

# OpenModelica

## OMShell and OMNotebook

### Introduction and Exercises

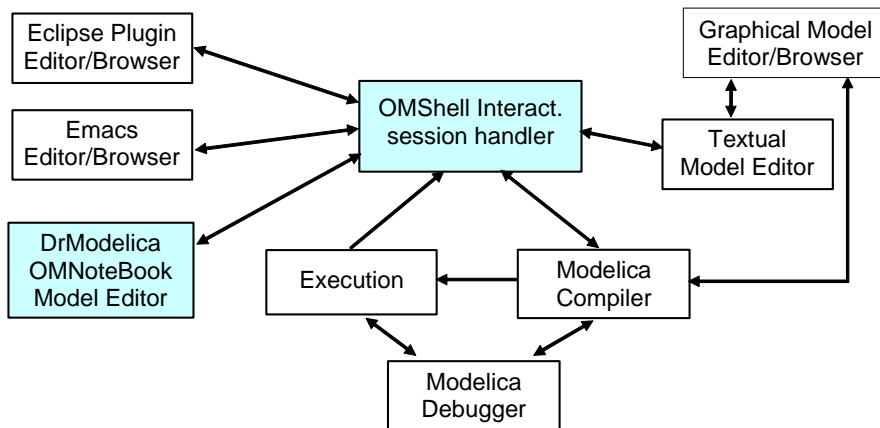
Peter Fritzson, Adrian Pop, Peter Aronsson

OpenModelica Course at INRIA, 2006 06 08

## OMShell and OMNotebook OpenModelica End-User Subsystems

- OMShell – interactive session handler for Modelica scripting
- OMNotebook – interactive electronic notebook for Modelica teaching (with DrModelica), scripting, and documentation
- OpenModelica Compiler (OMC) – compiles and executes/simulates Modelica models
- ptplot package (from Berkeley) make plots

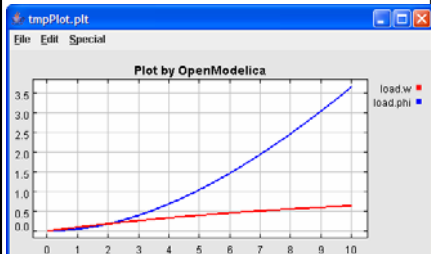
## OpenModelica Environment Architecture



## Interactive Session Handler – on dcmotor Example (Session handler called OMShell – OpenModelica Shell)

```
>>simulate(dcmotor,startTime=0.0,stopTime=10.0)
>>plot({load.w,load.phi})
```

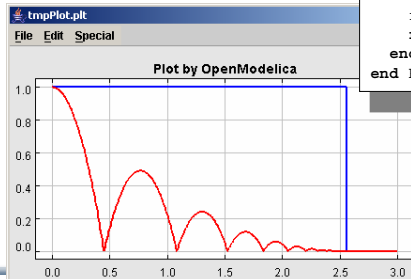
```
model dcmotor
  Modelica.Electrical.Analog.Basic.Resistor r1(R=10);
  Modelica.Electrical.Analog.Basic.Inductor il;
  Modelica.Electrical.Analog.Basic.EMF emf1;
  Modelica.Mechanics.Rotational.Inertia load;
  Modelica.Electrical.Analog.Basic.Ground g;
  Modelica.Electrical.Analog.Sources.ConstantVoltage v;
equation
  connect(v.p,r1.p);
  connect(v.n,g.p);
  connect(r1.n,il.p);
  connect(il.n,emf1.p);
  connect(emf1.n,g.p);
  connect(emf1.flange_b,load.flange_a);
end dcmotor;
```



## Event Handling by OpenModelica – BouncingBall

```
>>simulate(BouncingBall,  
           stopTime=3.0);  
>>plot({h, flying});
```

```
model BouncingBall  
  parameter Real e=0.7 "coefficient of restitution";  
  parameter Real g=9.81 "gravity acceleration";  
  Real h(start=1) "height of ball";  
  Real v "velocity of ball";  
  Boolean flying(start=true) "true, if ball is flying";  
  Boolean impact;  
  Real v_new;  
equation  
  impact=h <= 0.0;  
  der(v)=if flying then -g else 0;  
  der(h)=v;  
  when {h <= 0.0 and v <= 0.0, impact} then  
    v_new=if edge(impact) then -e*pre(v) else 0;  
    flying=v_new > 0;  
    reinit(v, v_new);  
  end when;  
end BouncingBall;
```

