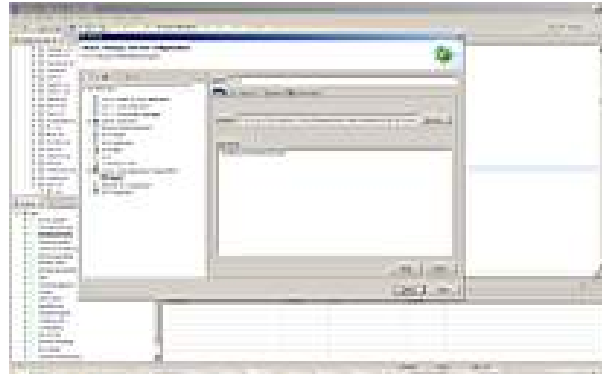
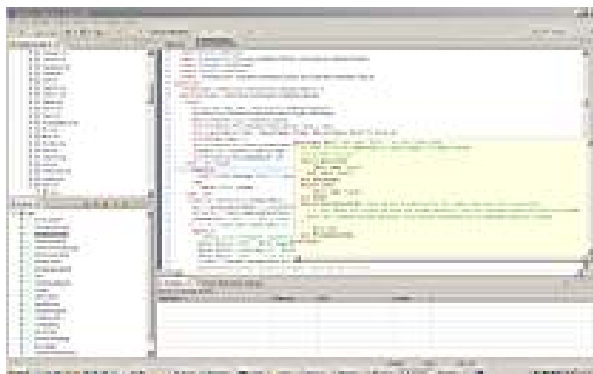
# What is OpenModelica? (I)

- Advanced Interactive Modelica compiler (OMC)
  - Supports most of the Modelica Language v. 2.2 and v. 3.1
- Basic environments for creating models
  - OMShell - an interactive command handler
  - OMNotebook - a literate programming notebook
  - 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
  - Now working on UML/SysML integration and better debugging
  - *Used heavily for OpenModelica development*
  - Used in 6 OpenModelica Development Courses (INRIA, PELAB)





# What is OpenModelica? (III)

- Open-source community services
  - Website and Support Forum
  - Version-controlled source base
  - Bug database (unfortunately)
  - Development courses

Welcome to OpenModelica

http://www.openmodelica.org/

Admin Dicts EU Eclipse SHit Firma Fiske Work Weather Media Modelica adrho RML Soft Other bookmarks

# OpenModelica

Log in Create an account

HOME DEVELOPER FORUM DOWNLOAD CONTACT US WORKSHOP RESEARCH search...

### Top information

**New OpenModelica website is up.**  
The new OpenModelica website is up and running.

### Registration

Please register if you download and install Open Modelica. Why? We would like to inform you about new releases of Open Modelica! We want be informed who is using it and the kind of usage. Your information will be not be distributed to third parties!

**Note:** It may take a while to be registered as we check the information we receive to fight the spam on our mailing lists.

Thank you for your patience.

### Introduction

Tuesday, 15 December 2009 08:58

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 complete Open Source Modelica modeling, compilation and simulation environment based on free software distributed in binary and source code form. We invite researchers and students, or any interested developer to participate in the project.

### Latest news

Feb 5: OpenModelica Release 1.5.0 RC2

Jan 28: OMScheme release available for download

Dec 14: OpenModelica Release 1.5.0 RC1

Dec 14: Open Master Theses

Dec 14: Open Positions

### Upcoming Events

OpenModelica Workshop 2010

Register yourself to get information about new releases.  
Participate in the OpenModelicaInterest mailing list.  
Help us: get the latest source code or nightly-build and report bugs!  
To learn about Modelica, read a book or a tutorial about Modelica®.

Log Messages - C:\bin\cygwin\home\adrho\dev\OpenModelica

From: 2007-08-26 To: 2007-12-18

| Revision | Author   | Date                           | Message  |
|----------|----------|--------------------------------|--|
| 2983     | adrho    | 15:19:01, den 18 december 2007 | - updates to OMSShell project to base it on OMSDev   |
| 2982     | adrho    | 15:18:59, den 18 december 2007 | - These are local settings or user files, they are not needed.                                     |
| 2981     | adrho    | 14:48:37, den 18 december 2007 | - updated OMSShell.exe to agree with the latest git libraries                                      |
| 2980     | adrho    | 01:25:56, den 18 december 2007 | - Linux test suite fixes: now all the tests succeed  |
| 2979     | adrho    | 10:50:31, den 7 december 2007  | - small cosmetic change  |
| 2977     | adrho    | 11:15:58, den 30 november 2007 | - depend update  |
| 2976     | adrho    | 11:15:58, den 30 november 2007 | - updated the msc/omc runtime to the latest version - also the ts-queue (reserved) priority        |
| 2975     | hoisu    | 13:09:13, den 29 november 2007 | - If the desired output interval was smaller than 0.001 not output was given except for the first  |
| 2968     | adrho    | 21:11:33, den 25 november 2007 | - fixed the input path to msc2311.lib: \$(OMDev)\lib\msc-vin32-misc                                |
| 2967     | x06nerer | 16:51:33, den 13 november 2007 | * Added some features, e.g. a line counter, error links * Saved files now correctly set an initial |
| 2966     | krsta    | 16:25:22, den 13 november 2007 | * A new MetaModelica related testcase  |
| 2965     | krsta    | 16:23:56, den 13 november 2007 | * Minor Changes in MetaModelica list handling  |
| 2964     | krsta    | 16:29:05, den 13 november 2007 | * Minor change in meta_modelica.h  |
| 2963     | krsta    | 17:01:13, den 13 november 2007 | * Respect to OS workspaces if class name is called   |

updated the msc/omc runtime to the latest version  
+ allow the to-space (reserved) only when a major GC happen  
+ timers for GC  
+ help text for the runtime when the executable is called with -help

| Action   | Path  | Copy from path | Revision |
|----------|---|----------------|----------|
| Modified | /junk/Compiler/VC7/mi/Runtime/runtime/vcpro               |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/arr-add.c    |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/arr-create.c |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/arr-length.c |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/arr-lib.c    |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/arr-sh.c     |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/arr-seb.c    |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/arr-update.c |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/arr-vec.c    |                |          |
| Added    | /junk/Compiler/VC7/mi/Runtime/runtime/common/arr.c        |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/bool-and.c   |                |          |
| Deleted  | /junk/Compiler/VC7/mi/Runtime/runtime/common/bool-or.c    |                |          |

Hide unrelated changed paths

Show | Next 100 | Stop on copy/replace

Statistics Help OK

OpenModelica Development Server powered by codeBeamer Enterprise

My Start Projects Wiki Documents Trackers Reports Forums Chats Builds Source Code Members

Projects > OpenModelica > Trackers > Bug

Tracker: Bug (190) Tracker View: Open New Filter: GO more details...

| ID   | Tracker | Summary   | Status      | Resolution |
|------|---------|---|-------------|------------|
| 1163 | [1] Bug | parameterized = true should be treated as a constant during runtime   | --          | --         |
| 1162 | [1] Bug | Backend: Fixed attribute not working properly for variables   | New         | --         |
| 1161 | [1] Bug | Wrong integer arithmetics   | Resolved    | FIXED      |
| 1160 | [1] Bug | getComponentAnnotations() and getLibComponentAnnotations() API are not working with Modelica standard library 3.2 | --          | --         |
| 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] Validating 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 Axis is trying to override final variable in class   | Resolved    | FIXED      |
| 1154 | [1] Bug | Linking of rapid constructor in modification fails (from MathCore)  | Resolved    | --         |

# What is OpenModelica? (IV)

- **An incubator platform for research**
  - 4 PhDs since 2004 (Debugging, Parallelization, PDEs Extensions)
  - 15 Master's theses since 2004
  - Both the students and the project benefit
- **Master theses at PELAB 2006-2010**
  - 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
- **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)
- **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

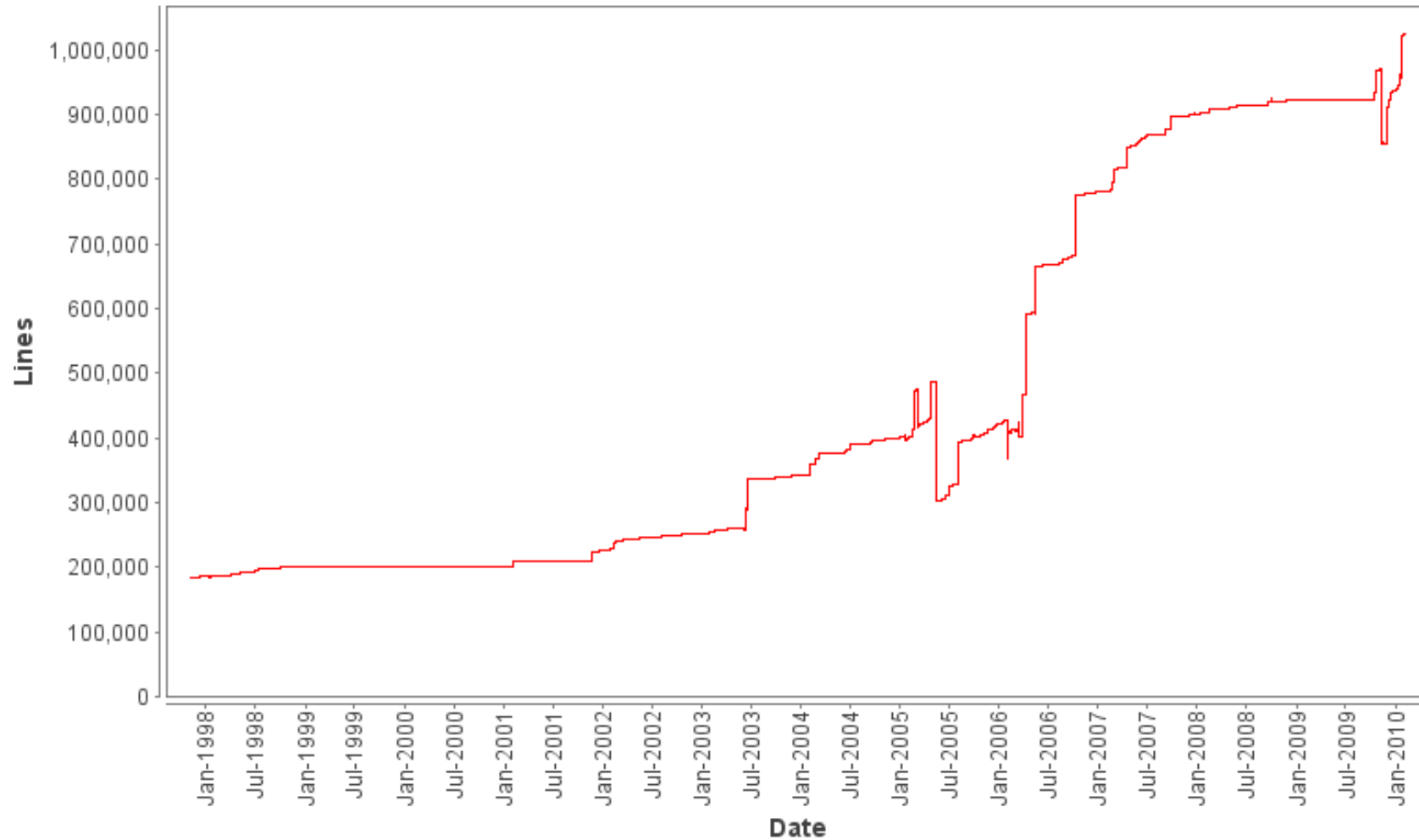
# OpenModelica Roadmap – Past & Present

2009 – 2010

- 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)
  - Braking of constraint connection graph (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**

