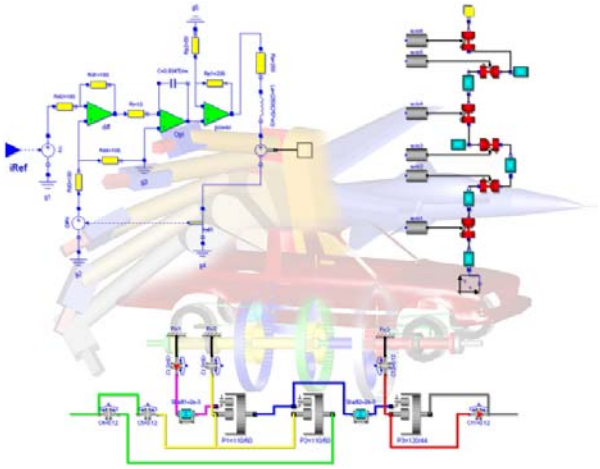
Technical Overview of OpenModelica and its Development Environment

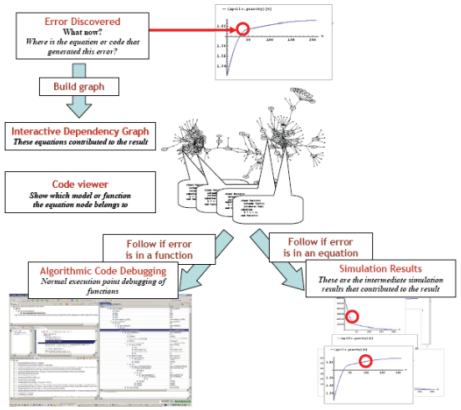
Adrian Pop

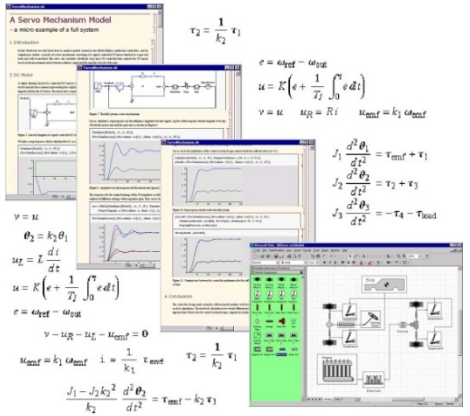
2010-02-08

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

www.OpenModelica.org







OPEN SOURCE MODELICA CONSORTIUM

OpenModelica

MODELICA

pelab

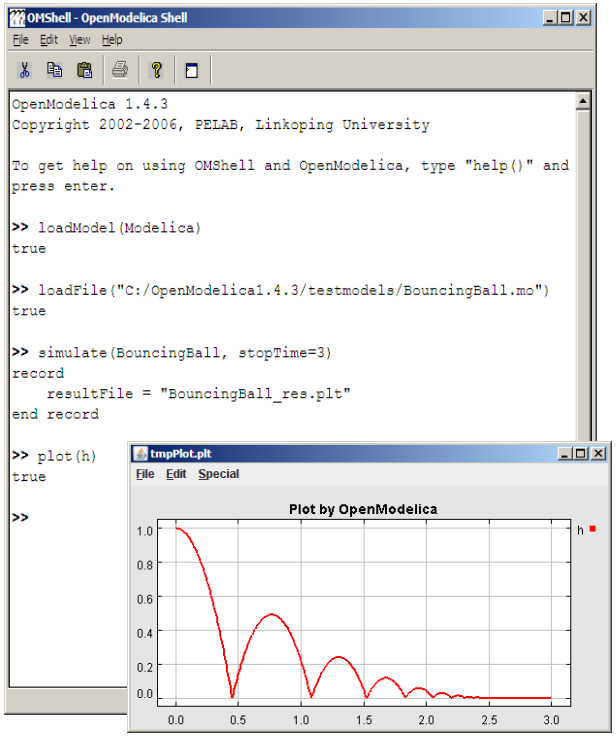
LINKÖPINGS UNIVERSITET

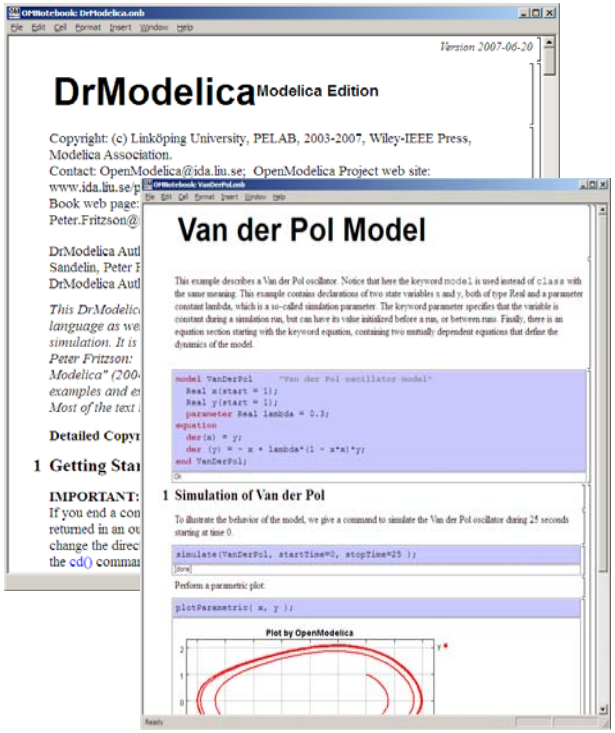
Outline

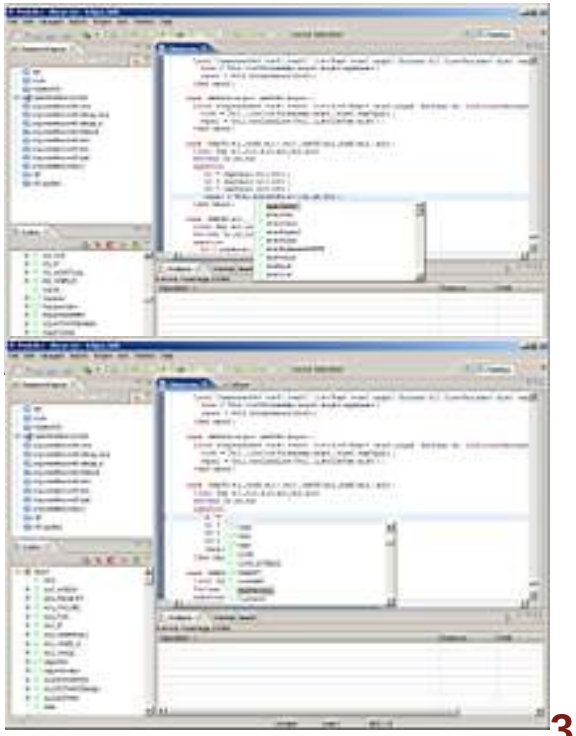
- OpenModelica
  - What is OpenModelica?
  - The past and present
- OpenModelica Technical Overview
  - OMC, OMShell, OMNotebook, SimForge
- OpenModelica Development Environment
  - MetaModelica
  - The Eclipse Environment
- OpenModelica Latest Developments (2009-2010)

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

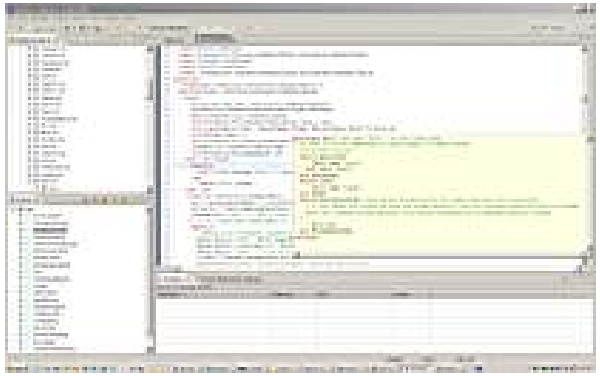


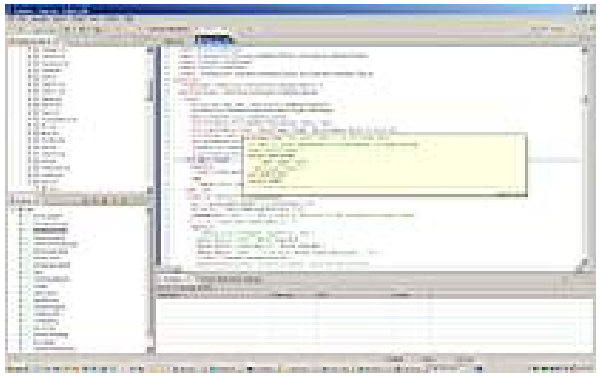
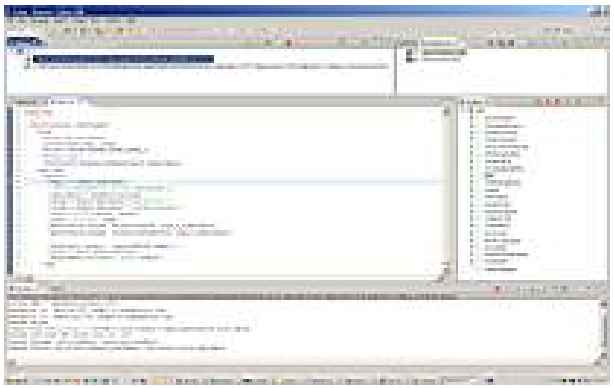
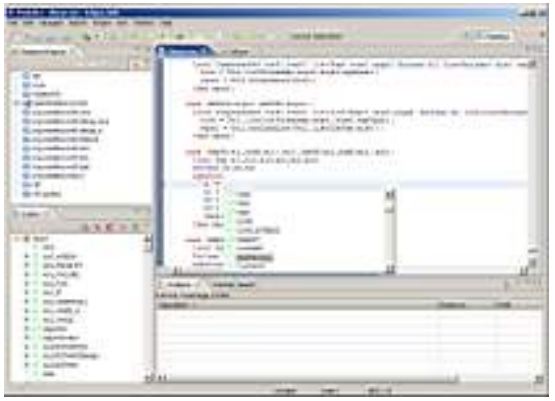




What Is OpenModelica? (II)

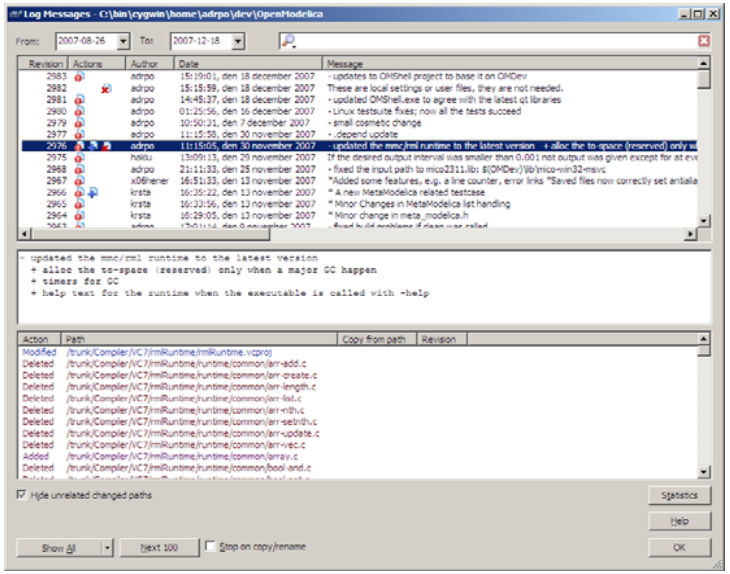
- Advanced Eclipse-based Development Environment
- **Modelica Development Tooling (MDT) - started in 2005**
  - Code Assistance, Debugging, Outline & a lot more
  - Now working on UML/SysML integration and better debugging
  - *Used heavily for OpenModelica development*
  - Used in 6 OpenModelica Development Courses (INRIA, PELAB)

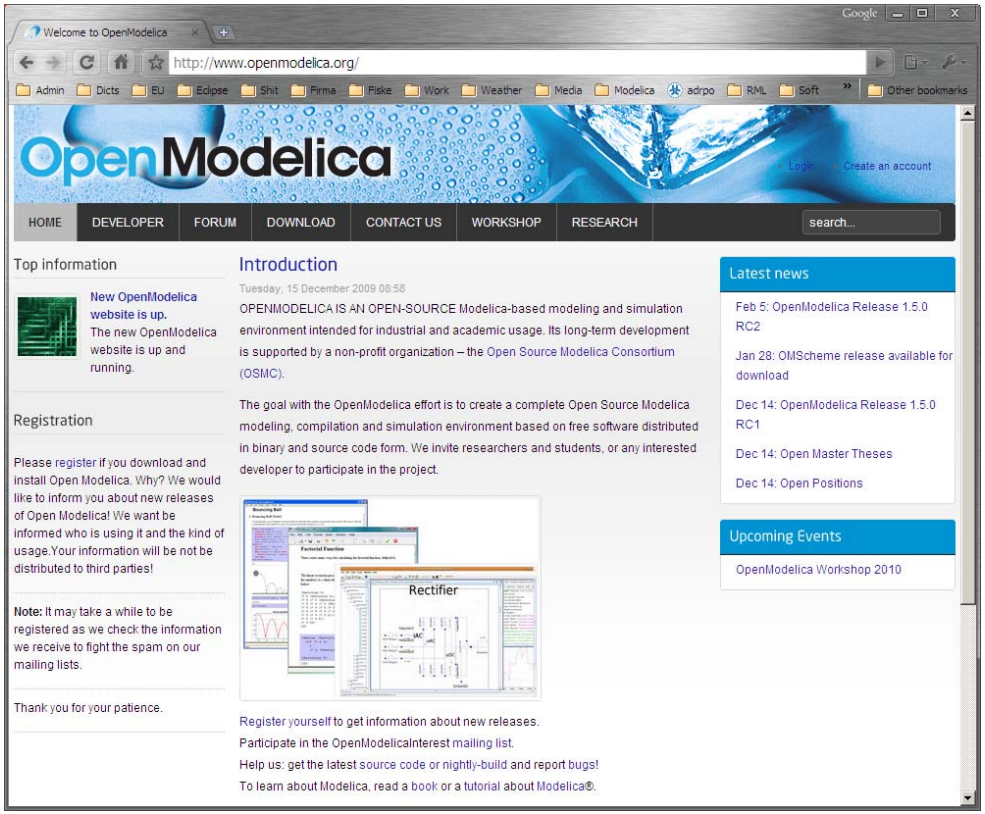


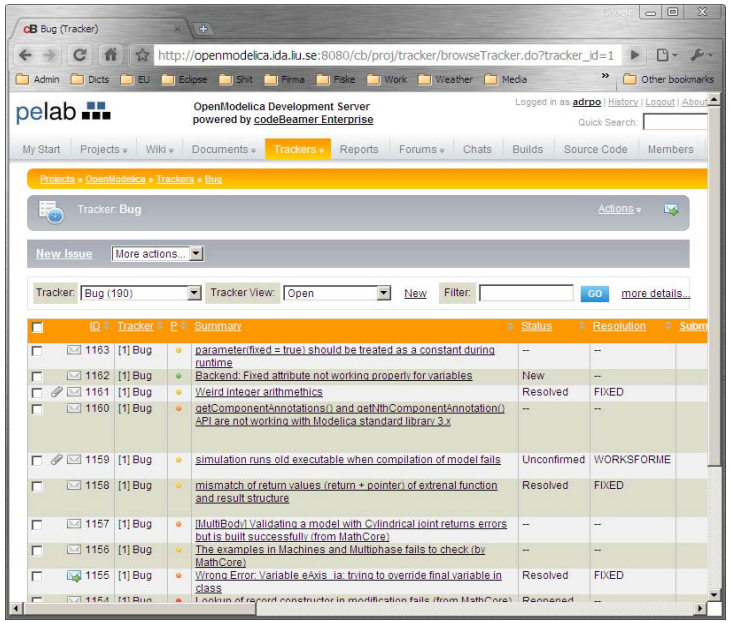





What is OpenModelica? (III)

