

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) y = 2.0 * z
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

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

Equations

Equations Browser

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

Depends

Variable
x

Equation Operations

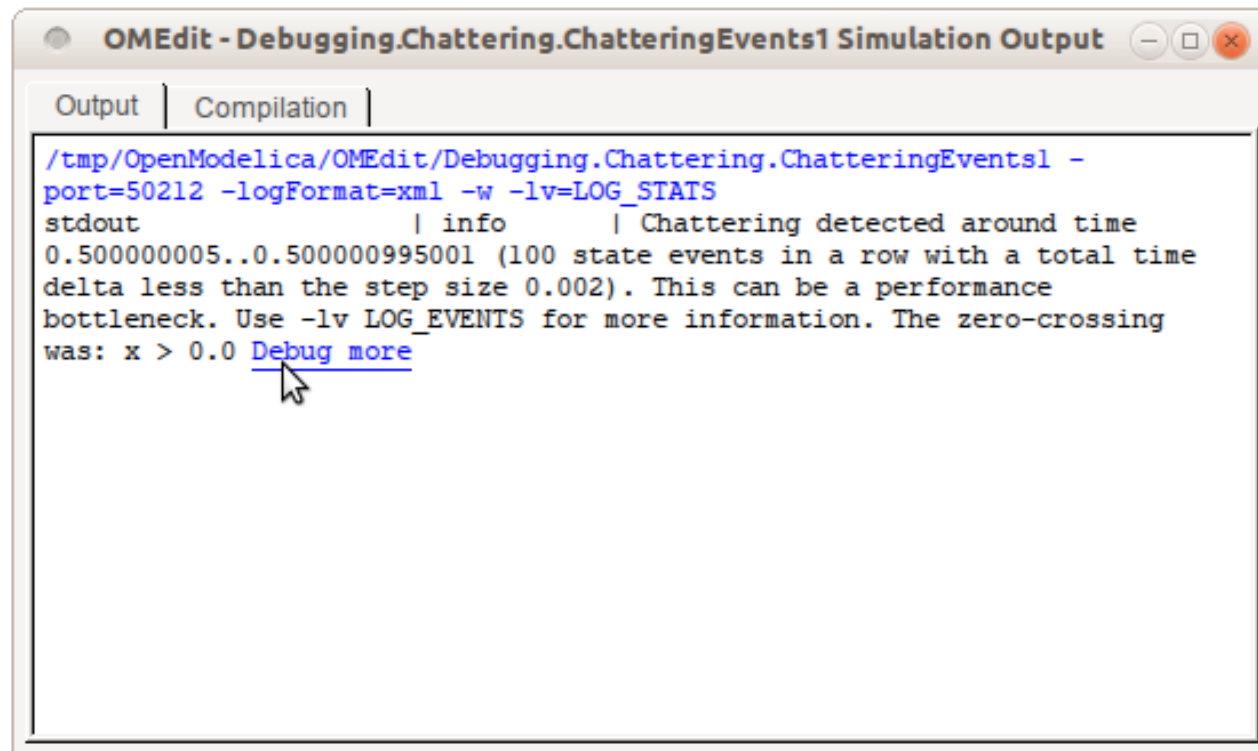
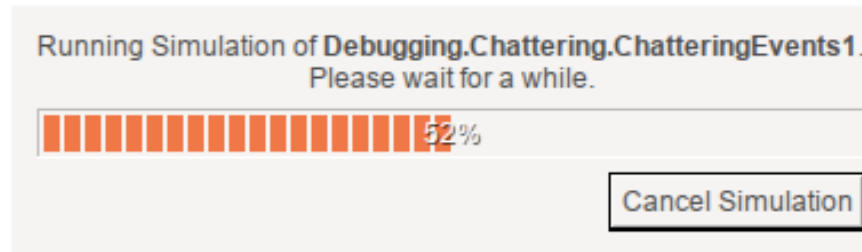
Operations

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;

equation

z = if x > 0 then -1 else 1;
y = 2*z;

Error Indication – Simulation Slows Down

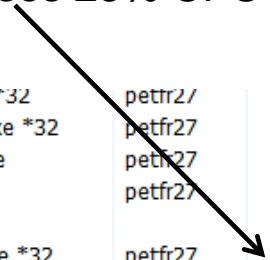


Exercise – Equation-based Model Debugger

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
- Click on the button Debug more (see prev. slide)
- Possibly start task manager and look at CPU. Then click stop simulation button