

Modelica

Rychloúvod do jazyka

jezekfi1@fel.cvut.cz

Použity části materiálů Petera Fritzsona

Model a simulace

- Jaký je rozdíl mezi modelem, simulantem a simulátorem?
- Model – abstrakce reálného objektu či systému
- Simulant je mdlého rozumu
- Simulátor napodobuje chování

Proč simulovat?

- minule proč, dnes jak

Historie

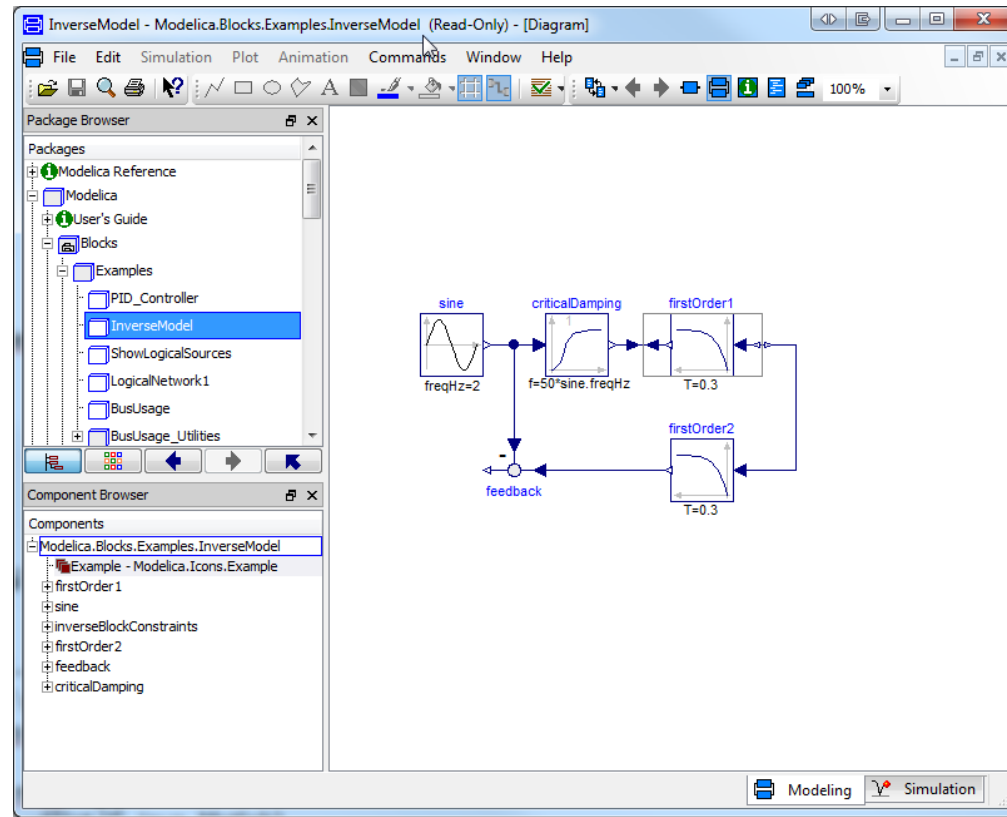
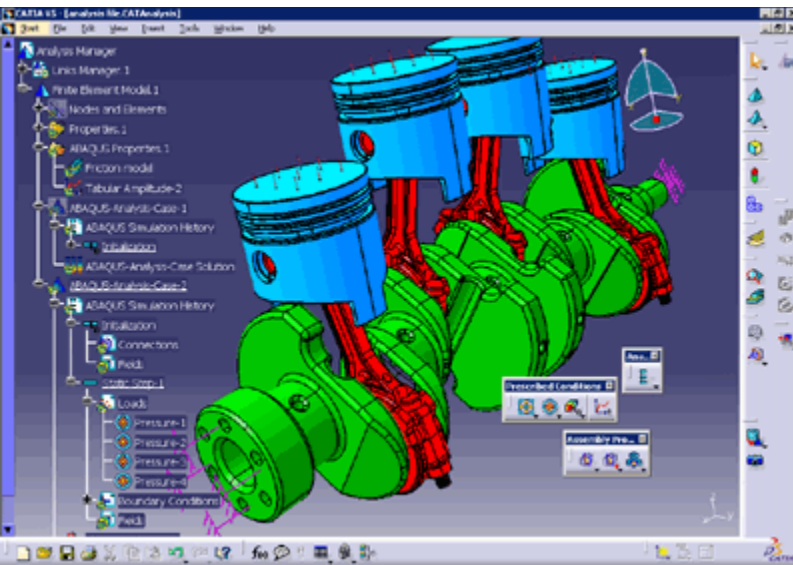
- University of Linköping, Švédsko
- od 1996, první aplikace 2000
- modelica standard library (v3.2)
 - Přes 2000 modelů a funkcí
- Modelica Association
 - DLR, Modelon, Dassault, Maplesoft, MathCore ...