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







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

OpenModelica Roadmap - Past

2008 – Further work on the compiler

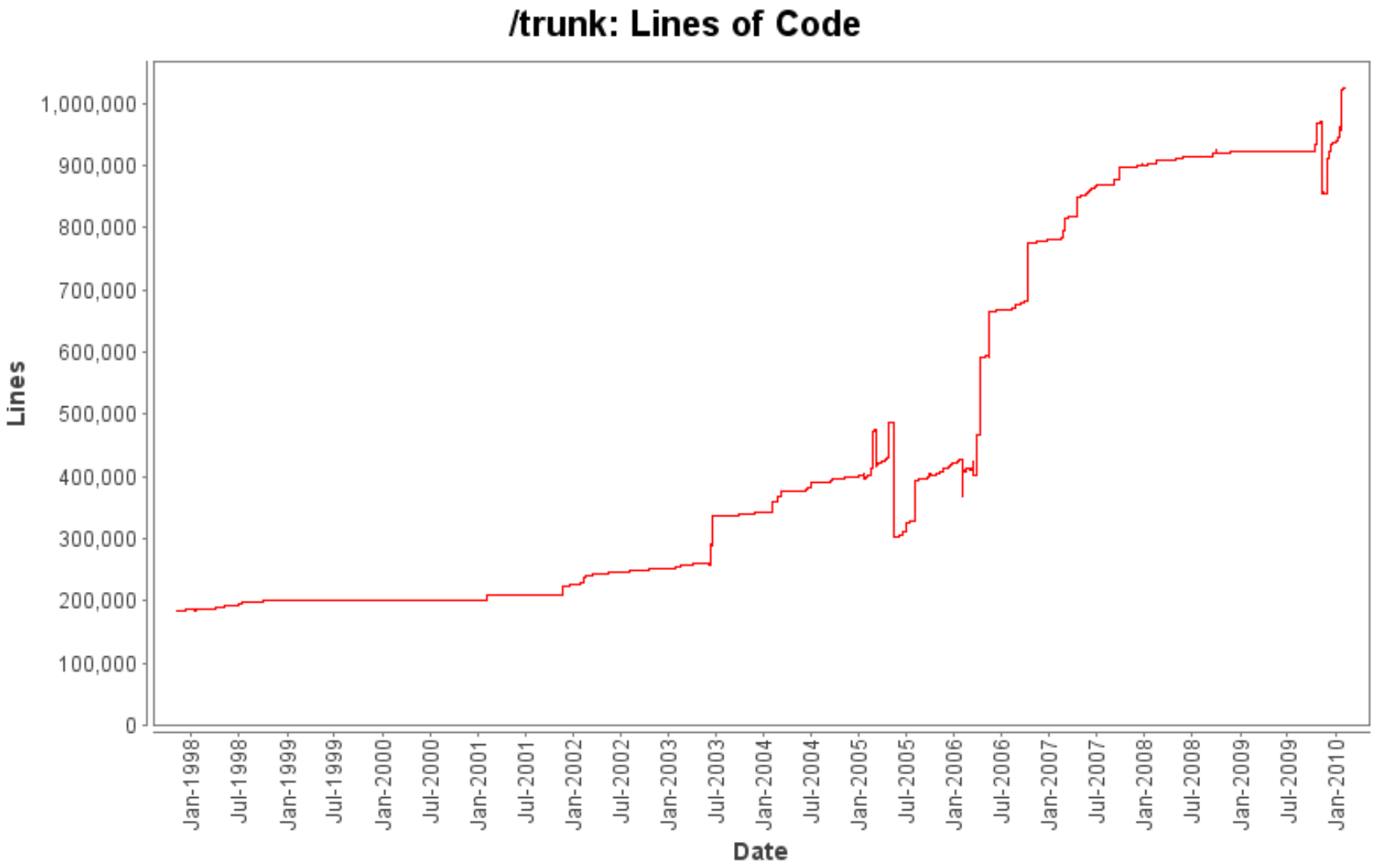
- Release 1.4.4 and 1.4.5
  - Linux, Mac, Solaris, Windows
- New Solver Interface
- Refactoring
- Dynamic loading of functions
- Merging of MathCore front-end code
- 744 commits in Subversion
- Much more other things I don't remember

OpenModelica Roadmap – Past & Present

2009 – 2010

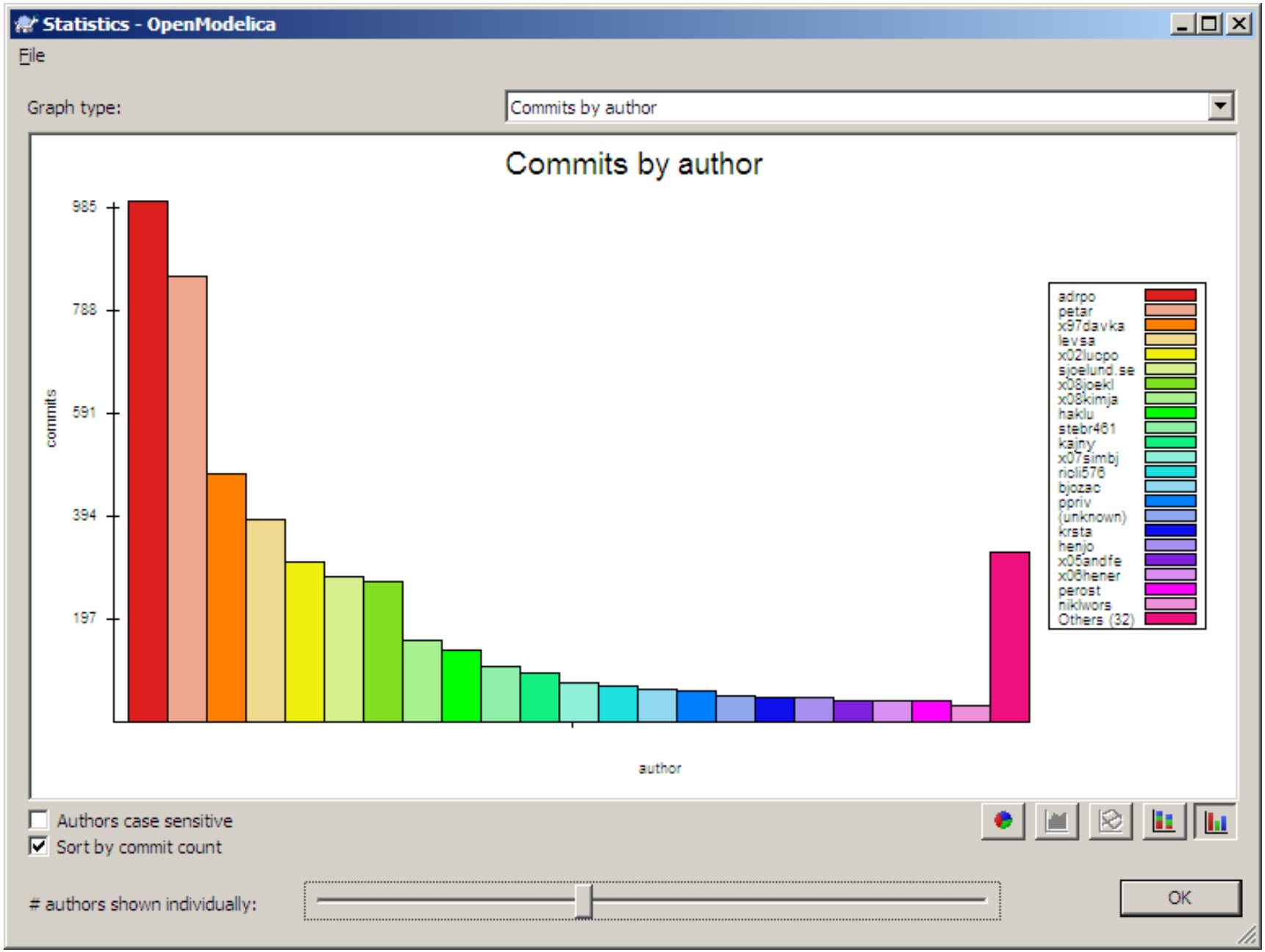
- Work mainly happened in OSMC (partially on a non-public branch)
- Front-end
  - Refactoring (OSMC)
  - Enumerations (OSMC)
  - Java Interface and Booststrapping (Martin Sjölund)
  - MultiBody flattening (OSMC)
  - Braking of constraint connection graph (VTT + OSMC)
  - Support for Modelica 3.x and 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)

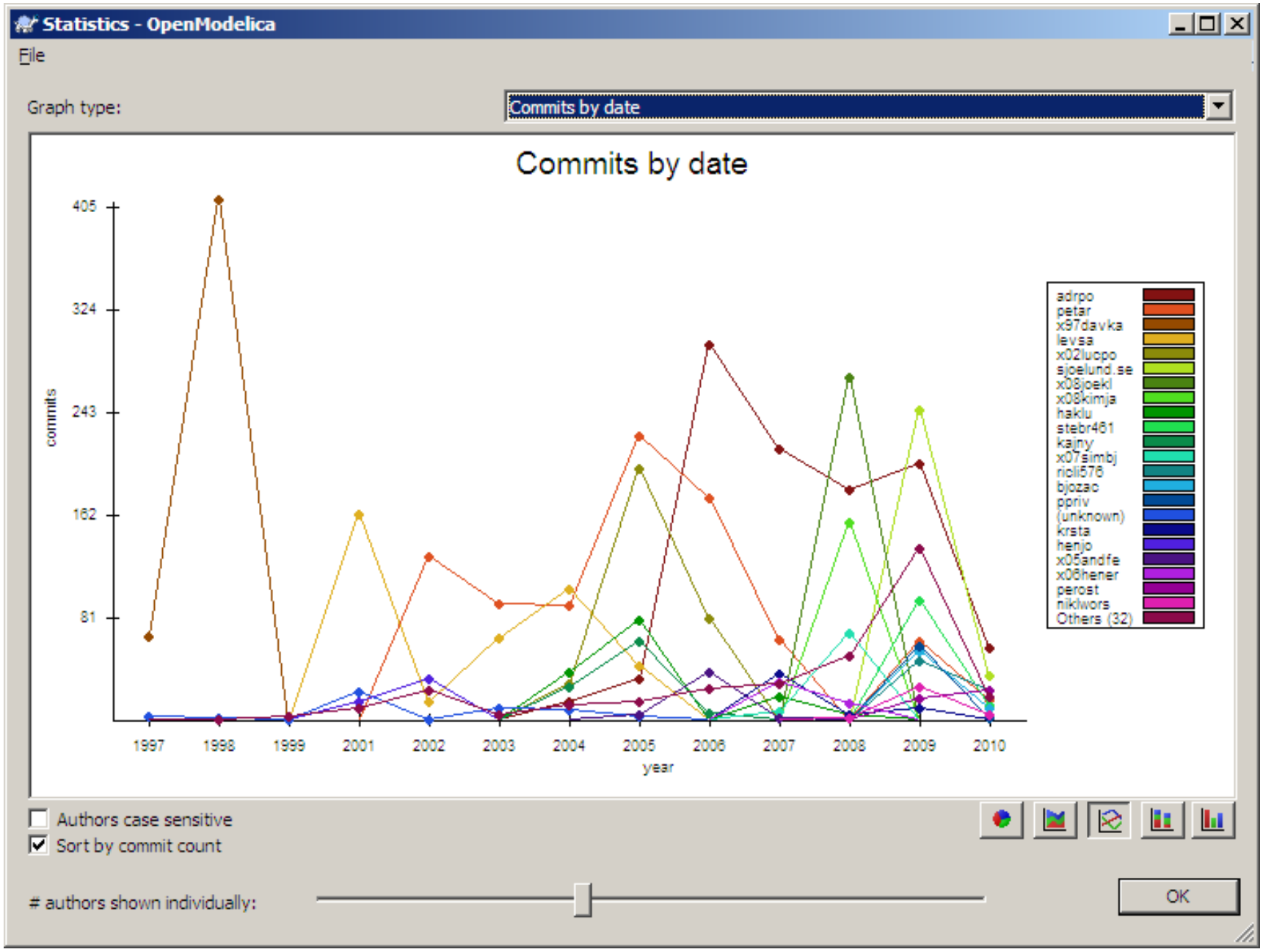


- Mature code base
- ~ 1000K lines of code, doubled since 2005

OpenModelica Statistics (II)



OpenModelica Statistics (III)



