
Debugging

Need for Debugging Tools

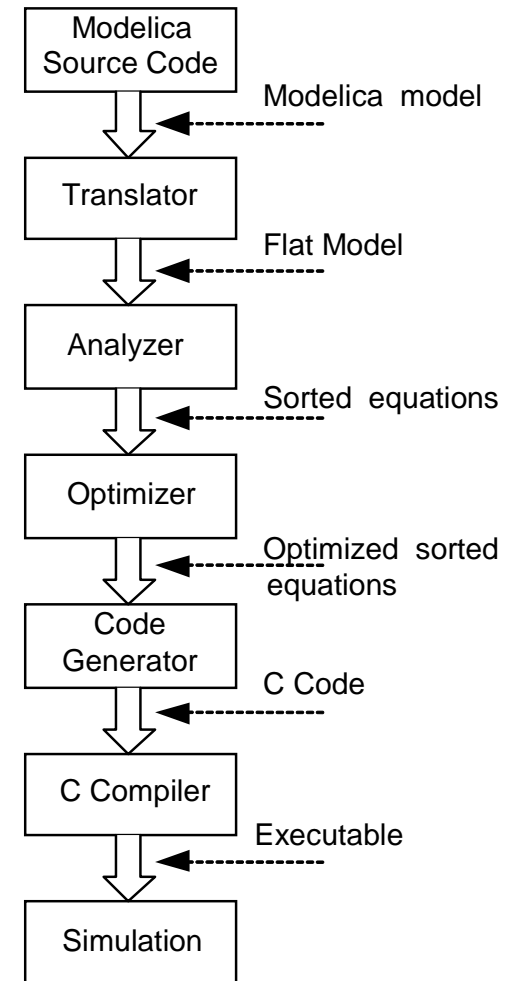
Map Low vs High Abstraction Level

- A **major part** of the total **cost** of software projects is due to testing and debugging
- US-Study 2002:
Software errors cost the US economy **annually ~60 Billion \$**
- **Problem: Large Gap in Abstraction Level**
from **Equations** to **Executable Code**
- Example error message (hard to understand)
Error solving nonlinear system 132
time = 0.002
residual[0] = 0.288956
x[0] = 1.105149
residual[1] = 17.000400
x[1] = 1.248448
...

Model Compiler Translation Phases Extended with Debugging

- Include debugging support within the translation process

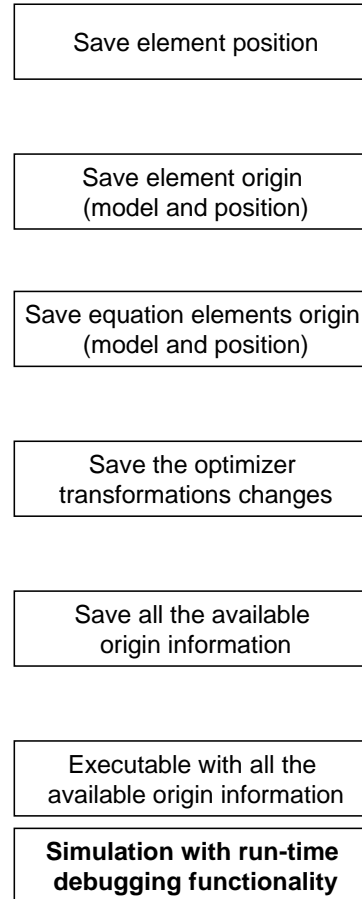
Normal Translation Process



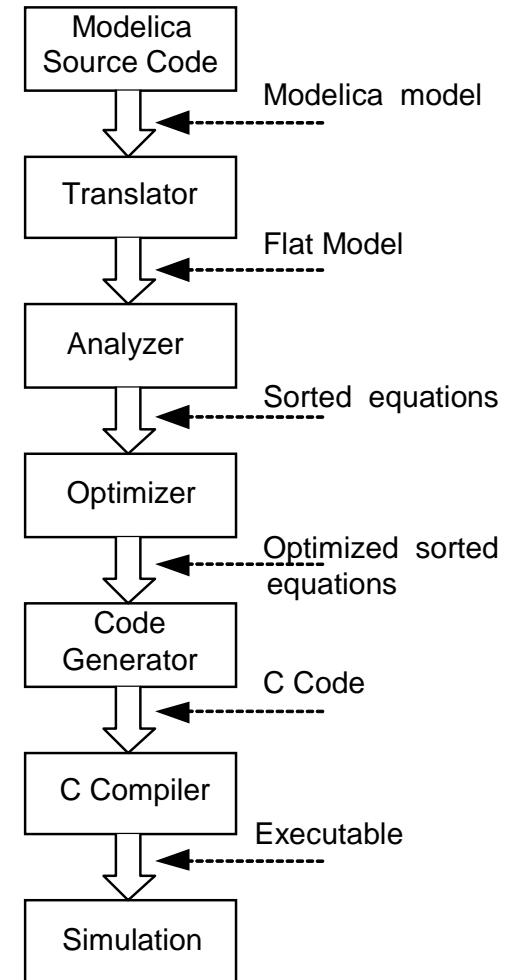
Model Compiler Translation Phases Extended with Debugging