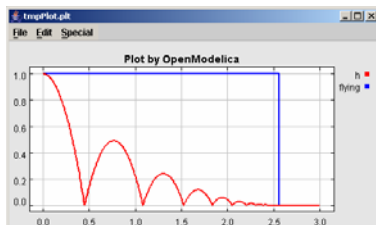


5 Peter Fritzon

## Run Scripts in OpenModelica

- RunScript command interprets a .mos file
- .mos means MModelica Script file
- Example:

```
>> runScript("sim_BouncingBall.mos")
```



The file `sim_BouncingBall.mos` :

```
loadFile("BouncingBall.mo");  
simulate(BouncingBall, stopTime=3.0);  
plot({h, flying});
```

6 Peter Fritzon

## Some OMShell Exercises

- Start OMShell (e.g. from the start menu)
- Look at the examples in the OpenModelicaUsersGuide.
- Try evaluating a small expression, e.g.  $2+3$ , push return
- type help()
- try command completion, e.g. type loadM followed by tab
- fill in Modelica (i.e. loadModel(Modelica))
- Use the file menu->LoadModel and load the dcmotor (./testmodels)
- type sim followed by tab; fill in dcmotor
- type plot followed by tab; fill in some variable(s) to plot (look at the users guide)
- type in a small function and call it

## OpenModelica OMNotebook Electronic Notebook with DrModelica

- Primarily for teaching
- Interactive electronic book
- Platform independent
- OMNotebook Does not need Mathematica

OMNotebook: DrModelica.omnb  
Version 2006-04-11

### DrModelica Modelica Edition

Copyright (c) Linköping University, PELAB, 2003-2006, Wiley-IEEE Press, Modelica Association.  
Contact: OpenModelica@sida.ln.se, OpenModelica Project web site:  
www.sida.ln.se/projects/OpenModelica  
Book web page: www.mathcore.com/drModelica, Book author: Peter Fritzon@sida.ln.se

DrModelica Authors (2003 version) Susanna Mozum, Eva-Lena Ljungqvist Swedelin, Peter Fritzon, Peter Buzza  
DrModelica Authors (2003 and later updates) Peter Fritzon

*This DrModelica notebook has been developed to facilitate learning the Modelica language as well as providing an introduction to object-oriented modeling and simulation. It is based on and is supplementary material to the Modelica book: Peter Fritzon: "Principles of Object-Oriented Modeling and Simulation with Modelica" (2004), 940 pages, Wiley-IEEE Press, ISBN 0-471-47163-1. All of the examples and exercises in DrModelica and the page references are from that book. Most of the text in DrModelica is also based on that book.*

#### Detailed Copyright and Acknowledgment Information

#### Getting Started Using OMNotebook

#### OpenModelica commands

#### Berkeley license OpenModelica

### 1 A Quick Tour of Modelica

#### 1.1 Getting Started - First Basic Examples

There is a long tradition that the first sample program in any computer language is a trivial program printing the string "Hello, World!" (p. 19 in Peter Fritzon's book). Since Modelica is an equation based language, printing a string does not make much sense. Instead, our Hello World Modelica program solves a trivial differential equation. The second example shows how you can write a model that solves a [Differential Algebraic Equation System](#) (p. 19). In the [Van der Pol](#) (p. 22) example declaration as well as initialization and prefix usage are shown in a slightly more complicated way.

#### 1.2 Classes and Instances

In Modelica objects are created implicitly just by [Declaring Instances of Classes](#) (p. 26). Almost anything in Modelica is a class, but there are some keywords for specific use of the class concept, called

## Interactive Contents in DrModelica Contains Examples and Exercises from Modelica Book

OPNotebook: DrModelica(rev).nh

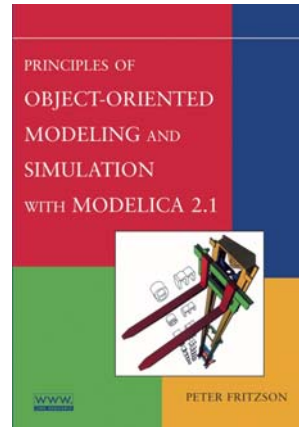
This notebook is developed to facilitate the learning of the Modelica language. It is a supplementary material to Peter Fritzon's book "Principles of Object-Oriented Modeling and Simulation with Modelica" (Wiley-IEEE Press, 2003) so the page references below are from this book.

**Getting Started**

- A Quick Tour of Modelica
- Classes, Types and Declarations
- Inheritance, Modifications and Generics
- Components, Connectors and Connections
- Literals, Operators and Expressions
- Arrays
- Equations
- Algorithms and Functions
- Packages
- Annotations, Units and Quantities
- System Modeling Methodology and Continuous Model Representation
- Modeling Discrete Events and Hybrid Systems
- Basic Laws of Nature
- Application Examples

Ready

Recent Book, 2004:



## Cells with both Text and Graphics

- Java must be installed for plotting to work

OPNotebook: HelloWorld.nh

### First Basic Class

**1 HelloWorld**

The program contains a declaration of a class called HelloWorld with two fields and one equation. The first field is the variable  $x$  which is initialized to a start value 2 at the time when the simulation starts. The second field is the variable  $a$ , which is a constant that is simulated to 2 at the beginning of the simulation. Such a constant is prefixed by the keyword parameter in order to indicate that it is constant during simulation but is a model parameter that can be changed between simulations.