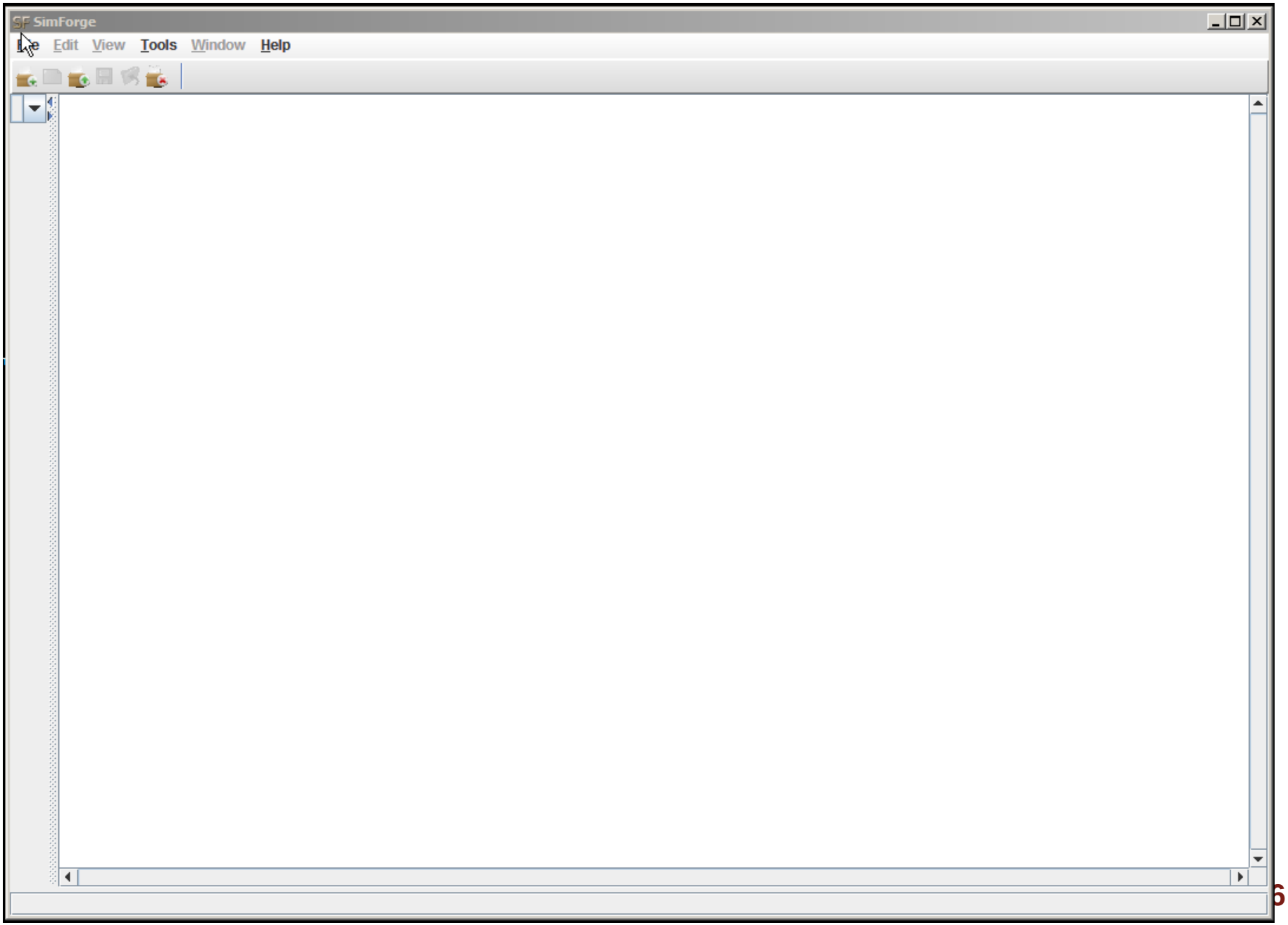
Outline

- 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)

OMShell & OMNotebook

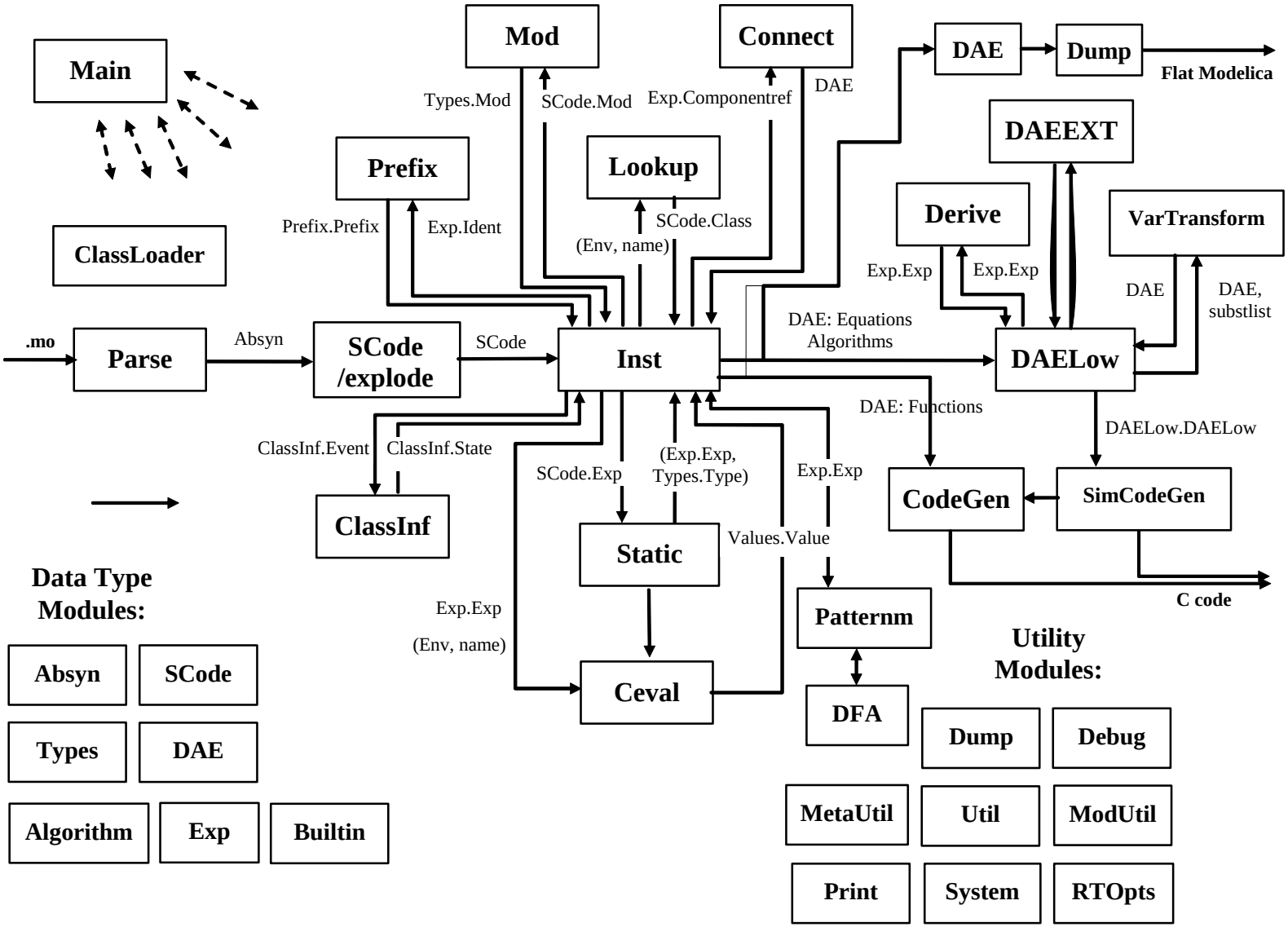
- Demo?

