# “D6.3.2 – Prototype and Evaluation of an Efficient Integrated Debugger in OpenModelica”

**“Sub WP 6.3: Debugging and performance profiling of system models in operation”**

**“Work Package 6: Modeling and simulation services”**

**MODRIO (11004)**

| | |
|---|---|
| **Version** | 1.0 |
| **Date** | 2016-04-26 |

This document: Evaluation of the Integrated Debugger

**Authors**

| | |
|---|---|
| Åke Kinnander | Siemens TU |

## Summary

The goal of this task is to investigate and evaluate the OpenModelica (OM) integrated debugger. The testing has been hampered by the fact that the algorithmic code debugger was not possible to run in OM version 1.9.6 under Windows 10. (This was working in the 1.9.3 release but is apparently temporarily out of order). The following conclusion were made from the test with the transformational debugger for equation models:

1. The debugger works well to find zero denominators that are parameters.

2. The debugger does not come into play automatically if the zero denominator is only a momentarily value, as the solver managed work around such time points in so far tested simulations. However, it catches the problem in the simulation output window (SOW), and gives a message that by clicking opens the TDW which displays the concerned equation. However, there is a risk that this is unnoticed as the solver continues and could generate a lot of consequential or other messages that could hide the zero denominator messages. It would be preferable if the SOW could aggregate messages of the same type into one, expandable, line, thereby giving a better overview of the all the types of messages the simulation has generated.

3. A zero denominator caused by structural model errors, like connection to not used connectors (this should not pass the model checking) the debugger points to the causing equation. One could not ask more of the transformational debugger, but the OM model check or model building could be made to prevent such mistakes.

4. At numerical problems causing long execution times the debugger points to the equations that has problems, but to understand the exact problem, plots of variables could be necessary – hence the result file should always be generated, regardless if the simulation is interrupted by solver or manually. This is not the case in the tested version for all the tests.

## Glossary

| | |
|---|---|
| OM | OpenModelica |
| PHF | Prescribed Heat Flow |
| SOW | Simulation Output Window |
| TDW | Transformational Debugger Window |

**Contents**

# 1 Introduction

In order to investigate different types of errors that could be expected to occur, a small and simple evaporator model is used. This is fetched from a larger model used for transient analysis of combined power plants by Siemens Industrial Turbomachinery AB in Finspång. The following errors are to be investigated:

1. Division by zero
2. Errors in the average logarithmic temperature difference used for heat transfer calculation:
    a. Inlet temperature difference =0
    b. Inlet temperature difference=outlet temperature difference
3. Boiling in the evaporator that causes halt of simulation progress by much too small time steps
4. Various test of bad initial values, with variation of pressure, temperatures, flows and masses in the different parts of the process.

The model selected is a simplified model of an error free model, hence the above test will be deliberately inserted and the debugging tool will only be examined by its outputs, while a sharper application for a real model development where errors are unknown and the debugger support for identifying them will be more apparent, will be carried out later. The reason for this is the limited time for testing available, and that a sharp application will only provide stochastic errors and could thereby not be planned in time.

# 2 Model for the evaluation

The following model is to be used for the investigation (Figure 1):

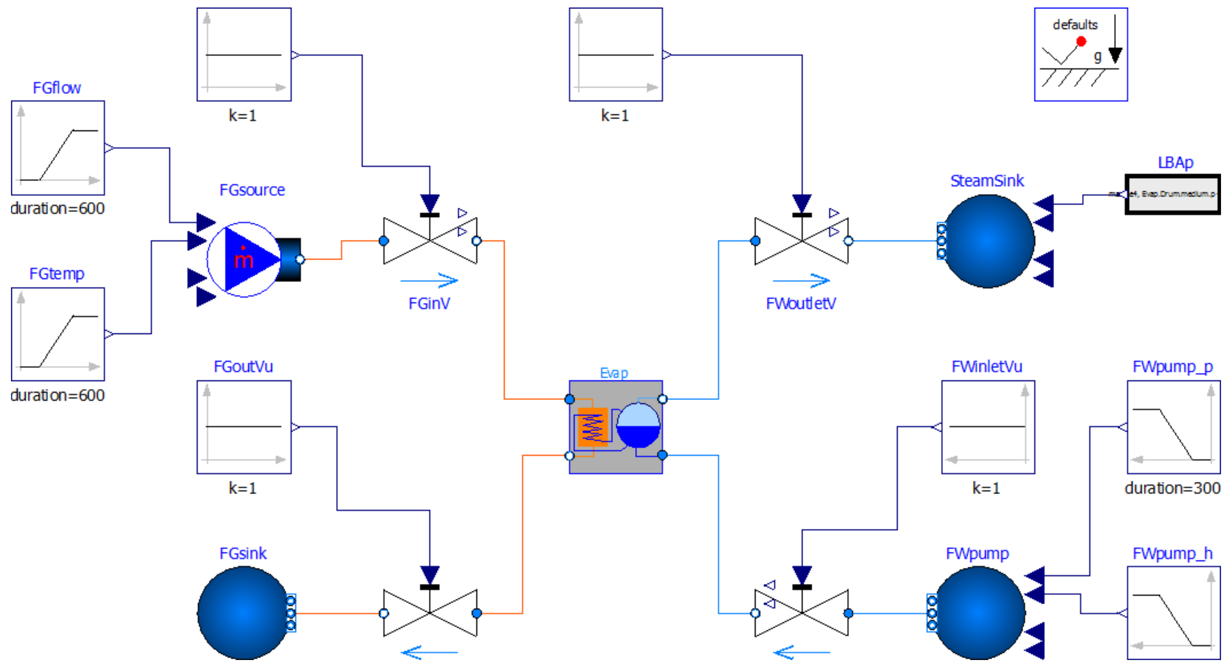


**Figure 1. Evaporator test model**

It consists of an evaporator model that has flue gases as heating source and water as coolant, producing dry steam to the steam sink. The steam production is decided by the heat from the flue gases, the enthalpy (temperature and pressure) of the water source (FWpump), and the steam extraction to the steam sink (SteamSink) that in turn is tracking the evaporators drum pressure with a negative bias of 0.1 to 1 bar.

The model has 1110 equations.

The evaporator model is designed according to Figure 2.

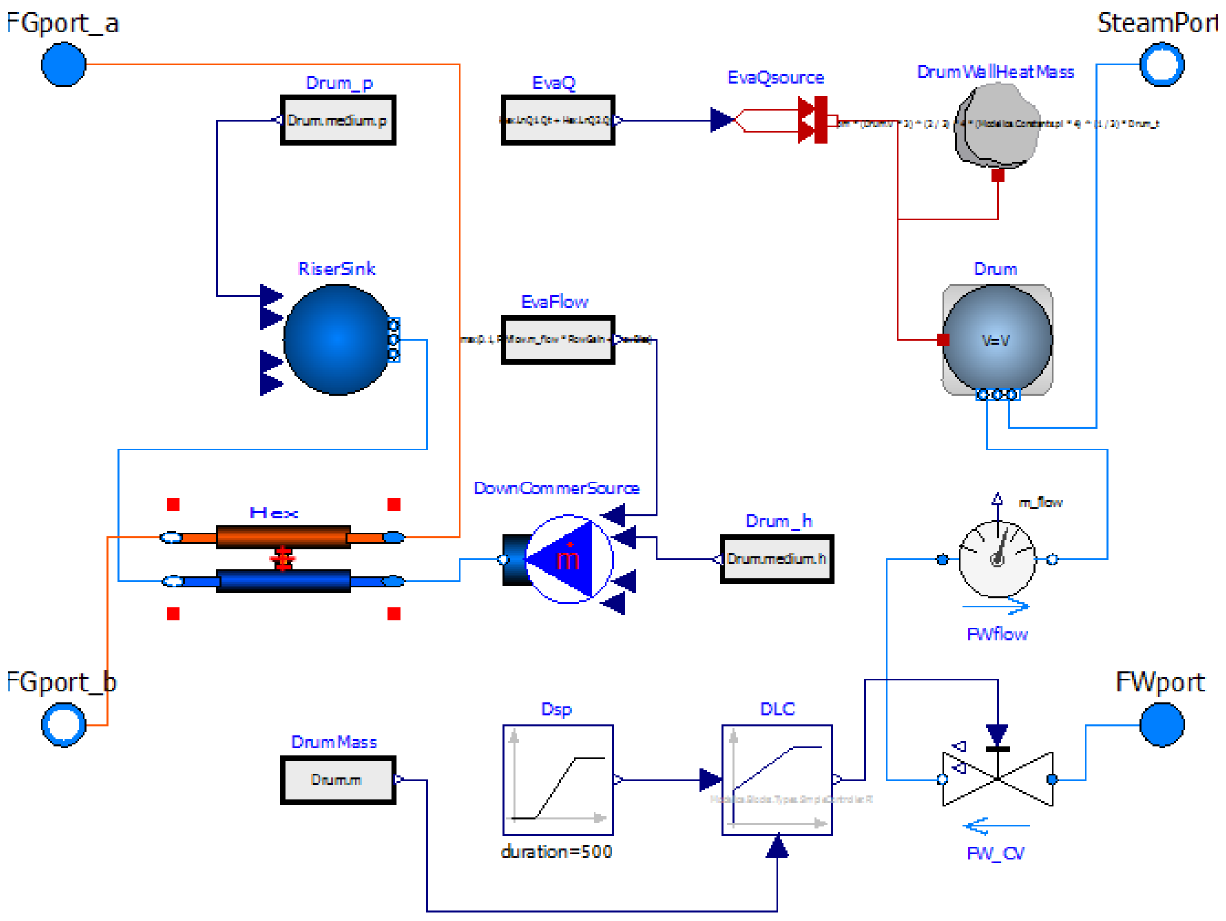


**Figure 2. Evaporator model**

The evaporator model has a drum model (Drum), a heat exchanger (Hex) and a level controller (DLC) that controls level by a control valve (FW_CV) . The level controller is no actual water level controller in length units (m), instead it controls the amount of water (kg) in the drum. The thermal capacity of the drum metals is represented of a heat capacity model (DrumWallHeatMass) but the insulation towards surroundings are assumed to be perfect, i.e. no heat losses to the outside of the drum.

This model has 809 equations.

The drum model is the volume model from the Fluid library manipulated to only let steam exit, i.e. always perfect separation.

The heat exchanger is as according to Figure 3.