- Additional step to provide needed debugging information

Debugging Translation Process Additional Steps



Normal Translation Process



Example Symbolic Transformations with Compiler Debug Trace

- Complicated to understand source of some errors
- Efficient trace of transformations

Example: $0 = y + \text{der}(x * \text{time} * z); \quad z = 1.0;$

(1) substitution:

$y + \text{der}(x * (\text{time} * z))$

\Rightarrow

$y + \text{der}(x * (\text{time} * 1.0))$

(2) simplify:

$y + \text{der}(x * (\text{time} * 1.0))$

\Rightarrow

$y + \text{der}(x * \text{time})$

**(3) expand derivative
(symbolic diff):**

$y + \text{der}(x * \text{time})$

\Rightarrow

$y + (x + \text{der}(x) * \text{time})$

(4) solve:

$0.0 = y + (x + \text{der}(x) * \text{time})$

\Rightarrow

$\text{der}(x) = ((-y) - x) / \text{time}$

Properties of Transformation Trace

- Most equations have very **few** transformations on them
- Most of the interesting equations have a few
 - Still rather readable
- Some extra care to handle Modelica variable aliasing
- Very **efficient** implementation, max 1% overhead

MSL 3.1 MultiBody DoublePendulum

# Ops	Frequency	Comment
0	457	Parameters
1	89	Dummy eq & know var
2	720	Alias vars
3	479	Alias vars
4	124	Alias after simplify
5	25	Alias after simplify
6	99	Alias after simplify
7	55	Scalar eq
8	37	...
9	110	...
10	72	...
11	12	...
12	25	...
13	35	...
14	3	Known constant after many replacements
21	27	World object (3x3 matrix with many occurrences of aliased vars)

OpenModelica Equation Model Debugger

Variables View

Variables Browser	Defined In Equations	Used In Equations
Case Sensitive	Index	Index
frame		
boxBody1		
body		
frame_a		
R		
T		

Equations View

Equations Browser	Defines	Depends
Index	Variable	Variable
819	world.frame_b.f[2]	boxBody1.frame_b.R.T[1,2]
820		boxBody1.frame_b.R.T[2,2]
821		revolute1.frame_b.f[1]
822		revolute1.frame_b.f[2]
823		
824		
825		
826		
827		
828		
829		
830		

Source View

```

317 // relationships between
318 quantities of frame_a and of
319 frame_b
320 frame_b.r_0 = frame_a.r_0;
321 if rooted(frame_a.R) then
322   R_rel =
323   Frames.planarRotation(e,
324   phi_offset + phi, w);
325   frame_b.R =
326   Frames.absoluteRotat
327   a.R, R_rel);
328   frame_b.f = -
329   Frames.resolve1(R_re
330   frame_b.t);
331   else
332   R_rel =
333   Frames.planarRotatio
334   phi_offset + phi, w)
335   frame_a.R =
336   Frames.absoluteRotat
337   b.R, R_rel);
338   frame_b.f = -
339   Frames.resolve1(R_re
340   frame_a.f);
341   frame_b.t = -
342   Frames.resolve1(R_re
343   frame_a.t);
344   end if;
  
```

Showing equation transformations of a model:

0 = y + der(x * time * z); z = 1.0;

(1) substitution:
 $y + \text{der}(x * (\text{time} * z))$
 \Rightarrow
 $y + \text{der}(x * (\text{time} * 1.0))$

(2) simplify:
 $y + \text{der}(x * (\text{time} * 1.0))$
 \Rightarrow
 $y + \text{der}(x * \text{time})$

(3) expand derivative (symbolic diff):
 $y + \text{der}(x * \text{time})$
 $\Rightarrow y + (x + \text{der}(x) * \text{time})$

(4) solve:
 $0.0 = y + (x + \text{der}(x) * \text{time})$
 \Rightarrow
 $\text{der}(x) = ((-y) - x) / \text{time}$
 $\text{time} < 0$

Mapping run-time error to source model position

Transformations Browser – EngineV6 Overview (11 116 equations in model)

Activities OMEdit Tue 12:06 sv ⏪ 🔍 📄 = Martin Sjöblund

OMEdit - Transformational Debugger

/tmp/OpenModelica_marsj/OMEdit/Modelica.Mechanics.MultiBody.Examples.Loops.EngineV6_info.xml

Variables

Variables Browser

Case Sensitive Regular Expression

Expand All Collapse All

Variables	Comment	Line	Location
phi	Exter...phi	6616	/usr/li...onal.mo
phi	Relat...ame_b	260	/usr/li...ints.mo
phi_offset	Relat...+ phi	242	/usr/li...ints.mo
Crank1	Absol...frame	11	/usr/li...mes.mo
body	Trans...frame	10	/usr/li...mes.mo
phi	Dumm...body	805	/usr/li...arts.mo
phi[1]	Dumm...body	805	/usr/li...arts.mo
phi[2]	Dumm...body	805	/usr/li...arts.mo
phi[3]	Dumm...body	805	/usr/li...arts.mo
phi_d	= der(phi)	809	/usr/li...arts.mo
phi_d[1]	= der(phi)	809	/usr/li...arts.mo
phi_d[2]	= der(phi)	809	/usr/li...arts.mo

Defined In Equations

Index	Type	Equation
587	initial	(nonlinear)
5016	regular	(nonlinear)