Nástroje Modelica



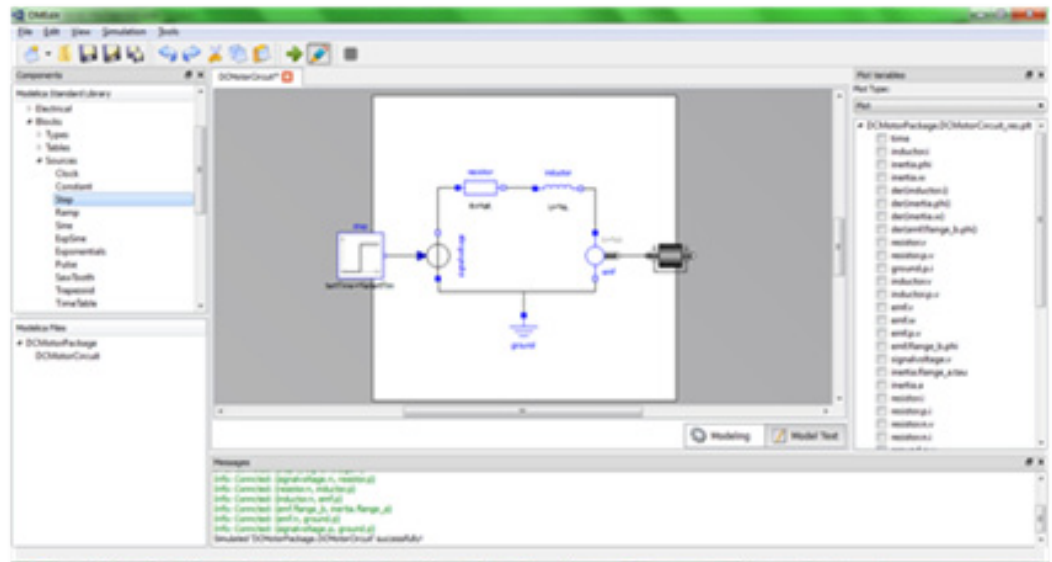
Dymola

- 3ds, nyní Dassault
- Integrace do 3D CAD/CAM/CAE prostředí Catia
- “industry standard”



Openmodelica

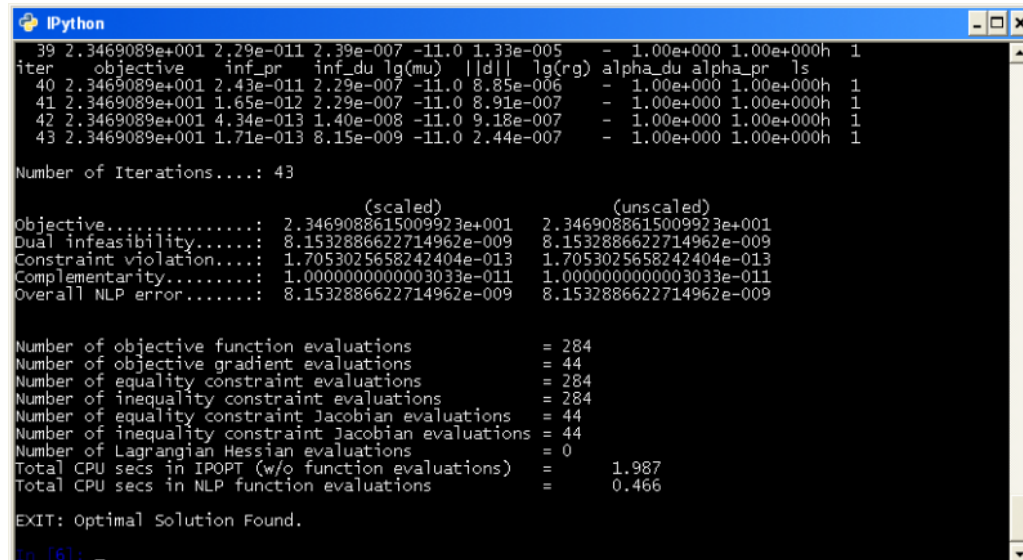
- Opensource platforma
- Openmodelica.org
- OMEdit, SimForge, OMNotebook, OMOptim
- Rychle se vyvíjí
- Pomožme s tím!



MathModelica (mathcore, nyní Wolfram)

