TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Maschinenwesen, Professur für Baumaschinen und Fördertechnik

Simulating MultiBody Applications with OpenModelica

– Current Status -

Christian Schubert, Dresden University

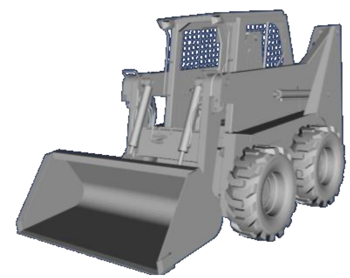
Linköping,
04.02.2013

Outline

1. Introduction
2. Feature Status
3. Performance by looking at Examples
    a. Examples
    b. Translation
    c. Compilation
    d. Simulation
4. Possible Future Directions

Applications

Models

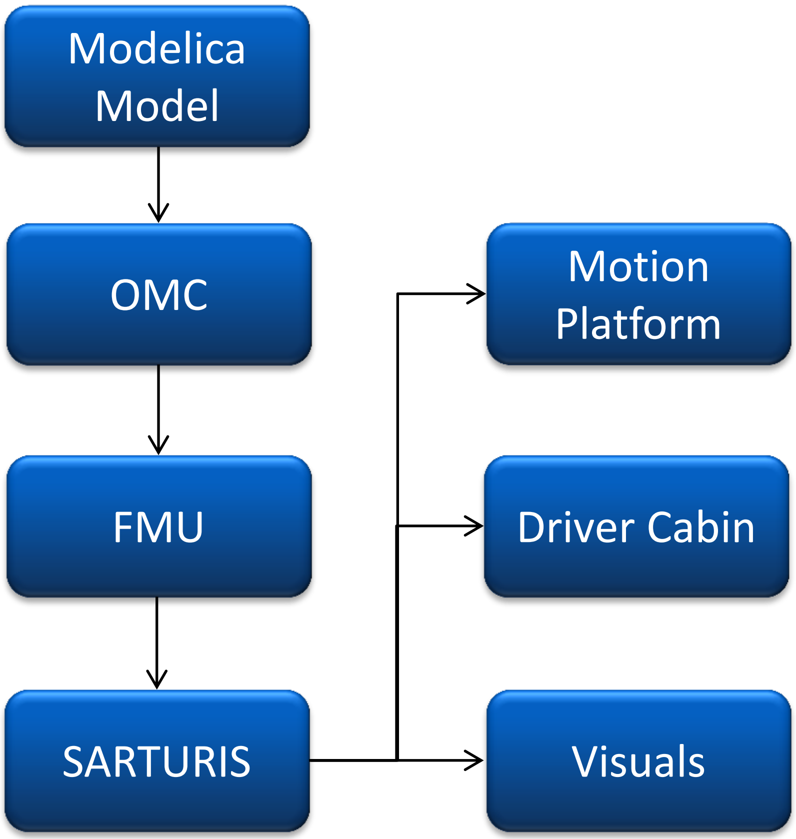


TECHNISCHE UNIVERSITÄT DRESDEN

Applications

Simulator

Toolchain



=> Performance is most important to us

Required Features

Handling of Records, Vectors, Matrices	✔
Overconstrained Connection Graph	✔
Annotations: Inline, (__Dymola_)InlineAfterIndexReduction, derivative, noDerivative, Evaluate	✔
Dynamic State Selection	~
Handling of replaceable gravityFunction	✘

Peformance/Comfort Features

Tearing Reducing size of sparse blocks	✔ / ~
Nonlinear Solver (Kinematic Loops)	~
Analytic Jacobians	~
Robust and fast Event Handling	~
Support for large models	~
Visualization	~ / ✘