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);
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
```

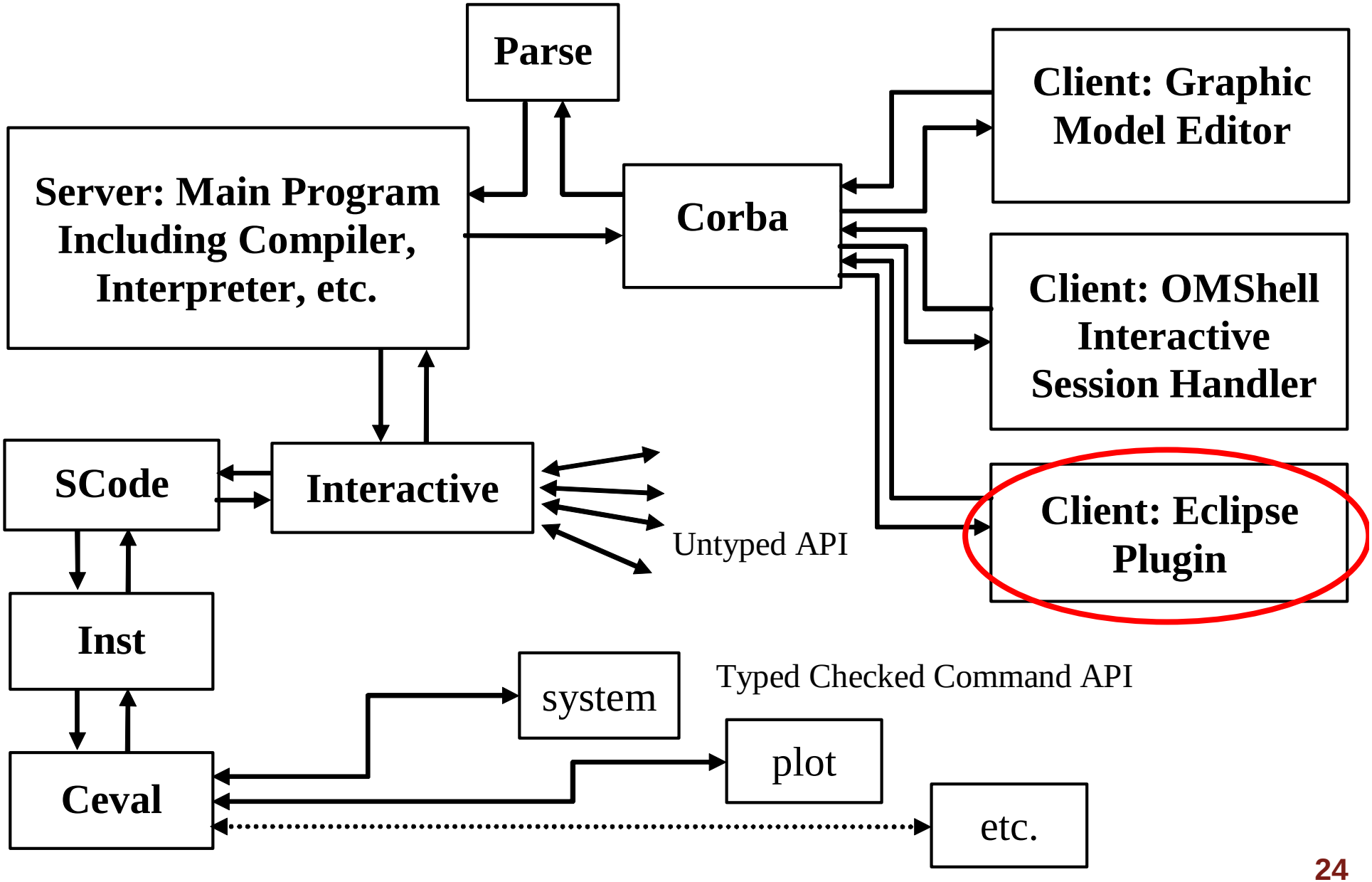
Outline

- 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)

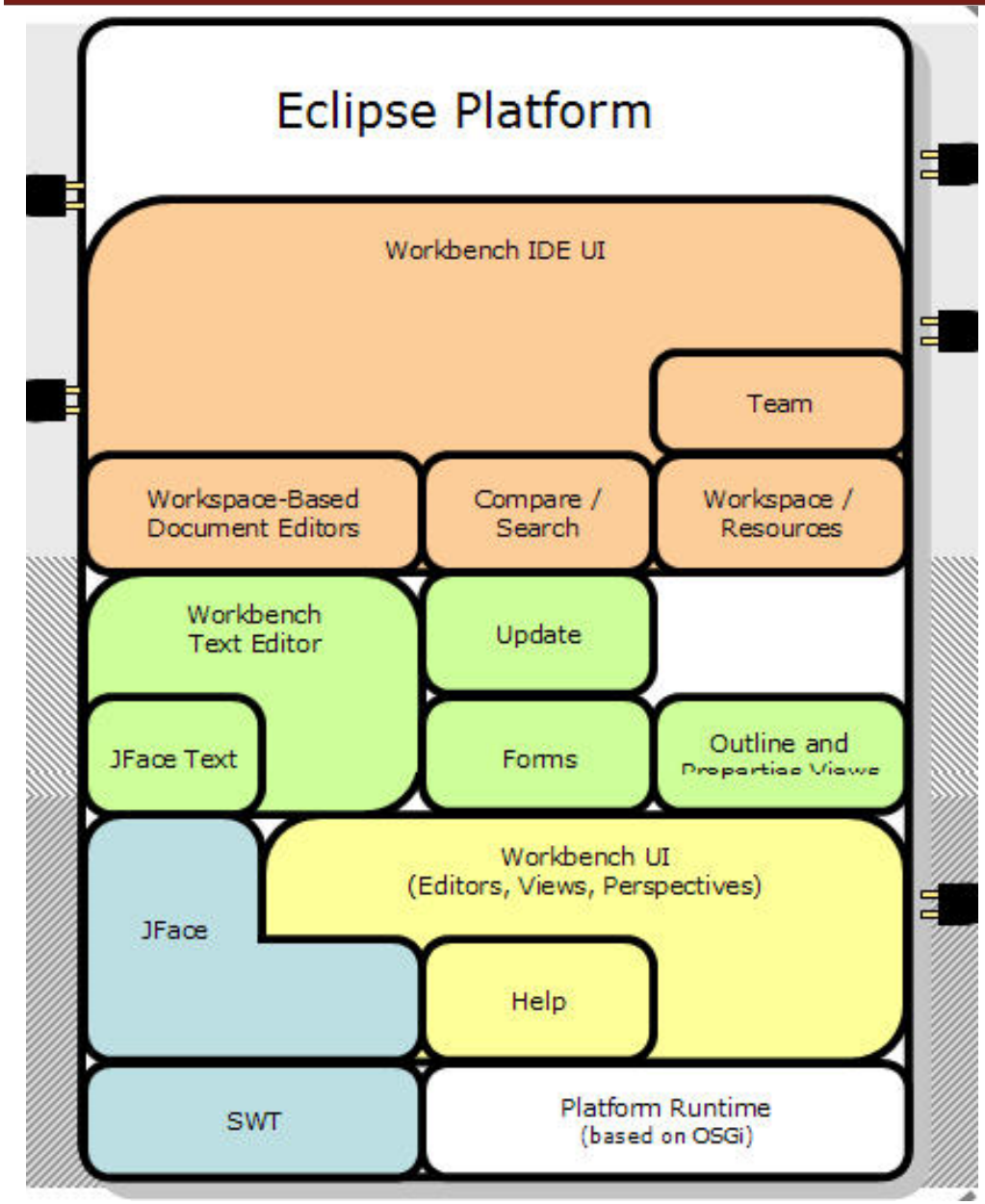
MetaModelica

- 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

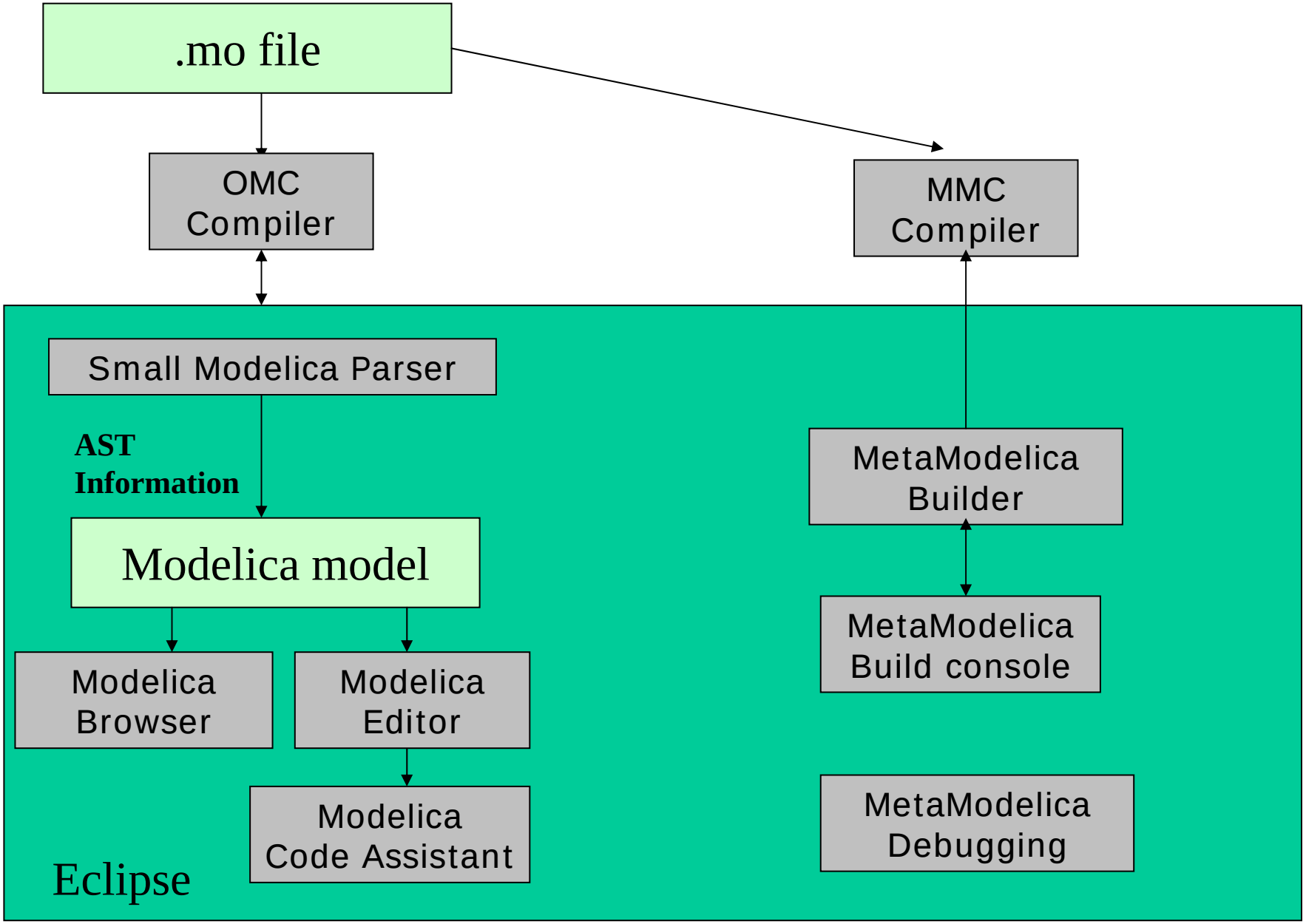
Modelica Editor

Modelica Code Assistant

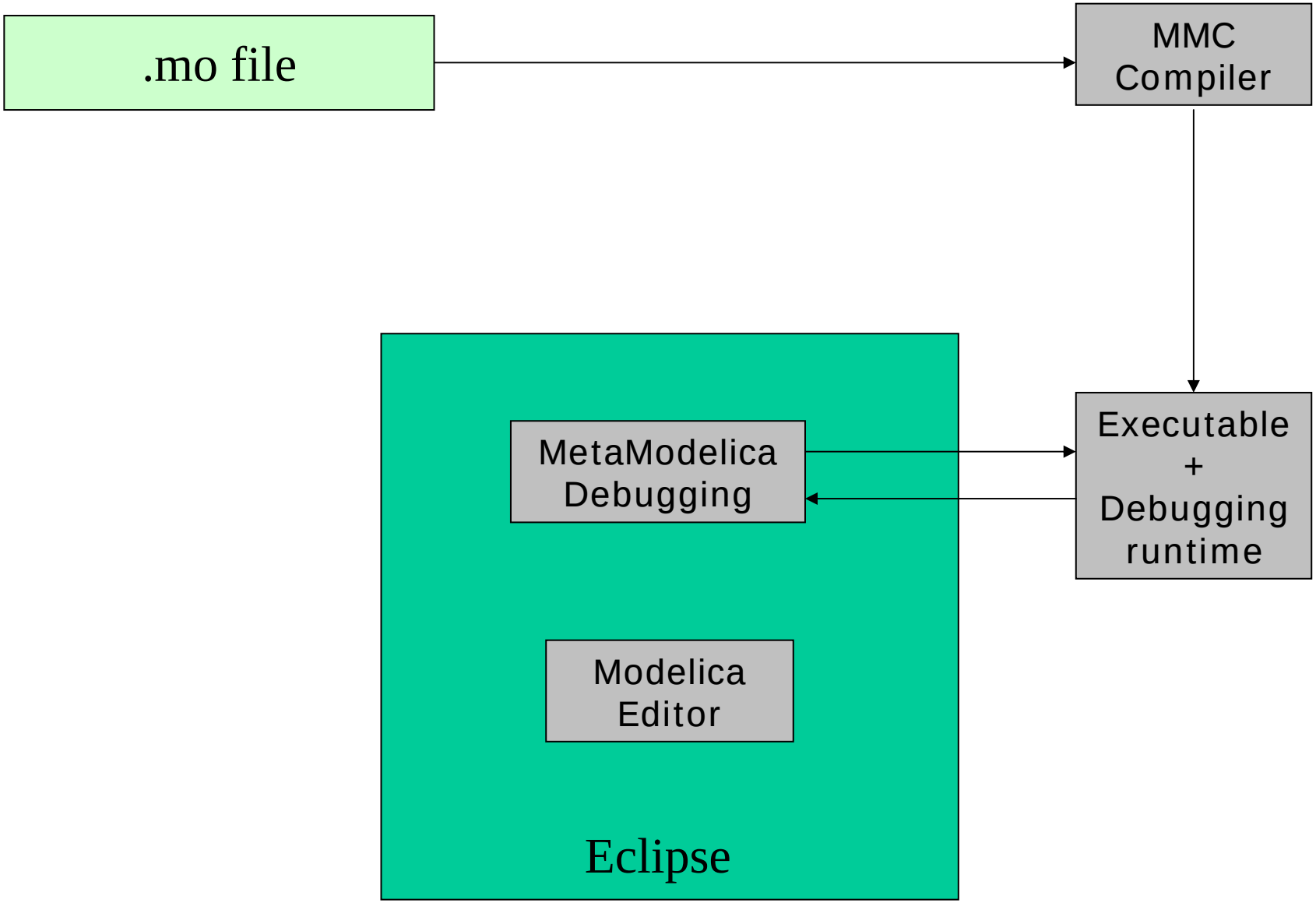
MetaModelica Debugging

Modelica Perspective

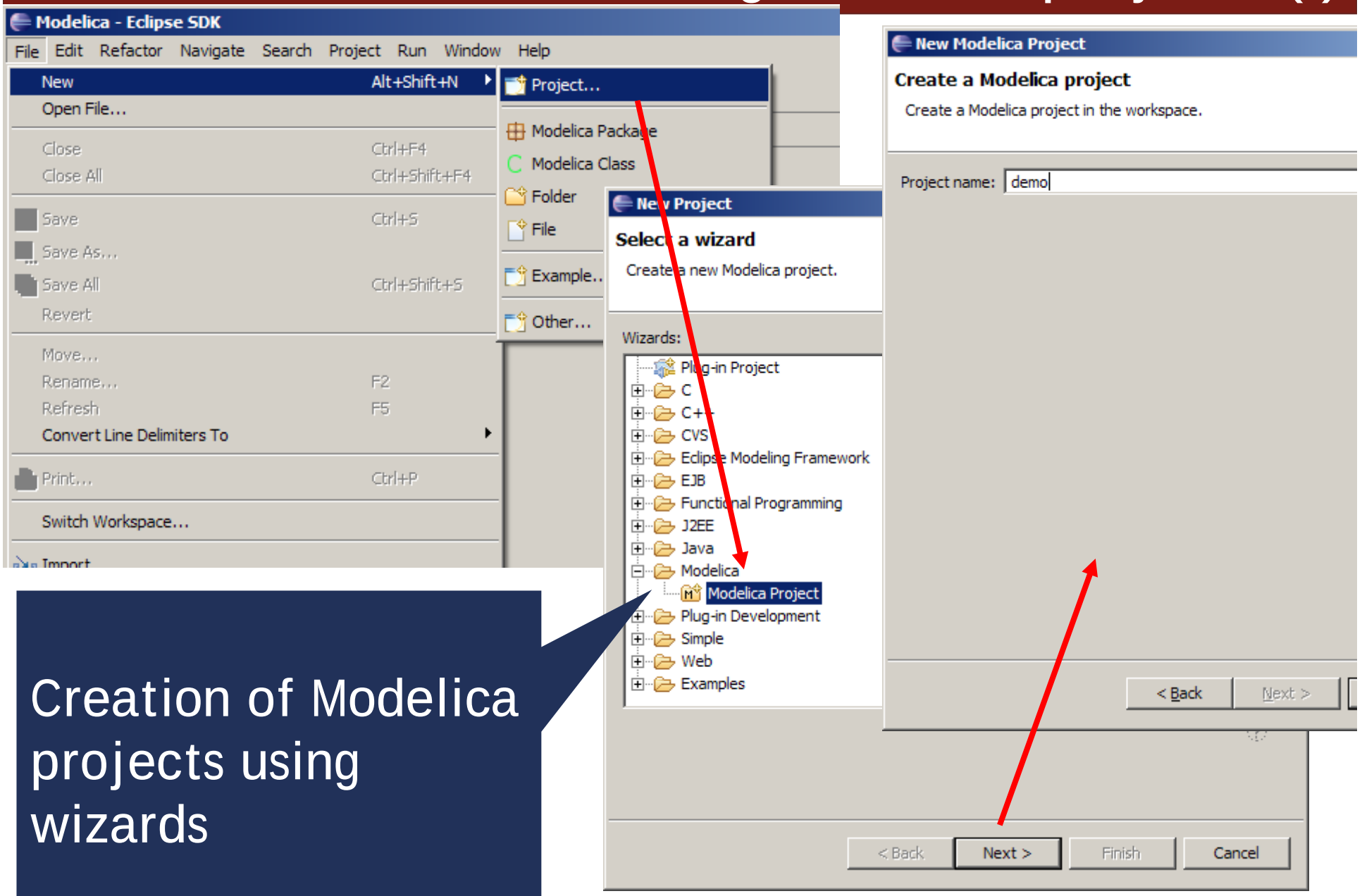
The MDT Eclipse Environment (II)