Jmodelica

- Open-source
- Modelon AB
- Optimica
- Více k optimalizaci modelů



```
IPython
39 2.3469089e+001 2.29e-011 2.39e-007 -11.0 1.33e-005 - 1.00e+000 1.00e+000h 1
iter objective inf_pr inf_du lg(mu) ||d|| lg(rg) alpha_du alpha_pr ls
40 2.3469089e+001 2.43e-011 2.29e-007 -11.0 8.85e-006 - 1.00e+000 1.00e+000h 1
41 2.3469089e+001 1.65e-012 2.29e-007 -11.0 8.91e-007 - 1.00e+000 1.00e+000h 1
42 2.3469089e+001 4.34e-013 1.40e-008 -11.0 9.18e-007 - 1.00e+000 1.00e+000h 1
43 2.3469089e+001 1.71e-013 8.15e-009 -11.0 2.44e-007 - 1.00e+000 1.00e+000h 1

Number of Iterations....: 43

                                (scaled)                                (unscaled)
Objective.....: 2.3469088615009923e+001 2.3469088615009923e+001
Dual infeasibility.....: 8.1532886622714962e-009 8.1532886622714962e-009
Constraint violation....: 1.7053025658242404e-013 1.7053025658242404e-013
Complementarity.....: 1.0000000000003033e-011 1.0000000000003033e-011
Overall NLP error.....: 8.1532886622714962e-009 8.1532886622714962e-009

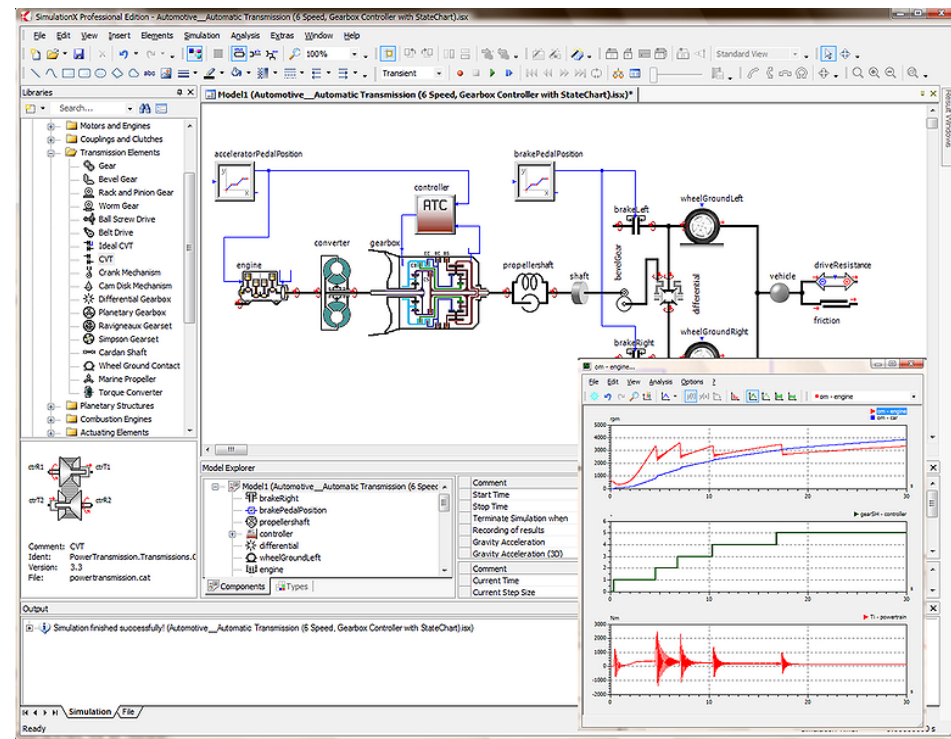
Number of objective function evaluations      = 284
Number of objective gradient evaluations     = 44
Number of equality constraint evaluations     = 284
Number of inequality constraint evaluations   = 284
Number of equality constraint Jacobian evaluations = 44
Number of inequality constraint Jacobian evaluations = 44
Number of Lagrangian Hessian evaluations    = 0
Total CPU secs in IPOPT (w/o function evaluations) = 1.987
Total CPU secs in NLP function evaluations   = 0.466

EXIT: Optimal Solution Found.

In [6]:
```

SimulationX

- ITI (DE)
- Pohony, hydraulika, termodynamika
- Podporuje i jazyk Modelica

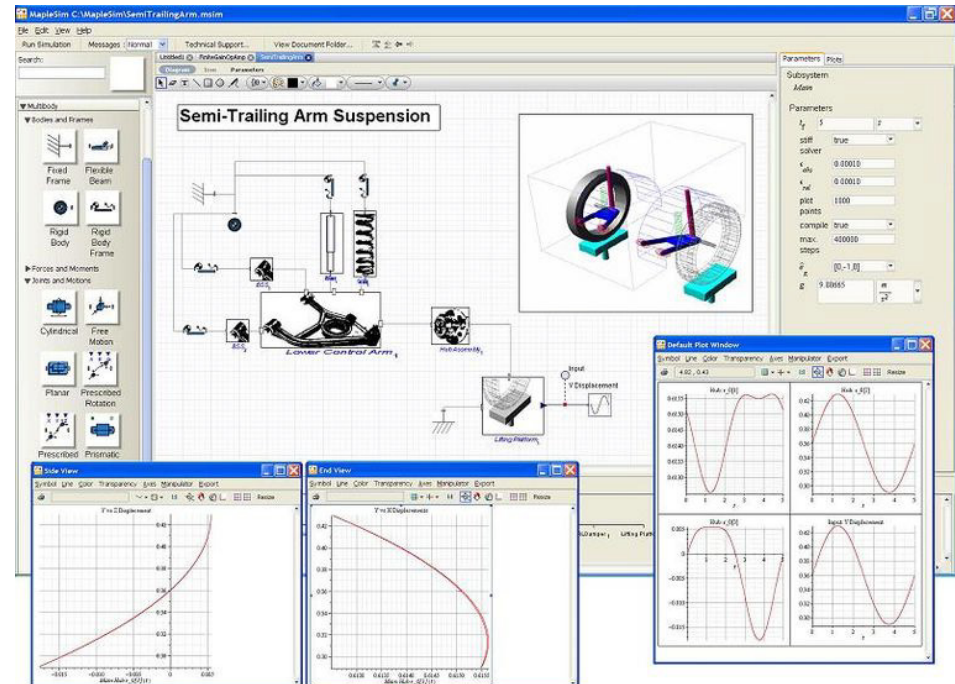


Maplesim

- Optimalizace rovnic pomocí Maple
- Rozsáhlý simulační framework
- Podporuje i komponenty z Modelicy

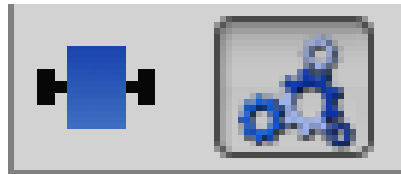
„Modelica definitely matters and you'll be hearing a lot more from us on this topic.“