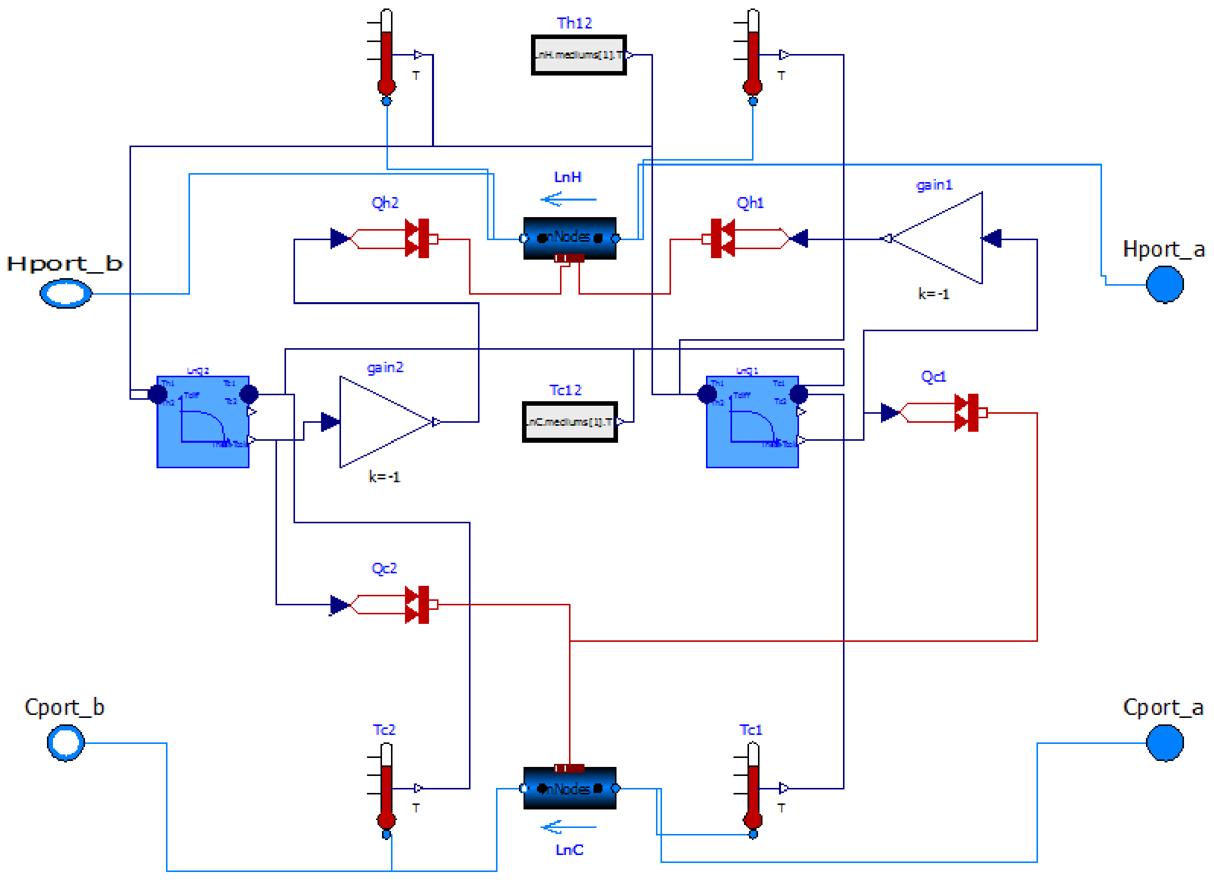


**Figure 3. Heat exchanger**

The heat exchanger has a pipe model (LnH) that corresponds to the flue gas channel and a pipe model (LnC) that corresponds to the pipe bundle carrying the water to be heated. The heat exchanger has parallel flows. In this simplified model the overall heat transfer coefficient (J/m$^2$/K) is a constant, i.e. it takes not configuration or medium properties into account. The driving temperature difference is calculated for respectively inlet and outlet sections of the pipe bundles (i.e. they configured with 2 nodes each) by the models LnQ1 and LnQ2. The temperatures are measured besides inlets and outlets for both pipe models also in the middle (Tc12 and Th12).

This model has 580 equations.

Finally, the heat calculation models LnQ1 and 2 are according to Figure 4 (all other model are from standard Modelica library)

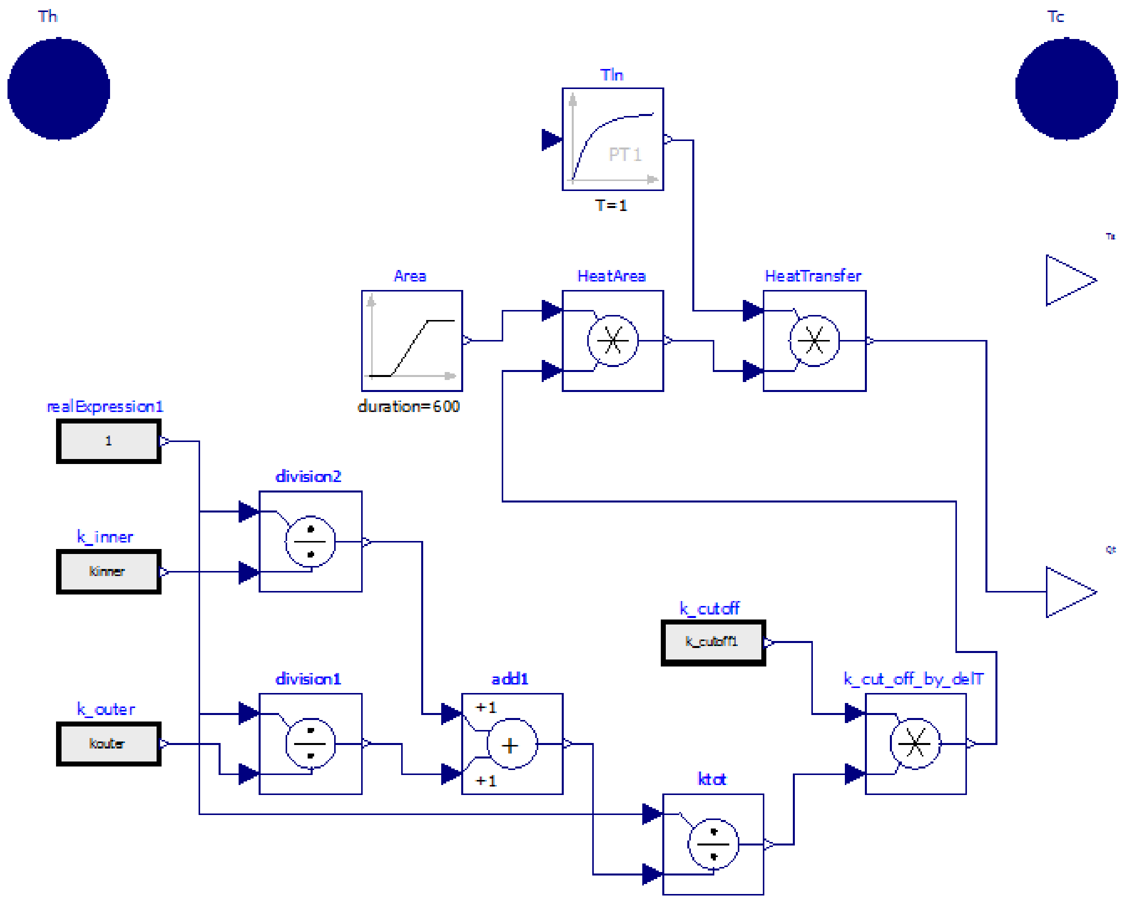


**Figure 4 Heat transfer calculation model**

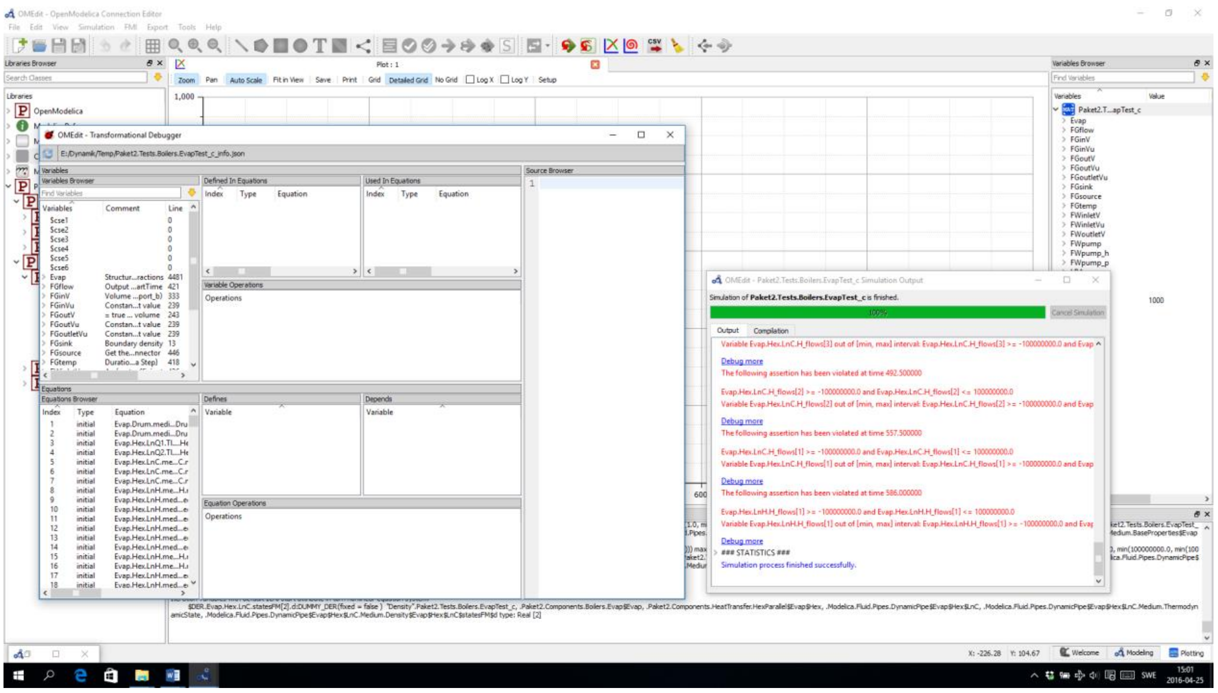
The model has low pass filter with an input connected in the text layer where the logarithmic temperature difference from the connectors Th and Tc which both are connecting both inlet and outlet temperatures respectively medium side, is calculated. The output of the model is the calculated heat transfer (W). The overall heat transfer coefficient ktot is calculated from parameters representing the heat transfer coefficient from flue gas to metal, k_outer, and from metal to water, k_inner. In the present full size boiler model those parameters are replaced with connectors that provides more accurate values, based on medium and flow properties calculated in separate models. The heat transfer area is a ramp with selectable duration and height values, to be adopted to what’s needed from initializing aspect (duration of suitable size respectively to the real heat transfer area).

This model contributes with 39 equations of the total of 1110.