The MDT Eclipse Environment (III)

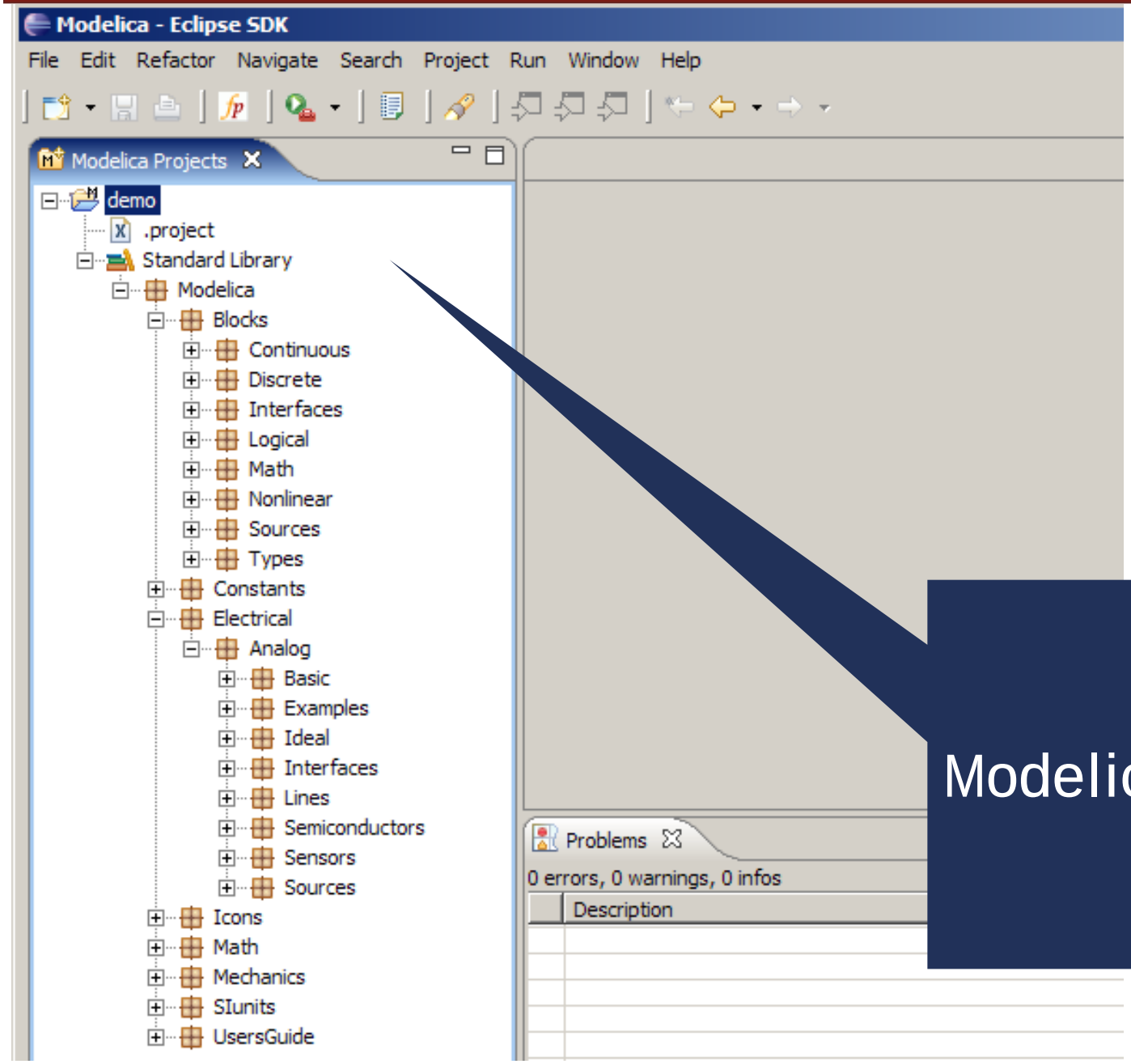


Creating Modelica projects (I)



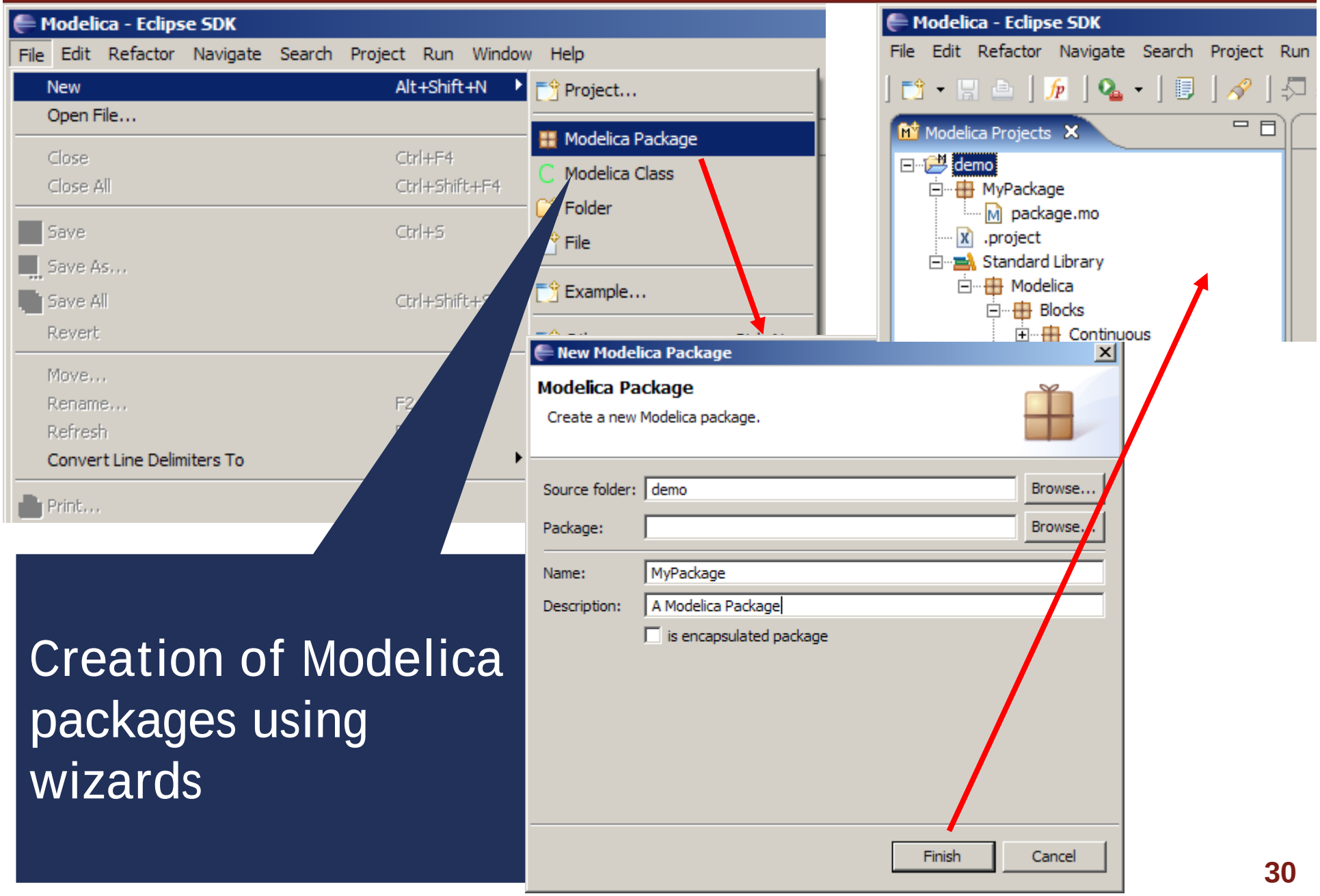
Creation of Modelica projects using wizards

Creating Modelica projects (II)



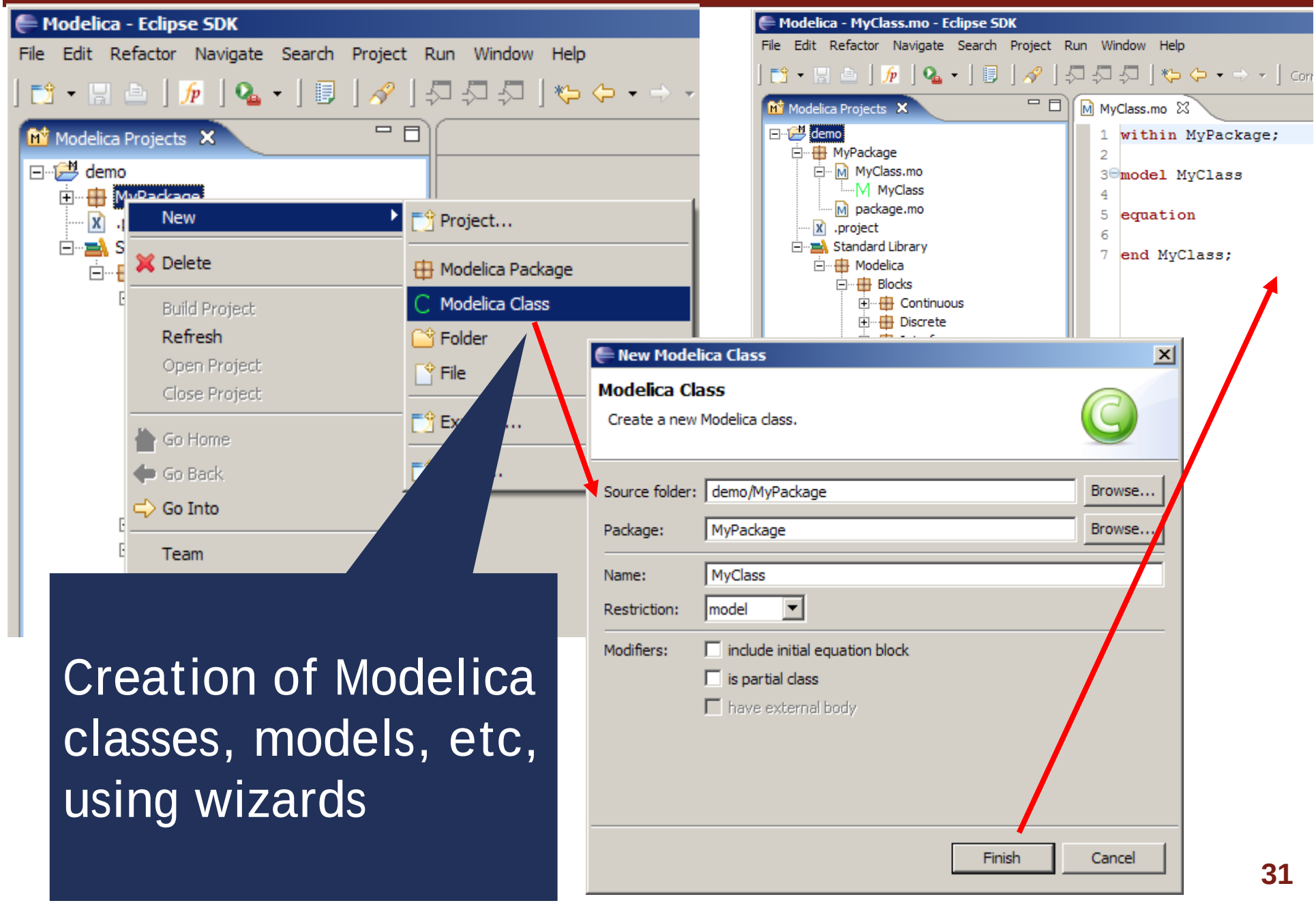
Modelica project

Creating Modelica packages



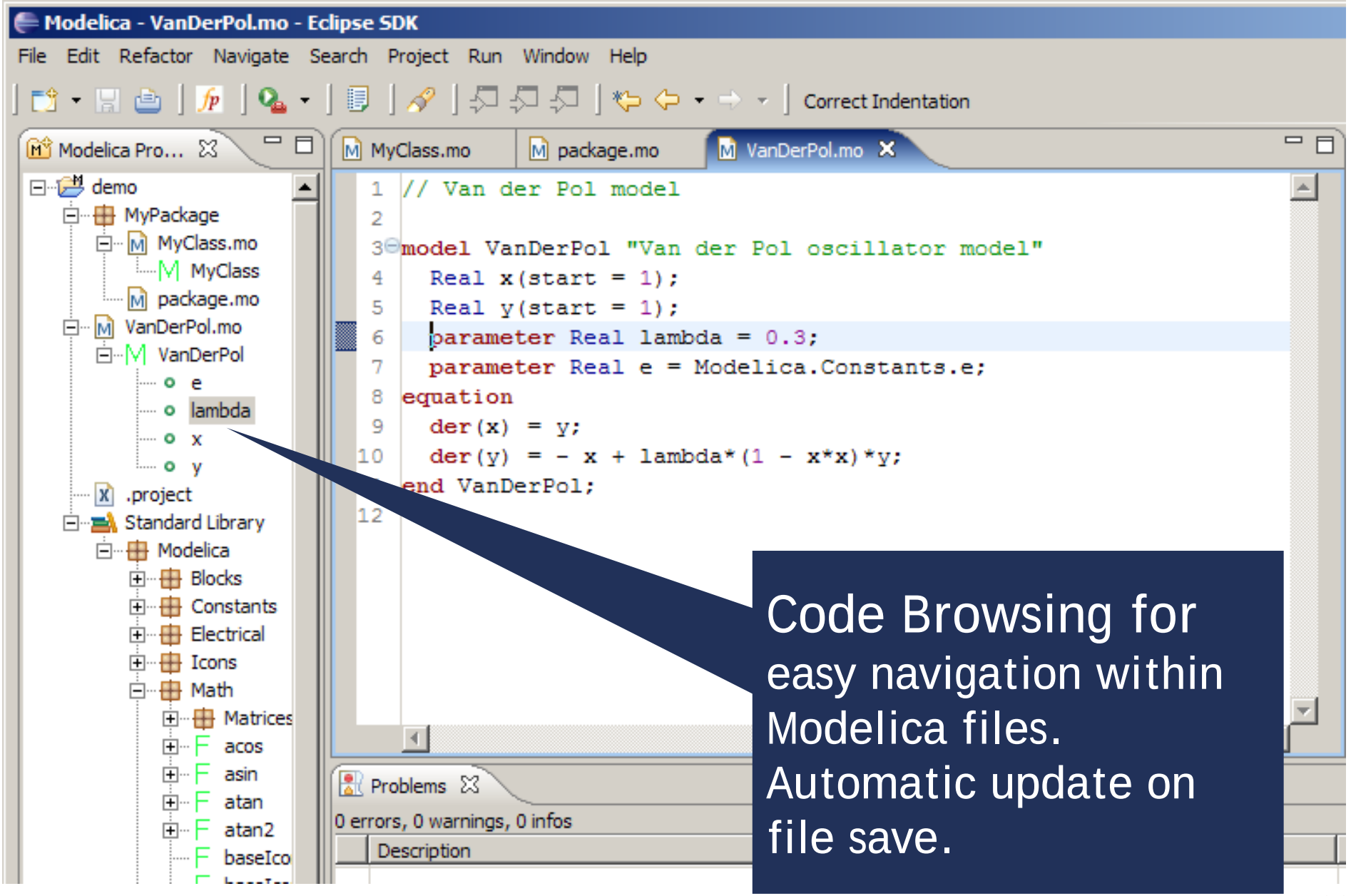
Creation of Modelica packages using wizards

Creating Modelica classes



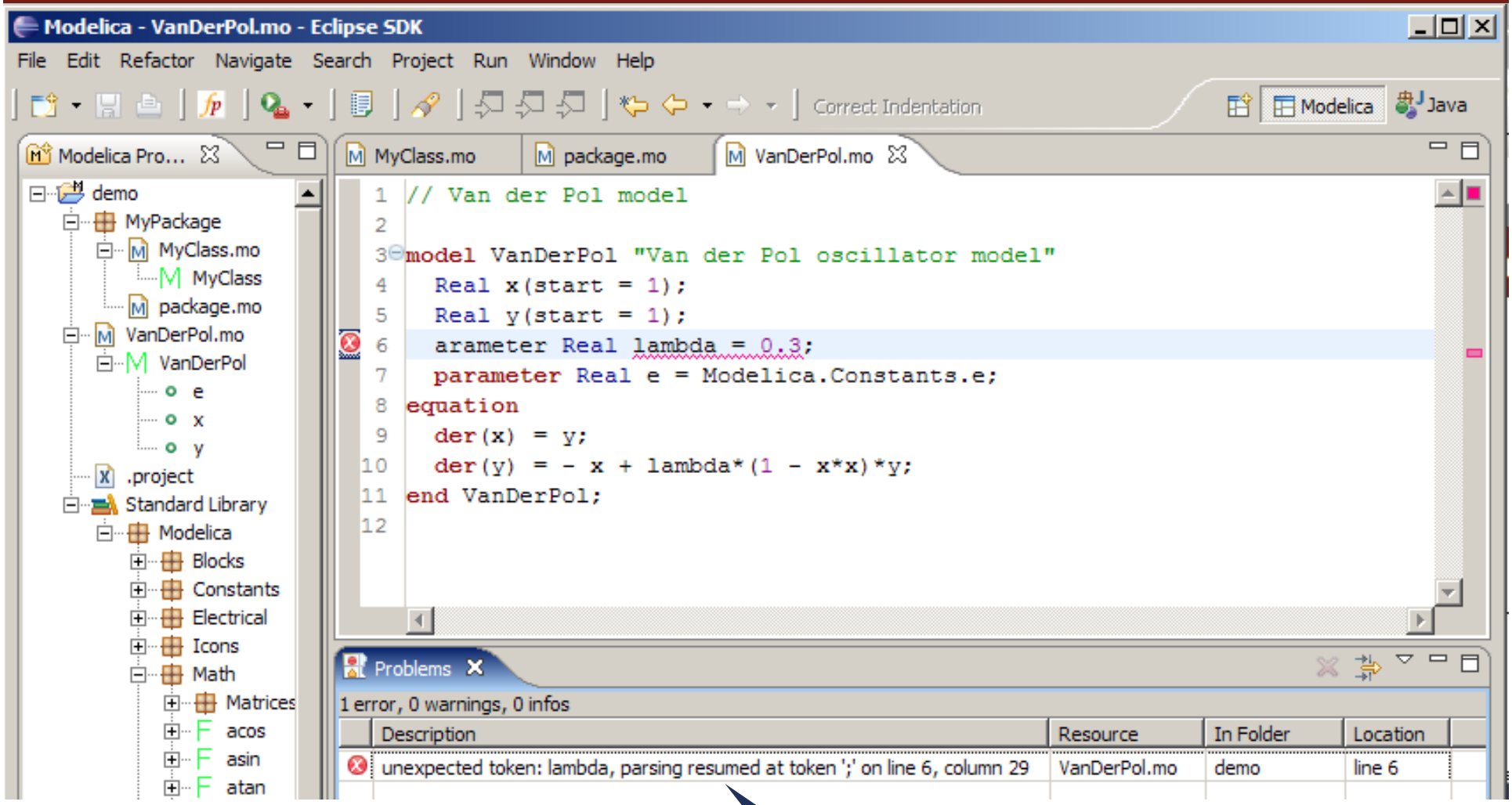
Creation of Modelica classes, models, etc, using wizards

Code browsing



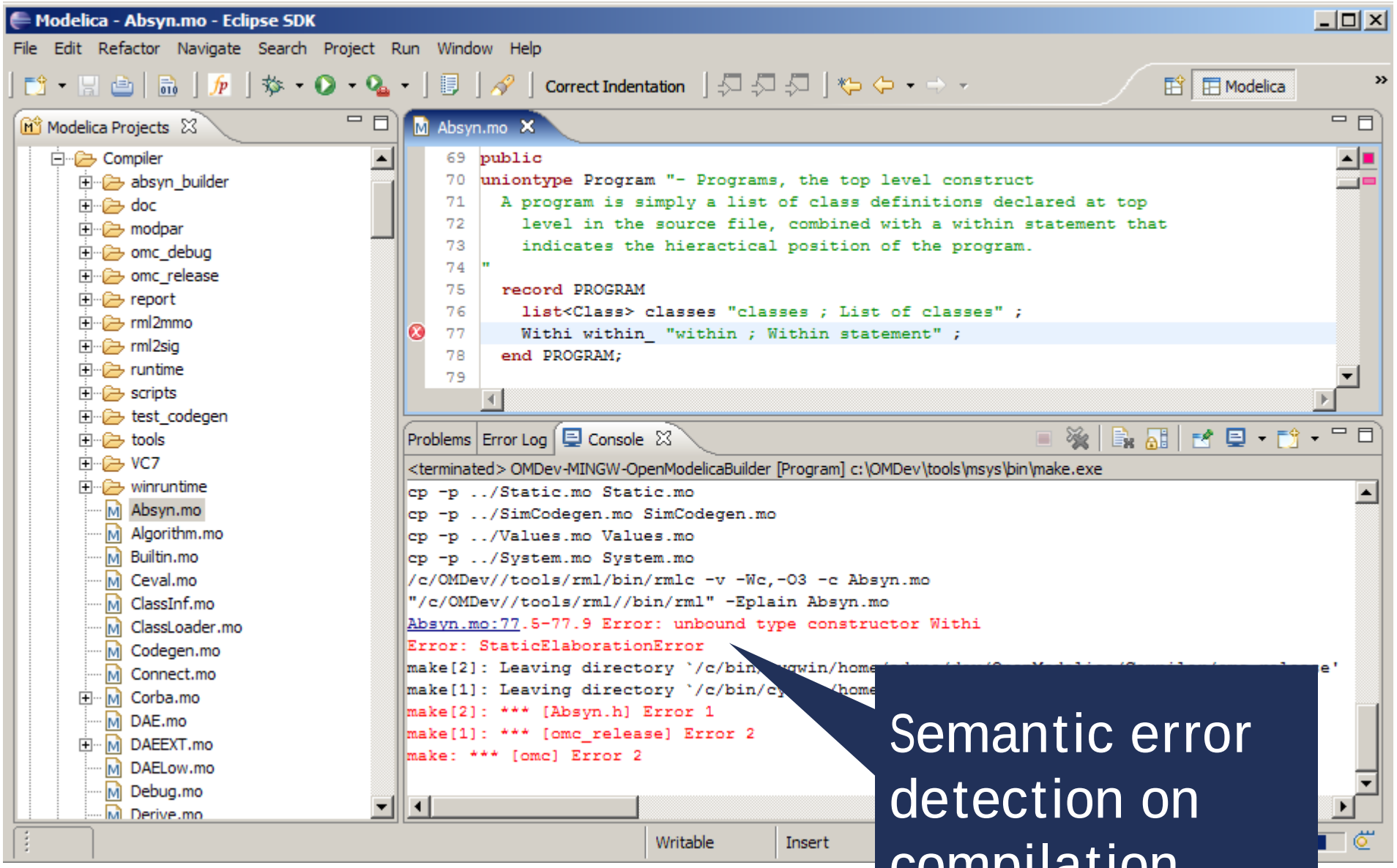