Current Status

Necessary features implemented

Workaround for gravityFunction

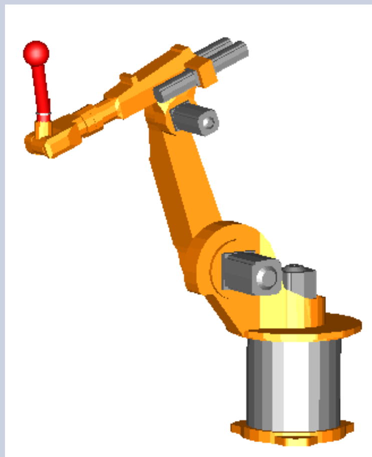
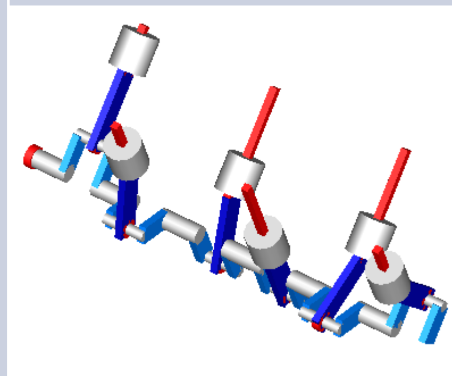
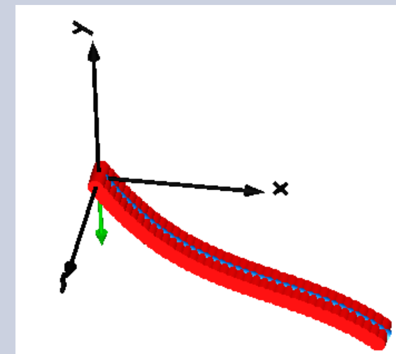
⇒ Multibody should work

⇒ One test from MSL 3.2.1 fails:
Elementary.UserDefinedGravityField

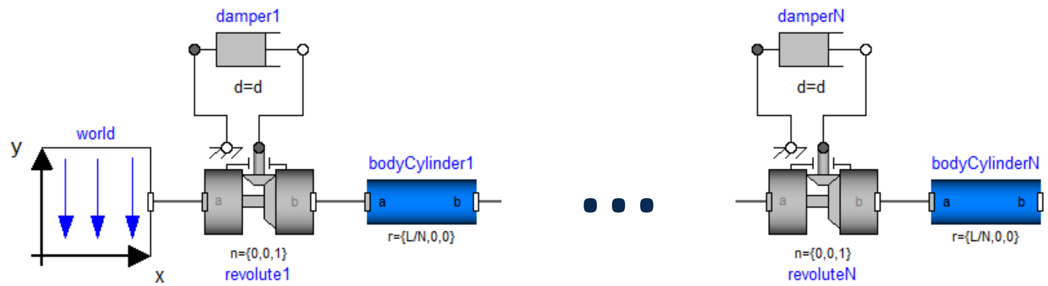
Performance/Comfort functions not finished

Let us see how they perform

Examples

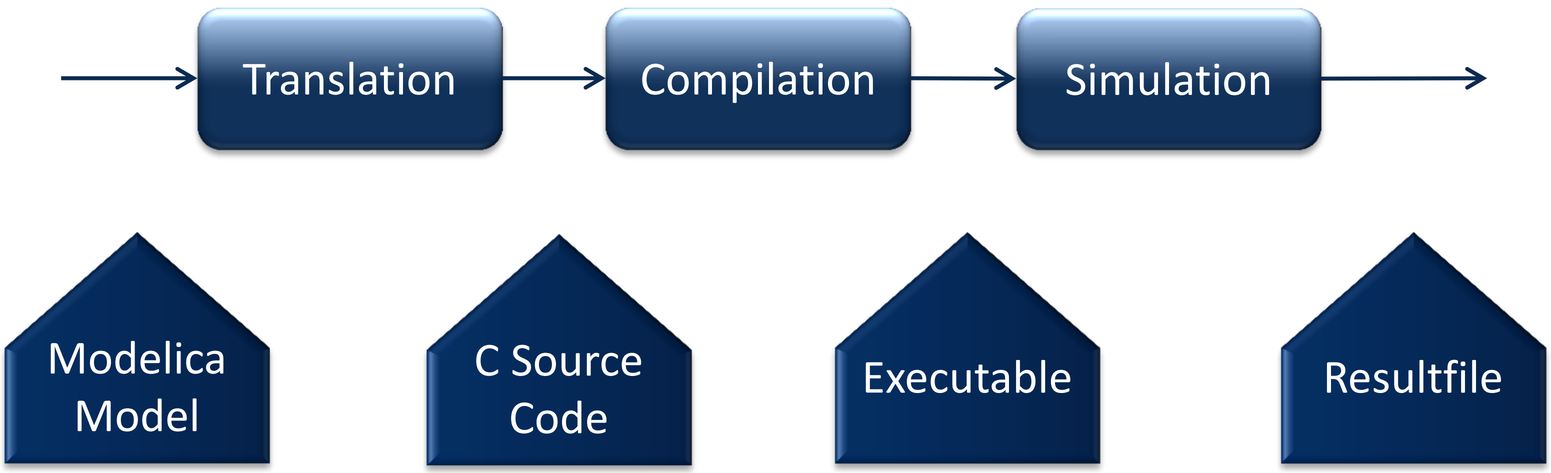
	RobotR3	EngineV6	Pendulum N
Picture			
#Vars	4921	12491	≈ 242 N
#States	36	4	2 N
#NIs	0	6	0
Linear	137 -> 6	322 -> 31	≈ 12 N -> N