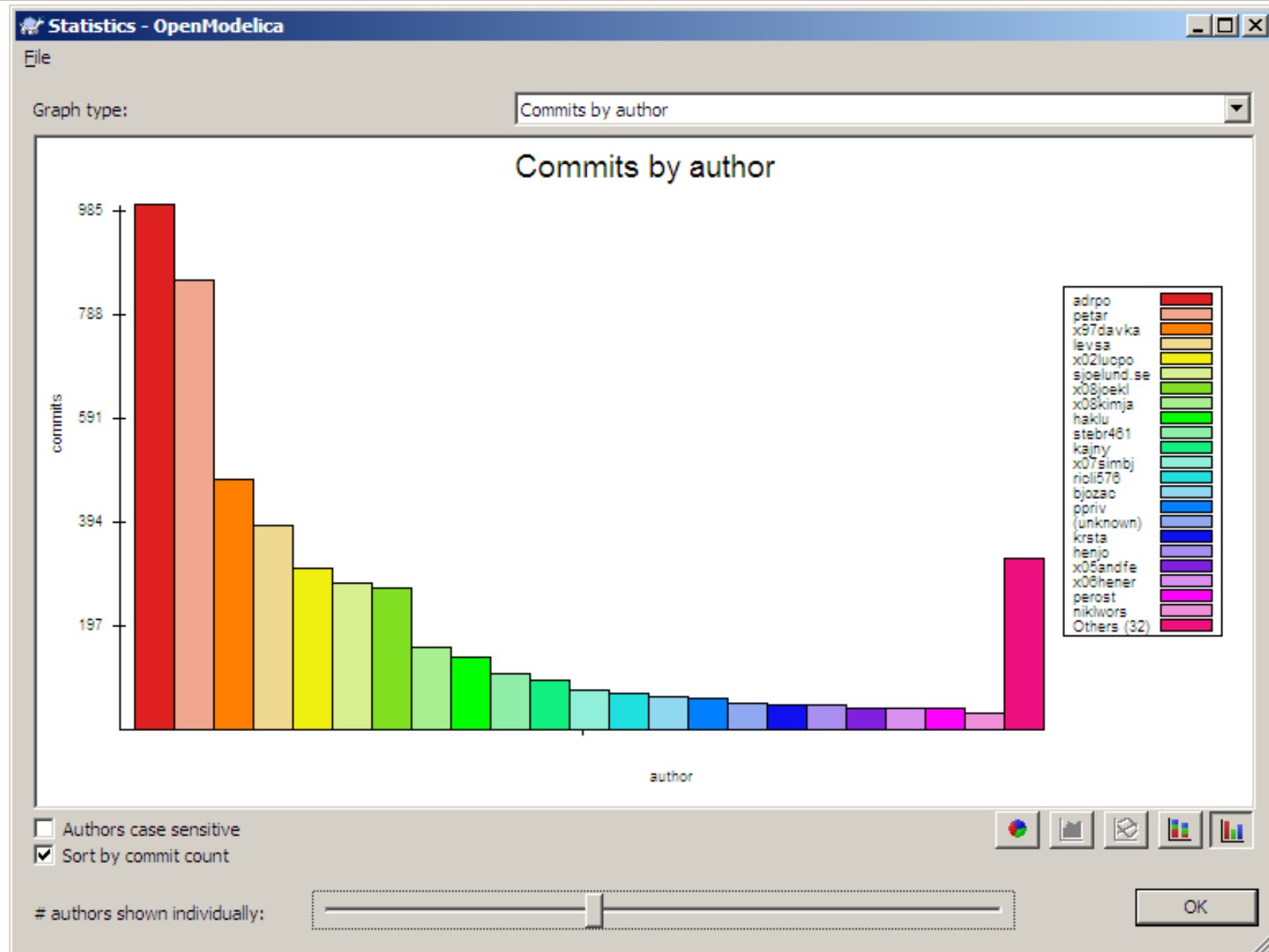
# OpenModelica Statistics (I)