Code Browsing for easy navigation within Modelica files. Automatic update on file save.

Error detection (I)

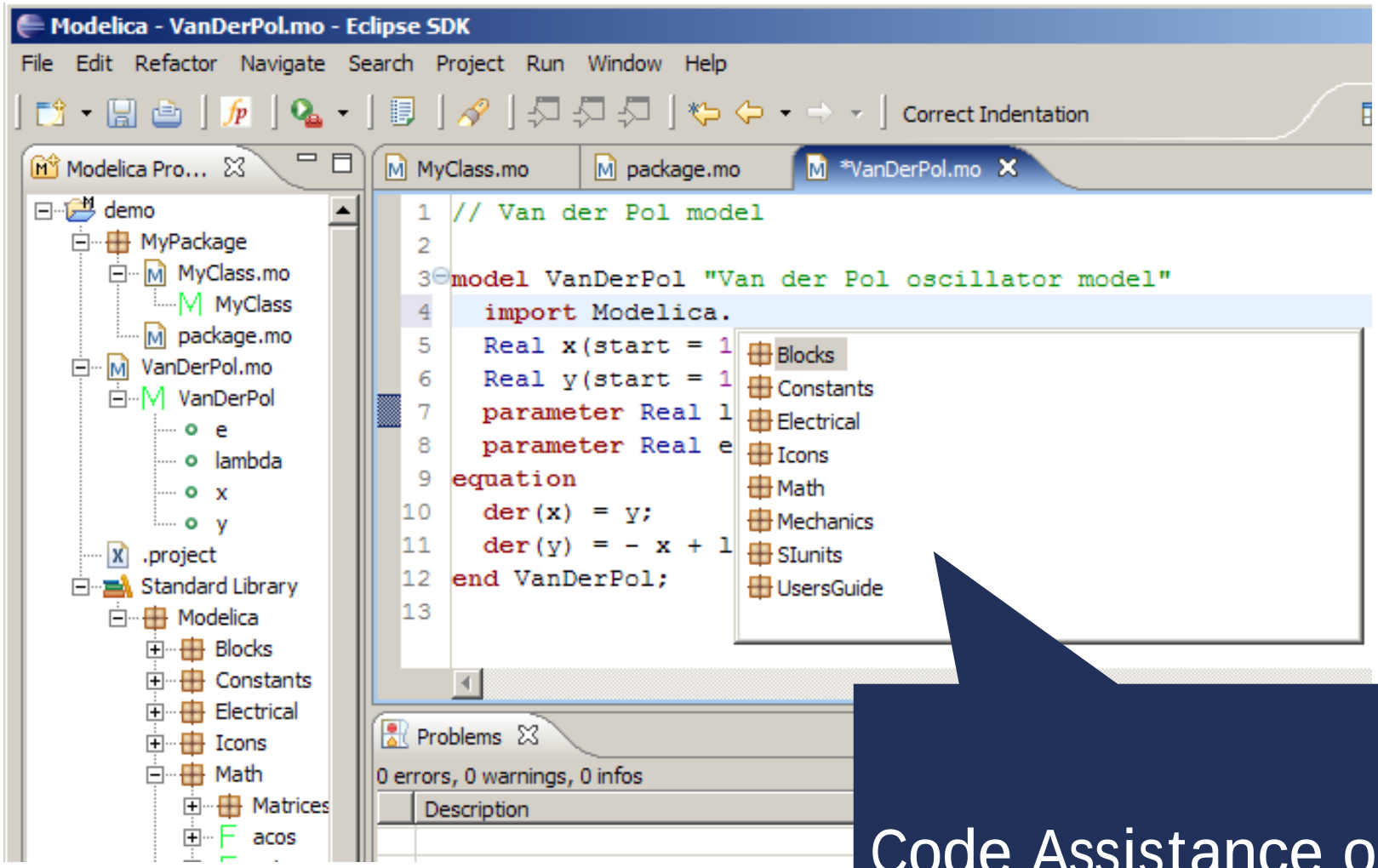


Parse error detection on file save

Error detection (II)

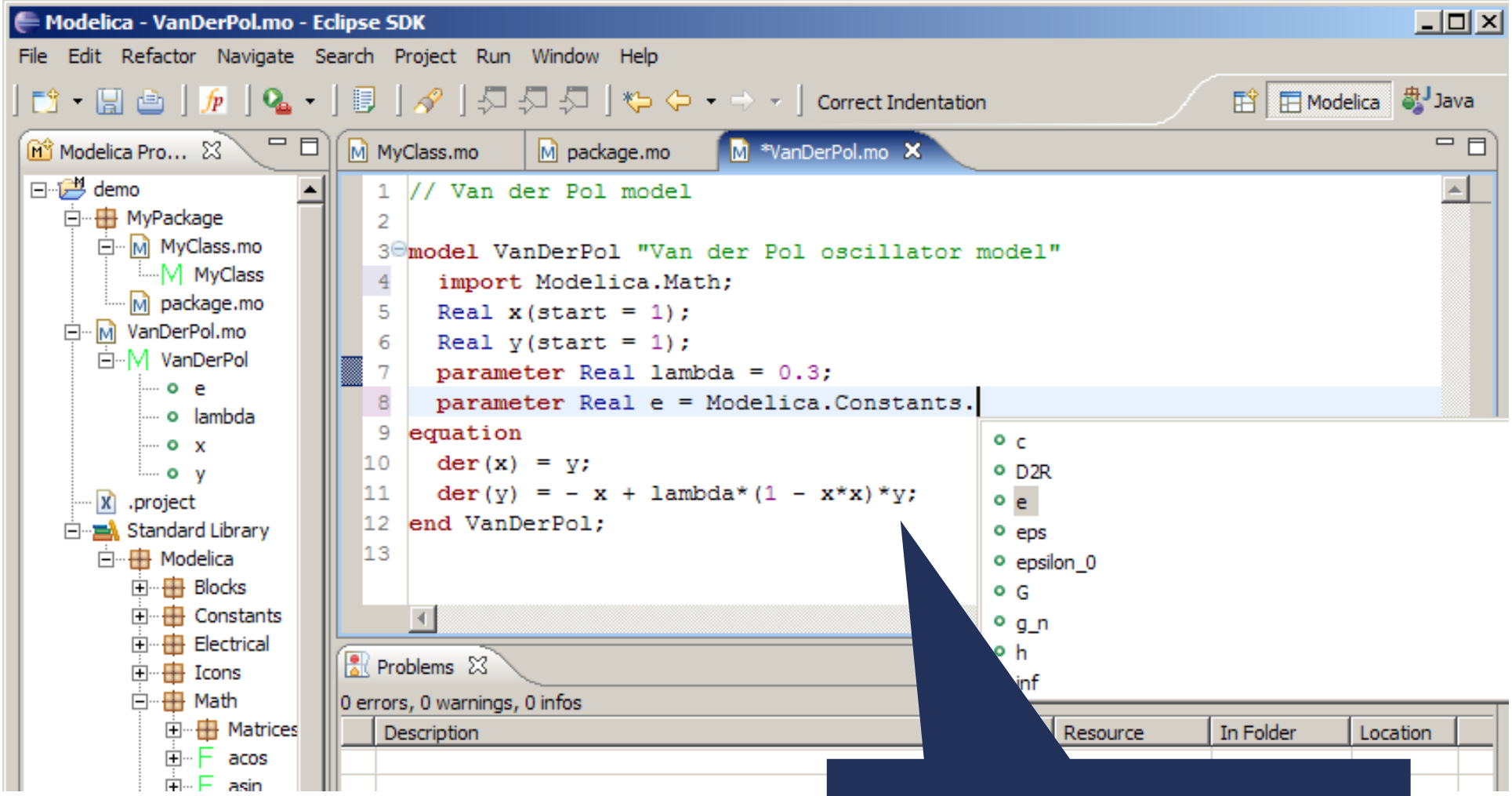


Code assistance (I)



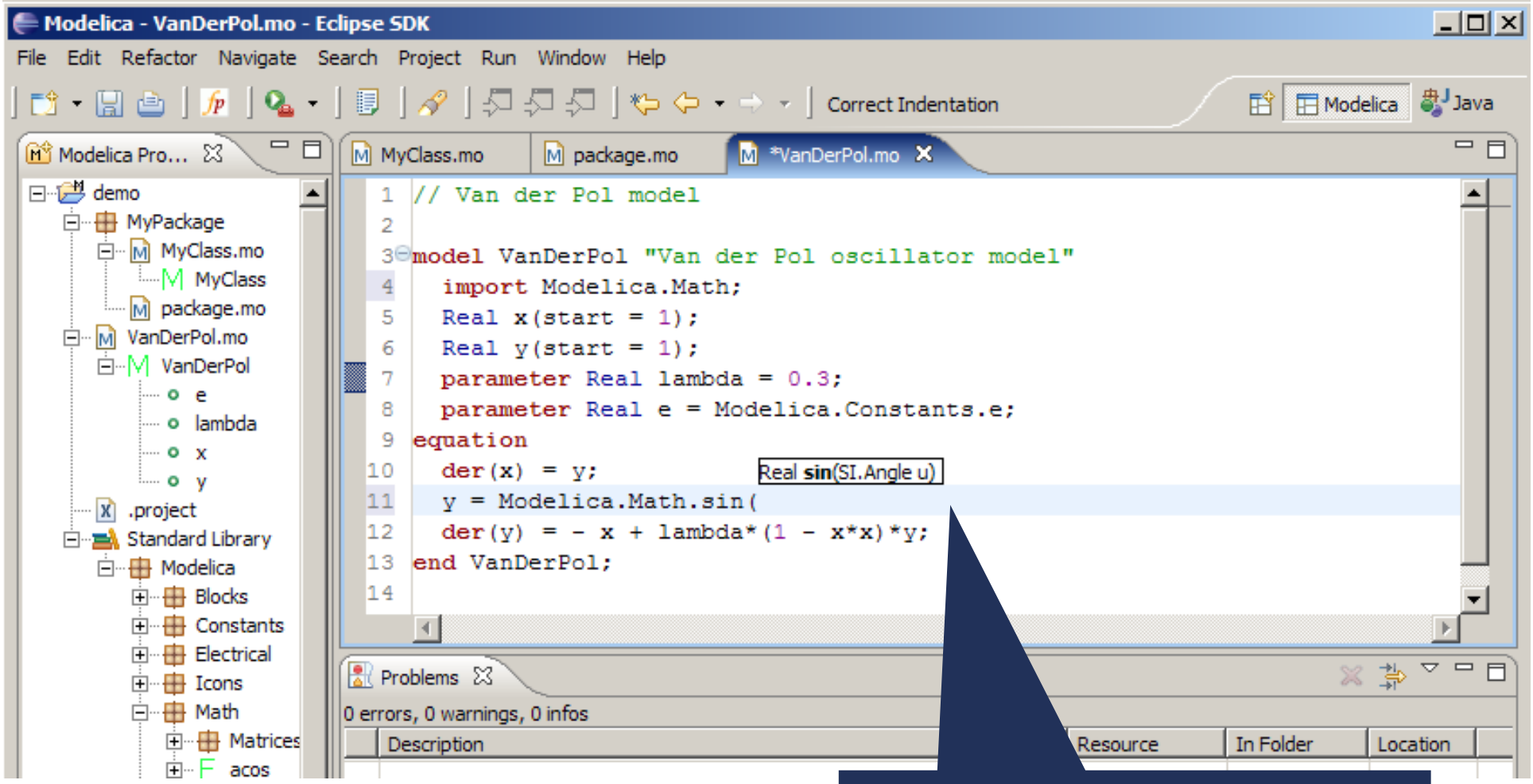
Code Assistance on imports

Code assistance (II)



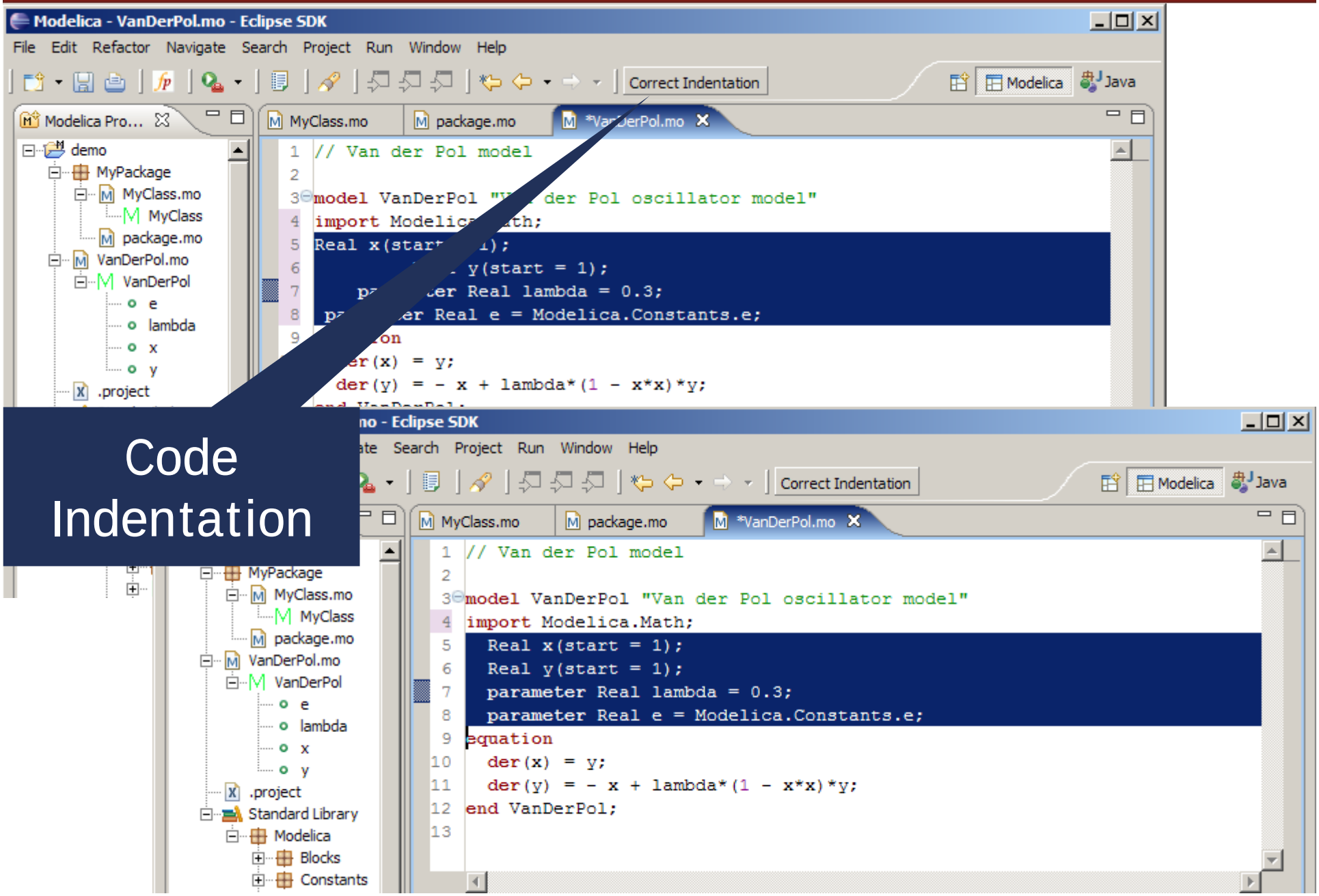
Code Assistance on assignments

Code assistance (III)



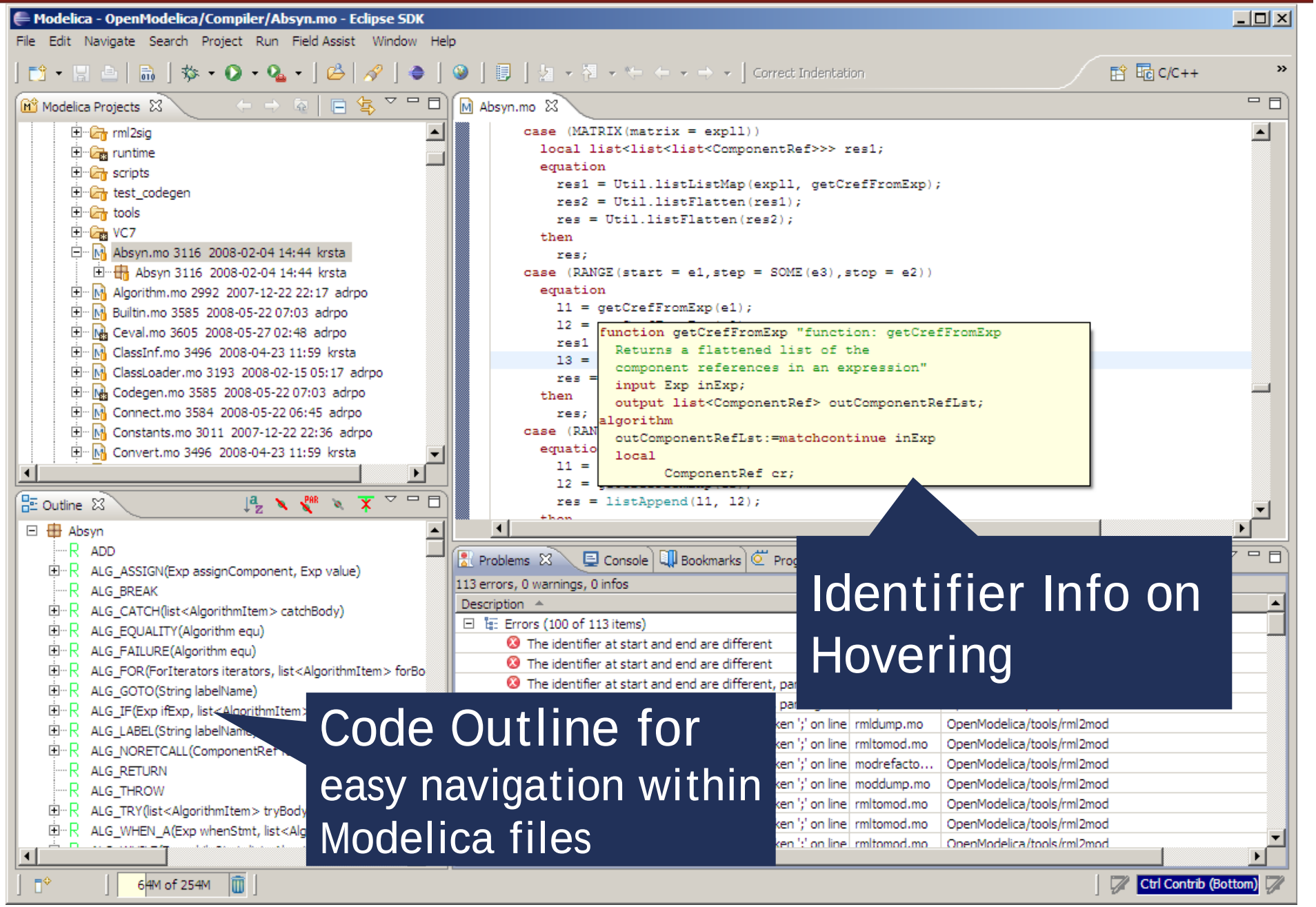
Code Assistance on function calls

Code indentation