Used In Equations

Index	Type	Equation
...	regular	(assignment) cylinder...cos(cylinder3.B2.phi)
...	regular	(assignment) cylinder3... sin(cylinder3.B2.phi)
...	regular	(assignment) cylinder...sin(cylinder3.B2.phi)
...	regular	(assignment) cylinder...cos(cylinder3.B2.phi)
...	regular	(assignment) der(cyl...der3.Rod.body.w_a[1]
...	regular	(assignment) der(cyl...der3.Rod.body.w_a[1]
...	regular	(assignment) der(cyl...der3.Rod.body.w_a[1]

Source Browser

/usr/lib/omlibrary/Modelica 3.2.1/Mechanics/MultiBody/Jo

```

386 Connections.branch(frame_a.R,
387 frame_b.R);
388
389 assert(cardinality(frame_a) > 0,
390 "Connector frame_a of revolute
391 joint is not connected");
392
393 assert(cardinality(frame_b) > 0,
394 "Connector frame_b of revolute
395 joint is not connected");
396
397
398 angle = phi_offset + phi;
399 w = der(phi);
400 a = der(w);
401
402
403 // relationships between quantities
404 of frame_a and of frame_b
405 frame_b.r_0 = frame_a.r_0;
406
407
408 if rooted(frame_a.R) then
409   R_rel = Frames.planarRotation(e,
410 phi_offset + phi, w);
411   frame_b.R =
412 Frames.absoluteRotation(frame_a.R,
413 R_rel);
414   frame_a.f = -
415 Frames.resolve(R_rel, frame_b.f);
416   frame_a.t = -
417 Frames.resolve(R_rel, frame_b.t);
418 else
419   R_rel = Frames.planarRotation(-e,
420 phi_offset + phi, w);
421   frame_a.R =
422 Frames.absoluteRotation(frame_b.R,
423 R_rel);
424   frame_b.f = -
425 Frames.resolve(R_rel, frame_a.f);
426   frame_b.t = -
427 Frames.resolve(R_rel, frame_a.t);
428 end if;
429
430 // d'Alemberts principle
431 tau = -frame_b.t*e;
432
433 // Connection to internal

```

Equations

Equations Browser

Inc	Type	Equation
...	regular	(assignment) cylind...ylinder3.Cylinder.s
...	regular	(assignment) cylind...linder3.gasForce.L
...	regular	(assignment) cylind...linder3.gasForce.x
...	regular	(assignment) cylind...linder3.gasForce.V
...	regular	(assignment) cylind...linder3.gasForce.L
...	regular	(assignment) cylind...linder.s else 1e-06
...	regular	(assignment) cylind...k2.frame_b.R.T[2,3]
...	regular	(linear, r_rel_a = Fra...r_0 - frame_a.r_0);
...	regular	(linear, frame_b.r_0 =... (s_offset + s));
...	regular	(assignment) cylind...linder3.gasForce.x
...	regular	(assignment) cylind...linder3.gasForce.p
...	regular	(assignment) cylind...r3.gasForce.d ^ 2.0
...	regular	(assignment) cylind...linder3.gasForce.k
...	regular	(assignment) cylind...ody.w_a[1] - load.w
...	regular	(assignment) der(c...r3.Rod.body.w_a[1]

Defines

Variable
der(cylinder3.B2.R_rel.T[3,3])

Depends

Variable
cylinder3.B2.phi
cylinder3.Rod.body.w_a[1]

