

# Integrated Modelica Debugging

Martin Sjölund

Programming Environments Laboratory (PELAB)  
Department of Computer and Information Science  
Linköping University

2014-02-03

# Debugging a Modelica model

- ▶ Many things influence simulation
- ▶ No unique translation of equations to executable code
- ▶ Selection of state variables
- ▶ Sorting, matching
- ▶ Numerical methods
- ▶ Mapping errors back to source code

## Creating a model with an error

```
model ChatteringEvents1 "Exhibits chattering after t =
    0.5, with generated events"
  Real x(start = 1, fixed = true);
  Real y;
  Real z;
equation
  z = if x > 0 then -1 else 1;
  y = 2 * z;
  der(x) = y;
  annotation(Documentation(info = "<html>
<p>After t = 0.5, chattering takes place, due to the
  discontinuity in the right hand side of the first
  equation.</p>
<p>Chattering can be detected because lots of tightly
  spaced events are generated. The feedback to the user
  should allow to identify the equation from which the
  zero crossing function that generates the events
  originates.</p>
</html>"), experiment(StopTime = 1));
end ChatteringEvents1;
```

## Simulating models with errors

- ▶ No error-message
- ▶ Cryptic error-message
- ▶ Simulation progressing slowly
  - ▶ Even fixed-step Euler solver is affected!
- ▶ Debugging is time-consuming



## Simulating models with errors

- ▶ No error-message
- ▶ Cryptic error-message
- ▶ Simulation progressing slowly
  - ▶ Even fixed-step Euler solver is affected!
- ▶ Debugging is time-consuming
- ▶ **You might not even find the error**



## Compiler generates XML of performed transformations

```
<solved>
  <lhs>revolute2.frame_b.t[2]</lhs>
  <rhs>-0</rhs>
</solved>
<simplify>
  <before>-revolute2.frame_b.t[2] = 0.2929079999999999 *
    0.0 + 0.008315999999999999 * boxBody2.body.w_a[3] *
    0.0 + 0.25 * -0 + -0.2929079999999999 * 0.0 *
    boxBody2.body.w_a[3]</before>
  <after>-revolute2.frame_b.t[2] = 0.0</after>
</simplify>
<!-- most things snipped -->
<flattening>
  <original>frame_a.t = I * z_a + cross(w_a, I * w_a) +
    cross(r_CM, frame_a.f);</original>
  <flattened>boxBody2.body.frame_a.t[2] = boxBody2.body.I
    [2,1] * ... - boxBody2.body.r_CM[1] * boxBody2.body.
    frame_a.f[3];</flattened>
</flattening>
```

## Simulating a model with an error

Running Simulation of Debugging.Chattering.ChatteringEvents1.  
Please wait for a while.



Cancel Simulation

OMEdit - Debugging.Chattering.ChatteringEvents1 Simulation Output

Output | Compilation

```
/tmp/OpenModelica/OMEdit/Debugging.Chattering.ChatteringEvents1 -  
port=50212 -logFormat=xml -w -lv=LOG_STATS  
stdout | info | Chattering detected around time  
0.500000005..0.500000995001 (100 state events in a row with a total time  
delta less than the step size 0.002). This can be a performance  
bottleneck. Use -lv LOG_EVENTS for more information. The zero-crossing  
was: x > 0.0 Debug more
```



# The Transformations Browser

The screenshot displays the OMEdit - Transformational Debugger interface. The main window title is "OMEdit - Transformational Debugger" and the file path is "/tmp/OpenModelica/OMEdit/Debugging.Chattering.ChatteringEvents1\_info.xml".

**Variables Browser:** Shows a list of variables with their comments and line numbers. Variable **z** is highlighted in orange, corresponding to line 9.

Variables	Comment	Line
x		7
y		8
z		9

**Defined In Equations:** A table showing the origin of variables. Variable **z** is defined in equation 4, which is highlighted in orange.