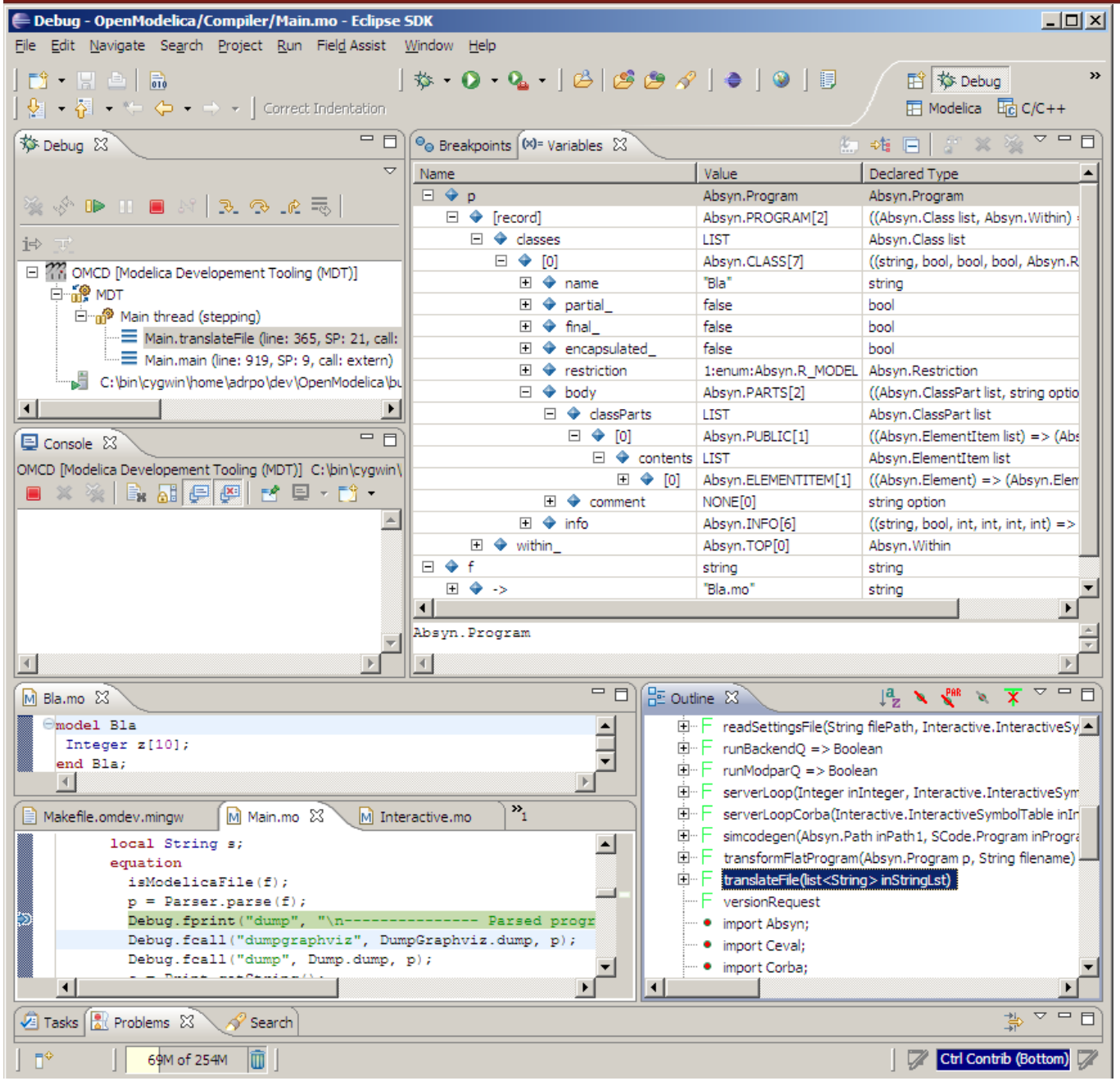


Code Indentation

Code Outline and Hovering Info

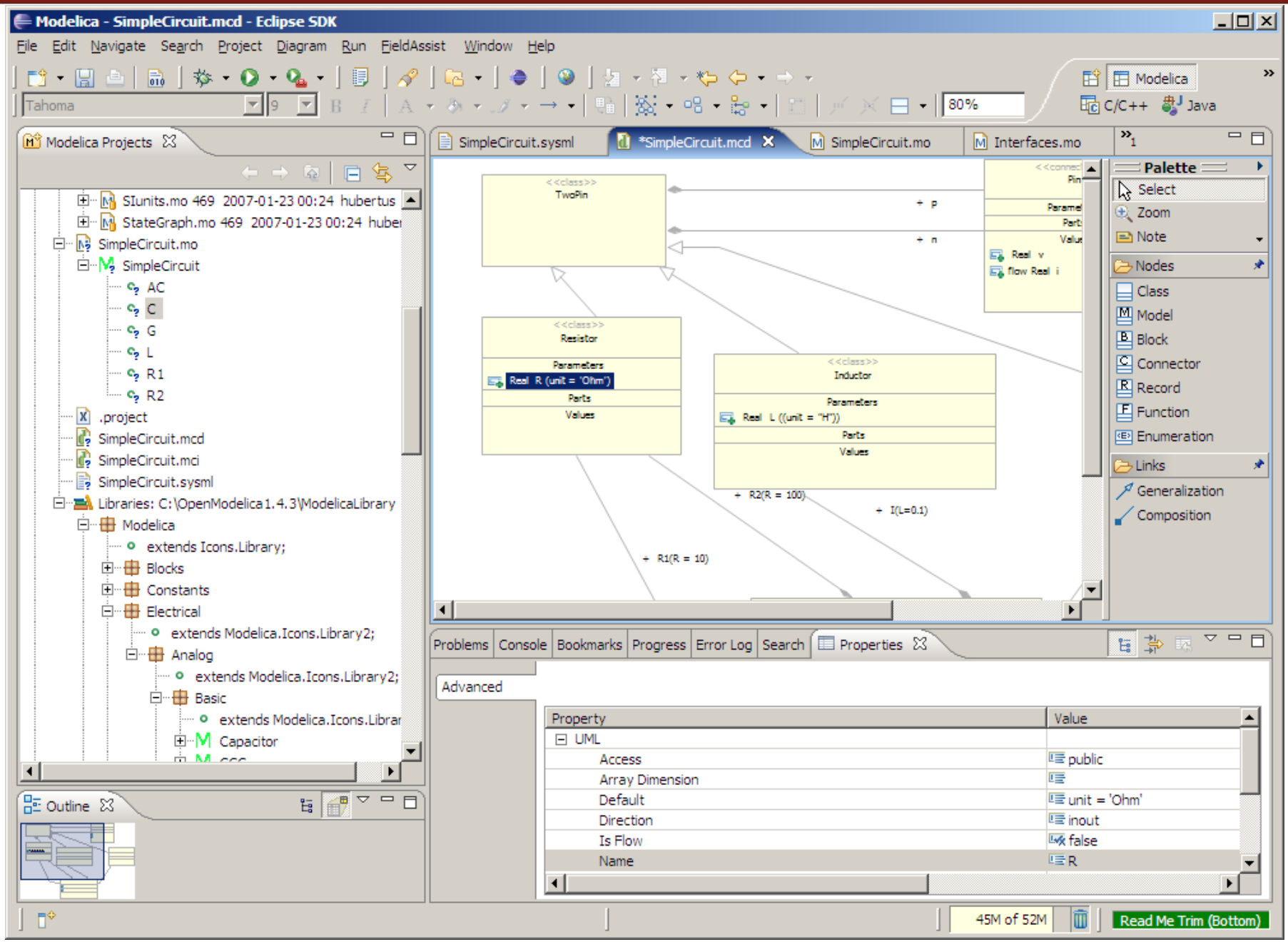


Eclipse Debugging Environment

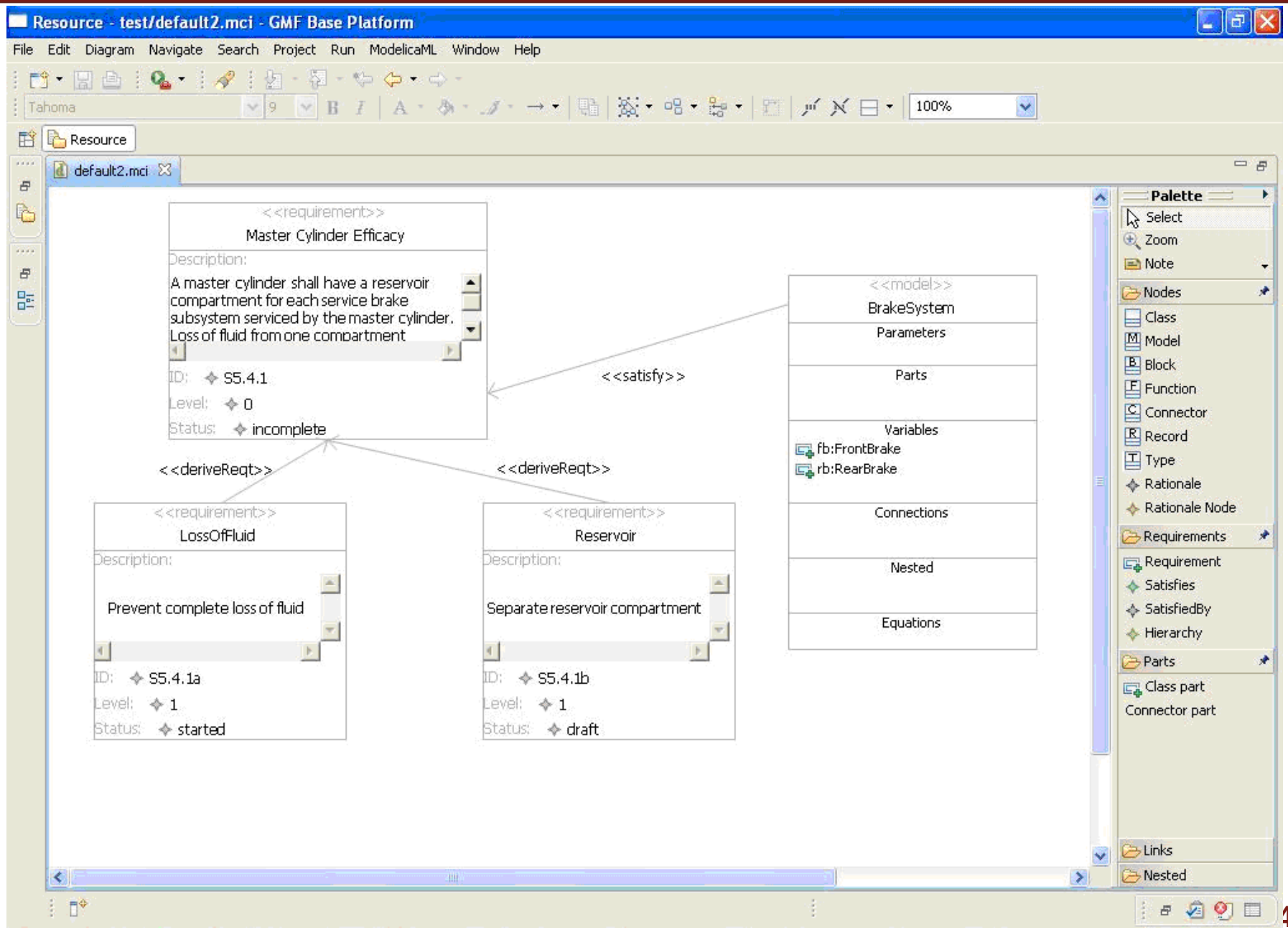


- Type information for all variables
- Browsing of complex data structures

Eclipse environment for ModelicaML



Requirements Modeling in Eclipse



Outline

- 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

MultiBody library

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 ovelapping 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

Media library