Pendulum with N bodies

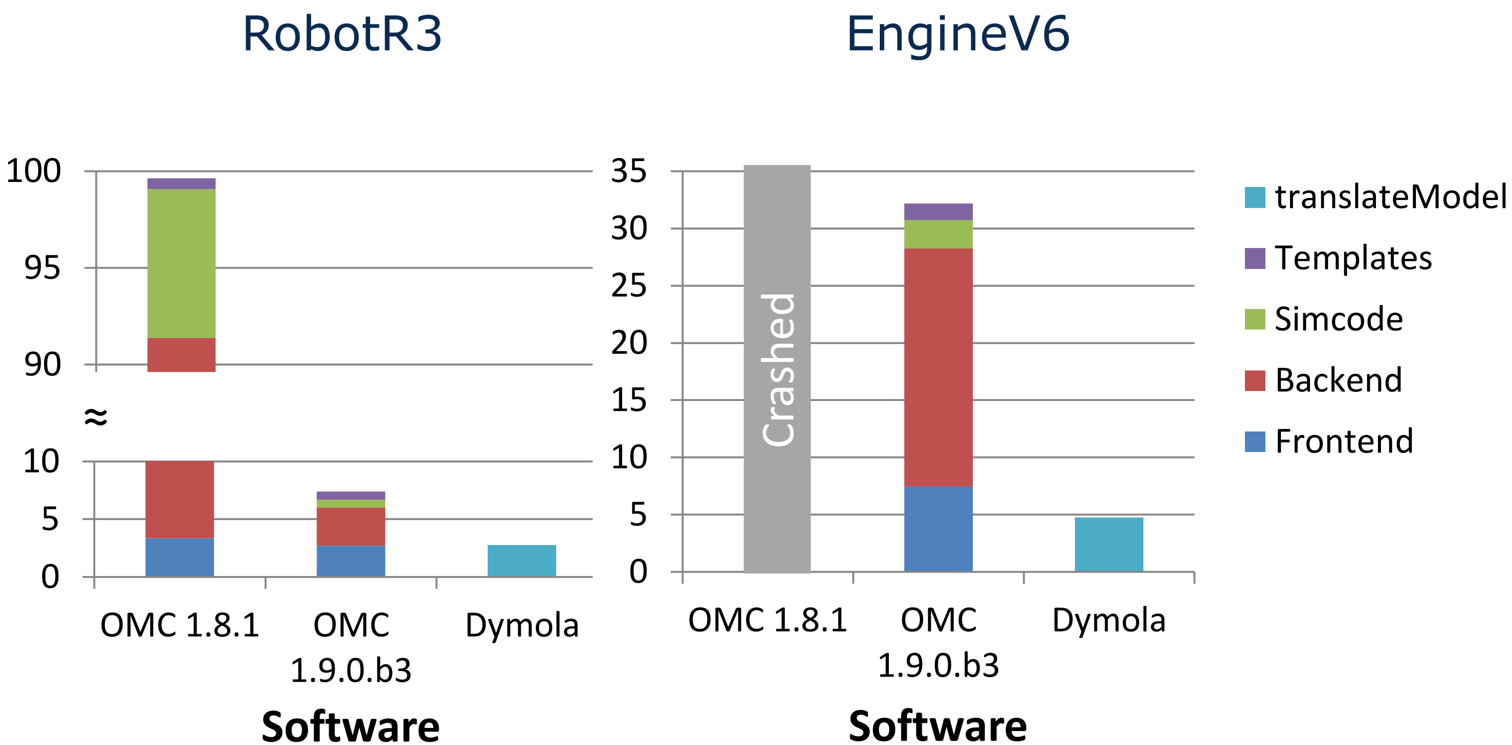



```
model Pendulum_N
  constant Integer N = 2;
  inner World world;
  Revolute revolute[N];
  BodyCylinder bodyCylinder[N];
  Damper damper[N];
equation
  connect(world.frame_b, revolute[1].frame_a);
  connect(revolute.frame_b, bodyCylinder.frame_a);
  connect(damper.flange_a, revolute.support);
  connect(damper.flange_b, revolute.axis);
  connect(bodyCylinder[1:N-1].frame_b, revolute[2:N].frame_a);
end Pendulum_N;
```