/trunk: Lines of Code

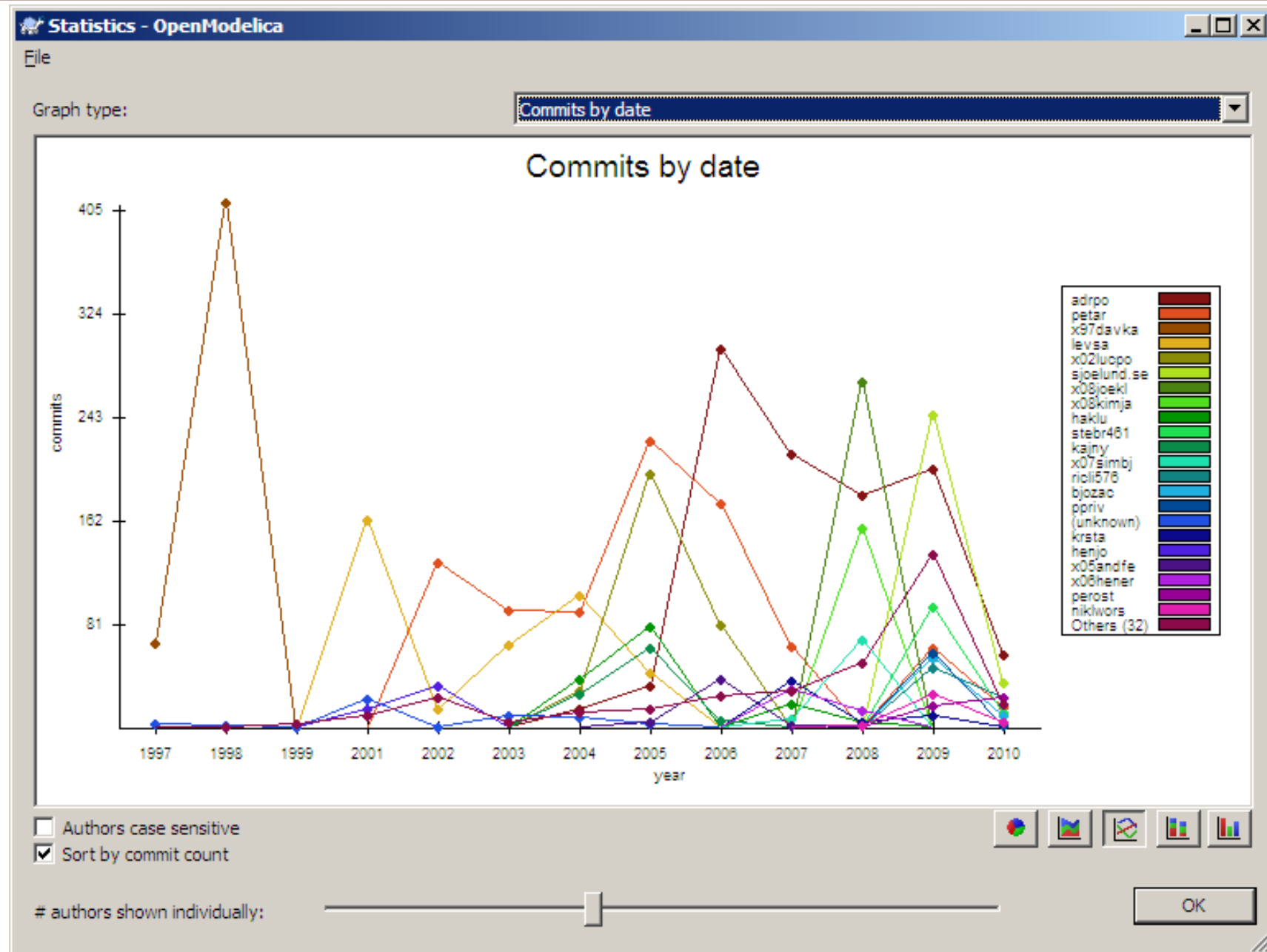


- Mature code base
- ~ 1000K lines of code, doubled since 2005

# OpenModelica Statistics (II)



# OpenModelica Statistics (III)

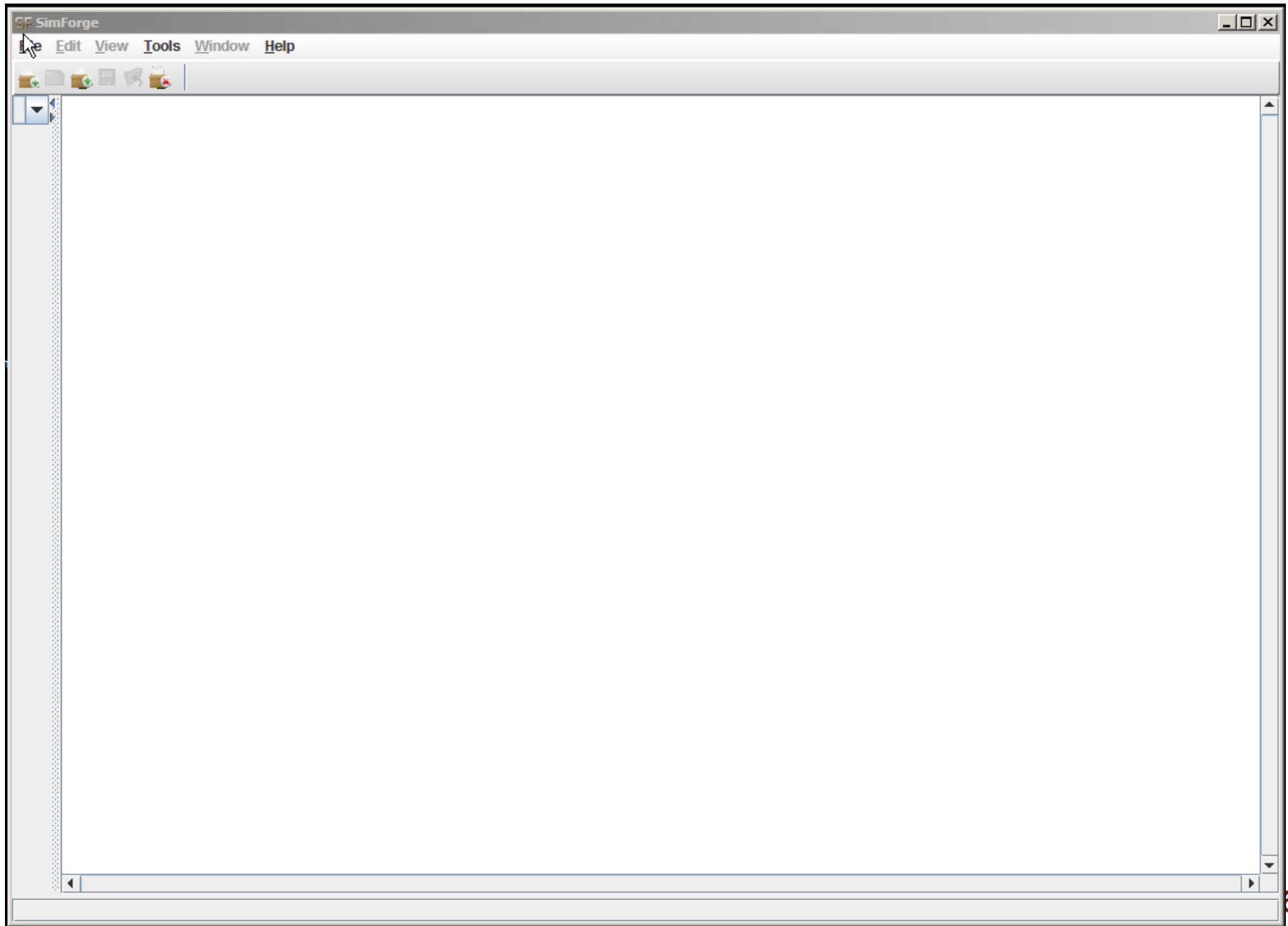




- OpenModelica
  - What is OpenModelica?
  - The past and present
- OpenModelica Technical Overview
  - OMC, OMSHELL, OMNotebook, SimForge
- OpenModelica Development Environment
  - MetaModelica
  - The Eclipse Environment
- OpenModelica Latest Developments (2009-2010)

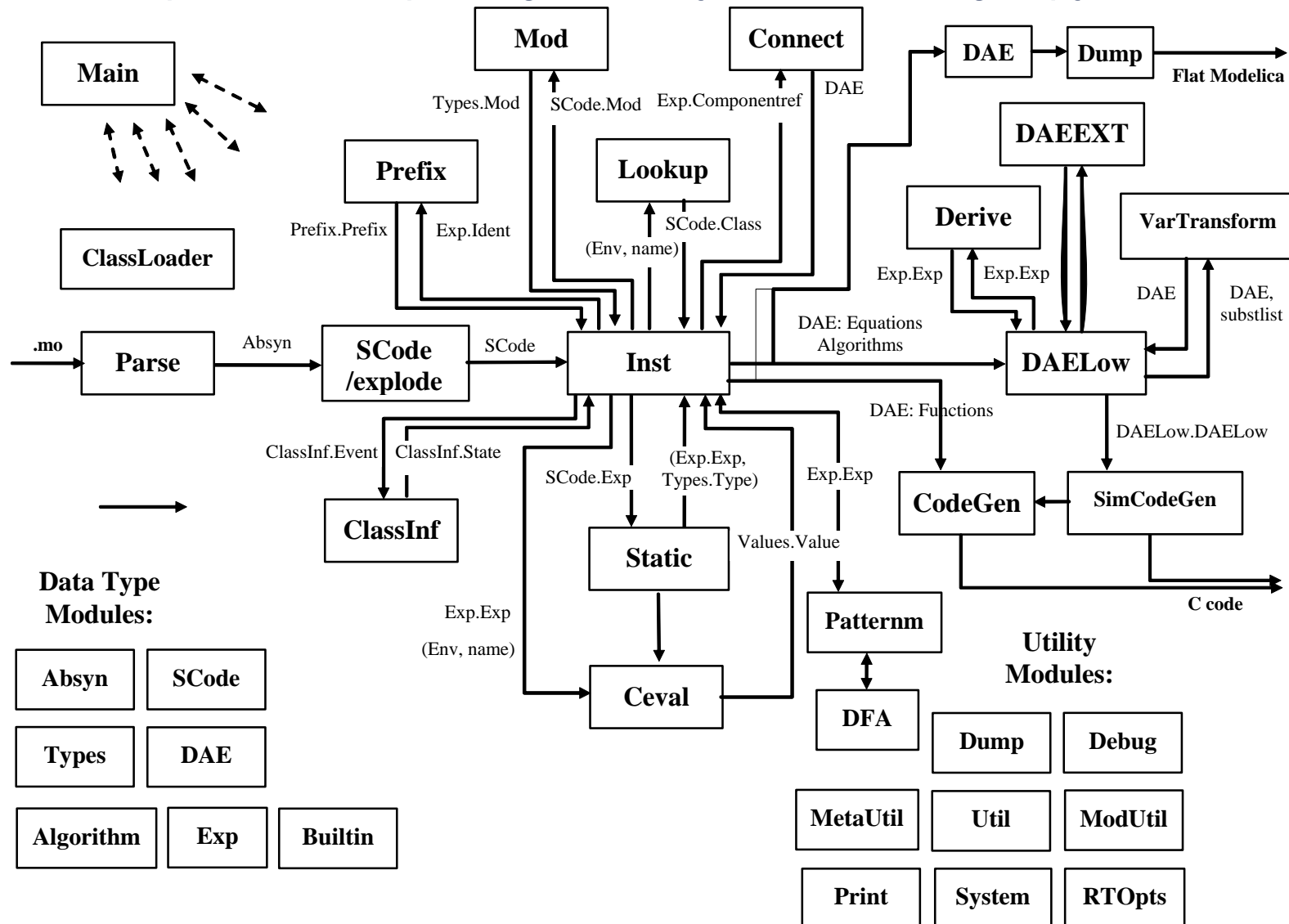
- Demo?

# SimForge - Demo? Maybe a movie!



# The OMC Compiler

- Implemented mainly in MetaModelica and C/C++
- The compiler has 91 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);
```

```
// instantiate the simplified code
```

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

```
// Transform all if equations to if expressions
```

```
dae2 = DAE.transformIfEqToExpr(dae1);
```

```
// Retrieve the last class name from the AST. This class will be instantiated.
```

```
lastClassName = Absyn.lastClassname(ast);
```

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

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



# Simulation Runtime Overview

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

# Simulation Runtime Main

## 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);  
deInitializeDataStruc(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

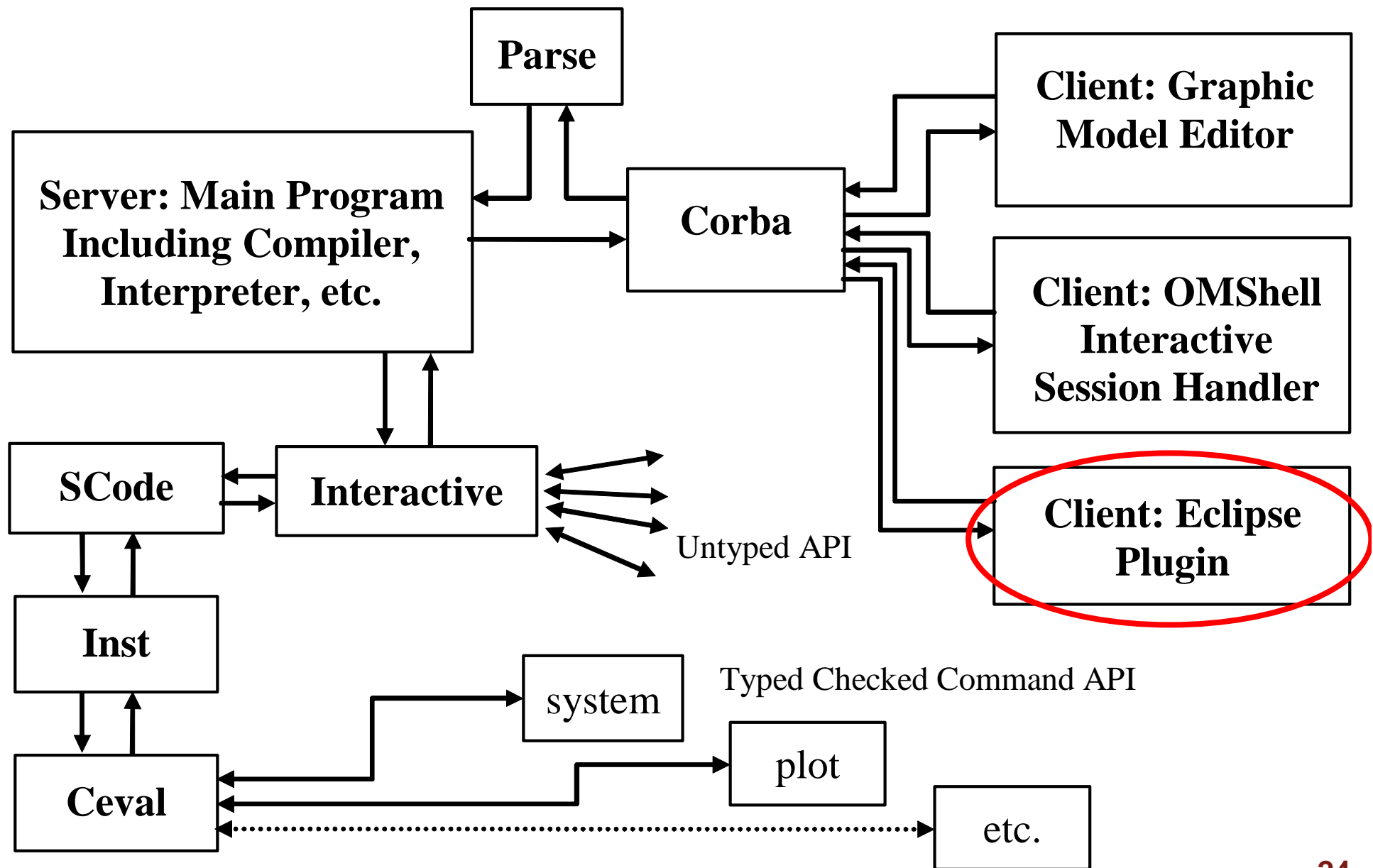
bound_parameters
initial_function
functionODE
functionDAE_output
functionDAE_output2
function_updateDependent
s
functionDAE_res
function_zeroCrossing
```