– Vice-President of Research and Development, MapleSim



Modelica a prostředí

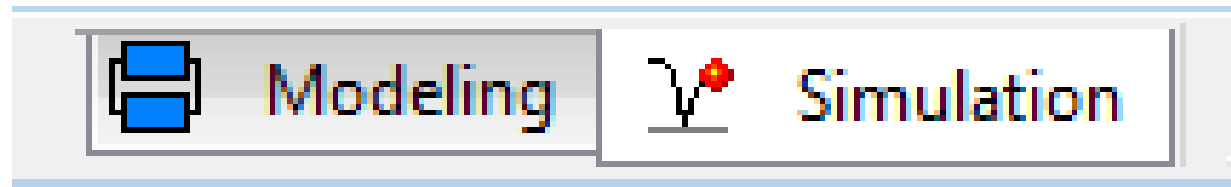
- Grafické zobrazení - modelování - prostředí – proprietary



- textové zobrazení - modelica - // non-proprietary //



- simulace - prostředí – proprietary



Jaký jazyk je Modelica?

- Deklarativní
- Multidoménový
- Objektový – vše je třída
- Vizuální
- Efektivní
- Neproprietární

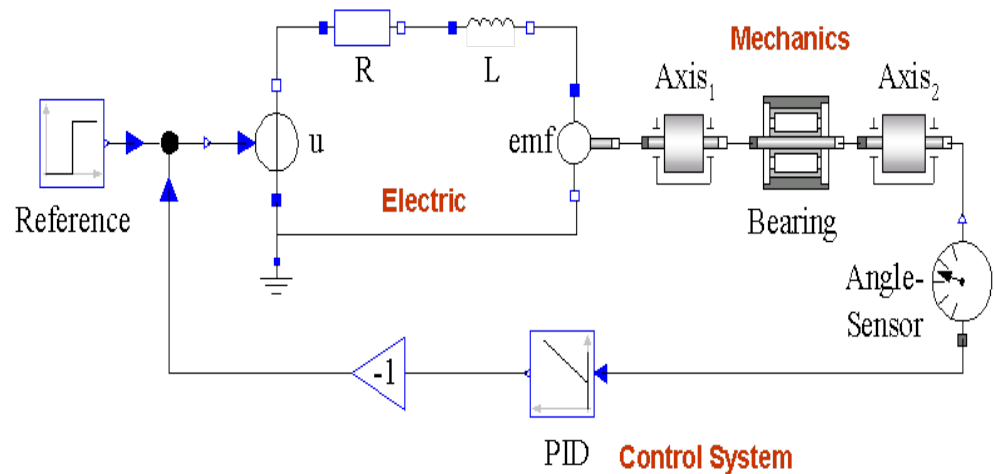
Deklarativní

- co se má udělat (např SQL)
- Vs. Imperativní – jak se to má udělat (C, java, Matlab)
- Equation based
- Akauzální

```
class VanDerPol "Van der Pol oscillator model"  
  Real x(start = 1) "Descriptive string for x";  
  Real y(start = 1) "y coordinate";  
  parameter Real lambda = 0.3;  
equation  
  der(x) = y;  
  der(y) = -x + lambda*(1 - x*x)*y;  
end VanDerPol;
```

Multidoménový

- Domény
 - Elektrické
 - Mechanické
 - Termodynamické
 - Hydraulické
 - Biologické
 - .. definovatelné



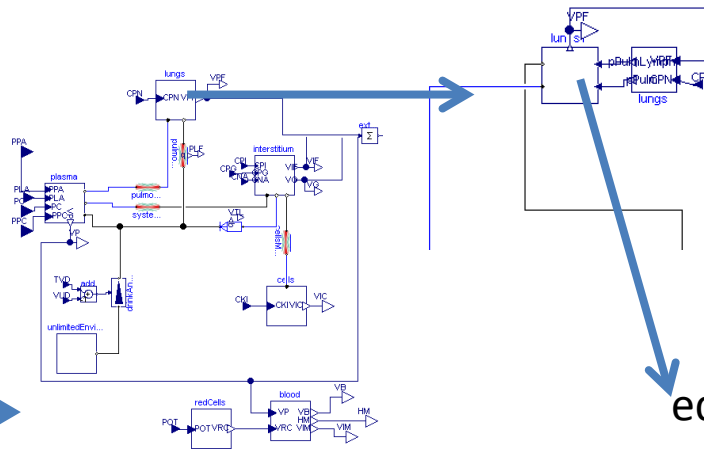
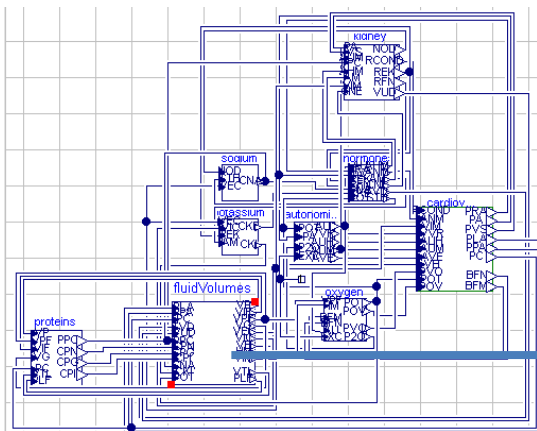
Objektový