# 3 Debugger tests

## 3.1 Activation of debugger

Debugger is activated by setting the flag « Launch transformational debugger ». After a successful simulation the output windows are containing following information (Figure 5).



```
OMEdit - Paket2.Tests.Boilers.EvapTest_c Simulation Output
Simulation of Paket2.Tests.Boilers.EvapTest_c is finished.
100%
Output   Compilation
Variable Evap.Hex.LnC.H_flows[3] out of [min, max] interval: Evap.Hex.LnC.H_flows[3] >= -100000000.0 and Evap
Debug more
The following assertion has been violated at time 492.500000
Evap.Hex.LnC.H_flows[2] >= -100000000.0 and Evap.Hex.LnC.H_flows[2] <= 100000000.0
Variable Evap.Hex.LnC.H_flows[2] out of [min, max] interval: Evap.Hex.LnC.H_flows[2] >= -100000000.0 and Evap
Debug more
The following assertion has been violated at time 557.500000
Evap.Hex.LnC.H_flows[1] >= -100000000.0 and Evap.Hex.LnC.H_flows[1] <= 100000000.0
Variable Evap.Hex.LnC.H_flows[1] out of [min, max] interval: Evap.Hex.LnC.H_flows[1] >= -100000000.0 and Evap
Debug more
The following assertion has been violated at time 586.000000
Evap.Hex.LnH.H_flows[1] >= -100000000.0 and Evap.Hex.LnH.H_flows[1] <= 100000000.0
Variable Evap.Hex.LnH.H_flows[1] out of [min, max] interval: Evap.Hex.LnH.H_flows[1] >= -100000000.0 and Evap
Debug more
### STATISTICS ###
Simulation process finished successfully.
```

**Figure 5 Information from OM with debugger activated at successful simulation**

The Simulation Output window (SOW) contains assertion violation messages that are false, because the enthalpy flow H (W) has too narrow range in the Standard Modelica library. It ought to be at least 10 times as big. This violation has no influence on the simulation result (might there be an unnecessary delay?). The window shows with a green bar that 100 % of simulation is done and the blue text that it has been successful.

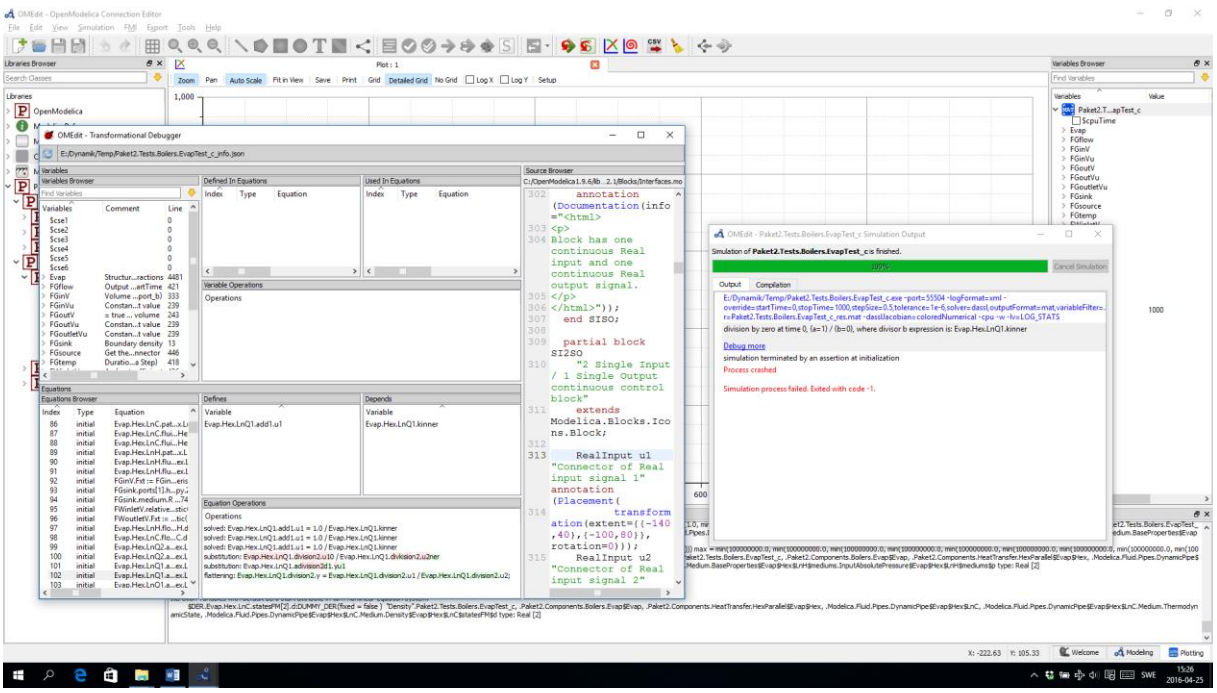
The debugger (transformational debugger window, TDW) shows all variables in the variable browsers window and all equations in the equation browser window, as found in the simulation code. All other frames in the debugger are empty.

## 3.2 Division by zero

### 3.2.1 By parameter setting

The test is done by setting parameter k_inner to zero.

The SOW displays following messages (Figure 6).



OMEdit - Paket2.Tests.Boilers.EvapTest_c Simulation Output

Simulation of **Paket2.Tests.Boilers.EvapTest_c** is finished.

```
E:/Dynamik/Temp/Paket2.Tests.Boilers.EvapTest_c.exe -port=55504 -logFormat=xml -
override=startTime=0,stopTime=1000,stepSize=0.5,tolerance=1e-6,solver=dassl,outputFormat=mat,variableFilter=.
r=Paket2.Tests.Boilers.EvapTest_c_res.mat -dasslJacobian=coloredNumerical -cpu -w -lv=LOG_STATS
division by zero at time 0, (a=1) / (b=0), where divisor b expression is: Evap.Hex.LnQ1.kinner

Debug more
simulation terminated by an assertion at initialization
Process crashed

Simulation process failed. Exited with code -1.
```

Defines — Variable: Evap.Hex.LnQ1.add1.u1

Depends — Variable: Evap.Hex.LnQ1.kinner

Equation Operations

```
solved: Evap.Hex.LnQ1.add1.u1 = 1.0 / Evap.Hex.LnQ1.kinner
solved: Evap.Hex.LnQ1.add1.u1 = 1.0 / Evap.Hex.LnQ1.kinner
solved: Evap.Hex.LnQ1.add1.u1 = 1.0 / Evap.Hex.LnQ1.kinner
substitution: Evap.Hex.LnQ1.division2.u10 / Evap.Hex.LnQ1.division2.u2ner
substitution: Evap.Hex.LnQ1.adivision2d1.yu1
flattening: Evap.Hex.LnQ1.division2.y = Evap.Hex.LnQ1.division2.u1 / Evap.Hex.LnQ1.division2.u2;
```

Figure 6 Division by zero by parameter setting

The Simulation Output window (SOW) gives the required information that simulation crashed at initialization due to an assertion that avoids division by zero and this is caused by k_inner=0.

The debugger window looks as before but after clicking debug more in the SOW it looks as in Figure 6.

The equation browser marks initial equation with index 102: Evap.Hex.LnQ1.add1.u1:= DIVISION(1.0, Evap.Hex.LnQ1.kinner), which is the same information as in SOW.

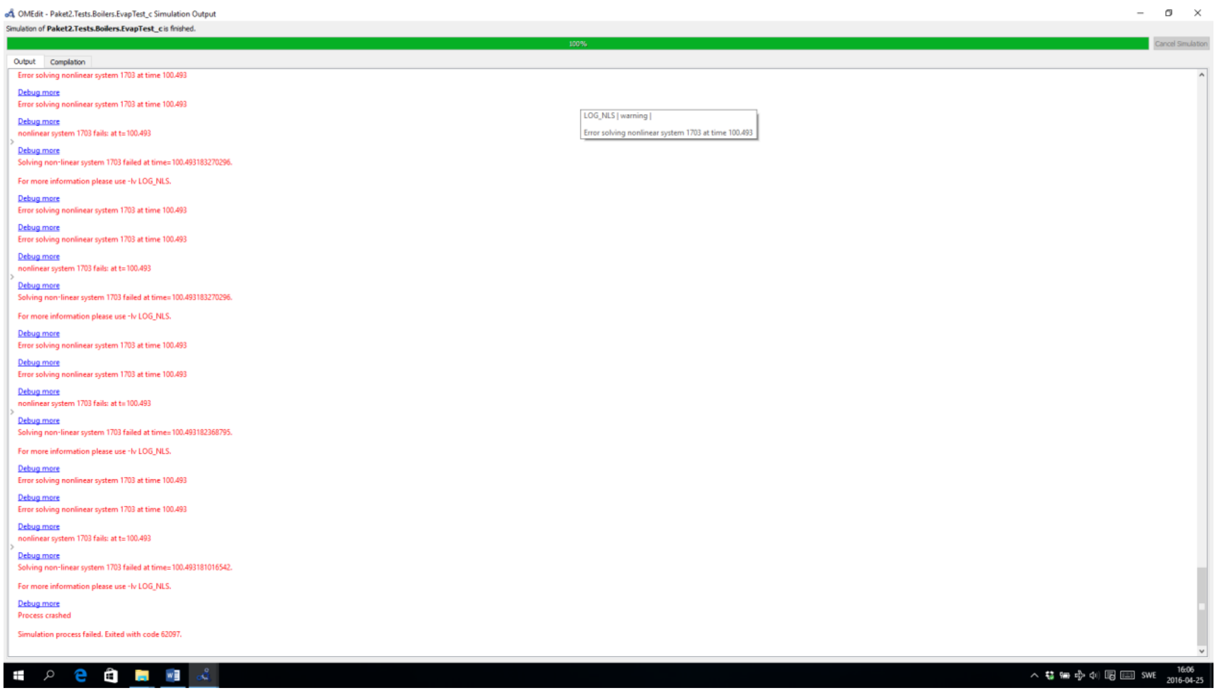
The frame denoted Defines give the variable that becomes undefined by the zero division. The frame denoted Depends give the variable name Evap.Hex.LnQ1.kinner.

**Conclusion**

For this error the debugger gives all information necessary.

### 3.2.2 By time function

The k_inner variable is replaced by a time function that ramps it down to zero in 100 seconds. This results in a never ending (?) simulation where SOW gives the messages in Figure 7.