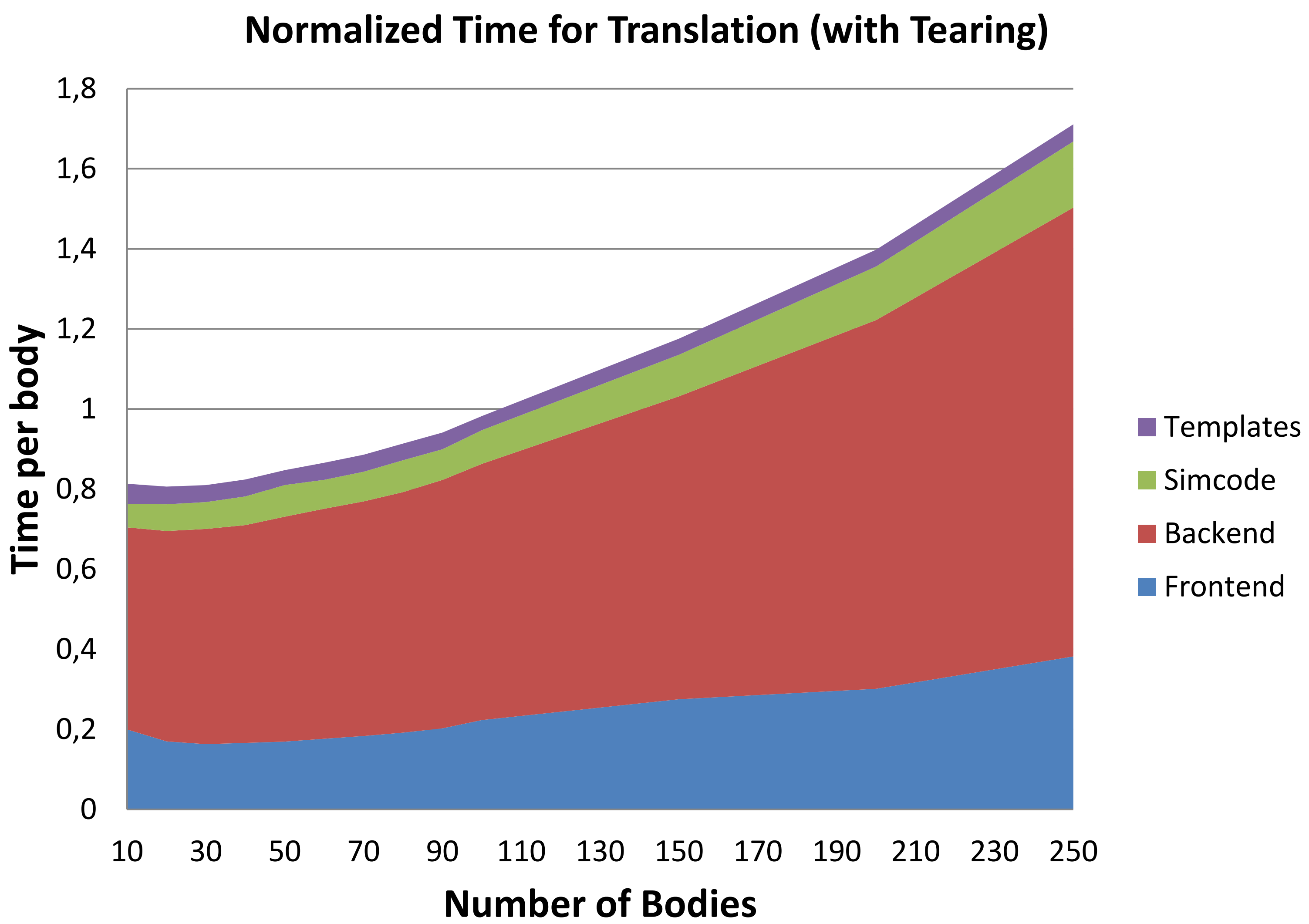
Division into three steps



Translation



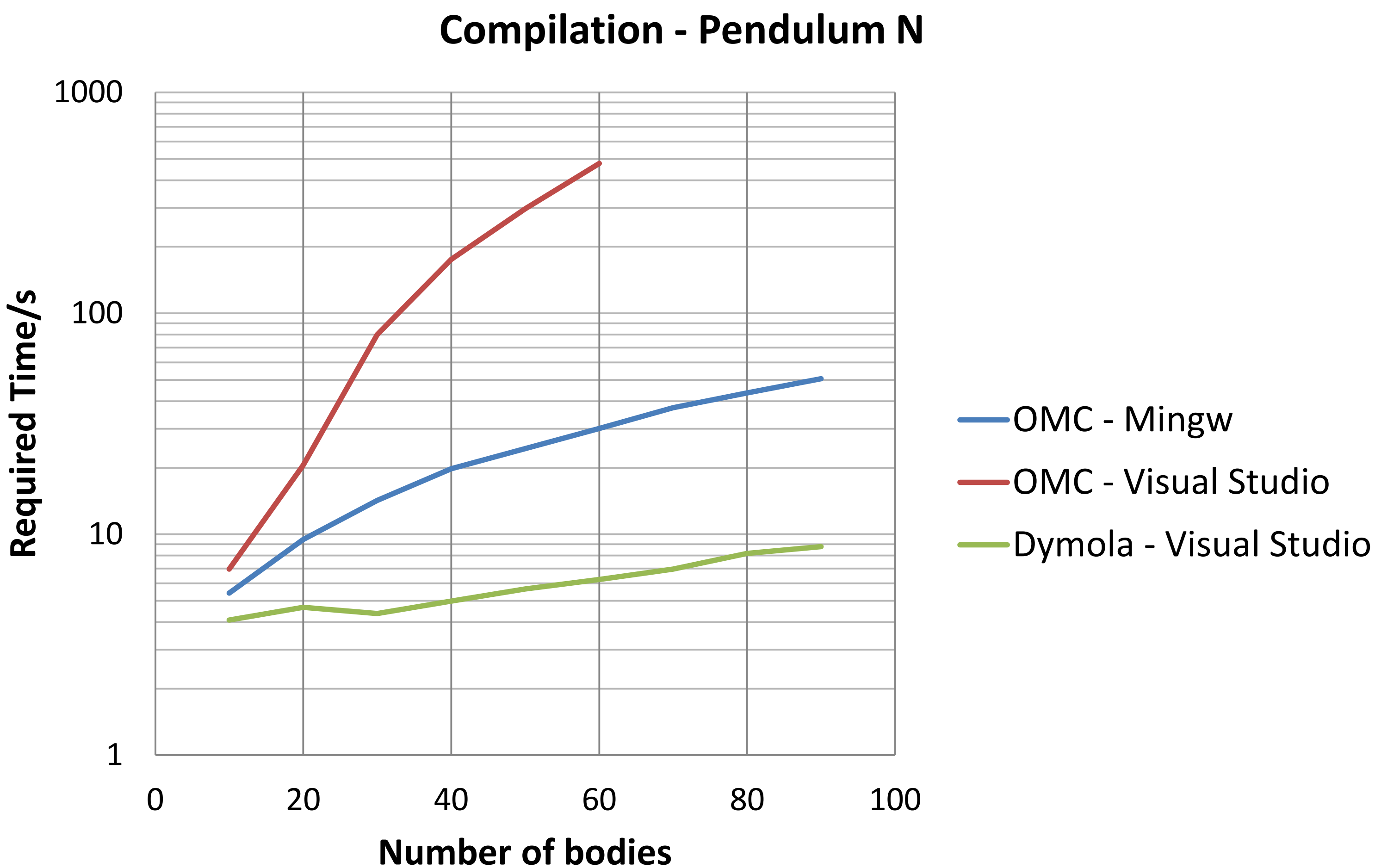
Translation



Translation

- Translation lies within factor 10 of Dymola
- Translation scales with $O(N^2)$
- Problems have been identified
  and are being fixed