- vše je třída
- Strong-typed (nemíchají se typy dohromady)
- Dědičnost
- Objektový jazyk co do hierarchického návrhu

```
class VanDerPol "Van der Pol oscillator model"  
  Real x(start = 1) "Descriptive string for x";  
  Real y(start = 1) "y coordinate";  
  parameter Real lambda = 0.3;  
equation  
  der(x) = y;  
  der(y) = -x + lambda*(1 - x*x)*y;  
end VanDerPol;
```


Vizuální návrh systému

- Hierarchická struktura
- „dole“ rovnice



equation

$$\text{der}(\text{volume}) = q1.q + q2.q;$$

$$q1.\text{pressure} = \text{pressure1};$$

$$q2.\text{pressure} = \text{pressure2};$$

Efektivní

- Efektivita simulace srovnatelná s C
- // při kompilaci překlad do C

Neproprietární

- „ne-soukromý“ ~ veřejný
- Modelica Association
- Společný vývoj dalších verzí

Organization

[Austrian Institute of Technology, Austria](#)

[DLR - Institute of Robotics and Mechatronics, Germany](#)

[Dassault Systèmes AB, Sweden](#)

[Dassault Systèmes SA, France](#)

[LMS Imagine S.A.](#)

[Equa Simulation Technology, Sweden](#)

[IDA PELAB, Sweden](#)

[Institut für Thermodynamik, Universität Braunschweig, Germany](#)

[Institute of Thermo-Fluid Dynamics, Hamburg University of Technology, Germany](#)

[Maplesoft, Canada](#)

[MathCore Engineering AB, Sweden](#)

[Modelon AB, Sweden](#)

[XRG Simulation GmbH, Germany](#)

Co to znamená Equation based modelling

- Vyjádření systému jen pomocí rovnic
- Triviální – $a=3$
- Jednoduché – $a = 2b + 3$
- ODE – obyčejné diferenciální rovnice
 - Jen 1. řádu
- Stejný počet neznámých a rovnic (na všech úrovních!)
- Systém musí mít řešení (singularity!)

- V současné době žádné PDE
 - Čili žádné FEM (finite element method)
 - ani CFD (computational fluid dynamics),
 - Je ale možné využít výsledků

Typy modelů

- modely statické vs dynamické
- spojité vs. Diskrétní
- kvalitativní vs kvantitativní

Statické modely

- Soustava rovnic
- Výsledek nezávisí na čase
- Systém musí mít řešení
- Stejný počet neznámých a rovnic

Model Menza

parameter Integer početVydanychJidel = 30;

parameter Real pravdepodobnostSalmonelózy = 0.5

Real početNakažených Jidel:

equation

početNakažených Jidel = početVydanychJidel
*pravdepodobnostSalmonelózy ;

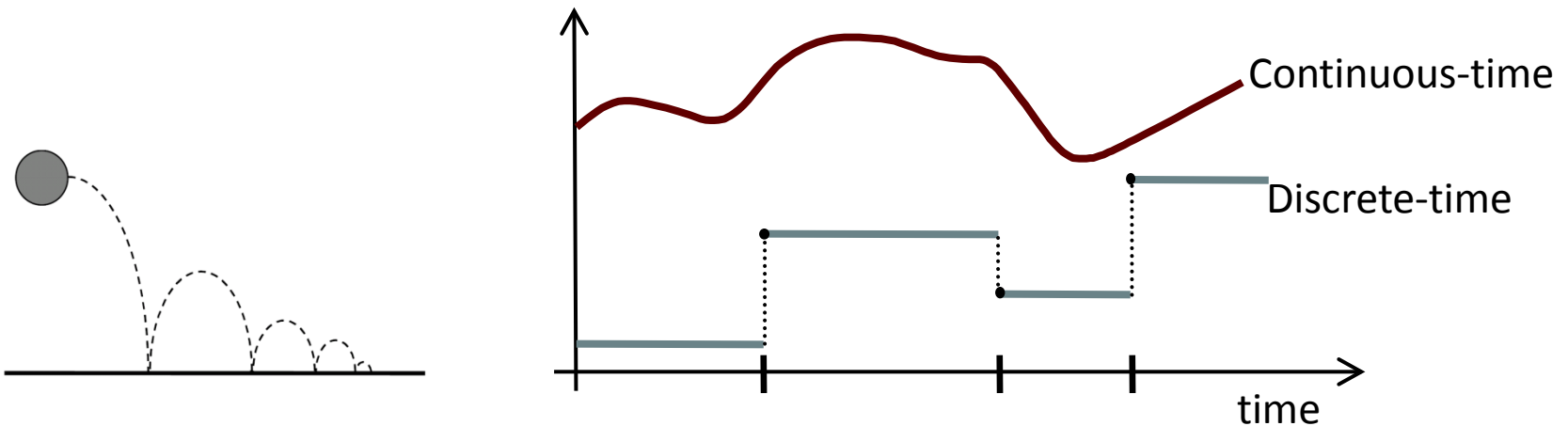
end

Dynamické spojité modely

- změna je derivace
- Soustavy ODE
- Řetězení derivací
 - NE:
 $a = \text{der}(\text{der}(x))$
 - ANO:
 $a = \text{der}(v);$
 $v = \text{der}(x);$
 - Model padající Klavír
 $x = m * g;$
 $g = \text{der}(v);$
 $v = \text{der}(x)$