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

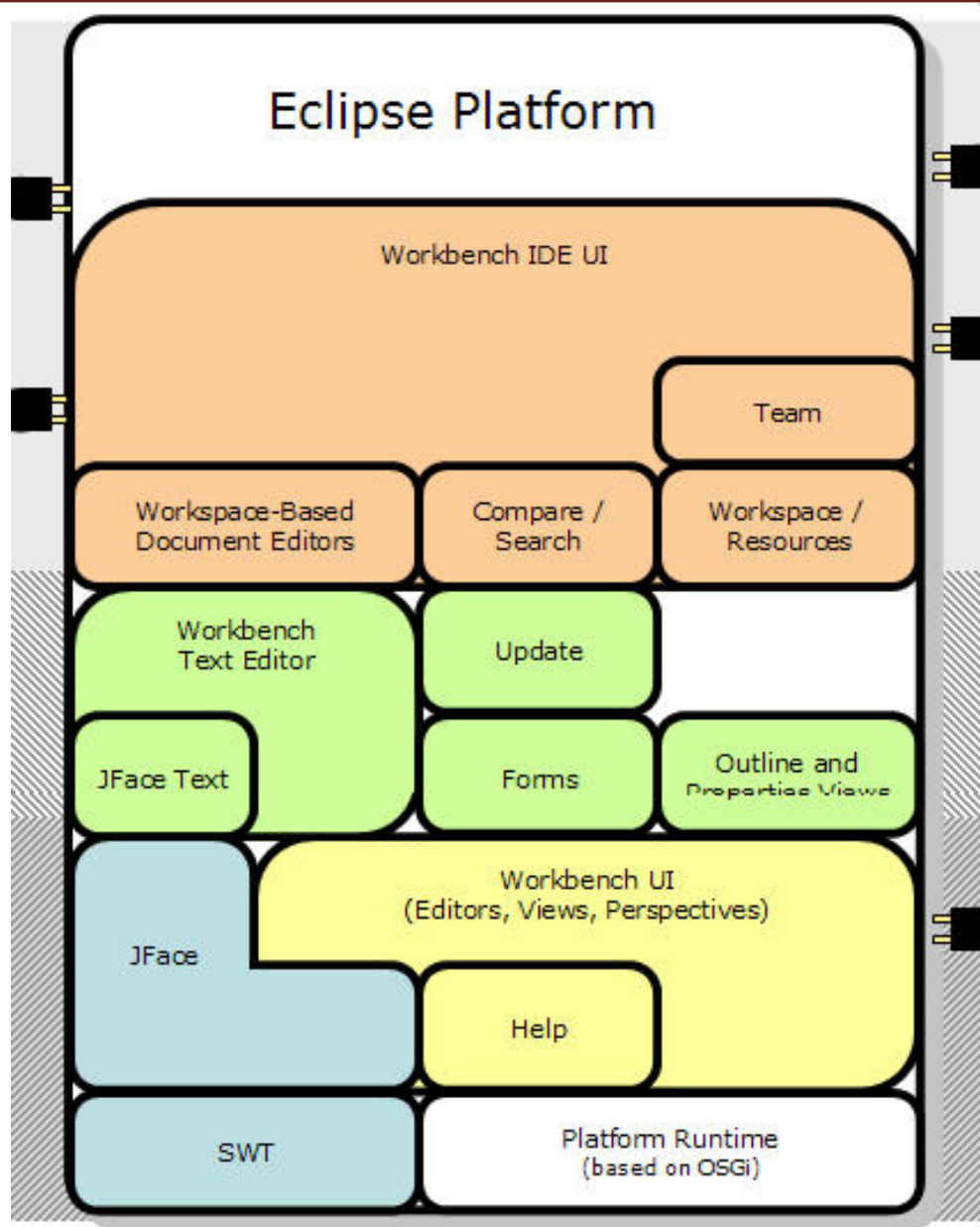
- **OMC**
  - Implemented mainly in MetaModelica and C/C++
- **Modelica**
  - classes, models, records, functions, packages
  - behavior is defined by equations or/and functions
  - equations
    - differential algebraic equations and conditional equations
- **MetaModelica extensions**
  - local equations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