OMEdit - Paket2.Tests.Boilers.EvapTest_c Simulation Output

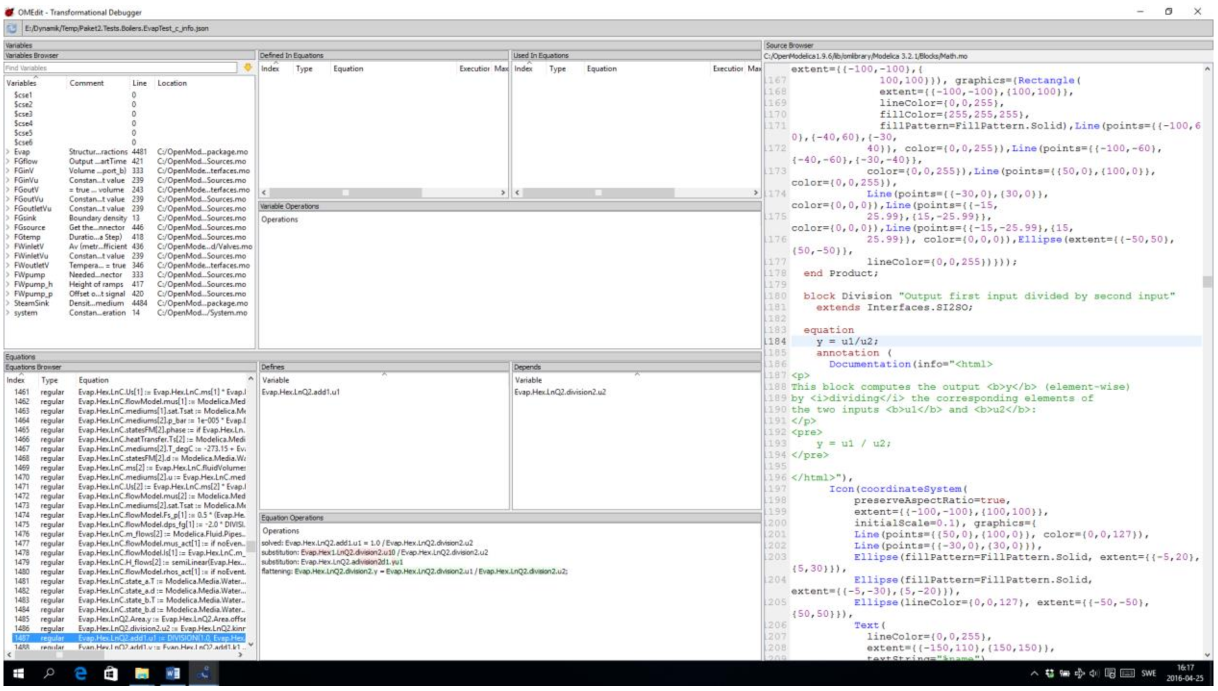
Simulation of **Paket2.Tests.Boilers.EvapTest_c** is finished.

```
Error solving nonlinear system 1703 at time 100.493
Debug more
Error solving nonlinear system 1703 at time 100.493
Debug more
nonlinear system 1703 fails: at t=100.493
Debug more
Solving non-linear system 1703 failed at time=100.493183270296.
For more information please use -lv LOG_NLS.
Debug more
Error solving nonlinear system 1703 at time 100.493
Debug more
Error solving nonlinear system 1703 at time 100.493
Debug more
nonlinear system 1703 fails: at t=100.493
Debug more
Solving non-linear system 1703 failed at time=100.493183270296.
For more information please use -lv LOG_NLS.
Debug more
Error solving nonlinear system 1703 at time 100.493
Debug more
Error solving nonlinear system 1703 at time 100.493
Debug more
nonlinear system 1703 fails: at t=100.493
Debug more
Solving non-linear system 1703 failed at time=100.493182368795.
For more information please use -lv LOG_NLS.
Debug more
Error solving nonlinear system 1703 at time 100.493
Debug more
Error solving nonlinear system 1703 at time 100.493
Debug more
nonlinear system 1703 fails: at t=100.493
Debug more
Solving non-linear system 1703 failed at time=100.493181016542.
For more information please use -lv LOG_NLS.
Debug more
Process crashed
Simulation process failed. Exited with code 62097.
```

```
LOG_NLS | warning |
Error solving nonlinear system 1703 at time 100.493
```

**Figure 7 SOW after manual interruption of simulation that has ramp that gives division by zero**

The solver manages to pass the 100 s time point where kinner is zero and a division by zero occurs. No plots are available but the ramp proceeds to negative values for kinner. The solver has skipped the exact 100 s time point, but then continued into other problems, due to the negative kinner value. On the passage it has however produced two messages about zero division at time 100 when they occurred. In the case of the user being unaware of the division problem, the large amount of output in SOW hides those messages. Scanning through the SOW window they are found and clicking them gives the TDW appearance according to Figure 8.



OMEdit - Transformational Debugger

E:/Dynamik/Temp/Paket2.Tests.Boilers.EvapTest_c_info.json

| Variables | Comment | Line | Location |
|---|---|---|---|
| $cse1 | | 0 | |
| $cse2 | | 0 | |
| $cse3 | | 0 | |
| $cse4 | | 0 | |
| $cse5 | | 0 | |
| $cse6 | | 0 | |
| Evap | Structur...ractions | 4481 | C:/OpenMod...package.mo |
| FGflow | Output ...artTime | 421 | C:/OpenMod...Sources.mo |
| FGinV | Volume ...port_b) | 333 | C:/OpenMode...terfaces.mo |
| FGinVu | Constan...t value | 239 | C:/OpenMod...Sources.mo |
| FGoutV | = true ... volume | 243 | C:/OpenMode...terfaces.mo |
| FGoutVu | Constan...t value | 239 | C:/OpenMod...Sources.mo |
| FGoutletVu | Constan...t value | 239 | C:/OpenMod...Sources.mo |
| FGsink | Boundary density | 13 | C:/OpenMod...Sources.mo |
| FGsource | Get the...nnector | 446 | C:/OpenMod...Sources.mo |
| FGtemp | Duratio...a Step) | 418 | C:/OpenMod...Sources.mo |
| FWinletV | Av (metr...fficient | 436 | C:/OpenMode...d/Valves.mo |
| FWinletVu | Constan...t value | 239 | C:/OpenMod...Sources.mo |
| FWoutletV | Tempera... = true | 346 | C:/OpenMode...terfaces.mo |
| FWpump | Needed...nector | 333 | C:/OpenMod...Sources.mo |
| FWpump_h | Height of ramps | 417 | C:/OpenMod...Sources.mo |
| FWpump_p | Offset o...t signal | 420 | C:/OpenMod...Sources.mo |
| SteamSink | Densit...medium | 4484 | C:/OpenMod...package.mo |
| system | Constan...eration | 14 | C:/OpenMod.../System.mo |

Defined In Equations

| Index | Type | Equation | Executior | Max |
|---|---|---|---|---|

Used In Equations

| Index | Type | Equation | Executior | Max |
|---|---|---|---|---|

Variable Operations

Operations

| Index | Type | Equation |
|---|---|---|
| 1461 | regular | Evap.Hex.LnC.Us[1] := Evap.Hex.LnC.ms[1] * Evap.I |
| 1462 | regular | Evap.Hex.LnC.flowModel.mus[1] := Modelica.Med |
| 1463 | regular | Evap.Hex.LnC.mediums[1].sat.Tsat := Modelica.Mo |
| 1464 | regular | Evap.Hex.LnC.mediums[2].p_bar := 1e-005 * Evap.I |
| 1465 | regular | Evap.Hex.LnC.statesFM[2].phase := if Evap.Hex.Ln. |
| 1466 | regular | Evap.Hex.LnC.heatTransfer.Ts[2] := Modelica.Medi |
| 1467 | regular | Evap.Hex.LnC.mediums[2].T_degC := -273.15 + Ev. |
| 1468 | regular | Evap.Hex.LnC.statesFM[2].d := Modelica.Media.Wa |
| 1469 | regular | Evap.Hex.LnC.ms[2] := Evap.Hex.LnC.fluidVolumes |
| 1470 | regular | Evap.Hex.LnC.mediums[2].u := Evap.Hex.LnC.med |
| 1471 | regular | Evap.Hex.LnC.Us[2] := Evap.Hex.LnC.ms[2] * Evap.I |
| 1472 | regular | Evap.Hex.LnC.flowModel.mus[2] := Modelica.Med |
| 1473 | regular | Evap.Hex.LnC.mediums[2].sat.Tsat := Modelica.Mo |
| 1474 | regular | Evap.Hex.LnC.flowModel.Fs_p[1] := 0.5 * (Evap.He. |
| 1475 | regular | Evap.Hex.LnC.flowModel.dps_fg[1] := -2.0 * DIVISI. |
| 1476 | regular | Evap.Hex.LnC.m_flows[2] := Modelica.Fluid.Pipes.. |
| 1477 | regular | Evap.Hex.LnC.flowModel.mus_act[1] := if noEven.. |
| 1478 | regular | Evap.Hex.LnC.flowModel.ls[1] := Evap.Hex.LnC.m_ |
| 1479 | regular | Evap.Hex.LnC.H_flows[2] := semiLinear(Evap.Hex... |
| 1480 | regular | Evap.Hex.LnC.flowModel.rhos_act[1] := if noEvent. |
| 1481 | regular | Evap.Hex.LnC.state_a.T := Modelica.Media.Water... |
| 1482 | regular | Evap.Hex.LnC.state_a.d := Modelica.Media.Water... |
| 1483 | regular | Evap.Hex.LnC.state_b.T := Modelica.Media.Water.. |
| 1484 | regular | Evap.Hex.LnC.state_b.d := Modelica.Media.Water.. |
| 1485 | regular | Evap.Hex.LnQ2.Area.y := Evap.Hex.LnQ2.Area.offse |
| 1486 | regular | Evap.Hex.LnQ2.division2.u2 := Evap.Hex.LnQ2.kinr |
| 1487 | regular | Evap.Hex.LnQ2.add1.u1 := DIVISION(1.0, Evap.Hex. |
| 1488 | regular | Evap.Hex.LnQ2.add1.y := Evap.Hex.LnQ2.add1.k1 .. |

Defines

| Variable |
|---|
| Evap.Hex.LnQ2.add1.u1 |

Depends

| Variable |
|---|
| Evap.Hex.LnQ2.division2.u2 |

Equation Operations

```
Operations
solved: Evap.Hex.LnQ2.add1.u1 = 1.0 / Evap.Hex.LnQ2.division2.u2
substitution: Evap.Hex1.LnQ2.division2.u10 / Evap.Hex.LnQ2.division2.u2
substitution: Evap.Hex.LnQ2.adivision2d1.yu1
flattening: Evap.Hex.LnQ2.division2.y = Evap.Hex.LnQ2.division2.u1 / Evap.Hex.LnQ2.division2.u2;
```