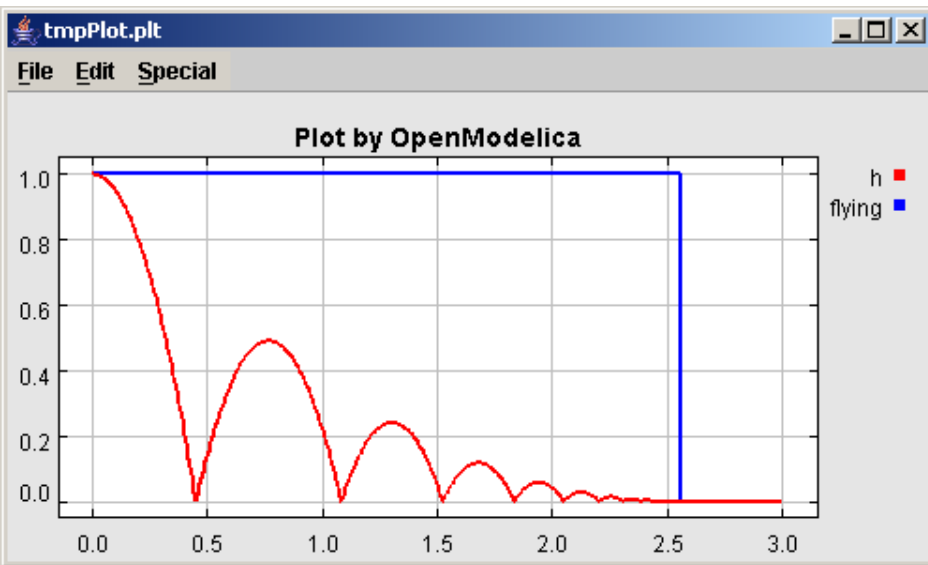
Diskrétní modely

- Měníme hodnoty skokově
- Stavový automat
- Podmínky If, when...



Hybridní modely

- Spojité + nespojitosti
- Např. dopad míčku na zem

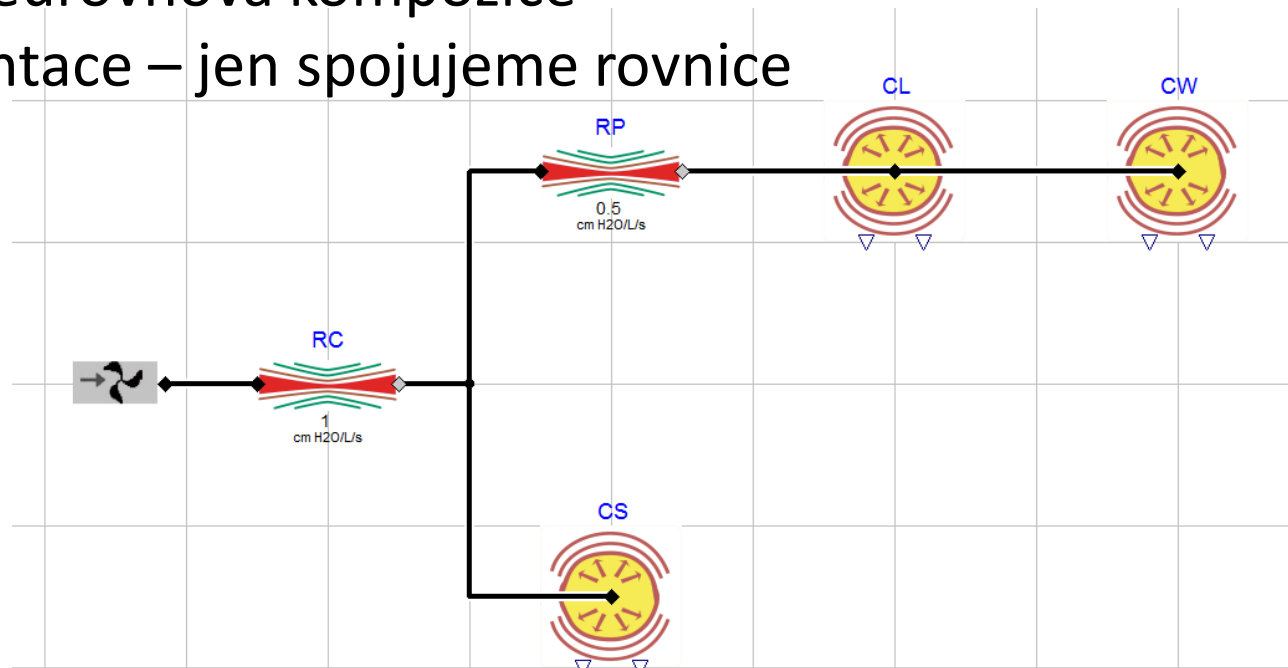


```
model BouncingBall
...
equation
  impact=h <= 0.0;
  der(v)=if flying then -g else 0;
  der(h)=v;
  when {h <= 0.0 and v <= 0.0, impact} then
    v_new=if edge(impact) then -e*pre(v) else 0;
    flying=v_new > 0;
    reinit(v, v_new);
  end when;
end BouncingBall;
```

- Co je nejbliže fyzice makrosvětá?