# OpenModelica Context



# The MDT Eclipse Environment (I)



**Modelica Browser**

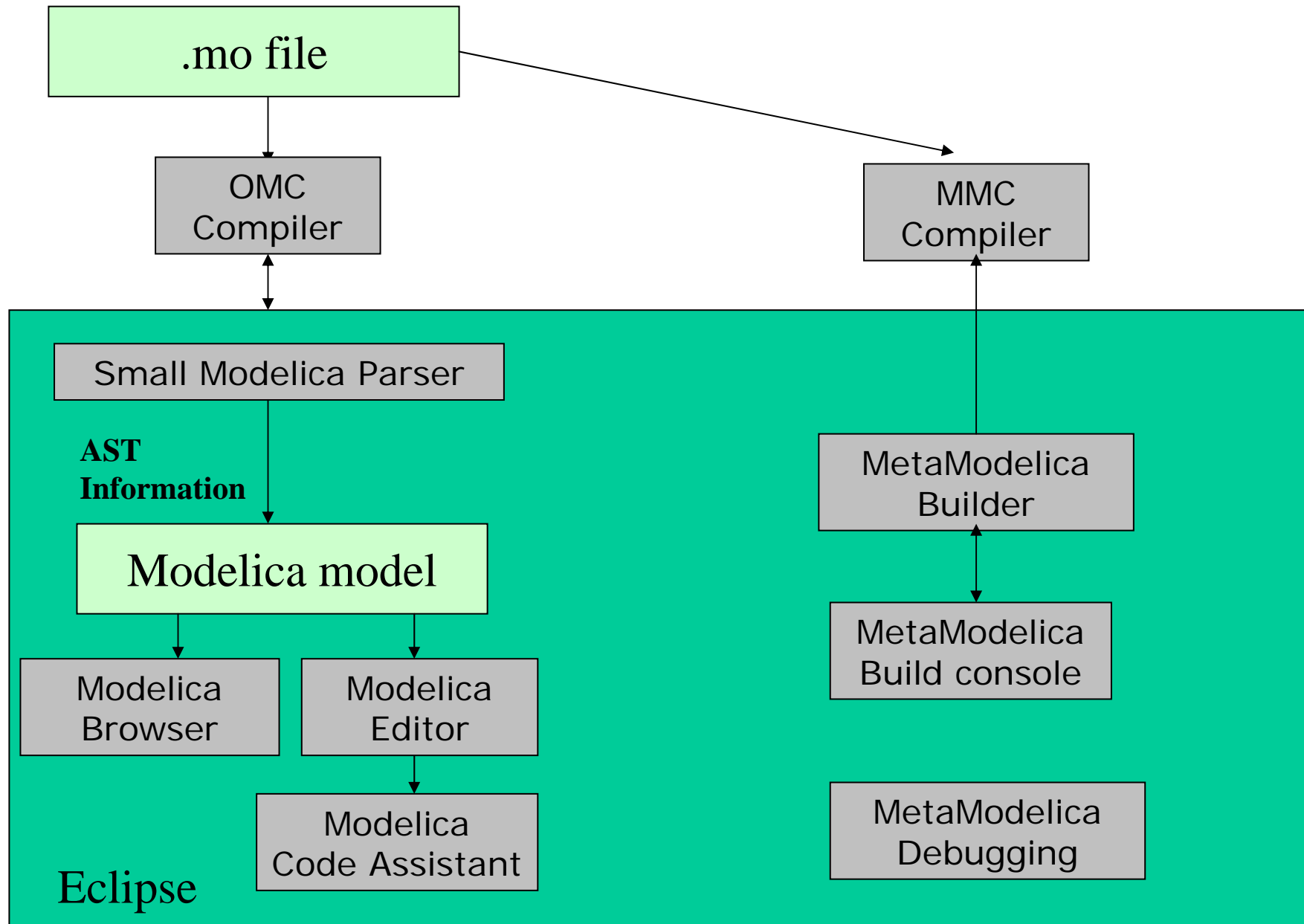
**Modelica Editor**

**Modelica Code Assistant**

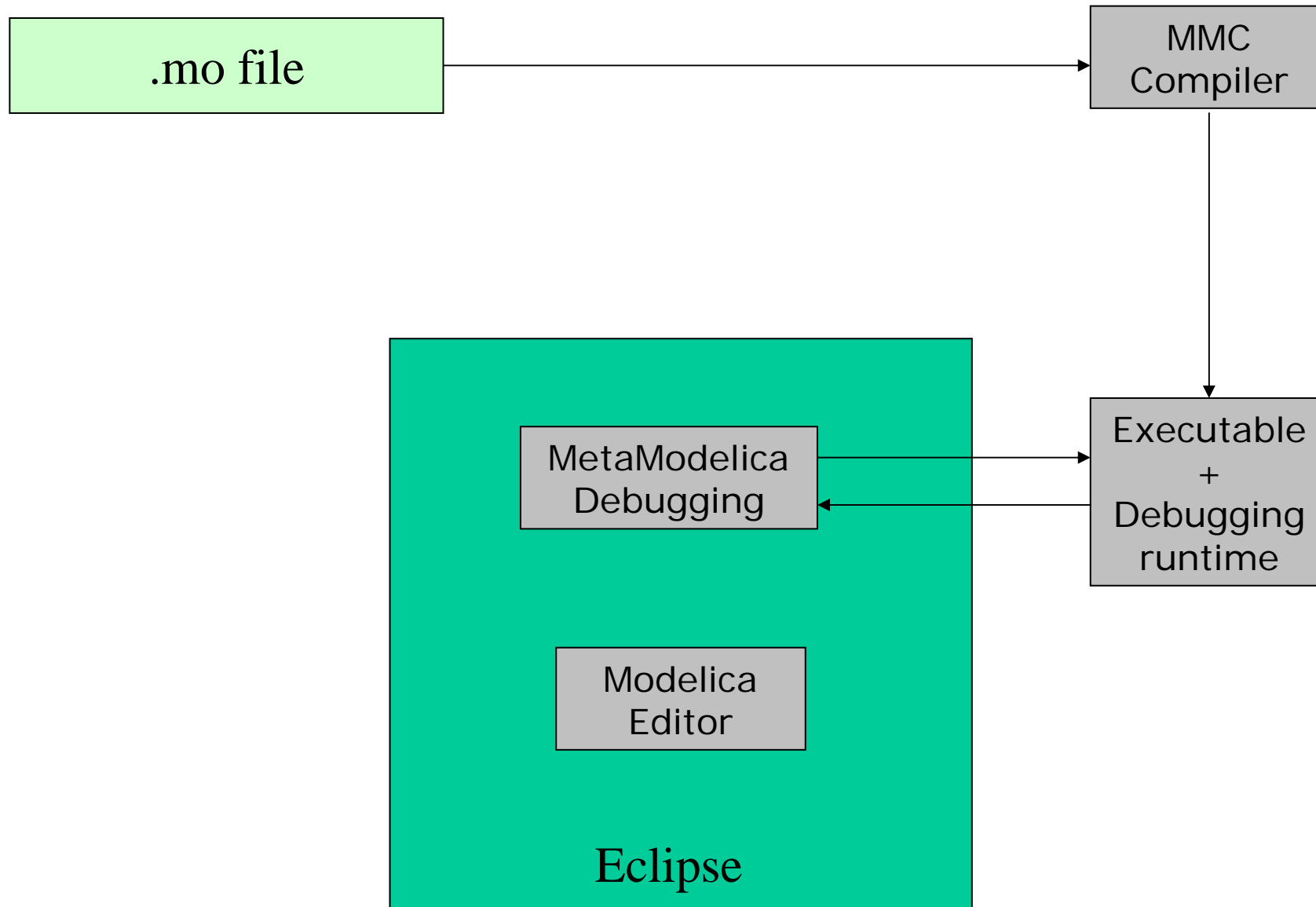
**MetaModelica Debugging**

**Modelica Perspective**

# The MDT Eclipse Environment (II)



# The MDT Eclipse Environment (III)



# Creating Modelica projects (I)

The screenshot illustrates the steps to create a Modelica project in Eclipse. On the left, the Eclipse SDK menu is open, showing the 'New' menu with 'Project...' selected. A red arrow points from 'Project...' to the 'New Project' dialog box. In this dialog, 'Modelica Project' is selected under the 'Wizards' list. Another red arrow points from 'Modelica Project' to the 'New Modelica Project' wizard. The wizard shows the 'Project name' field with 'demo' entered. At the bottom of the wizard, the 'Next >' button is highlighted with a red arrow.

**Modelica - Eclipse SDK**

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**

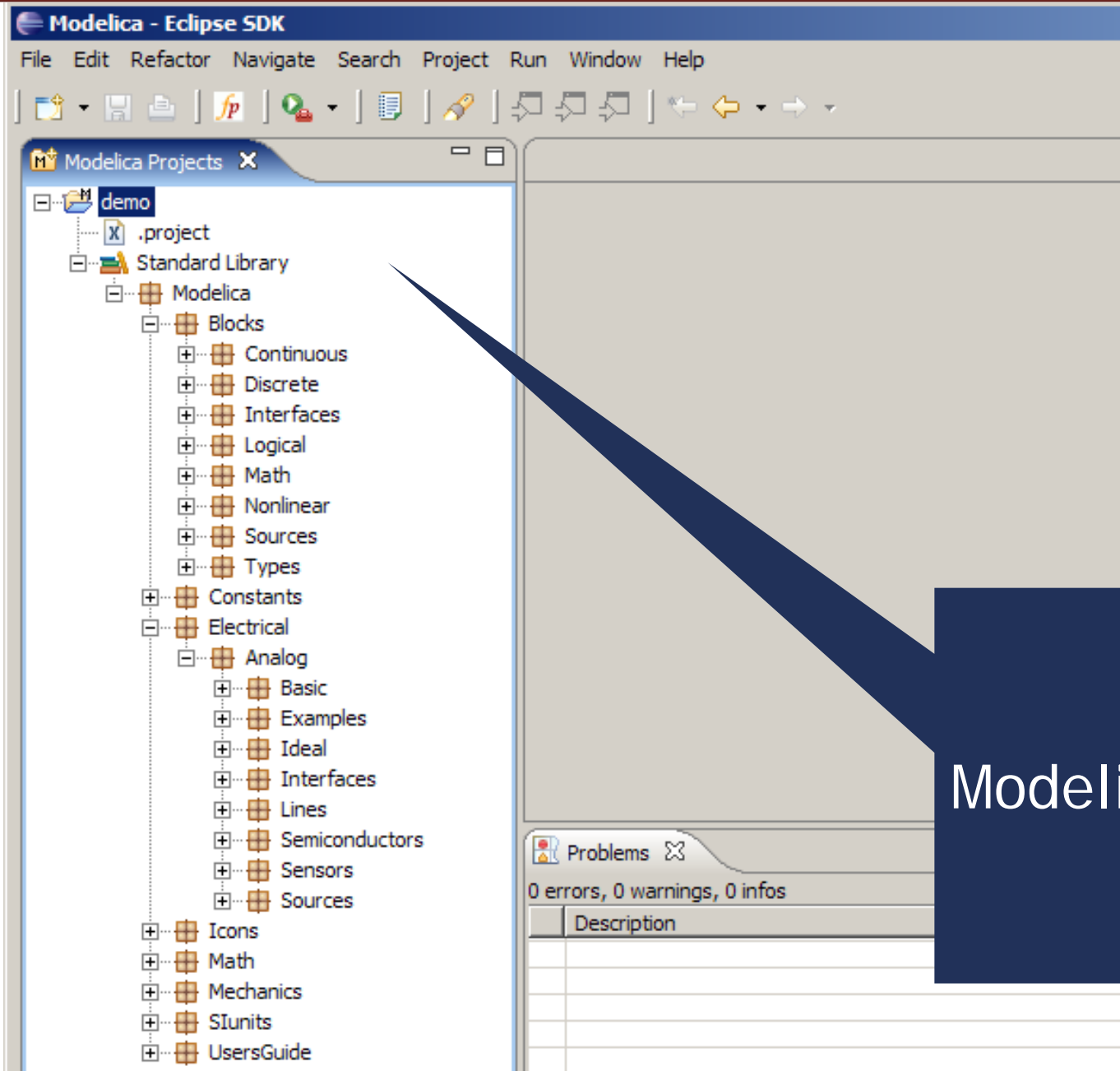
Create a Modelica project in the workspace.

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 SDK interface for creating a new Modelica package. The 'New' menu is open, and the 'Modelica Package' option is selected. The 'New Modelica Package' wizard 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. A blue callout box on the left contains the text: 'Creation of Modelica packages using wizards'.

# Creating Modelica classes

The image shows the Eclipse IDE interface for Modelica. The 'Modelica Projects' view on the left shows a project named 'demo' with a sub-package 'MyPackage'. A context menu is open over 'MyPackage', and the 'New Modelica Class' option is selected. The 'New Modelica Class' wizard dialog is displayed in the foreground, with the following settings:

- Source folder: demo/MyPackage
- Package: MyPackage
- Name: MyClass
- Restriction: model
- Modifiers:  include initial equation block,  is partial class,  have external body

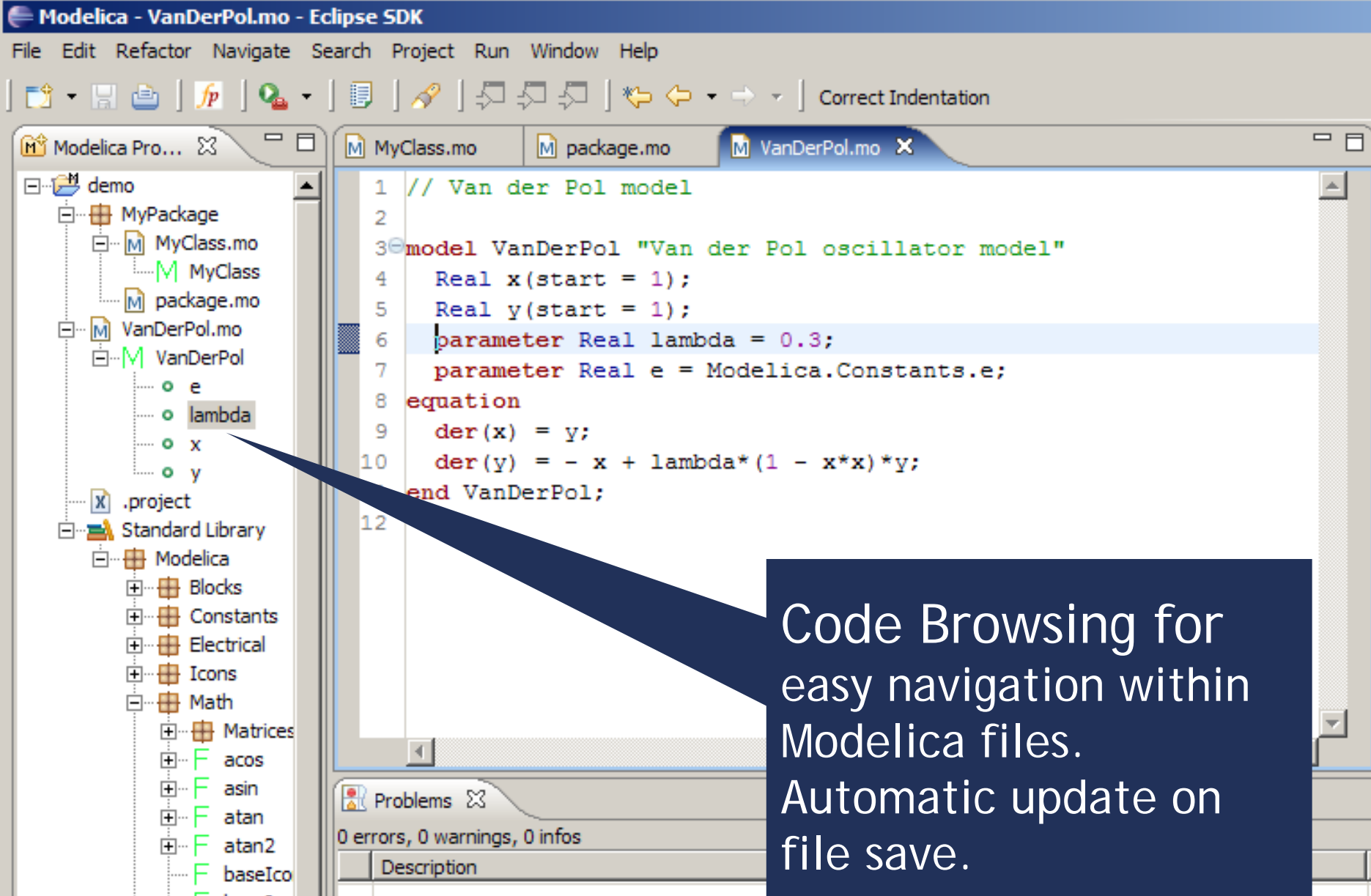
The 'Finish' button is highlighted with a red arrow. Another red arrow points from the 'Finish' button to the code editor on the right, which shows the following code:

```
1 within MyPackage;  
2  
3 model MyClass  
4  
5 equation  
6  
7 end MyClass;
```