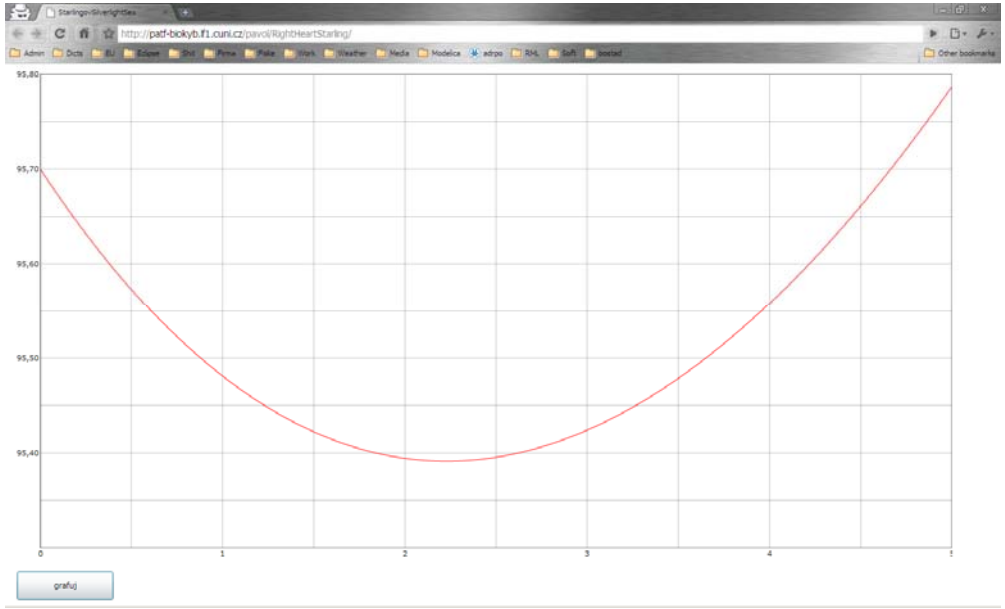
- The most evil Modelica standard library

- 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 redelclare replaceable function extends used to set constants in partial packages.

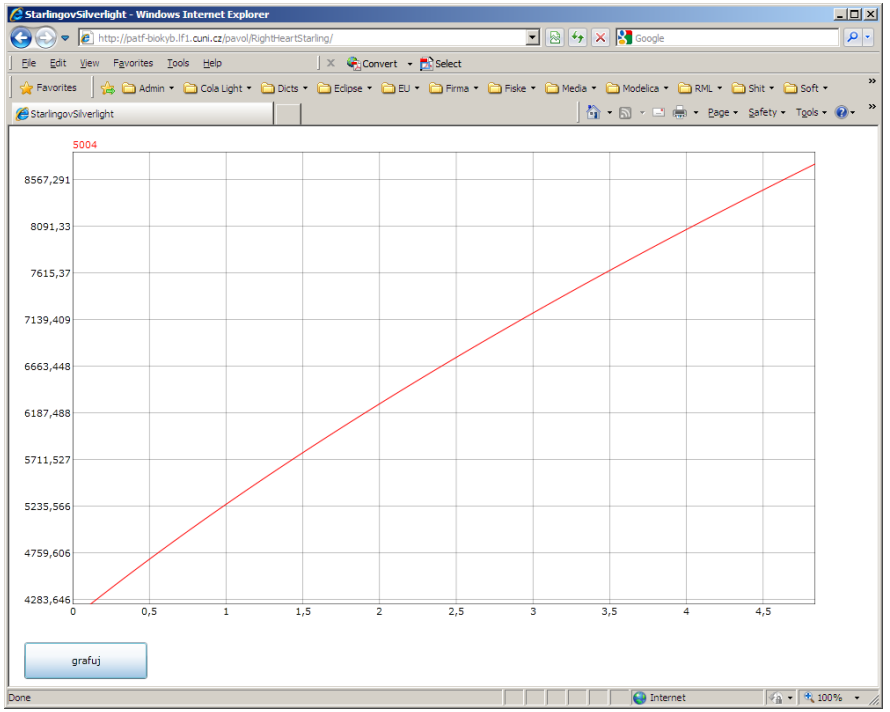
Template Code Generation

- 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!**

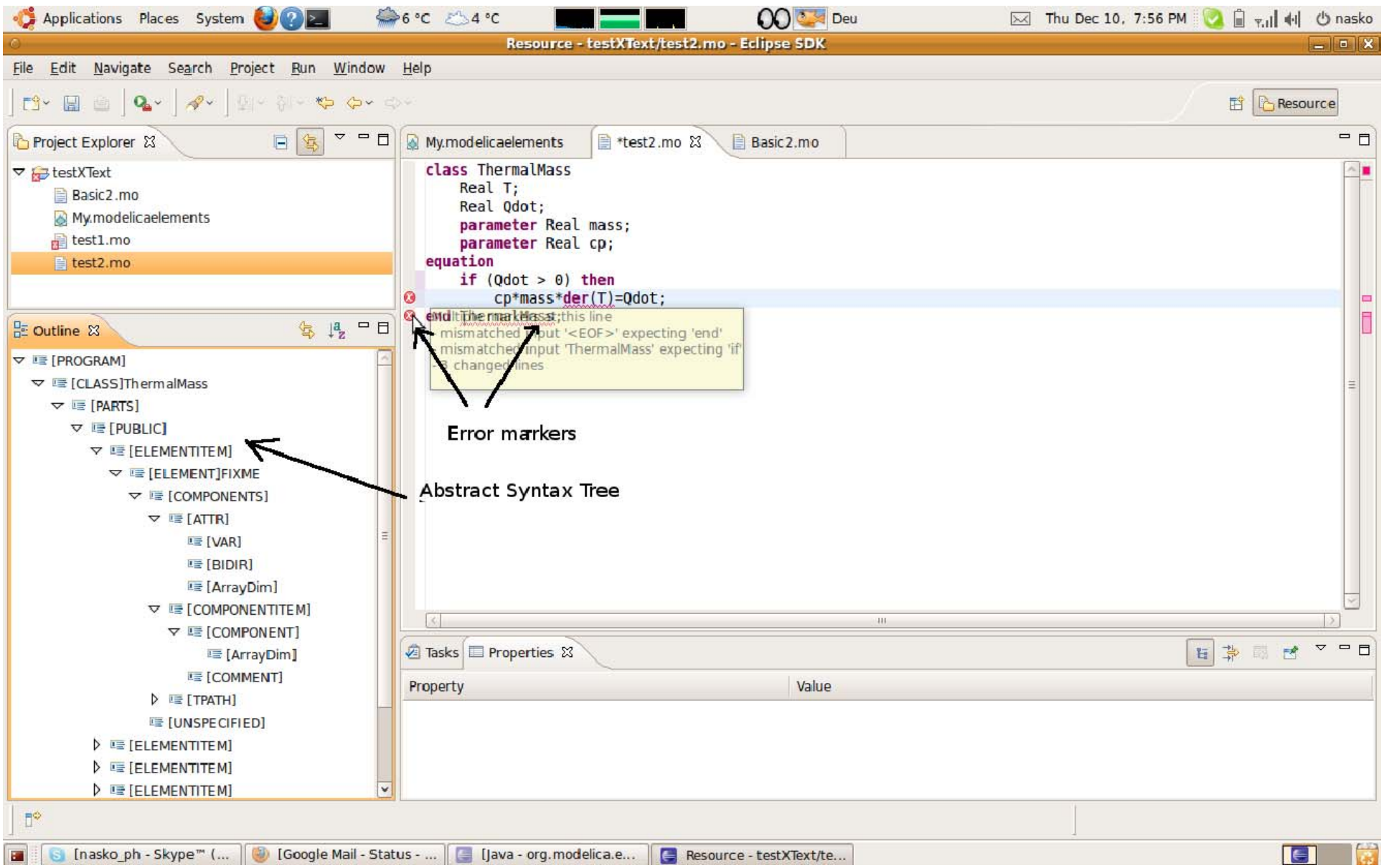


Input Pressure to Right Heart vs. Volume



Input Pressure to Right Heart vs. BlodFlow

Modelica Development Tooling based on Xtext



Thank You!
Questions?

OpenModelica Project
http://www.OpenModelica.org