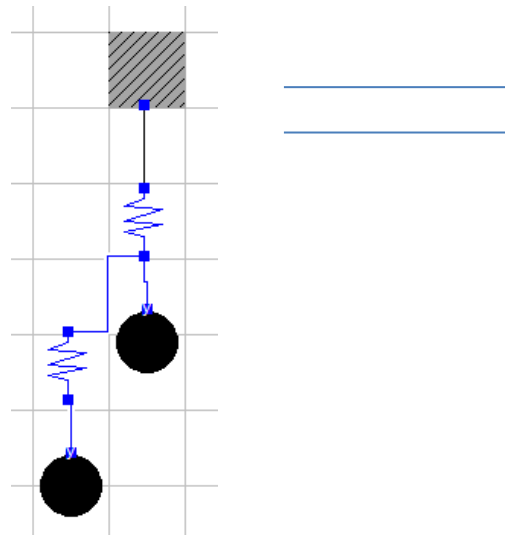
Pohled komponentového návrhu

- Schéma zapojení
- pohled - co ikona to fyzická komponenta - vlastně stavíme přímo blokový schéma
- Spojovací linky vyjadřují propojení mezi komponentami
- Uvnitř komponentů jsou rovnice popisující jejich chování
- Hierarchická víceúrovňová kompozice
- textová reprezentace – jen spojujeme rovnice



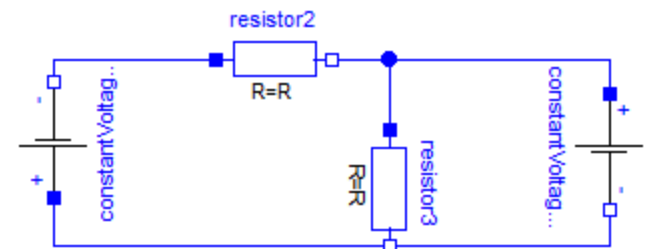
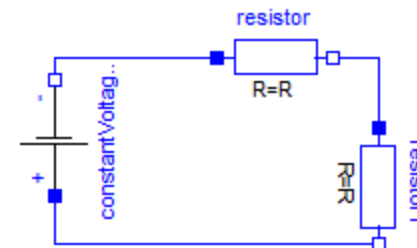
Drag n drop vs. Equation based

- Equation based
 - Zapisujeme rovnice v komponentě
- Drag'n'drop
 - Spojujeme komponenty
 - Parametrizace prvků



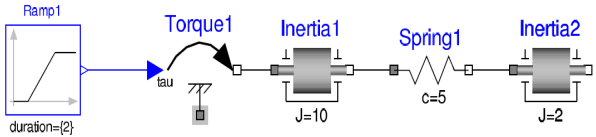
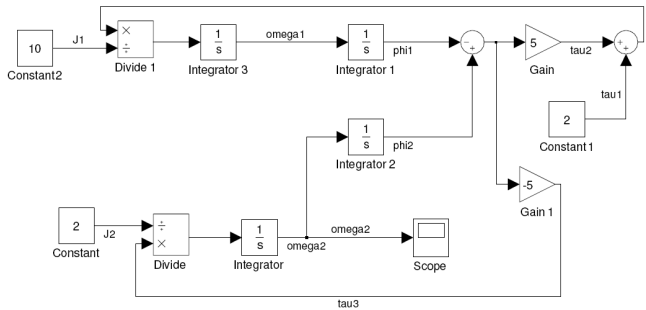
```
model system
fix fix1 annotation (Placement(transformation(extent={{-20,60},{0,80}})));
spring spring1
  annotation (Placement(transformation(extent={{-20,20},{0,40}})));
mass mass1(m=1, x(start=-1))
  annotation (Placement(transformation(extent={{-20,-12},{0,8}})));
spring spring2(k=2)
  annotation (Placement(transformation(extent={{-40,-18},{-20,2}})));
mass mass2(m=1, initPos=-1)
  annotation (Placement(transformation(extent={{-40,-50},{-20,-30}})));
equation
connect(fix1.position_y1, spring1.y1) annotation (Line(
  points={{-11,61},{-11,39}},
  color={0,0,0},
  smooth=Smooth.None));
connect(mass1.y, spring1.y2) annotation (Line(
  points={{-10,7},{-10,14},{-11,14},{-11,21}},
  color={0,0,255},
  smooth=Smooth.None));
connect(spring2.y1, spring1.y2) annotation (Line(
  points={{-31,1},{-20.5,1},{-20.5,21},{-11,21}},
  color={0,0,255},
  smooth=Smooth.None));
connect(mass2.y, spring2.y2) annotation (Line(
  points={{-30,-31},{-30,-17},{-31,-17}},
  color={0,0,255},
  smooth=Smooth.None));
en system;
```

- Co to je kauzalita a akauzalita
- Postup výpočtu je dán zápisem
 - $X = x + 1$
- Postup výpočtu je dán strukturou
 - Vyřeší kompilátor
- (postup výpočtu se mění)



Akauzální modelování

Pořadí výpočtu není dáno při modelování

	Akauzální	Kauzální
Vizuální návrh		
Rovnice	<p>Rovnice rezistoru: $R \cdot i = v;$</p>	<p>Kauzální možnosti $i := v/R;$ $v := R \cdot i;$ $R := v/i;$</p>

Syntax

- Struktura
- Klíčová slova
 - Package
 - Model
 - Connector
 - Equation
 - Initial equation
 - End
- Datové typy

Typy

- Real
 - spojitá
- Integer
- Boolean
- String
- Enumeration

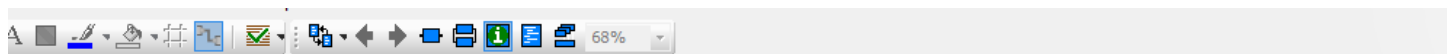
- Předpony:
 - Constant
 - constant Real x;
 - Parameter
 - parameter Boolean alive;
 - Discrete
 - discrete Real schody;
 - Input, output

Komentáře

- V kódu:
 - // inline komentář
 - /* blokový komentář*/
 - Pozor, v OMC zatím při kompilaci vypadává!