Creation of Modelica classes, models, etc, using wizards



# Code browsing



The screenshot shows the Eclipse IDE interface with the title bar "Modelica - VanDerPol.mo - Eclipse SDK". The menu bar includes File, Edit, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains icons for file operations and a "Correct Indentation" button. The left sidebar shows a project tree with "demo" containing "MyPackage" (with "MyClass.mo" and "package.mo") and "VanDerPol.mo" (with "VanDerPol" containing "e", "lambda", "x", and "y"). The "lambda" element is selected. The main editor shows the code for "VanDerPol.mo" with the following content:

```
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 on line 6 is highlighted in blue. A blue 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 "Problems" with "0 errors, 0 warnings, 0 infos" and a "Description" tab.

# Error detection (I)

The screenshot shows the Eclipse IDE with the following components:

- Modelica - VanDerPol.mo - Eclipse SDK** window title.
- 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 sub-packages 'MyPackage' and 'VanDerPol'. 'MyPackage' contains 'MyClass.mo' and 'package.mo'. 'VanDerPol' contains 'VanDerPol.mo' and 'VanDerPol'. The 'Standard Library' is also visible.
- 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   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 (bottom): 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 with the following components:

- Modelica Projects:** A tree view on the left showing a project structure with folders like 'Compiler', 'absyn\_builder', 'doc', 'modpar', etc., and files like 'Absyn.mo', 'Algorithm.mo', 'Builtin.mo', etc.
- Absyn.mo:** The main editor window showing the source code of the 'Absyn.mo' file. The code includes a 'public' block, a 'uniontype Program' definition, and a 'record PROGRAM' definition. Line 77 is highlighted with a red 'x' icon, indicating an error.
- Problems/Console:** The bottom panel shows the error log. The error message is: `Absyn.mo:77.5-77.9 Error: unbound type constructor Withi`. Below this, there are several lines of output from the compilation process, including 'make[2]: Leaving directory...' and 'make[1]: Leaving directory...'. A blue callout box points to the error message.

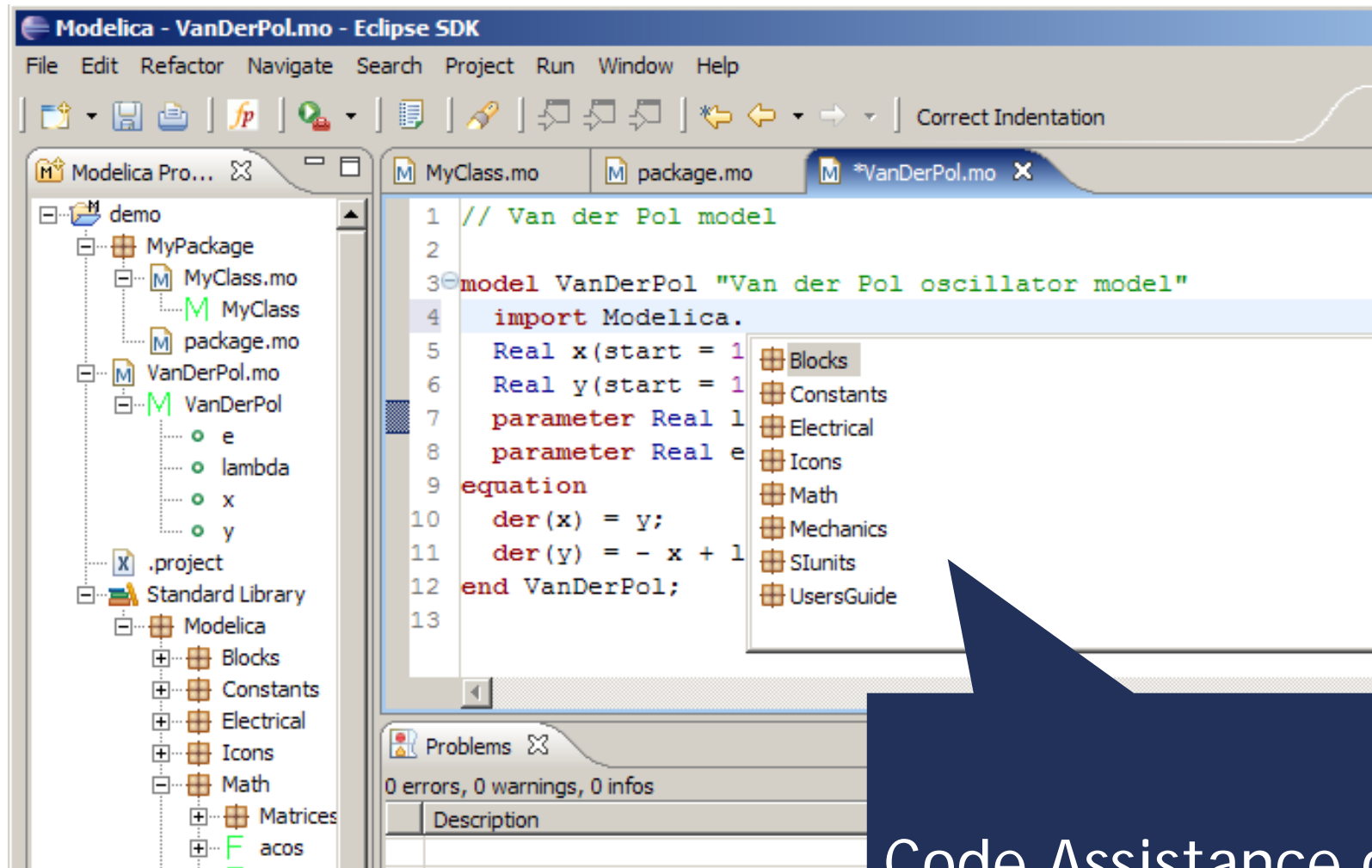
```
69 public
70 uniontype Program "- Programs, the top level construct
71   A program is simply a list of class definitions declared at top
72   level in the source file, combined with a within statement that
73   indicates the hieractical position of the program.
74 "
75 record PROGRAM
76   list<Class>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

**Problems/Console:**

```
<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/...
make[1]: Leaving directory `~/c/bin/...
make[2]: *** [Absyn.h] Error 1
make[1]: *** [omc_release] Error 2
make: *** [omc] Error 2
```