Equation Operations

Operations

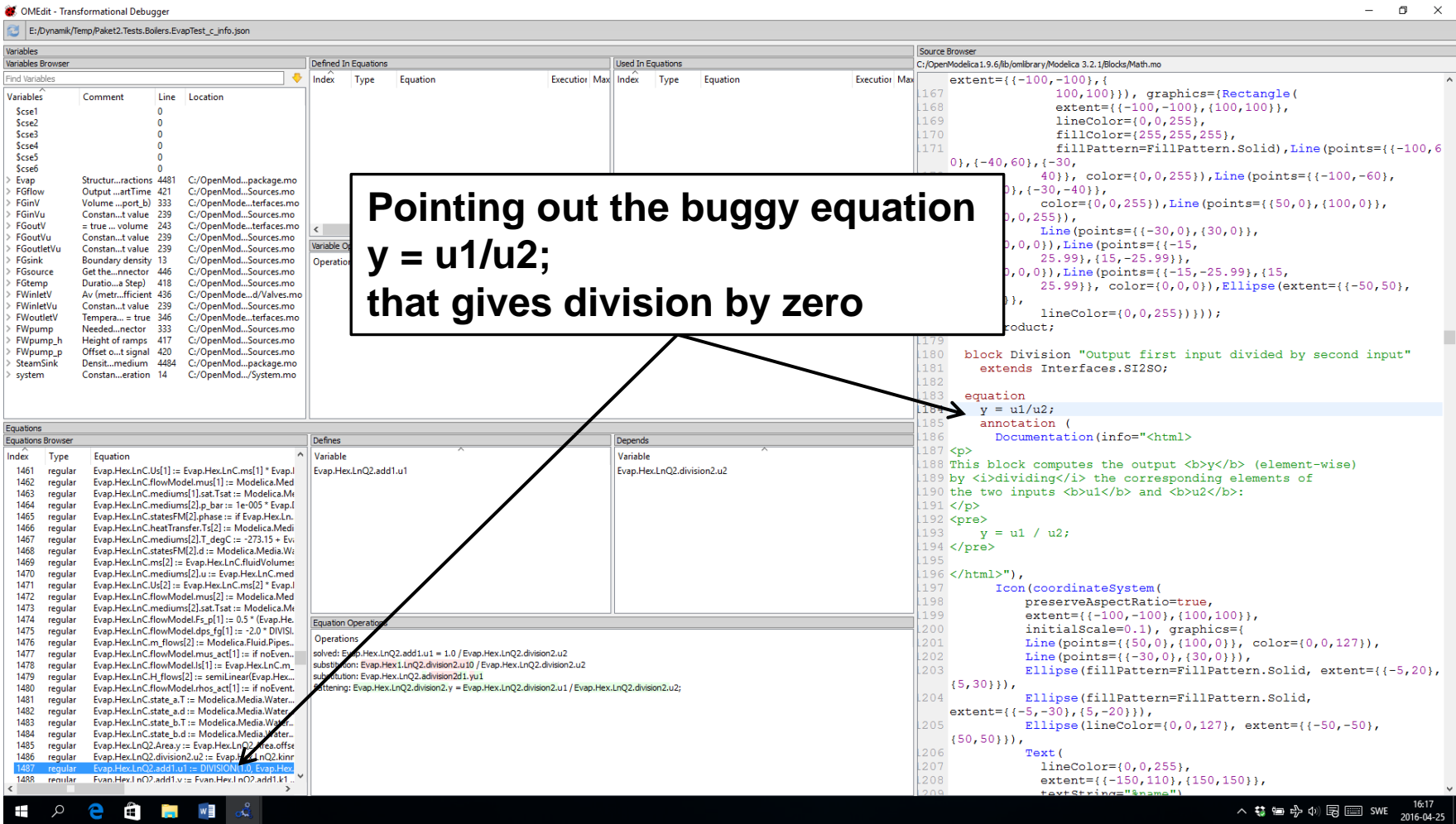
```

- solved: der(cylinder3.B2.R_rel.T[3,3]) = (-sin(cylinder3.B2.phi)) * cylinder3.Rod.body.w_a[1]
- substitute: (-sin(cylinder3.B2.phi)) * cylinder3.B2.w => (-sin(cylinder3.B2.phi)) * cylinder3.Rod.body.w_a[1]
- differentiate: dcos(cylinder3.B2.phi)/dt = (-sin(cylinder3.B2.phi)) * der(cylinder3.B2.phi)
- differentiate: d(cylinder3.B2.R_rel.T[3,3])/dt = der(cylinder3.B2.R_rel.T[3,3])
- scalarize(9): cylinder3.B2.R_rel.T = {{1.0, 0.0, 0.0}, [-0.0, c...B2.phi]] => cylinder3.B2.R_rel.T[3,3] = cos(cylinder3.B2.phi)
- simplify: cylinder3.B2.R_rel.T = {{1.0 * 1.0 + (1.0 - 1.0 * 1.0) * B2.phi}, {0.0, -sin(cylinder3.B2.phi), cos(cylinder3.B2.phi)}}
- substitute: {{cylinder3.B2.e[1] * cylinder3.B2.e[1] + (1.0 - cy...2.phi), 0.0 * 0.0 + (1.0 - 0.0 * 0.0) * cos(cylinder3.B2.phi)}}
- inline: cylinder3.B2.R_rel = Modelica.Mechanics.MultiBody...[2] * cylinder3.B2.w, cylinder3.B2.e[3] * cylinder3.B2.w)
- original: R_rel = Frames.planarRotation(e, phi_offset + phi, w); => flattened:

```

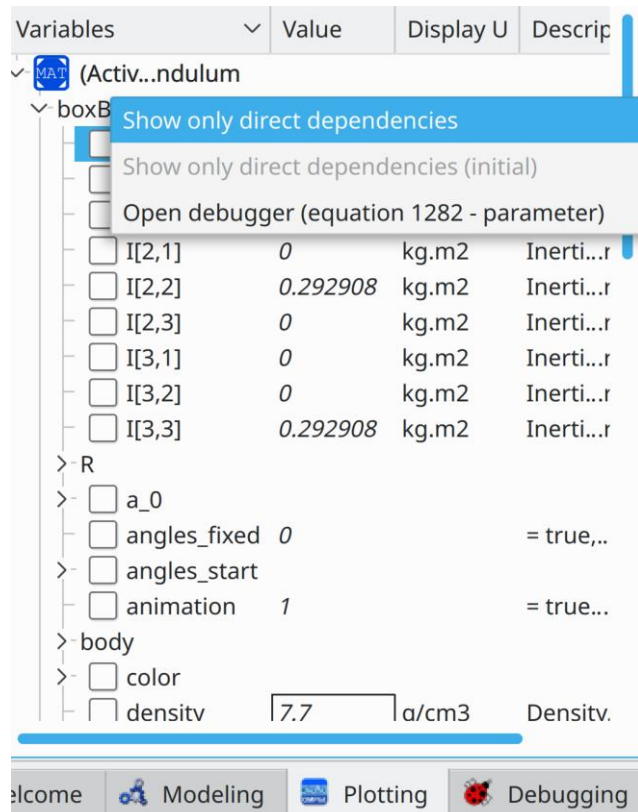

Equation Model Debugger on Siemens Model

(Siemens Evaporator test model, 1100 equations)