Compilation



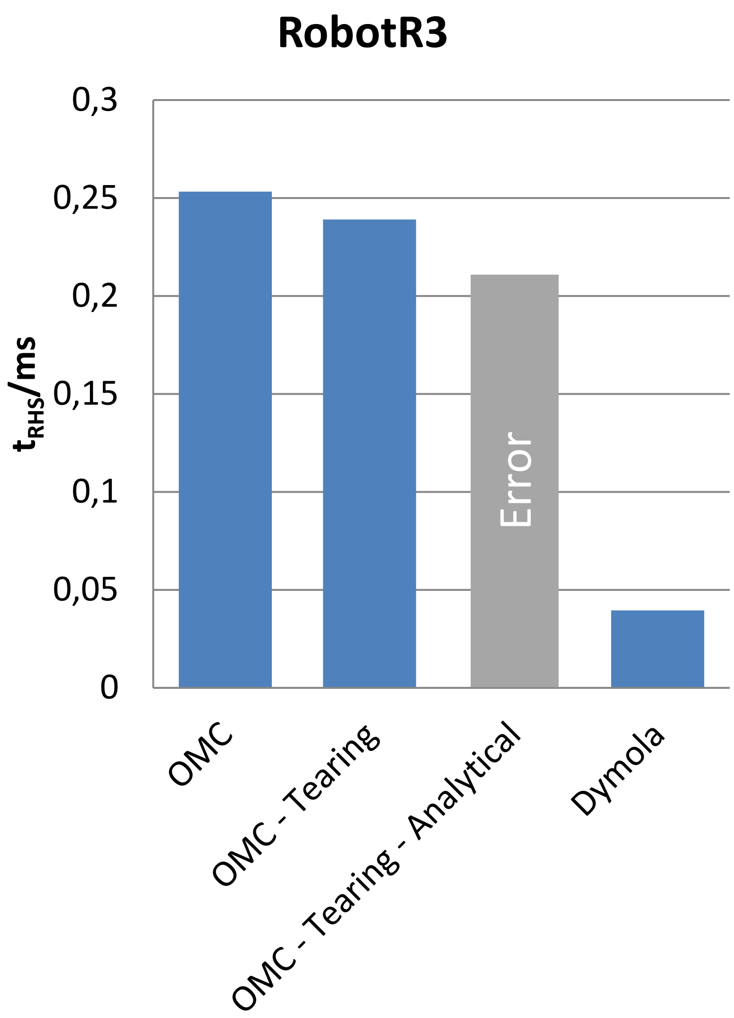
Compilation

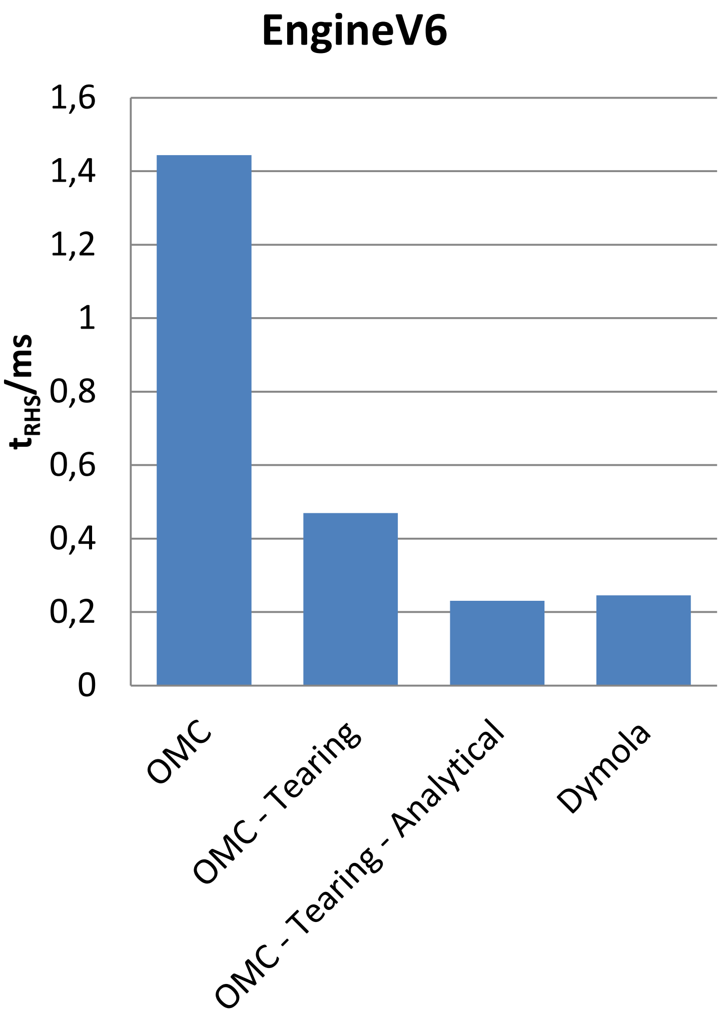
- Compilation as long as translation
- Compilation limits model size
- gcc faster than Visual Studio
- much slower than Dymola