Semantic error  
detection on  
compilation

# Code assistance (I)



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
```

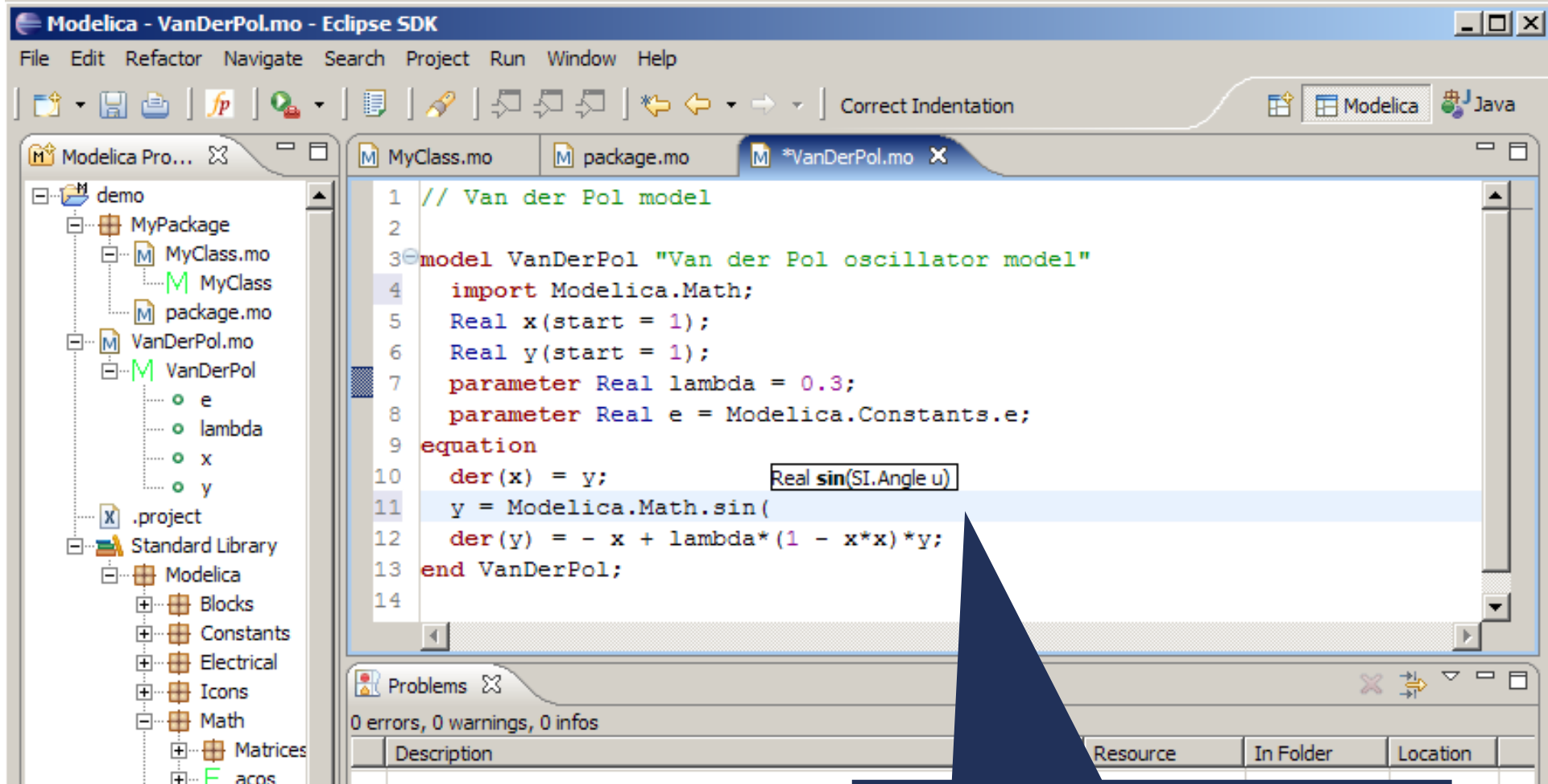
The cursor is positioned at the end of line 8, and a code completion popup is visible, listing the following options:

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

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

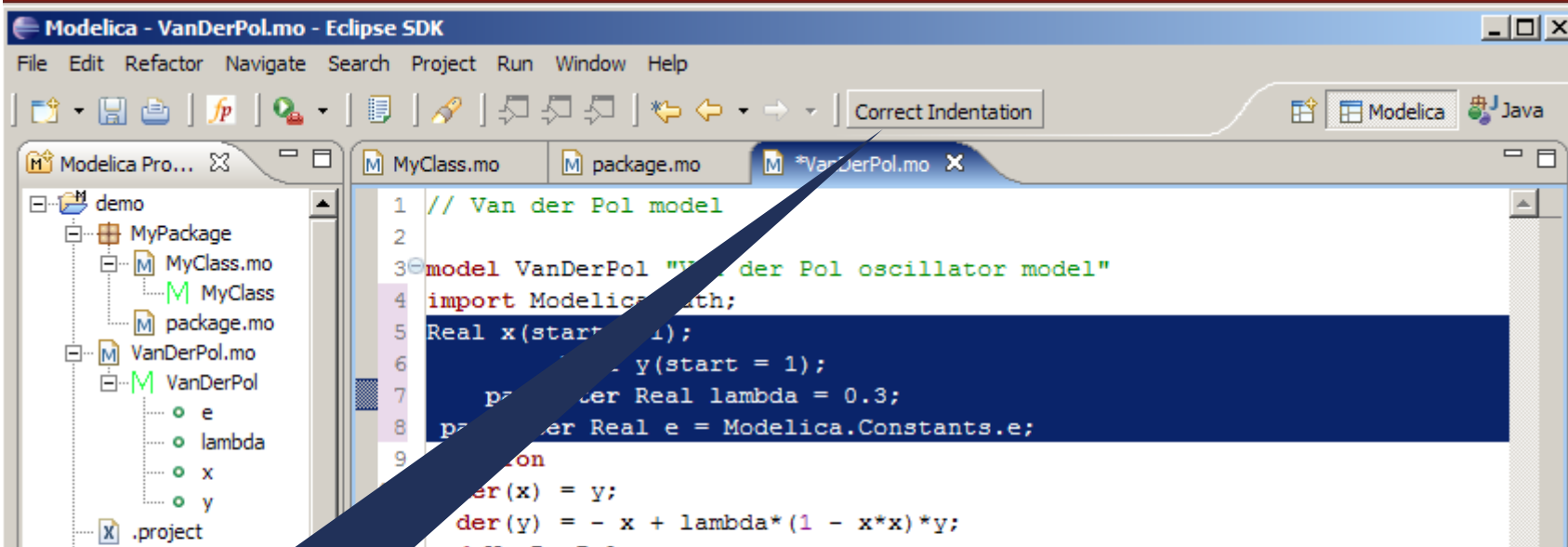
Code Assistance on assignments

# Code assistance (III)



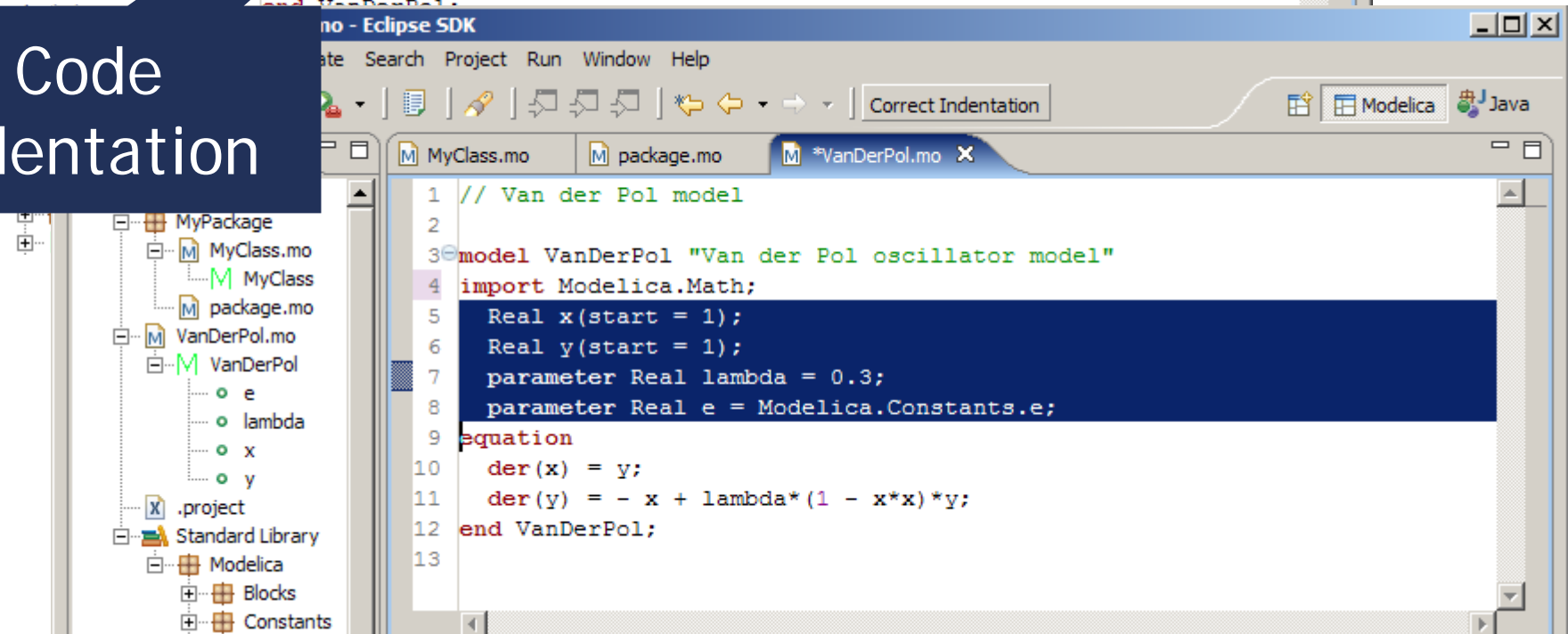
Code Assistance on  
function calls

# Code indentation



```
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
10 equation
11   der(x) = y;
12   der(y) = - x + lambda*(1 - x*x)*y;
13 end VanDerPol;
```

Code  
Indentation