Dokumentace

- "Popis modelu" `partial block SO "Single Output continuous control block"`
`extends BlockIcon;`
- "Popis proměnné" `RealOutput y "Connector of Real output signal"`
`end SO;`
- Objeví se v generované dokumentaci



Single Output continuous control block

Information

Block has one continuous Real output signal.

Extends from [BlockIcon](#) (Basic graphical layout of input/output block).

Connectors

Type	Name	Description
output RealOutput	y	Connector of Real output signal

Name: SO

Path: Modelica.Blocks.Interfaces.SO

Filename: C:/Program Files/Dymola 7.4/Modelica/Library/Modelica 3.1/Blocks/Interfaces.mo

Version: 3.1, 2009-08-14, build 6 (2010-01-17 20:15:49Z)

Třídy a základ dědičnosti

- Vše je třída
 - Package
 - Model
 - Connector

Dědičnost

- Každá třída podporuje vícenásobné dědění
- Keyword *extends*
- Keyword *partial* – „virtual“ / „abstract“
- Přebírá jak grafiku, tak objekty a proměnné
 - Univerzální ikonky

parent class to Color ←

restricted kind of class without equations ←

child class or subclass ←

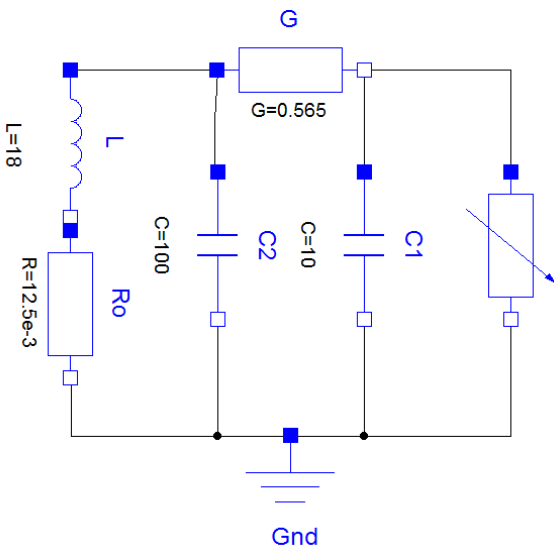
keyword denoting inheritance ←

```
record ColorData
  parameter Real red = 0.2;
  parameter Real blue = 0.6;
  Real green;
end ColorData;
```

```
class Color
  extends ColorData;
  equation
    red + blue + green = 1;
end Color;
```

Konektory

- Veličiny úsilí
 - $x_1 = x_2 = x_3 = \dots$
- Veličiny toku – keyword *flow*
 - $x_1 + x_2 + x_3 + \dots = 0$



connector Pin "Pin of an electrical component"
Modelica.SIunits.Voltage v "Potential at the pin"
flow Modelica.SIunits.Current i "Current flowing into the pin"
end Pin;

Anotace

- Definují zobrazení a pozice
 - Ikon
 - Diagramu
 - Propojení
- Dokumentace
- Vizuální návrh
- Vše ostatní krom matematických vlastností
 - Nemá vliv na funkci

```
connector Pin "Pin of an electrical component"  
Modelica.SIunits.Voltage v "Potential at the pin"  
flow Modelica.SIunits.Current i "Current flowing into the pin" annotation (  
  unassignedMessage="An electrical current cannot be uniquely calculated.");  
annotation (defaultComponentName="pin",  
  Icon(coordinateSystem(preserveAspectRatio=true, extent={{-100,-100},{100,  
    100}}), graphics={Rectangle(  
    extent={{-100,100},{100,-100}},  
    lineColor={0,0,255},  
    fillColor={0,0,255},  
    fillPattern=FillPattern.Solid)),  
  Diagram(coordinateSystem(preserveAspectRatio=true, extent={{-100,-100},{  
    100,100}}), graphics={Rectangle(  
    extent={{-40,40},{40,-40}},  
    lineColor={0,0,255},  
    fillColor={0,0,255},  
    fillPattern=FillPattern.Solid), Text(  
    extent={{-160,110},{40,50}},  
    lineColor={0,0,255},  
    textString="%name"))},  
  Documentation(revisions="<html>  
<ul>  
<li><i> 1998 </i>  
  by Christoph Clauss<br> initially implemented<br>  
</li>  
</ul>  
</html>"));  
end Pin;
```

Shrnutí

- Nástroje Modelica
- Vlastnosti Modelicy
- Co je vizuální návrh a co je equation based modelování
- Modelica umí simulovat spojité i diskrétní
- Bloky se spojují konektory
- K definici zobrazení se používají anotace

Bazar

- Nabízíme diplomku - Průtok kanylou pro ecmo
- Nechcete někdo pracovat v motole?
 - Biomedicínský inženýr, technik