New OM Debug function that can trace (and plot) which variables and equations influence a variable

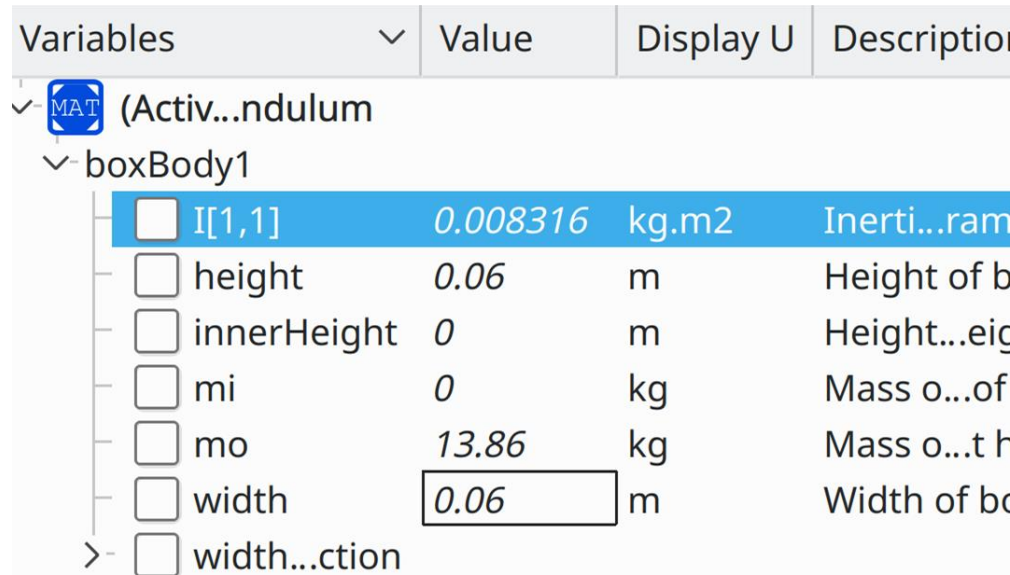
New menu choice to show direct dependencies



The screenshot shows the OpenModelica GUI with the 'Variables' window open. A context menu is displayed over the 'boxBody1' variable, with the option 'Show only direct dependencies' highlighted in blue. The menu also includes 'Show only direct dependencies (initial)' and 'Open debugger (equation 1282 - parameter)'. The background table lists various variables and their values.

Variables	Value	Display U	Descript
<input checked="" type="checkbox"/> MAT (Activ...ndulum			
<input checked="" type="checkbox"/> boxB			
<input type="checkbox"/> I[2,1]	0	kg.m2	Inerti...r
<input type="checkbox"/> I[2,2]	0.292908	kg.m2	Inerti...r
<input type="checkbox"/> I[2,3]	0	kg.m2	Inerti...r
<input type="checkbox"/> I[3,1]	0	kg.m2	Inerti...r
<input type="checkbox"/> I[3,2]	0	kg.m2	Inerti...r
<input type="checkbox"/> I[3,3]	0.292908	kg.m2	Inerti...r
> R			
<input type="checkbox"/> a_0			
<input type="checkbox"/> angles_fixed	0		= true,...
<input type="checkbox"/> angles_start			
<input type="checkbox"/> animation	1		= true...
> body			
<input type="checkbox"/> color			
<input type="checkbox"/> densitv	7.7	a/cm3	Densitv.

List of Variables directly influencing:



The screenshot shows the 'List of Variables directly influencing:' window. It displays a table of variables and their values, with the 'boxBody1' variable selected. The table lists variables like 'I[1,1]', 'height', 'innerHeight', 'mi', 'mo', 'width', and 'width...ction'.

Variables	Value	Display U	Description
<input checked="" type="checkbox"/> MAT (Activ...ndulum			
<input checked="" type="checkbox"/> boxBody1			
<input type="checkbox"/> I[1,1]	0.008316	kg.m2	Inerti...ram
<input type="checkbox"/> height	0.06	m	Height of b
<input type="checkbox"/> innerHeight	0	m	Height...eig
<input type="checkbox"/> mi	0	kg	Mass o...of
<input type="checkbox"/> mo	13.86	kg	Mass o...t h
<input type="checkbox"/> width	0.06	m	Width of b
> width...ction			

ABB Industry Use of OpenModelica Debugger

- ABB OPTIMAX® provides advanced model based control products for power generation and water utilities



- ABB: *“ABB uses several compatible Modelica tools, including OpenModelica, depending on specific application needs.”*
- ABB: *“OpenModelica provides outstanding debugging features that help to save a lot of time during model development.”*

Equation Debugging Summary

- Debugging **equation-based** models present new **challenges**
- **Equation** systems are **transformed** symbolically to a form hard for the user to understand
- Maintain and **explain** a **mapping** between the **low** level and the **high** level model
- **The first integrated static/dynamic debugger of any Modelica tool**

Debugging Example – Detecting Source of Chattering (excessive event switching) causing bad performance

OMEdit - Transformational Debugger

/tmp/OpenModelica_marsj/OMEdit/Debugging.Chattering.ChatteringEvents1_info.xml

Variables

Variables Browser

Find Variables

☐ Case Sensitive Regular Expression

Expand All Collapse All

Variables	Comment	Line	Location
x		7	/hom...g.
y		8	/hom...g.
z		9	/hom...g.