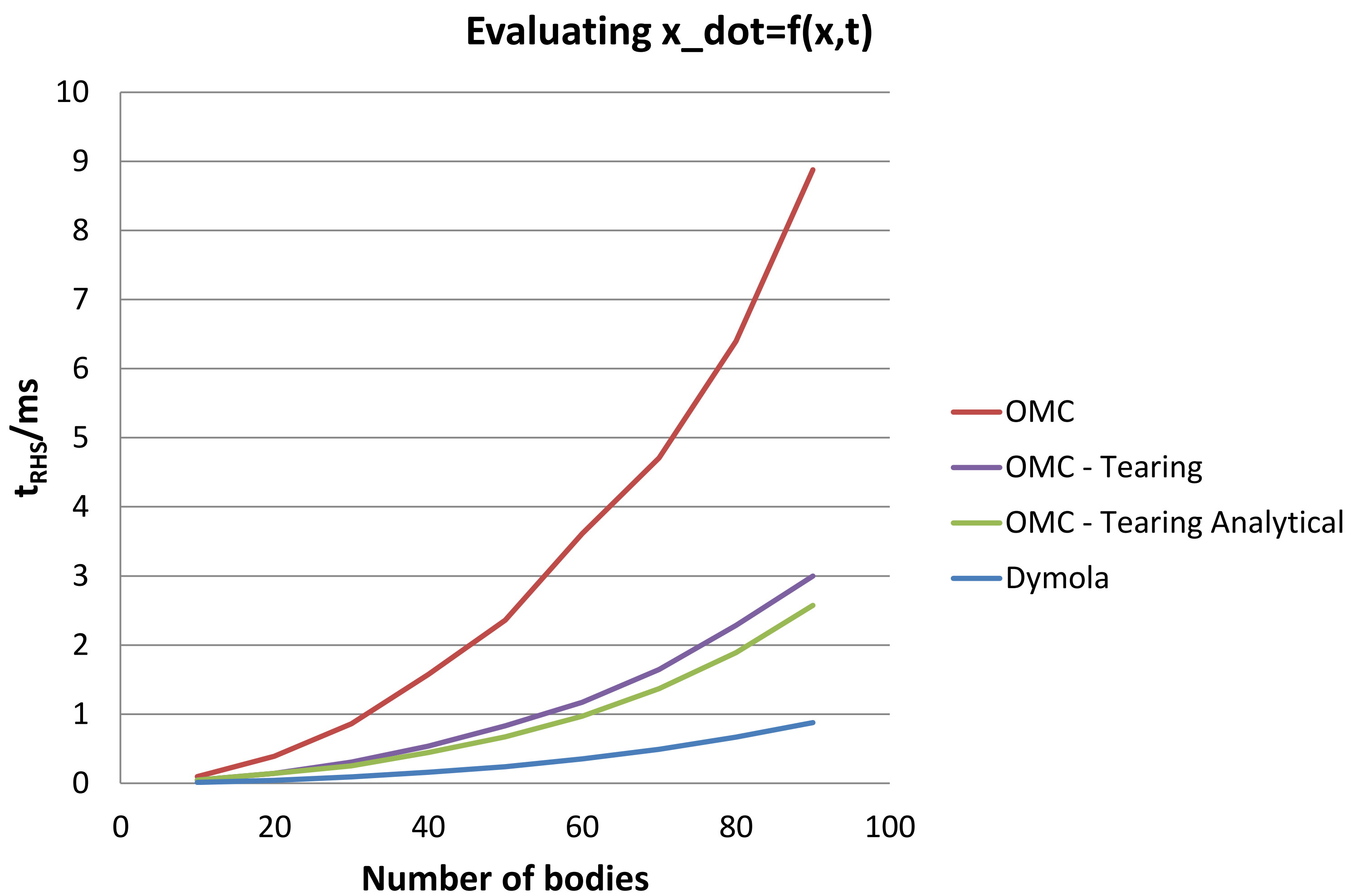
Tearing

- Deactivated by default for linear systems
    - +d=doLinearTearing
- Always treated as nonlinear system
    - => Numerical Jacobian
- Much faster/more robust with symbolic Jacobian
    - +d=doLinearTearing,NLSanalyticJacobian

Simulation


RobotR3
0,3
0,25
0,2
0,15
0,1
0,05
0
t_{RHS}/ms
Error
OMC
OMC - Tearing
OMC - Tearing - Analytical
Dymola


EngineV6
1,6
1,4
1,2
1
0,8
0,6
0,4
0,2
0
t_{RHS}/ms
OMC
OMC - Tearing
OMC - Tearing - Analytical
Dymola


Evaluating x_dot=f(x,t)
10
9
8
7
6
5
4
3
2
1
0
0
20
40
60
80
100
t_{RHS}/ms
Number of bodies
OMC
OMC - Tearing
OMC - Tearing Analytical
Dymola

EngineV6 ($\mathbf{f}(t,\mathbf{x})$ equally fast)

	OMC	Dymola
Simulation time[s]	9,72	5,18
steps	8081	7912
events	544	335
F-Evaluations	12273	24548
Jacobians	3338	3215

Not everything is counted!
Event detection has to be improved

Pendulum with 40 bodies ($\mathbf{f}(t, \mathbf{x})$ 3x slower)

	OMC	Dymola
Simulation time[s]	0,951	0,421
steps	257	240
events	0	0
F-Evaluations	364	1837
Jacobians	14	17

Possible Future Directions

Short term goals: Finishing
- Dynamic State Selection
- Tearing
- Analytic Jacobians

Mid term goal:
- Visualization
- Clean up OpenModelica-Trac

Long term goal:

A complete redesign of the generated code should be considered

TECHNISCHE UNIVERSITÄT DRESDEN

Thank you

»Wissen schafft Brücken.«

Christian Schubert, Dipl.-Ing, M.Sc.
Dresden University of Technology
Institute of Processing Machines and Mobile Machines
E-Mail: christian.schubert@tu-dresden.de
Tel.: +49 351 463-39278