Source Browser

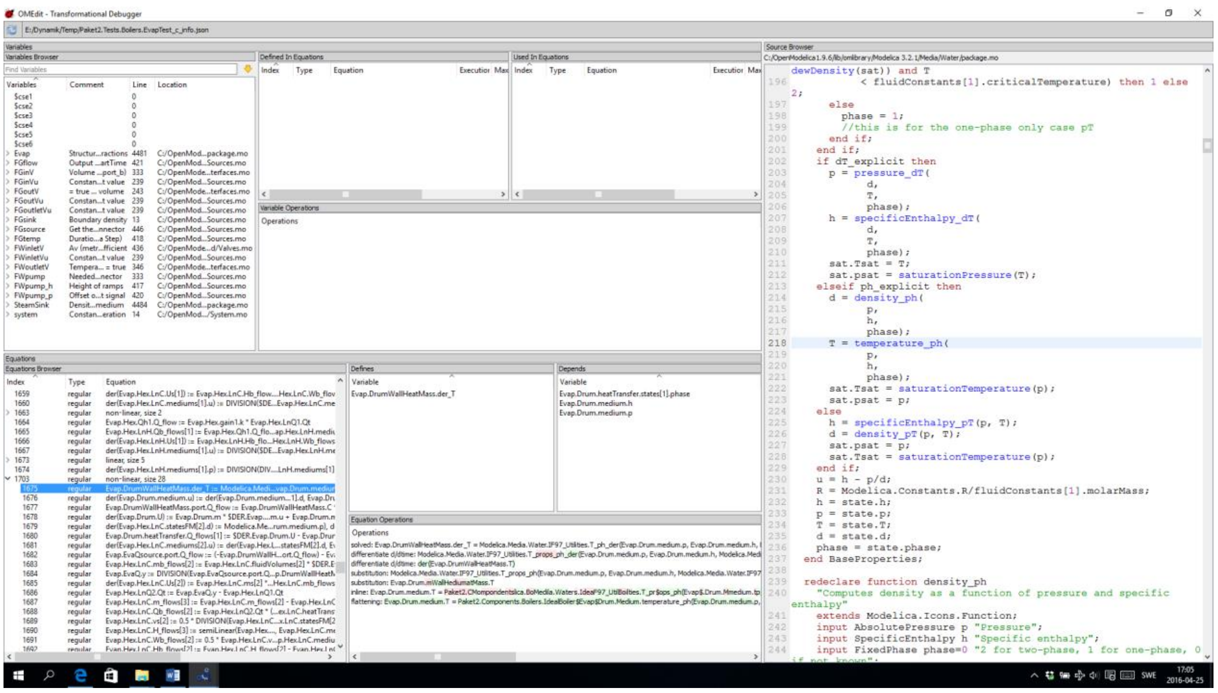
C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Blocks/Math.mo

```
extent={{-100,-100},{
                    100,100}}), graphics={Rectangle(
                    extent={{-100,-100},{100,100}},
                    lineColor={0,0,255},
                    fillColor={255,255,255},
                    fillPattern=FillPattern.Solid),Line(points={{-100,6
0},{-40,60},{-30,
                  40}}, color={0,0,255}),Line(points={{-100,-60},
{-40,-60},{-30,-40}},
                  color={0,0,255}),Line(points={{50,0},{100,0}},
color={0,0,255}),
                  Line(points={{-30,0},{30,0}},
color={0,0,0}),Line(points={{-15,
                  25.99},{15,-25.99}},
color={0,0,0}),Line(points={{-15,-25.99},{15,
                  25.99}}, color={0,0,0}),Ellipse(extent={{-50,50},
{50,-50}},
                  lineColor={0,0,255})}));
  end Product;

  block Division "Output first input divided by second input"
    extends Interfaces.SI2SO;

  equation
    y = u1/u2;
    annotation (
      Documentation(info="<html>
<p>
This block computes the output <b>y</b> (element-wise)
by <i>dividing</i> the corresponding elements of
the two inputs <b>u1</b> and <b>u2</b>:
</p>
<pre>
    y = u1 / u2;
</pre>

</html>"),
      Icon(coordinateSystem(
          preserveAspectRatio=true,
          extent={{-100,-100},{100,100}},
          initialScale=0.1), graphics={
          Line(points={{50,0},{100,0}}, color={0,0,127}),
          Line(points={{-30,0},{30,0}}),
          Ellipse(fillPattern=FillPattern.Solid, extent={{-5,20},
{5,30}}),
          Ellipse(fillPattern=FillPattern.Solid,
extent={{-5,-30},{5,-20}}),
          Ellipse(lineColor={0,0,127}, extent={{-50,-50},
{50,50}}),
          Text(
            lineColor={0,0,255},
            extent={{-150,110},{150,150}},
```

**Figure 8 TDW at ramp to zero of denominator**

Again this is an exact pointer to the error.

A problem is that without this knowledge the risk that the user will miss the division by zero due to the massive output of messages concerning a consecutive problem (Figure 7). An estimation is that the messages from time 100 to 100.493 are about 50 and seemed not to cease or automatically interrupt simulation, not at least within authors frame of patience. After manual interruption there are no stored variables to plot. Clicking the latest Debug More message gives a pointer into TDW to equation with index 1703, and a message, in the Defines frame, that derivatives of drum pressure and enthalpy and water pipes outlet volumes enthalpy is calculated. Equation 1703 has order 28, i.e. it consists of 28 other nonlinear equations with indexes ranging from 1675 to 1702. Clicking on of them changes the TDW (Figure 9).


OMEdit - Transformational Debugger

E:/Dynamik/Temp/Paket2.Tests.Boilers.EvapTest_c_info.json

Variables Browser

| Variables | Comment | Line | Location |
|---|---|---|---|
| $cse1 | | 0 | |
| $cse2 | | 0 | |
| $cse3 | | 0 | |
| $cse4 | | 0 | |
| $cse5 | | 0 | |
| $cse6 | | 0 | |
| Evap | Structur...ractions | 4481 | C:/OpenMod...package.mo |
| FGflow | Output ...artTime | 421 | C:/OpenMod...Sources.mo |
| FGinV | Volume ...port_b) | 333 | C:/OpenMode...terfaces.mo |
| FGinVu | Constan...t value | 239 | C:/OpenMod...Sources.mo |
| FGoutV | = true ... volume | 243 | C:/OpenMode...terfaces.mo |
| FGoutVu | Constan...t value | 239 | C:/OpenMod...Sources.mo |
| FGoutletVu | Constan...t value | 239 | C:/OpenMod...Sources.mo |
| FGsink | Boundary density | 13 | C:/OpenMod...Sources.mo |
| FGsource | Get the...nnector | 446 | C:/OpenMod...Sources.mo |
| FGtemp | Duratio...a Step) | 418 | C:/OpenMod...Sources.mo |
| FWinletV | Av (metr...fficient | 436 | C:/OpenMode...d/Valves.mo |
| FWinletVu | Constan...t value | 239 | C:/OpenMod...Sources.mo |
| FWoutletV | Tempera... = true | 346 | C:/OpenMode...terfaces.mo |
| FWpump | Needed...nector | 333 | C:/OpenMod...Sources.mo |
| FWpump_h | Height of ramps | 417 | C:/OpenMod...Sources.mo |
| FWpump_p | Offset o...t signal | 420 | C:/OpenMod...Sources.mo |
| SteamSink | Densit...medium | 4484 | C:/OpenMod...package.mo |
| system | Constan...eration | 14 | C:/OpenMod.../System.mo |

Defined In Equations

| Index | Type | Equation | Executior | Max |
|---|---|---|---|---|

Used In Equations

| Index | Type | Equation | Executior | Max |
|---|---|---|---|---|

Variable Operations

Operations

Equations Browser

| Index | Type | Equation |
|---|---|---|
| 1659 | regular | der(Evap.Hex.LnC.Us[1]) := Evap.Hex.LnC.Hb_flow....Hex.LnC.Wb_flov |
| 1660 | regular | der(Evap.Hex.LnC.mediums[1].u) := DIVISION($DE...Evap.Hex.LnC.me |
| 1663 | regular | non-linear, size 2 |
| 1664 | regular | Evap.Hex.Qh1.Q_flow := Evap.Hex.gain1.k * Evap.Hex.LnQ1.Qt |
| 1665 | regular | Evap.Hex.LnH.Qb_flows[1] := Evap.Hex.Qh1.Q_flo...ap.Hex.LnH.mediu |
| 1666 | regular | der(Evap.Hex.LnH.Us[1]) := Evap.Hex.LnH.Hb_flo...Hex.LnH.Wb_flows |
| 1667 | regular | der(Evap.Hex.LnH.mediums[1].u) := DIVISION($DE...Evap.Hex.LnH.me |
| 1673 | regular | linear, size 5 |
| 1674 | regular | der(Evap.Hex.LnH.mediums[1].p) := DIVISION(DIV....LnH.mediums[1] |
| 1703 | regular | non-linear, size 28 |
| 1675 | regular | Evap.DrumWallHeatMass.der_T := Modelica.Medi...vap.Drum.mediur |
| 1676 | regular | der(Evap.Drum.medium.u) := der(Evap.Drum.medium...1].d, Evap.Dru |
| 1677 | regular | Evap.DrumWallHeatMass.port.Q_flow := Evap.DrumWallHeatMass.C * |
| 1678 | regular | der(Evap.Drum.U) := Evap.Drum.m * $DER.Evap....m.u + Evap.Drum.n |
| 1679 | regular | der(Evap.Hex.LnC.statesFM[2].d) := Modelica.Me...rum.medium.p), d |
| 1680 | regular | Evap.Drum.heatTransfer.Q_flows[1] := $DER.Evap.Drum.U - Evap.Drur |
| 1681 | regular | der(Evap.Hex.LnC.mediums[2].u) := der(Evap.Hex.L...statesFM[2].d, E\ |
| 1682 | regular | Evap.EvaQsource.port.Q_flow := (-Evap.DrumWallH...ort.Q_flow) - Ev |
| 1683 | regular | Evap.Hex.LnC.mb_flows[2] := Evap.Hex.LnC.fluidVolumes[2] * $DER.E |
| 1684 | regular | Evap.EvaQ.y := DIVISION(Evap.EvaQsource.port.Q...p.DrumWallHeatN |
| 1685 | regular | der(Evap.Hex.LnC.Us[2]) := Evap.Hex.LnC.ms[2] *...Hex.LnC.mb_flows |
| 1686 | regular | Evap.Hex.LnQ2.Qt := Evap.EvaQ.y - Evap.Hex.LnQ1.Qt |
| 1687 | regular | Evap.Hex.LnC.m_flows[3] := Evap.Hex.LnC.m_flows[2] - Evap.Hex.LnC |
| 1688 | regular | Evap.Hex.LnC.Qb_flows[2] := Evap.Hex.LnQ2.Qt * (...ex.LnC.heatTrans |
| 1689 | regular | Evap.Hex.LnC.vs[2] := 0.5 * DIVISION(Evap.Hex.LnC...x.LnC.statesFM[2 |
| 1690 | regular | Evap.Hex.LnC.H_flows[3] := semiLinear(Evap.Hex..., Evap.Hex.LnC.me |
| 1691 | regular | Evap.Hex.LnC.Wb_flows[2] := 0.5 * Evap.Hex.LnC.v...p.Hex.LnC.mediu |
| 1692 | regular | Evap.Hex.LnC.Hb_flows[2] := Evap.Hex.LnC.H_flows[2] - Evap.Hex.Ln( |

Defines

| Variable |
|---|
| Evap.DrumWallHeatMass.der_T |

Depends

| Variable |
|---|
| Evap.Drum.heatTransfer.states[1].phase |
| Evap.Drum.medium.h |
| Evap.Drum.medium.p |

Equation Operations

```
Operations
solved: Evap.DrumWallHeatMass.der_T = Modelica.Media.Water.IF97_Utilities.T_ph_der(Evap.Drum.medium.p, Evap.Drum.medium.h,
differentiate d/dtime: Modelica.Media.Water.IF97_Utilities.T_props_ph_der(Evap.Drum.medium.p, Evap.Drum.medium.h, Modelica.Medi
differentiate d/dtime: der(Evap.DrumWallHeatMass.T)
substitution: Modelica.Media.Water.IF97_Utilities.T_props_ph(Evap.Drum.medium.p, Evap.Drum.medium.h, Modelica.Media.Water.IF97
substitution: Evap.Drum.mWallHediumatMass.T
inline: Evap.Drum.medium.T = Paket2.CMompondentslica.BoMedia.Waters.IdealF97_UtiBoilties.T_pr$ops_ph(Evap$.Drum.Mmedium.tp
flattening: Evap.Drum.medium.T = Paket2.Components.Boilers.IdealBoiler$Evap$Drum.Medium.temperature_ph(Evap.Drum.medium.p,
```