Defined In Equations

Inc	Type	Equation
2	initial	(assignment) x = 1.0
5	regular	(assignment) y = 2.0 * z

Used In Equations

Inc	Type	Equation
3	initial	(assignment) y = 2.0 * z
6	regular	(assignment) y = 2.0 * z

Variable Operations

Operations

Equations

Equations Browser

Inc	Type	Equation
1	initial	(assignment) x = 1.0
2	initial	(assignment) x = 1.0
3	initial	(assignment) y = 2.0 * z
4	initial	(assignment) der(x) = y
5	regular	(assignment) y = 2.0 * z
6	regular	(assignment) y = 2.0 * z
7	regular	(assignment) der(x) = y

Defines

Variable	Type
z	

Depends

Variable	Type
x	

Equation Operations

Operations

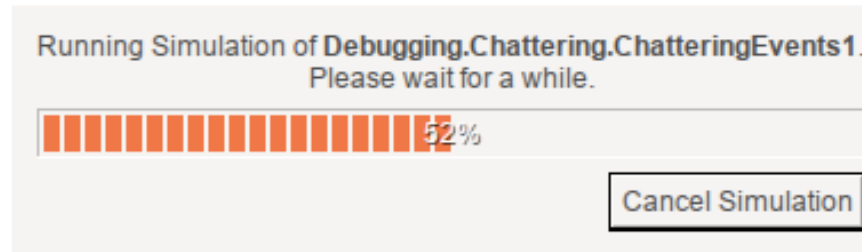
solved: z = if x > 0.0 then -1.0 else 1.0
original: z = if x > 0 then -1 else 1; => flattened: z = if x > 0.0 then -1.0 else 1.0;

Source Browser

/home/marsj/trunk/testsuite/openmodelica

```
1 within ;
2 package Debugging "Test
3 cases for debugging of
4 declarative models"
5
6 package Chattering "Models
7 with chattering behaviour"
8
9 model ChatteringEvents1
10 "Exhibits chattering
11 after t = 0.5, with
12 generated events"
13
14 Real x(start=1,
15 fixed=true);
16 Real y;
17 Real z;
18 equation
19 z = if x > 0 then -1
20 else 1;
21 y = 2*z;
22 der(x) = y;
23 annotation
24 (Documentation(info="<html>
25 <p>After t = 0.5, chattering
26 takes place, due to the
27 discontinuity in the right
28 hand side of the first
29 equation.</p>
30 <p>Chattering can be
31 detected because lots of
32 tightly spaced events are
33 generated. The feedback to
34 the user should allow to
35 identify the equation from
36 which the zero crossing
37 function that generates the
38 events originates.</p>
39 </html>"),
40 experiment(StopTime=1));
41 end ChatteringEvents1;
42
43 model ChatteringEvents2
44 "Exhibits chattering
45 after t = 0.422, with
46 generated events"
```

Error Indication – Simulation Slows Down

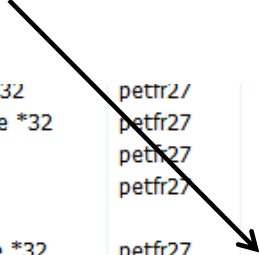


Exercise – Equation-based Model Debugger

In the model ChatteringEvents1, chattering takes place after $t = 0.5$, due to the discontinuity in the right hand side of the first equation. Chattering can be detected because lots of tightly spaced events are generated. The debugger allows to identify the (faulty) equation that gives rise to all the zero crossing events.

```
model ChatteringEvents1
  Real x(start=1, fixed=true);
  Real y;
  Real z;
equation
  z = noEvent(if x > 0 then -1 else 1);
  y = 2*z;
  der(x) = y;
end ChatteringEvents1;
```

Uses 25% CPU



acrotray.exe *32	pettr2/	00	9/6 K	A
AdobeARM.exe *32	petfr27	00	1,136 K	A
Bootcamp.exe	pettr27	00	1,448 K	B
conhost.exe	petfr27	00	1,300 K	C
csrss.exe		00	3,000 K	
DCSHelper.exe *32	petfr27	00	660 K	D
Debugging.Chattering...	petfr27	25	1,436 K	D
dllhost.exe	petfr27	00	2,224 K	C