```
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
10  equation
11    der(x) = y;
12    der(y) = - x + lambda*(1 - x*x)*y;
13 end VanDerPol;
```



# Code Outline and Hovering Info

The screenshot displays the Eclipse IDE with the following components:

- Project Explorer:** Shows a tree view of the 'Modelica Projects' folder, including subfolders like 'runtime', 'scripts', and 'tools', and files such as 'Absyn.mo', 'Algorithm.mo', 'Builtin.mo', etc.
- Code Editor:** Displays the source code of 'Absyn.mo'. A function definition is highlighted, and a tooltip provides its documentation: 'function: getCrefFromExp Returns a flattened list of the component references in an expression'. The code includes a 'case' statement for 'MATRIX' and 'RANGE', and a 'function' definition for 'getCrefFromExp'.
- Outline View:** Located at the bottom left, it shows a hierarchical list of symbols in the 'Absyn' package, including 'ADD', 'ALG\_ASSIGN', 'ALG\_BREAK', 'ALG\_CATCH', 'ALG\_EQUALITY', 'ALG\_FAILURE', 'ALG\_FOR', 'ALG\_GOTO', 'ALG\_IF', 'ALG\_LABEL', 'ALG\_NORET\_CALL', 'ALG\_RETURN', 'ALG\_THROW', 'ALG\_TRY', and 'ALG\_WHEN\_A'.
- Problems View:** Located at the bottom center, it shows 113 errors, 0 warnings, and 0 infos. The error description is 'The identifier at start and end are different'.

Two callout boxes highlight key features:

- Code Outline for easy navigation within Modelica files:** Points to the Outline view.
- Identifier Info on Hovering:** Points to the tooltip over the function definition.



# Eclipse Debugging Environment

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

The screenshot shows the Eclipse IDE with the following components:

- Debug Console:** Shows the execution flow, including 'Main.translateFile (line: 365, SP: 21, call: ...)' and 'Main.main (line: 919, SP: 9, call: extern)'. The current execution point is at line 919.
- Variables View:** Displays a tree structure of variables and their values. The root variable is 'p' of type 'Absyn.Program'. It contains a 'record' of type 'Absyn.PROGRAM[2]' with fields like 'classes', 'name', 'partial\_', 'final\_', 'encapsulated\_', 'restriction', 'body', 'classParts', 'contents', 'comment', 'info', and 'within\_'. The 'name' field is 'Bla', and the 'info' field is 'Absyn.INFO[6]'. The 'body' field is 'Absyn.PARTS[2]'.
- Console:** Shows the output of the program, including 'OMCD [Modelica Development Tooling (MDT)]' and 'C:\bin\cygwin\'. The output is currently empty.
- Outline:** Shows the project structure, including 'readSettingsFile(String filePath, Interactive.InteractiveSym...', '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)', and 'versionRequest'. The 'translateFile' method is highlighted.
- Tasks and Problems:** Shows the status of tasks and problems, with '69M of 254M' memory usage and 'Ctrl Contrib (Bottom)' button.

# Eclipse environment for ModelicaML

The screenshot shows the Eclipse IDE with the Modelica environment. The main window displays a UML class diagram with the following classes and relationships:

- TwoPin** (class): Parameters: P, n.
- Resistor** (class): Parameters: Real R (unit = 'Ohm').
- Inductor** (class): Parameters: Real L ((unit = "H")).

Relationships:

- TwoPin is associated with Resistor (parameter: + R1(R = 10)).
- TwoPin is associated with Inductor (parameter: + I(L = 0.1)).
- Resistor is associated with Inductor (parameter: + R2(R = 100)).

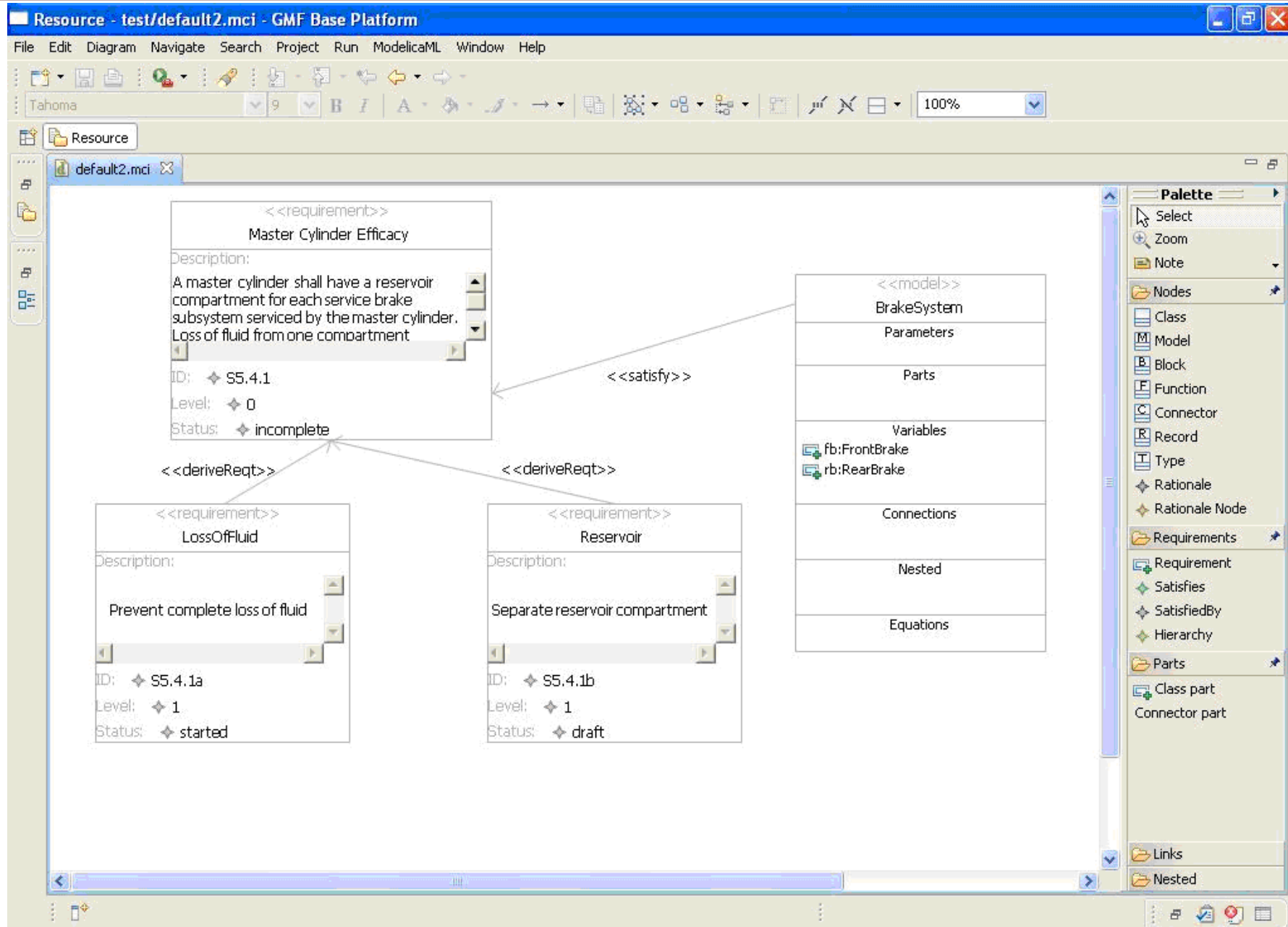
The left sidebar shows the project structure:

- Modelica Projects
  - SIunits.mo 469 2007-01-23 00:24 hubertus
  - StateGraph.mo 469 2007-01-23 00:24 huber
  - SimpleCircuit.mo
    - SimpleCircuit
      - AC
      - C
      - G
      - L
      - R.1
      - R.2
  - .project
  - SimpleCircuit.mcd
  - SimpleCircuit.mci
  - SimpleCircuit.sysml
  - Libraries: C:\OpenModelica 1.4.3\ModelicaLibrary
    - Modelica
      - extends Icons.Library;
      - Blocks
      - Constants
      - Electrical
        - extends Modelica.Icons.Library2;
        - Analog
          - extends Modelica.Icons.Library2;
          - Basic
            - extends Modelica.Icons.Librar
            - Capacitor
            - CCC

The bottom right shows the Properties window for the selected class:

| Property        | Value        |
|-----------------|--------------|
| UML             |              |
| Access          | public       |
| Array Dimension |              |
| Default         | unit = 'Ohm' |
| Direction       | inout        |
| Is Flow         | false        |
| Name            | R            |

# Requirements Modeling in Eclipse



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

# Latest Developments (2009-2010)

2009 - 2010

- 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)
  - Braking of constraint connection graph (VTT + OSMC)
  - Support for Modelica 3.x and 3.x annotations (OSMC)
  - Better structuring of the compiler (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**

## General Modelica issues solved

- array aliases (100%)
- enumerations (95%)
  - using enumerations as array indexes remains to be done
- inner outer with modifications on inner (95%)
  - works for constant modifications
  - started another implementation that uses the inner object directly

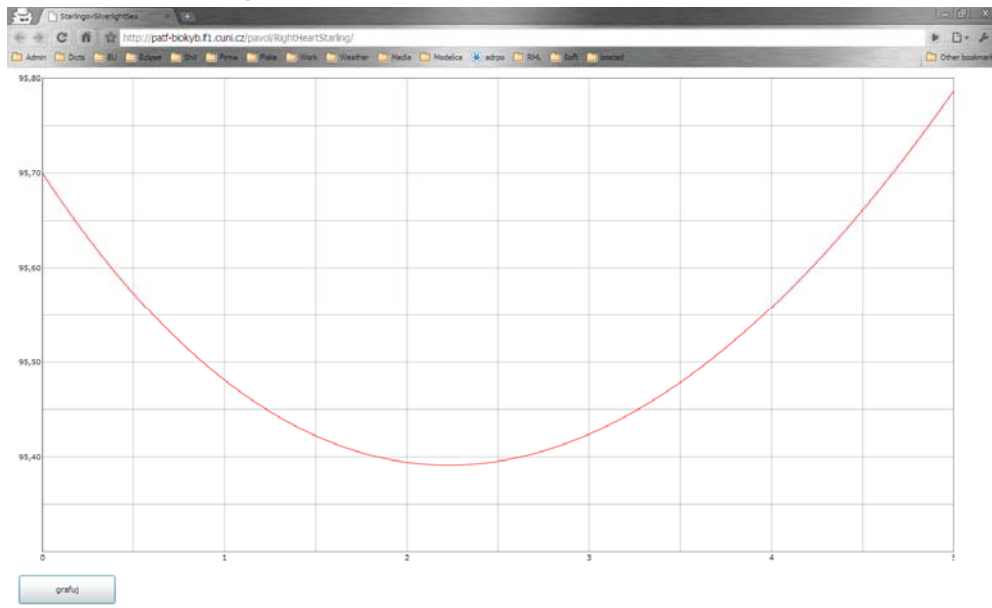
## MultiBody specific issues

- calling functions via component i.e. world.gravityAcceleration (100%)
  - not legal modelica and on plus the gravityAcceleration function is protected
- braking of over constrained connection graph (90%)
  - implemented by VTT (Hannu)
  - constraint types (100%)
  - some issues with inner/outer overlapping connection braking.
- performance issues (40%)
  - faster handling of inner outer
  - more caching in the compiler
- expandable connectors (90%)
  - Implementation as a phase before instantiation
  - Some small bugs still to be fixed

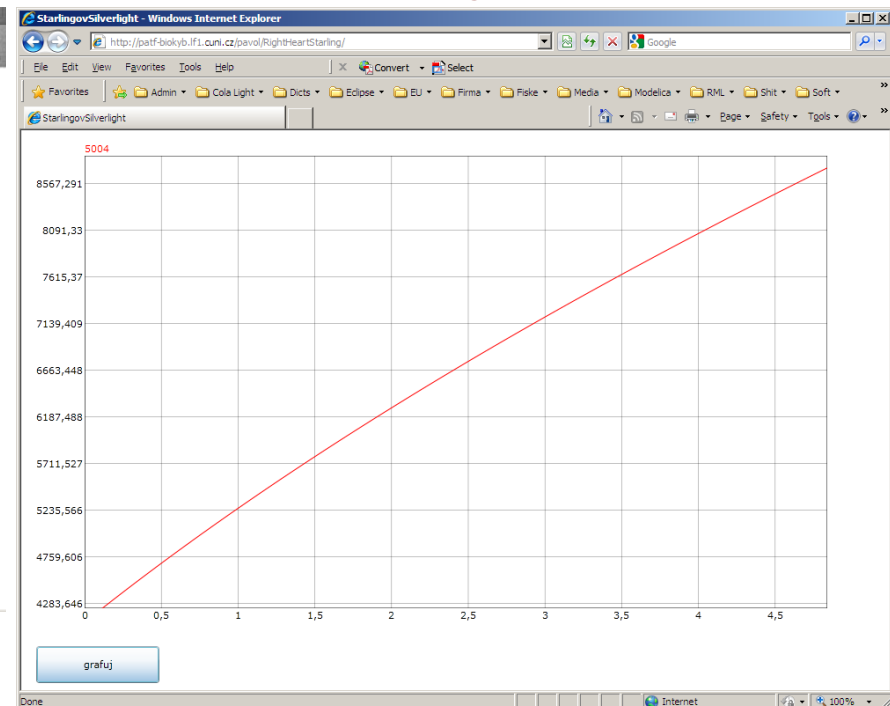
- The most evil Modelica standard library
- problems with partial functions in partial packages
- problems with full packages in partial packages used via the fully qualified path
- problems with redeclare replaceable model extends x.
- modifiers are in the wrong scope in the presence of redeclare replaceable model extends x.
- functions using redeclare replaceable function extends used to set constants in partial packages.

# Template Code Generation

- Pavol Privitzer,  
Charles University in Prague, Creative Connections s.r.o. Czech Republic
- Simple model of right heart Starling law
- IDA solver behind (in F#) linked with the code of the model (completely generated using the Susan template code generation in OpenModelica)
- **The generated model runs in the Browser on top of Silverlight!**



Input Pressure to Right Heart vs. Volume



Input Pressure to Right Heart vs. BlodFlow 47



# Modelica Development Tooling based on Xtext

The screenshot shows the Eclipse IDE interface for developing Modelica code using Xtext. The main editor displays the following code:

```
class ThermalMass
  Real T;
  Real Qdot;
  parameter Real mass;
  parameter Real cp;
equation
  if (Qdot > 0) then
    cp*mass*der(T)=Qdot;
  end if;
end ThermalMass;
```

An error tooltip is visible over the line `cp*mass*der(T)=Qdot;`, indicating a syntax error: `end of ThermalMass; this line mismatched input '<EOF>' expecting 'end' mismatched input 'ThermalMass' expecting 'if' 3 changed lines`. The error markers are highlighted with red 'x' icons in the left margin.

The Project Explorer on the left shows the project structure, and the Outline view on the right shows the Abstract Syntax Tree (AST) for the class, with an arrow pointing to the 'equation' block. The status bar at the bottom shows the current file is 'Resource - testXText/test2.mo - Eclipse SDK'.

**Error markers**

**Abstract Syntax Tree**

Thank You!  
Questions?

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