Source Browser

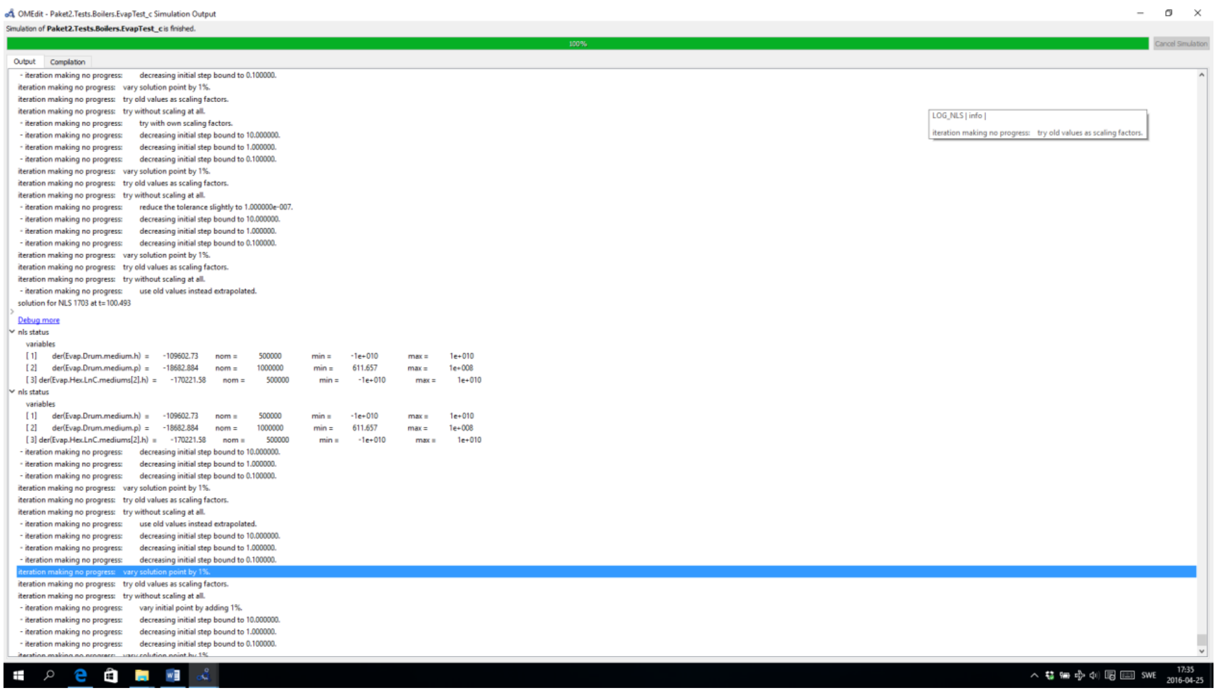
C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Media/Water/package.mo

```
dewDensity(sat)) and T
                < fluidConstants[1].criticalTemperature) then 1 else
2;
        else
          phase = 1;
          //this is for the one-phase only case pT
        end if;
      end if;
      if dT_explicit then
        p = pressure_dT(
              d,
              T,
              phase);
        h = specificEnthalpy_dT(
              d,
              T,
              phase);
        sat.Tsat = T;
        sat.psat = saturationPressure(T);
      elseif ph_explicit then
        d = density_ph(
              p,
              h,
              phase);
        T = temperature_ph(
              p,
              h,
              phase);
        sat.Tsat = saturationTemperature(p);
        sat.psat = p;
      else
        h = specificEnthalpy_pT(p, T);
        d = density_pT(p, T);
        sat.psat = p;
        sat.Tsat = saturationTemperature(p);
      end if;
      u = h - p/d;
      R = Modelica.Constants.R/fluidConstants[1].molarMass;
      h = state.h;
      p = state.p;
      T = state.T;
      d = state.d;
      phase = state.phase;
    end BaseProperties;

    redeclare function density_ph
      "Computes density as a function of pressure and specific
enthalpy"
      extends Modelica.Icons.Function;
      input AbsolutePressure p "Pressure";
      input SpecificEnthalpy h "Specific enthalpy";
      input FixedPhase phase=0 "2 for two-phase, 1 for one-phase, 0
```

**Figure 9 TDW after clicking one of the equations pointed out by SOW**

Each subordinate equation to 1703, starting with 1675 gives the defined variables and its dependencies and transformations made from original source file and the source browser points to its corresponding equation. Could this case be helped by this information? We are now searching for a link between the implanted error (negative heat transfer coefficient) and the messages and source code lines found by the debugger. A problem is definitely that there are no plots available so no values could be used for tracing the error. Characteristic for most the 28 equations pointed out is that they concern the medium properties of the drum and the Hex water pipes. Those equations are not written by author and thereby less interesting for corrective actions. Equations 1697 to 1702 are from the LnQ2 models that calculates the heat transfer. But without values there is little information to continue with.

Looking at SOW it recommends to use LOG_NLS.Setting. This flag in the simulation set up menu gives the following output from SOW (Figure 10)



Simulation of **Paket2.Tests.Boilers.EvapTest_c** is finished.

```
 - iteration making no progress:     decreasing initial step bound to 0.100000.
iteration making no progress:   vary solution point by 1%.
iteration making no progress:   try old values as scaling factors.
iteration making no progress:   try without scaling at all.
 - iteration making no progress:     try with own scaling factors.
 - iteration making no progress:     decreasing initial step bound to 10.000000.
 - iteration making no progress:     decreasing initial step bound to 1.000000.
 - iteration making no progress:     decreasing initial step bound to 0.100000.
iteration making no progress:   vary solution point by 1%.
iteration making no progress:   try old values as scaling factors.
iteration making no progress:   try without scaling at all.
 - iteration making no progress:     reduce the tolerance slightly to 1.000000e-007.
 - iteration making no progress:     decreasing initial step bound to 10.000000.
 - iteration making no progress:     decreasing initial step bound to 1.000000.
 - iteration making no progress:     decreasing initial step bound to 0.100000.
iteration making no progress:   vary solution point by 1%.
iteration making no progress:   try old values as scaling factors.
iteration making no progress:   try without scaling at all.
 - iteration making no progress:     use old values instead extrapolated.
solution for NLS 1703 at t=100.493
>
Debug more
v nls status
    variables
    [ 1]     der(Evap.Drum.medium.h)  =    -109602.73    nom =    500000     min =    -1e+010     max =    1e+010
    [ 2]     der(Evap.Drum.medium.p)  =    -18682.884    nom =    1000000    min =    611.657     max =    1e+008
    [ 3] der(Evap.Hex.LnC.mediums[2].h)  =    -170221.58    nom =    500000     min =    -1e+010     max =    1e+010
v nls status
    variables
    [ 1]     der(Evap.Drum.medium.h)  =    -109602.73    nom =    500000     min =    -1e+010     max =    1e+010
    [ 2]     der(Evap.Drum.medium.p)  =    -18682.884    nom =    1000000    min =    611.657     max =    1e+008
    [ 3] der(Evap.Hex.LnC.mediums[2].h)  =    -170221.58    nom =    500000     min =    -1e+010     max =    1e+010
 - iteration making no progress:     decreasing initial step bound to 10.000000.
 - iteration making no progress:     decreasing initial step bound to 1.000000.
 - iteration making no progress:     decreasing initial step bound to 0.100000.
iteration making no progress:   vary solution point by 1%.
iteration making no progress:   try old values as scaling factors.
iteration making no progress:   try without scaling at all.
 - iteration making no progress:     use old values instead extrapolated.
 - iteration making no progress:     decreasing initial step bound to 10.000000.
 - iteration making no progress:     decreasing initial step bound to 1.000000.
 - iteration making no progress:     decreasing initial step bound to 0.100000.
iteration making no progress:   vary solution point by 1%.
iteration making no progress:   try old values as scaling factors.
iteration making no progress:   try without scaling at all.
 - iteration making no progress:     vary initial point by adding 1%.
 - iteration making no progress:     decreasing initial step bound to 10.000000.
 - iteration making no progress:     decreasing initial step bound to 1.000000.
 - iteration making no progress:     decreasing initial step bound to 0.100000.
iteration making no progress:   vary solution point by 1%.
```

```
LOG_NLS | info |

iteration making no progress:   try old values as scaling factors.
```

**Figure 10 The SOW after activating logging of None Linear Solutions (log_NLS)**

Displaying the last nls status shows negative values of drum pressure and enthalpy. Those values should never become negative in this process. Negative values of those should render an automatic crash of simulation and a saving of all variables to the result file (.mat) - at least as the pressure should not be below 611 pa. However, it seems like the solver tries to find a solution (that will bring acceptable numerical errors in the solution) with a stamina that is unacceptably time consuming. Those negative values shows that the drum has totally wrong working conditions. But what has caused those unphysical conditions? All the repeated messages about the drum conditions are of little value as there are no result file to examine. For this case it is how ever not that many messages to scan through, so the chance of discovering the zero division messages at 100 s are good.

In case there is a permanent zero denominator after a certain time point then the debugger points directly to the causing equation.

**Conclusion**

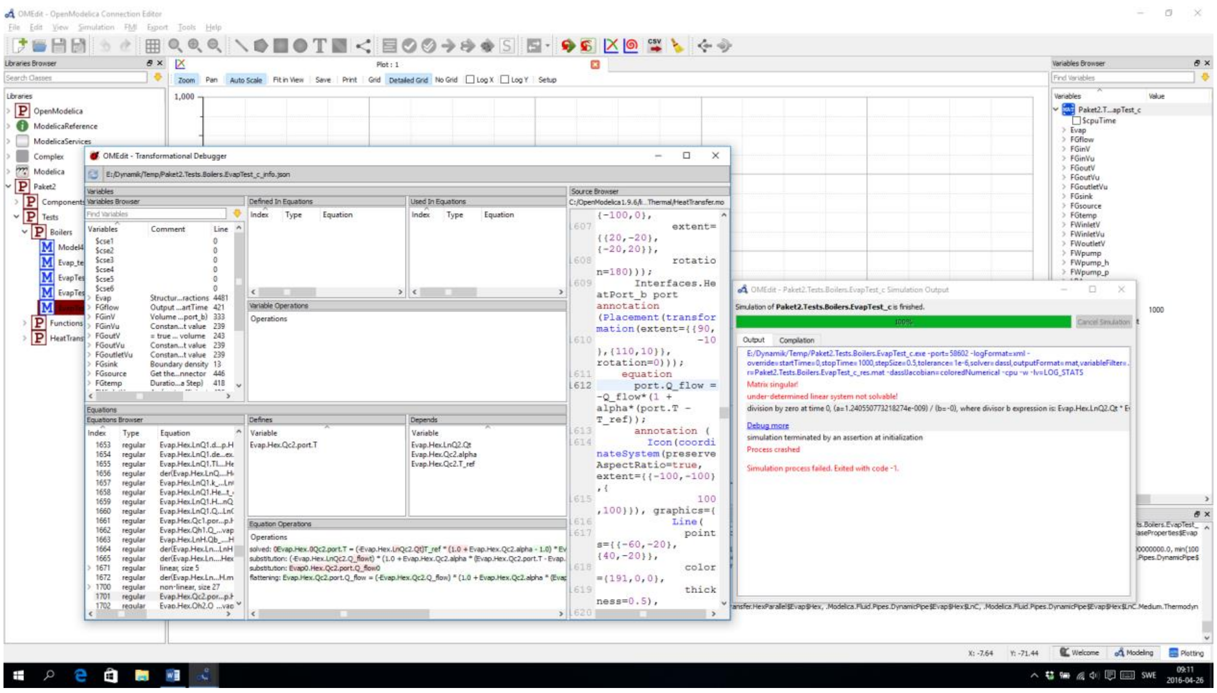
For a ramped denominator passing zero, the debugger is not optimal in case the solver manages to pass the critical point and that consecutive errors then hides the information from the user. A solution would be the option to let the user decide if division by zero should be accepted or not, i.e. the solver should then interrupt and save when any denominator having a passage of zero.