The Modelica program solves a typical differential equation:  $x' = -a * x$ . The variable  $x$  is a state variable that can change value over time. The  $x'$  is the time derivative of  $x$ .

```
class HelloWorld
  Real x(start = 2);
  parameter Real a = 2;
equation
  der(x) = - a * x;
end HelloWorld;
```

Ok

**2 Simulation of HelloWorld**

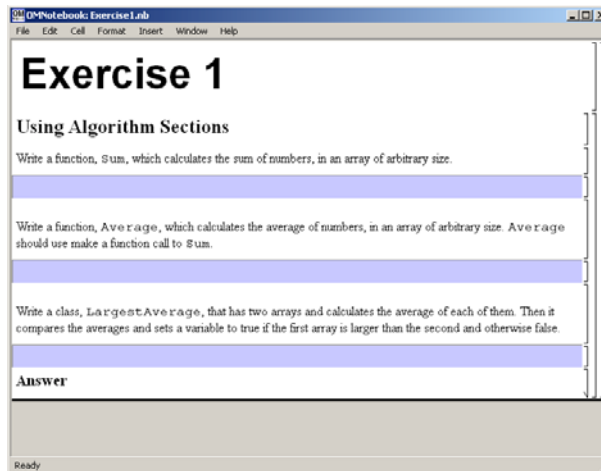
```
simulate( HelloWorld, startime=0, stopime=4 );
```

[done]

plot( x );

Ready

## Exercises and Answers in OMNotebook DrModelica



## Some OMNotebook Commands (see also OpenModelica Users Guide)

- Shift-return (evaluated a cell)
- File Menu (open, close, etc.)
- Text Cursor (vertical), Cell cursor (horizontal)
- Cell types: text cells & executable code cells
- Copy, paste, group cells
- Copy, paste, group text
- Command Completion (shift-tab)

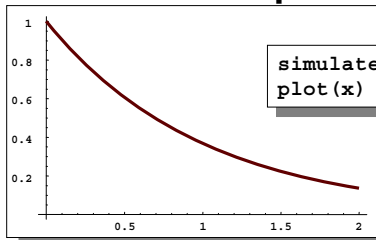
## Simplest Model – Hello World!

### A Modelica “Hello World” model

Equation:  $x' = -x$   
Initial condition:  $x(0) = 1$

```
class HelloWorld "A simple equation"  
  Real x(start=1);  
equation  
  der(x) = -x;  
end HelloWorld;
```

### Simulation in OpenModelica environment

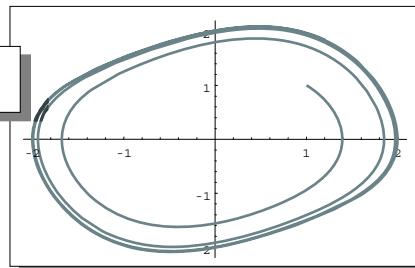


```
simulate(HelloWorld, stopTime = 2)  
plot(x)
```

## Example class: Van der Pol Oscillator

```
class VanDerPol "Van der Pol oscillator model"  
  Real x(start = 1) "Descriptive string for x"; // x starts at 1  
  Real y(start = 1) "y coordinate"; // y starts at 1  
  parameter Real lambda = 0.3;  
equation  
  der(x) = y; // This is the 1st diff equation //  
  der(y) = -x + lambda*(1 - x*x)*y; /* This is the 2nd diff equation */  
end VanDerPol;
```

```
simulate(VanDerPol, stopTime = 25)  
plotParametric(x, y)
```



## Small OMNotebook Exercise

- Locate the HelloWorld model in DrModelica using OMNotebook!
- Simulate and plot the example. Do a slight change in the model, re-simulate and re-plot.

```
class HelloWorld "A simple equation"  
  Real x(start=1);  
equation  
  der(x) = -x;  
end HelloWorld;
```

```
simulate(HelloWorld, stopTime = 2)  
plot(x)
```

- Locate the VanDerPol model in DrModelica and try it!
- Do some change and re-simulate

## Small OMNotebook Exercises cont.

- Select and copy a cell (tree to the right)
- Position the horizontal cell cursor: first click between cells; then click once more on top of the horizontal line
- Paste the cell
  
- Note: You can find most Users Guide examples in the UsersGuideExamples.onb in the testmodels directory