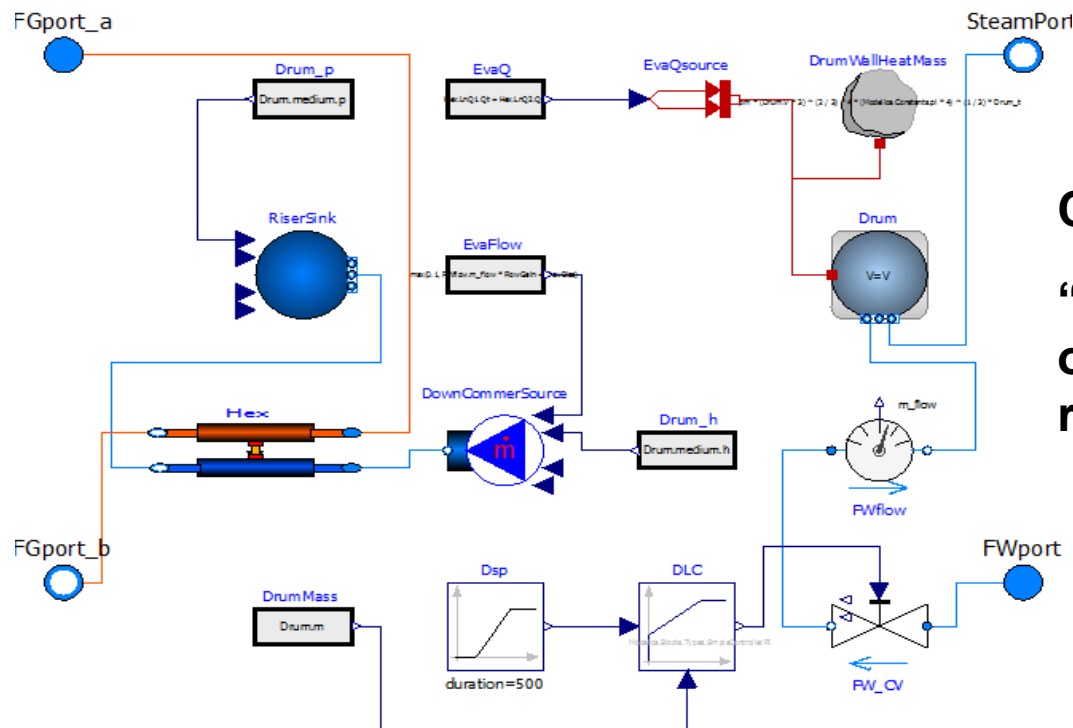
- Switch to OMEdit text view (click on text button upper left)
- Open the Debugging.mo package file using OMEdit
- Open subpackage Chattering, then open model ChatteringEvents1
- Simulate in debug mode (transformational debugger)
- Click on the button Debug more (see prev. slide)
- Possibly start task manager and look at CPU. Then click stop simulation button

Performance Analysis

Performance Profiling for faster Simulation

(Here: Profiling equations of Siemens Drum boiler model with evaporator)

- Measuring **performance** of equation blocks to find bottlenecks
 - Useful as input before model simplification for real-time applications
- Integrated with the debugger to **point out the slow equations**
- Suitable for **real-time profiling** (collect less information), or a complete view of all equation blocks and function calls