In /	Type	Equation
-1	initial	(assignm...lse 1.0
4	regular	(assignm...lse 1.0

**Used In Equations:** A table showing where variables are used. Variable **z** is used in equations 2 and 5.

In /	Type	Equation
-2	initial	(assignm...2.0 * z
5	regular	(assignm...2.0 * z

**Equations Browser:** A table listing all equations. Equation 4 is highlighted in orange.

In /	Type	Equation
-1	initial	(assignm...else 1.0
-2	initial	(assignm... 2.0 * z
-3	initial	(assignm...r(x) = y
4	regular	(assignm... else 1.0
5	regular	(assignm... 2.0 * z
6	regular	(assignm...r(x) = y

**Source Browser:** Displays the source code for the package Chattering. The code is as follows:

```
4 package Chattering
5 "Models with
6 chattering behaviour"
7 model
8 ChatteringEvents1
9 "Exhibits
10 chattering after t =
11 0.5, with generated
12 events"
13 Real x(start=1,
14 fixed=true);
15 Real y;
16 Real z;
17 equation
18 z = if x > 0
19 then -1 else 1;
20 y = 2*z;
21 der(x) = y;
22 annotation
23 (Documentation(info="
24 <html>
25 <p>After t = 0.5,
26 chattering takes
27 place, due to the
28 discontinuity in the
29 right hand side of
30 the first
31 equation.</p>
32 <p>Chattering can be
```



# The Transformations Browser (Overview)

The screenshot displays the OMEGA - Transformational Debugger interface, which is used for analyzing and transforming mathematical models. The interface is divided into several panes:

- Variables Browser:** Shows a list of variables defined in equations. The table below summarizes the variables shown in the image.
- Equations Browser:** Shows a list of equations defined in the model. The table below summarizes the equations shown in the image.
- Source Browser:** Shows the source code of the model, including package declarations, model definitions, and equations.

Eq.	Type	Equation
-1	initial	(assignment) z = f. 0.0 then -1 else 1.0
-2	initial	(assignment) y = 2.0 * z
-3	initial	(assignment) der(x) = y
1	regular	(assignment) z = f. 0.0 then -1 else 1.0
2	regular	(assignment) y = 2.0 * z
3	regular	(assignment) der(x) = y

  

Eq.	Type	Equation
-1	initial	(assignment) z = f.x = 0.0 then -1 else 1.0
-2	initial	(assignment) y = 2.0 * z
-3	initial	(assignment) der(x) = y
1	regular	(assignment) z = f.x = 0.0 then -1 else 1.0
2	regular	(assignment) y = 2.0 * z
3	regular	(assignment) der(x) = y

```
package Chattering "Models with chattering behaviour"
model ChatteringEvents1
  "Exhibits chattering after t = 0.5, with generated events"
  Real x(start=1, fixed=true);
  Real y;
  Real z;
  equation
  11 z = if x > 0 then -1 else 1;
  12 y = 2*z;
  13 der(x) = y;
  14 annotation (Documentation(info="<html>
  15 <p>After t = 0.5, chattering takes place, due to the discontinuity in the right
  16 hand side of the first equation.</p>
  17 <p>Chattering can be detected because lots of tightly spaced events are generated.
  18 The feedback to the user should allow to identify the equation from which the zero
  19 crossing function that generates the events originates.</p>
  20 </html>"), experiment(StopTime=1));
  end ChatteringEvents1;
  model ChatteringEvents2
  21 "Exhibits chattering after t = 0.422, with generated events"
  22 Real x(start=1, fixed=true);
  23 Real w(start=0, fixed=true);
  24 Real y;
  25 Real z;
  26 equation
  27 der(w) = -w + 1;
  28 z = if x > 0 then -1 else 1;
  29 y = 2*(z - w);
  30 der(x) = y;
  31 annotation (Documentation(info="<html>
  32 <p>After t = 0.5, chattering takes place, due to the discontinuity in the right
  33 hand side of the second equation.</p>
  34 <p>Chattering can be detected because lots of tightly spaced events are generated.
  35 The feedback to the user should allow to identify the equation from which the zero
  36 crossing function that generates the events originates.</p>
  37 </html>"), experiment(StopTime=1));
  end ChatteringEvents2;
  model ChatteringNoEvents1
  38 "Exhibits chattering after t = 0.5, without generated events"
  39 Real x(start=1, fixed=true);
  40 Real y;
  41 Real z;
  42 equation
  43 z = noEvent(if x > 0 then -1 else 1);
  44 y = 2*z;
  45 der(x) = y;
  46 annotation (Documentation(info="<html>
  47 <p>After t = 0.5, chattering takes place, due to the discontinuity in the right
  48 hand side of the first equation. The discontinuity does not generate state events,
  49 thanks to the noEvent operator.</p>
  50 <p>If a variable-step-size integration algorithm with error control is used, the
  51 time step will be reduced to very small values once the discontinuity is hit, and
  52 this can be detected by monitoring the value of time at each time step.</p>
  53 <p>Variable step size solvers usually allow to identify which state variables(s)
  54 are the cause of the discontinuity.</p>
  55 </html>"));
  end ChatteringNoEvents1;
```