Another proposal is if the SOW could discriminate or summarize the messages and just display one of each type at the top level and if user likes, he could brows each type to see how many and when they have occurred.

Just for interest the simulation is repeated up to 100.49 s to see how the process looks at the kinner zero passage. The process shows a flue gas outlet temperature that goes bananas and suddenly becomes the highest temperature in the process, and a natural thing is then to investigate the heat transfer calculation, i.e. it's a highly recommendable feature to have a result file available as soon as any progress in simulation has been achieved.

## 3.2.3 Division by zero due to mismatch in parameter settings

By deselection of the heat transfer use in the LnC pipe in the Hex model (Figure 3) the test model still checks OK, but now with only 1106 equations instead of 1110. Simulation gives following SOW and TDW (Figure 11).


Source Browser
C:/OpenModelica1.9.6/li...Thermal/HeatTransfer.mo

```
     {-100,0},
1607                extent=
     {{20,-20},
     {-20,20}},
1608              rotatio
     n=180)));
1609      Interfaces.He
     atPort_b port
     annotation
     (Placement(transfor
     mation(extent={{90,
1610                  -10
     },{110,10}},
     rotation=0)));
1611    equation
1612      port.Q_flow =
     -Q_flow*(1 +
     alpha*(port.T -
     T_ref));
1613      annotation (
1614        Icon(coordi
     nateSystem(preserve
     AspectRatio=true,
     extent={{-100,-100}
     ,{
1615                  100
     ,100}}), graphics={
1616              Line(
1617                point
     s={{-60,-20},
     {40,-20}},
1618                color
     ={191,0,0},
1619                thick
     ness=0.5),
```

Defines
Variable
Evap.Hex.Qc2.port.T

Depends
Variable
Evap.Hex.LnQ2.Qt
Evap.Hex.Qc2.alpha
Evap.Hex.Qc2.T_ref

Equation Operations
Operations
```
solved: 0Evap.Hex.0Qc2.port.T = (-Evap.Hex.LnQc2.Qt)T_ref * (1.0 + Evap.Hex.Qc2.alpha - 1.0) * Ev
substitution: (-Evap.Hex.LnQc2.Q_flowt) * (1.0 + Evap.Hex.Qc2.alpha * (Evap.Hex.Qc2.port.T - Evap.
substitution: Evap0.Hex.Qc2.port.Q_flow0
flattening: Evap.Hex.Qc2.port.Q_flow = (-Evap.Hex.Qc2.Q_flow) * (1.0 + Evap.Hex.Qc2.alpha * (Evap
```

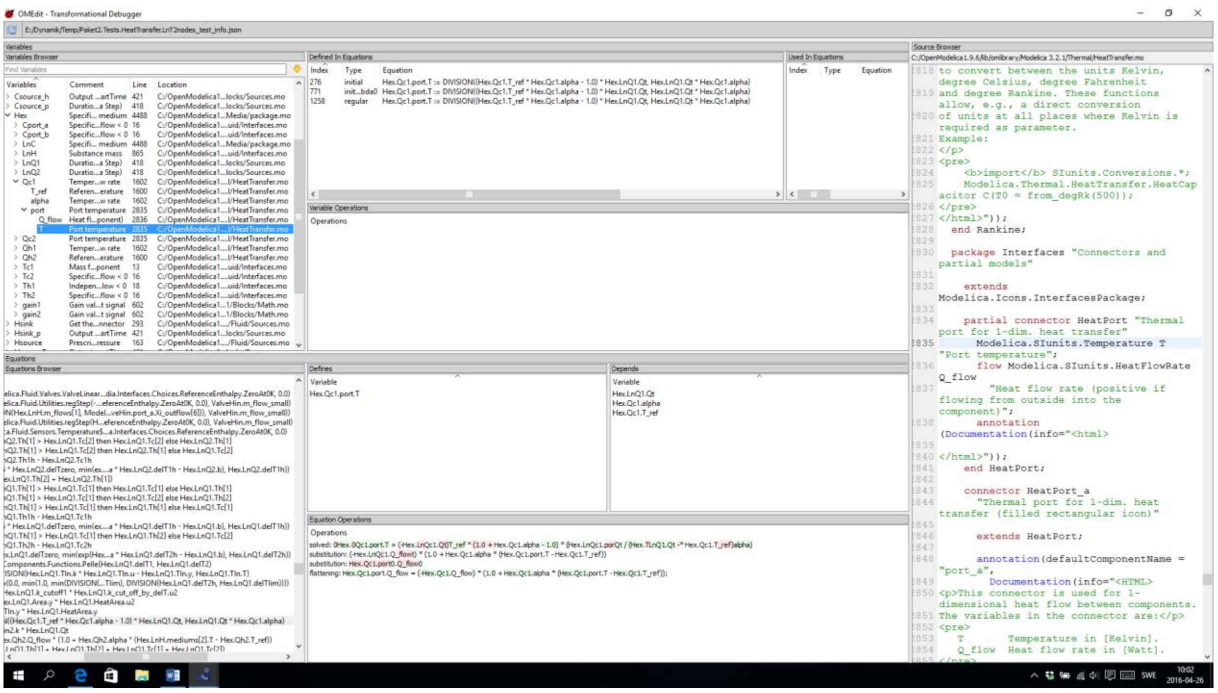
OMEdit - Paket2.Tests.Boilers.EvapTest_c Simulation Output
Simulation of **Paket2.Tests.Boilers.EvapTest_c** is finished.

```
E:/Dynamik/Temp/Paket2.Tests.Boilers.EvapTest_c.exe -port=58602 -logFormat=xml -
override=startTime=0,stopTime=1000,stepSize=0.5,tolerance=1e-6,solver=dassl,outputFormat=mat,variableFilter=.
r=Paket2.Tests.Boilers.EvapTest_c_res.mat -dasslJacobian=coloredNumerical -cpu -w -lv=LOG_STATS
Matrix singular!
under-determined linear system not solvable!
division by zero at time 0, (a=1.240550773218274e-009) / (b=-0), where divisor b expression is: Evap.Hex.LnQ2.Qt * E
Debug more
simulation terminated by an assertion at initialization
Process crashed

Simulation process failed. Exited with code -1.
```

OMEdit - OpenModelica Connection Editor
OMEdit - Transformational Debugger
E:/Dynamik/Temp/Paket2.Tests.Boilers.EvapTest_c_info.json
Variables Browser
Equations Browser
Libraries Browser
Variables Browser
Output
Compilation
100%
Cancel Simulation

**Figure 11 Outputs after no use of heat transfer specified but corresponding heat transfer connectors any way connected**

The simulation crashes at initialization due to division with zero where denominator is involving variables Qt and alpha. A check of the model reveals that alpha is the heat transfer coefficient to surroundings and is deliberately zero. I.e. this model is impossible to run although it checks OK. The debugger points to an equation 1258. Marking that equation points to a source code line 1612 where the Modelica standard library (MSL) component PrescribedHeatFlow (PHF) equals the heat ports heat flow (Q_Flow) to the connector input Q_Flow (the prescribed flow). This equation contains also a dependence on alpha, but with alpha=0 the equation should be OK: port.Q_Flow=-Q_Flow. The PHF model calculates its internal temperature in order to be able to have the prescribed heat flow by this equation (index 1258). This temperature should be the same as the connected pipes temperature as there is no thermal resistance in the connection itself so why is it calculated in this way? One could not be sure that any user would understand that this equation is unexpected and caused by a wrong parameter setting in the pipe model. It would have been better if there were a consistency check between connectors and there use in the program. One way would be not to show the connector unless it is activated, or give an error message if it is anyway connected. Could the variables browser shed some light over the problem? The variables browser is presented in Figure 12, after the port temperature has been clicked.



E:/Dynamik/Temp/Paket2.Tests.HeatTransfer.LnT2nodes_test_info.json

| Variables | Comment | Line | Location |
|---|---|---|---|
| Csource_h | Output ...artTime | 421 | C:/OpenModelica1...locks/Sources.mo |
| Csource_p | Duratio...a Step) | 418 | C:/OpenModelica1...locks/Sources.mo |
| Hex | Specifi... medium | 4488 | C:/OpenModelica1...Media/package.mo |
| Cport_a | Specific...flow < 0 | 16 | C:/OpenModelica1....uid/Interfaces.mo |
| Cport_b | Specific...flow < 0 | 16 | C:/OpenModelica1....uid/Interfaces.mo |
| LnC | Specifi... medium | 4488 | C:/OpenModelica1...Media/package.mo |
| LnH | Substance mass | 865 | C:/OpenModelica1....uid/Interfaces.mo |
| LnQ1 | Duratio...a Step) | 418 | C:/OpenModelica1...locks/Sources.mo |
| LnQ2 | Duratio...a Step) | 418 | C:/OpenModelica1...locks/Sources.mo |
| Qc1 | Temper...w rate | 1602 | C:/OpenModelica1....l/HeatTransfer.mo |
| T_ref | Referen...erature | 1600 | C:/OpenModelica1....l/HeatTransfer.mo |
| alpha | Temper...w rate | 1602 | C:/OpenModelica1....l/HeatTransfer.mo |
| port | Port temperature | 2835 | C:/OpenModelica1....l/HeatTransfer.mo |
| Q_flow | Heat fl...ponent) | 2836 | C:/OpenModelica1....l/HeatTransfer.mo |
| T | Port temperature | 2835 | C:/OpenModelica1....l/HeatTransfer.mo |
| Qc2 | Port temperature | 2835 | C:/OpenModelica1....l/HeatTransfer.mo |
| Qh1 | Temper...w rate | 1602 | C:/OpenModelica1....l/HeatTransfer.mo |
| Qh2 | Referen...erature | 1600 | C:/OpenModelica1....l/HeatTransfer.mo |
| Tc1 | Mass f...ponent | 13 | C:/OpenModelica1....uid/Interfaces.mo |
| Tc2 | Specific...flow < 0 | 16 | C:/OpenModelica1....uid/Interfaces.mo |
| Th1 | Indepen...low < 0 | 18 | C:/OpenModelica1....uid/Interfaces.mo |
| Th2 | Specific...flow < 0 | 16 | C:/OpenModelica1....uid/Interfaces.mo |
| gain1 | Gain val...t signal | 602 | C:/OpenModelica1...1/Blocks/Math.mo |
| gain2 | Gain val...t signal | 602 | C:/OpenModelica1...1/Blocks/Math.mo |
| Hsink | Get the...nnector | 293 | C:/OpenModelica1..../Fluid/Sources.mo |
| Hsink_p | Output ...artTime | 421 | C:/OpenModelica1...locks/Sources.mo |
| Hsource | Prescri...ressure | 163 | C:/OpenModelica1..../Fluid/Sources.mo |

