### Integrated Modelica Debugging

Martin Sjölund

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

2014-02-03

```
model ChatteringEvents1 "Exhibits chattering after t = 0.5
             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>
After t = 0.5, chattering takes place, due to the disconductory of th
Chattering can be detected because lots of tightly space
</html>"), experiment(StopTime = 1));
end ChatteringEvents1;
```

# 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 v:
 Real z;
equation
  z = if x > 0 then -1 else 1;
  y = 2 * z;
 der(x) = y;
  annotation(Documentation(info = "<html>
After t = 0.5, chattering takes place, 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 feedback to the user
    should allow to identify the equation from which the
    zero crossing function that generates the events
    originates. 
</html>"), experiment(StopTime = 1));
end ChatteringEvents1;
```

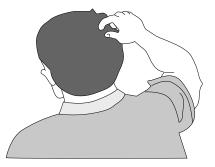
# Simulating models with errors

- 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

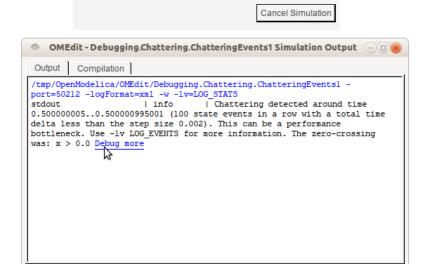


Compiler generates XML of performed transformations

```
<solved>
 <lhs>revolute2.frame_b.t[2]</lhs>
 < rhs > -0 < /rhs >
</solved>
<simplify>
 <before>-revolute2.frame b.t[2] = 0.2929079999999999 *
     0.0 + 0.00831599999999999 * boxBody2.body.w_a[3] *
      0.0 + 0.25 * -0 + -0.292907999999999 * 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.



# The Transformations Browser (Equations)

Transformations B	rowser 🗗	×				
Variables View	Equation View /tmp/OpenModelica/OMEdit/Debugging.Chattering.ChatteringEvents1_info.xml					
Equation Index: Search						
Defines	Depends Equation Operations					
Variable 🛆	Variable         Operations           x         solved: z = if x > 0.0 then -1 else 1.0					
	- original: z = if x > 0 then -1 else 1; => flattened:z = if x > 0.0 then -1 else 1.0;					
Operation : sol	ved: z = if x > 0.0 then -1 else 1.0 $$\clubsuit$$					

# The Transformations Browser (Variables)

Transformations Brows	ser			₽×			
Variables View Equation View /tmp/OpenModelica/OMEdit/Debugging.Chattering.ChatteringEvents1_info.xml							
Variables	Variable Types	Variable Operations	Defined In Equations	Used In Equations			
Variable 🛆 Cor	Types	Operations	Index 🛆 Typ	e Index 🛆 Type			
der(x)	L Real		-1 initia	al – 2 initial			
x			4 regu	ula 5 regular			
у							
z							
<u>↓</u> →			13-				
Index : 4							
Type : regular Equation : (assignment) z = if x > 0.0 then -1 else 1.0							
				<b>v</b>			

### 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