# The Transformations Browser (Nonlinear system)

Equations		
Equations Browser		
Ind /	Type	Equation
-16	regular	(assignment) $p1 = dp\_valve + p2$
-17	regular	(assignment) $\text{sqrt\_dp} = \text{DIVISION}(dp\_valve, (dp\_valve \wedge 2.0 + \dots k/\text{testsuite}/\text{openmodelica}/\text{debugging}/\text{Debugging.mo Line: 145})$
-18	regular	(assignment) $w\_pump = Kv * \text{sqrt\_dp}$
-19	regular	(assignment) $dp\_pump = p1 - patm$
-20	regular	(residual) $dp\_pump + a1 * w\_pump \wedge 2.0 - dp0$
21	regular	(nonlinear)
...	regular	(assignment) $p1 = dp\_valve + p2$
...	regular	(assignment) $\text{sqrt\_dp} = \text{DIVISION}(dp\_valve, (dp\_valve \wedge 2.0 + \dots k/\text{testsuite}/\text{openmodelica}/\text{debugging}/\text{Debugging.mo Line: 145})$
...	regular	(assignment) $w\_pump = Kv * \text{sqrt\_dp}$
...	regular	(assignment) $dp\_pump = p1 - patm$
...	regular	(residual) $dp\_pump + a1 * w\_pump \wedge 2.0 - dp0$
-22	regular	(assignment) $W = a2 + a3 * w\_pump$
-23	regular	(assignment) $h1 = \text{DIVISION}(W - (-w\_pump) * h0, w\_pump, \#\text{SHARED\_LITERAL\_2}(\text{String})\#)$
-24	regular	(assignment) $T1 = \text{DIVISION}(h1 - (-cp) * Tref, cp, \#\text{SHARED\_LITERAL\_3}(\text{String})\#)$
-25	regular	(assignment) $\text{der}(y) = \text{DIVISION}(w\_extra + w\_pump, \rho * A, \#\text{SHARED\_LITERAL\_4}(\text{String})\#)$
-26	regular	(assignment) $\eta = dp\_pump * \text{DIVISION}(w\_pump, W * \rho, \#\text{SHARED\_LITERAL\_5}(\text{String})\#)$
-27	par...ter	(assignment) $\tau = 1.0$
-28	par...ter	(assignment) $h0 = cp * (T0 - Tref)$
-29	par...ter	(statement) $\text{assert}(Tref \geq 0.0, \text{"Variable Tref out of [min, max] interval: Tref} \geq 0.0 \text{ has value: " + String}(Tref, "g"))$ ;
-30	par...ter	(statement) $\text{assert}(\rho \geq 0.0, \text{"Variable rho out of [min, max] interval: rho} \geq 0.0 \text{ has value: " + String}(\rho, "g"))$ ;
-31	par...ter	(statement) $\text{assert}(T0 \geq 0.0, \text{"Variable T0 out of [min, max] interval: T0} \geq 0.0 \text{ has value: " + String}(T0, "g"))$ ;
-32	par...ter	(statement) $\text{assert}(T1 \geq 0.0, \text{"Variable T1 out of [min, max] interval: T1} \geq 0.0 \text{ has value: " + String}(T1, "g"))$ ;

## Integration of Tools

- ▶ OpenModelica Compiler, info.xml files
- ▶ Simulation executables, xml output format
- ▶ OMEdit, reads simulation output and info.xml. Generates and opens the correct view

## Future Work

- ▶ More specialised views (nonlinear/etc)
- ▶ Better diffs (colour-coded/etc)
- ▶ Trace more operations (tearing/etc)
- ▶ Integration with plotting / result-files
- ▶ Integration with the profiler