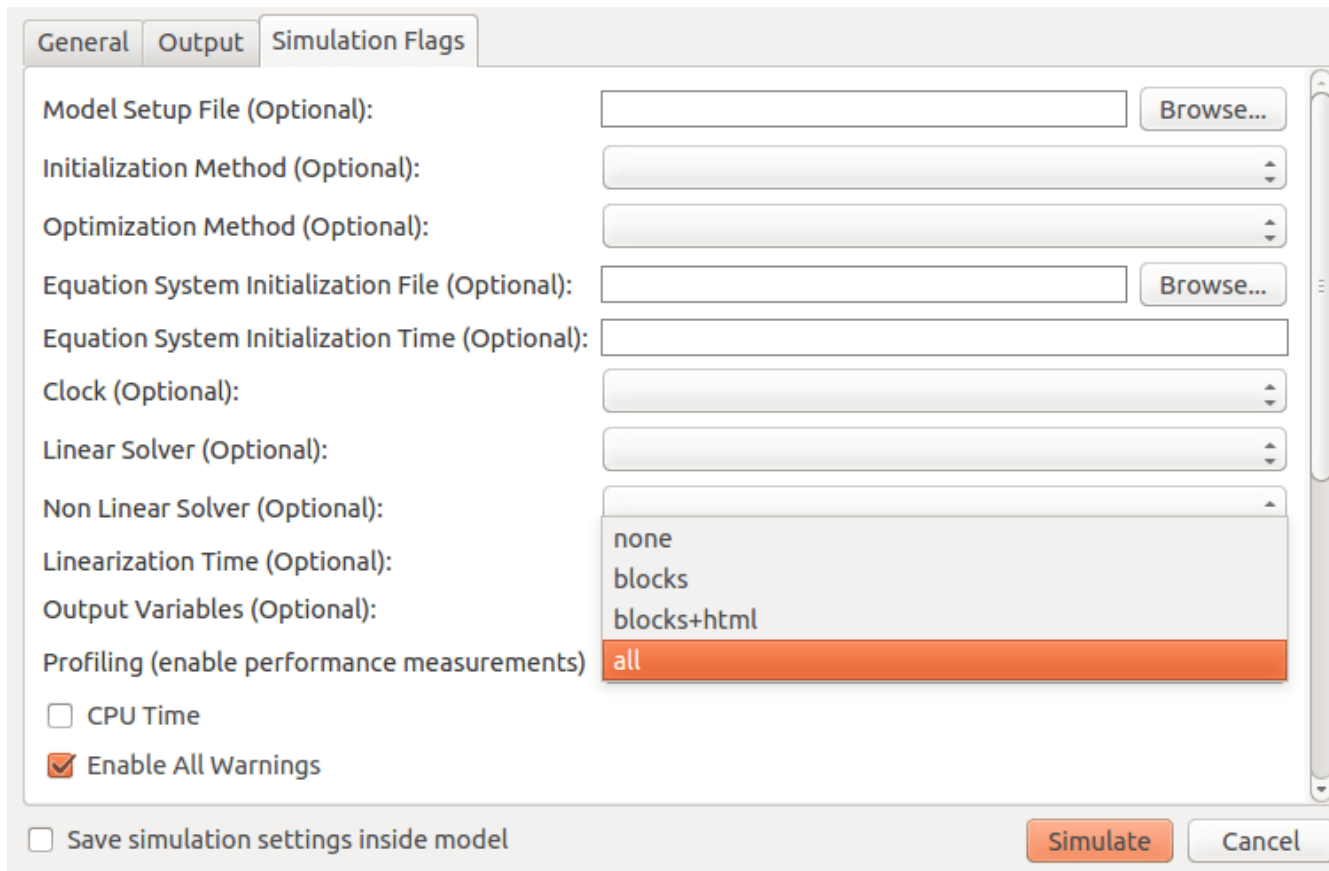


Conclusion from the evaluation:

“...the profiler makes the process of performance optimization radically shorter.”

Using the Performance Profiler on DoublePendulum

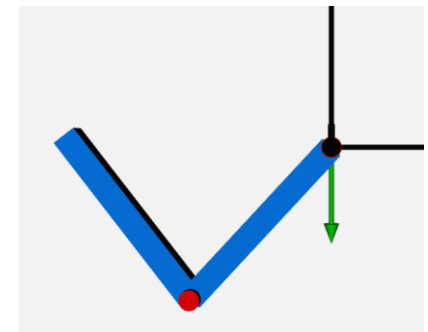
- When running a simulation from OMEdit, it is possible to enable profiling information, which can be combined with the [transformations browser](#).



Set this in
simulation
Setup



DoublePendulum in
MultiBody library



Using the Performance Profiler on the DoublePendulum model

- When profiling the DoublePendulum example from MSL, the following output below is a typical result. This information clearly shows which system takes longest to simulate (a linear system, where most of the time overhead probably comes from initializing LAPACK over and over).

Equations Browser							Defines
Index	Type	Equation	Executions	Max time	Time	Fraction ▲	Variable ▼
+ 876	regular	linear, size 2	4602	0.000199	0.0582	86.2%	damper.a_rel revolute2.frame_b.f[2]
- 836	regular	(assignment) revolute2.R_rel.T[2,2] = cos(revolute2.phi)	1534	8.25e-05	0.000491	0.728%	
- 837	regular	(assignment) revolute2.R_rel.T[2,1] = -sin(revolute2.phi)	1534	7.29e-05	0.000422	0.625%	
- 841	regular	(assignment) boxBody1.frame_...[2,1] = -sin(damper.phi_rel)	1534	7.1e-05	0.000395	0.585%	
- 840	regular	(assignment) boxBody1.frame_...T[2,2] = cos(damper.phi_rel)	1534	7.08e-05	0.000361	0.535%	
- 839	regular	(assignment) revolute2.R_rel.T[1,1] = cos(revolute2.phi)	1534	7.33e-05	0.000303	0.449%	
- 842	regular	(assignment) boxBody1.frame_b.R.T[1,2] = sin(damper.phi_rel)	1534	7.45e-05	0.000303	0.449%	
- 838	regular	(assignment) revolute2.R_rel.T[1,2] = sin(revolute2.phi)	1534	7.11e-05	0.0003	0.444%	
- 849	regular	(assignment) boxBody1.frame_...T[1,1] = cos(damper.phi_rel)	1534	7.29e-05	0.000286	0.424%	
- 827	regular	(assignment) revolute1.tau = (-damper.d) * revolute1.w	1534	6.84e-05	0.000274	0.406%	

Performance Profiler Exercise

- Try the profiler on this model. Results in Equations Browser, enlarge the window, click on Fraction to sort in ascending/descending order.

```

model ProfilingTest
  function f
    input Real r;
    output Real o = sin(r);
  end f;
  String s = "abc";
  Real x = f(x) "This is x";
  Real y(start=1);
  Real z1 = cos(z2);
  Real z2 = sin(z1);
equation
  der(y) = time;
end ProfilingTest;
  
```

Index	Type	Equation	Execution	Max time	Time	Fraction
21	regular	non-linear, unknown variables: 1	1006	7.24e-05	0.000664	44.6%
19	regular	non-linear (torn), unknown variables: 1	1006	7.87e-05	0.000635	42.7%
22	regular	(assign) der(y) := time	540	8e-07	1.71e-05	1.15%
9	initial	non-linear (torn), unknown variables: 1	8	5.3e-06	5.3e-06	0.356%
11	initial	non-linear, unknown variables: 1	3	1.5e-06	1.5e-06	0.101%
2	initial	(assign) s := "abc"	2	2e-07	2e-07	0.0134%
1	initial	(assign) y := \$START.y	1	1e-07	1e-07	0.00672%
12	initial	(alias) 22	0	0	0	0%
23	parameter	(alias) 2	0	0	0	0%