Defined In Equations

| Index | Type | Equation |
|---|---|---|
| 276 | initial | Hex.Qc1.port.T := DIVISION((Hex.Qc1.T_ref * Hex.Qc1.alpha - 1.0) * Hex.LnQ1.Qt, Hex.LnQ1.Qt * Hex.Qc1.alpha) |
| 771 | init...bda0 | Hex.Qc1.port.T := DIVISION((Hex.Qc1.T_ref * Hex.Qc1.alpha - 1.0) * Hex.LnQ1.Qt, Hex.LnQ1.Qt * Hex.Qc1.alpha) |
| 1258 | regular | Hex.Qc1.port.T := DIVISION((Hex.Qc1.T_ref * Hex.Qc1.alpha - 1.0) * Hex.LnQ1.Qt, Hex.LnQ1.Qt * Hex.Qc1.alpha) |

Used In Equations

| Index | Type | Equation |
|---|---|---|

Variable Operations

Operations

Equations Browser

```
elica.Fluid.Valves.ValveLinear...dia.Interfaces.Choices.ReferenceEnthalpy.ZeroAt0K, 0.0)
elica.Fluid.Utilities.regStep(-...eferenceEnthalpy.ZeroAt0K, 0.0), ValveHin.m_flow_small)
N(Hex.LnH.m_flows[1], Model...veHin.port_a.Xi_outflow[6])), ValveHin.m_flow_small))
elica.Fluid.Utilities.regStep(H...eferenceEnthalpy.ZeroAt0K, 0.0), ValveHin.m_flow_small)
:a.Fluid.Sensors.Temperature$...a.Interfaces.Choices.ReferenceEnthalpy.ZeroAt0K, 0.0)
nQ2.Th[1] > Hex.LnQ1.Tc[2] then Hex.LnQ1.Tc[2] else Hex.LnQ2.Th[1]
nQ2.Th[1] > Hex.LnQ1.Tc[2] then Hex.LnQ2.Th[1] else Hex.LnQ1.Tc[2]
nQ2.Th1h - Hex.LnQ2.Tc1h
 * Hex.LnQ2.delTzero, min(ex....a * Hex.LnQ2.delT1h - Hex.LnQ2.b), Hex.LnQ2.delT1h))
ex.LnQ1.Th[2] + Hex.LnQ2.Th[1])
Q1.Th[1] > Hex.LnQ1.Tc[1] then Hex.LnQ1.Tc[1] else Hex.LnQ1.Th[1]
Q1.Th[1] > Hex.LnQ1.Tc[1] then Hex.LnQ1.Tc[2] else Hex.LnQ1.Th[2]
nQ1.Th[1] > Hex.LnQ1.Tc[1] then Hex.LnQ1.Th[1] else Hex.LnQ1.Tc[1]
nQ1.Th1h - Hex.LnQ1.Tc1h
 * Hex.LnQ1.delTzero, min(ex....a * Hex.LnQ1.delT1h - Hex.LnQ1.b), Hex.LnQ1.delT1h))
nQ1.Th[1] > Hex.LnQ1.Tc[1] then Hex.LnQ1.Th[2] else Hex.LnQ1.Tc[2]
nQ1.Th2h - Hex.LnQ1.Tc2h
x.LnQ1.delTzero, min(exp(Hex...a * Hex.LnQ1.delT2h - Hex.LnQ1.b), Hex.LnQ1.delT2h))
Components.Functions.Pelle(Hex.LnQ1.delT1, Hex.LnQ1.delT2)
ISION(Hex.LnQ1.Tln.k * Hex.LnQ1.Tln.u - Hex.LnQ1.Tln.y, Hex.LnQ1.Tln.T)
(0.0, min(1.0, min(DIVISION(...Tlim), DIVISION(Hex.LnQ1.delT2h, Hex.LnQ1.delTlim))))
Hex.LnQ1.k_cutoff1 * Hex.LnQ1.k_cut_off_by_delT.u2
ex.LnQ1.Area.y * Hex.LnQ1.HeatArea.u2
Tln.y * Hex.LnQ1.HeatArea.y
N((Hex.Qc1.T_ref * Hex.Qc1.alpha - 1.0) * Hex.LnQ1.Qt, Hex.LnQ1.Qt * Hex.Qc1.alpha)
in2.k * Hex.LnQ1.Qt
ex.Qh2.Q_flow * (1.0 + Hex.Qh2.alpha * (Hex.LnH.mediums[2].T - Hex.Qh2.T_ref))
.LnQ1.Th[1] + Hex.LnQ1.Th[2] + Hex.LnQ1.Tc[1] + Hex.LnQ1.Tc[2])
```

Defines

| Variable |
|---|
| Hex.Qc1.port.T |

Depends

| Variable |
|---|
| Hex.LnQ1.Qt |
| Hex.Qc1.alpha |
| Hex.Qc1.T_ref |

Equation Operations

```
Operations
solved: 0Hex.0Qc1.port.T = (-Hex.LnQc1.Qt)T_ref * (1.0 + Hex.Qc1.alpha - 1.0) * (Hex.LnQc1.porQt / (Hex.TLnQ1.Qt -* Hex.Qc1.T_ref)alpha)
substitution: (-Hex.LnQc1.Q_flowt) * (1.0 + Hex.Qc1.alpha * (Hex.Qc1.port.T - Hex.Qc1.T_ref))
substitution: Hex.Qc1.port0.Q_flow0
flattening: Hex.Qc1.port.Q_flow = (-Hex.Qc1.Q_flow) * (1.0 + Hex.Qc1.alpha * (Hex.Qc1.port.T - Hex.Qc1.T_ref));
```

Source Browser

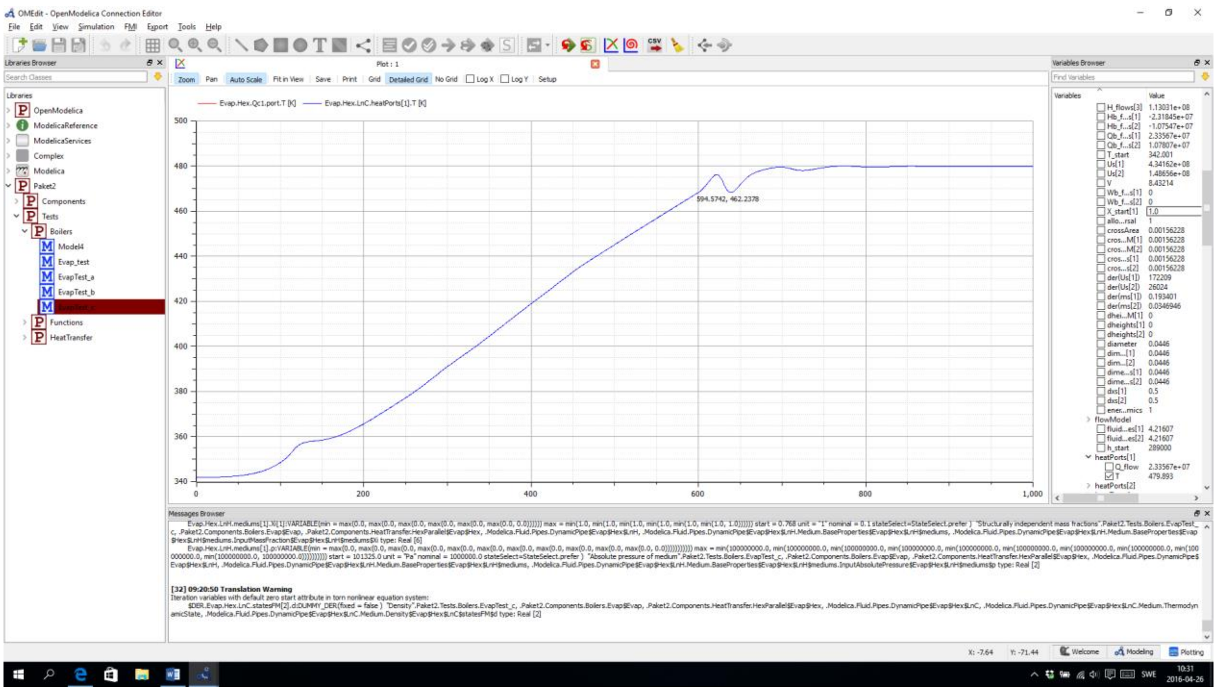
C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Thermal/HeatTransfer.mo

```
2818 to convert between the units Kelvin,
     degree Celsius, degree Fahrenheit
2819 and degree Rankine. These functions
     allow, e.g., a direct conversion
2820 of units at all places where Kelvin is
     required as parameter.
2821 Example:
2822 </p>
2823 <pre>
2824     <b>import</b> SIunits.Conversions.*;
2825     Modelica.Thermal.HeatTransfer.HeatCap
     acitor C(T0 = from_degRk(500));
2826 </pre>
2827 </html>"));
2828   end Rankine;
2829
2830   package Interfaces "Connectors and
     partial models"
2831
2832     extends
     Modelica.Icons.InterfacesPackage;
2833
2834     partial connector HeatPort "Thermal
     port for 1-dim. heat transfer"
2835       Modelica.SIunits.Temperature T
     "Port temperature";
2836       flow Modelica.SIunits.HeatFlowRate
     Q_flow
2837         "Heat flow rate (positive if
     flowing from outside into the
     component)";
2838       annotation
     (Documentation(info="<html>
2839
2840 </html>"));
2841     end HeatPort;
2842
2843     connector HeatPort_a
2844       "Thermal port for 1-dim. heat
     transfer (filled rectangular icon)"
2845
2846       extends HeatPort;
2847
2848       annotation(defaultComponentName =
     "port_a",
2849         Documentation(info="<HTML>
2850 <p>This connector is used for 1-
     dimensional heat flow between components.
2851 The variables in the connector are:</p>
2852 <pre>
2853   T       Temperature in [Kelvin].
2854   Q_flow  Heat flow rate in [Watt].
```

**Figure 12 Information form variables browser**

It gives the same information added with the initial equations. A question is why is not the debugger pointing to the initial equations indexes 276 and 771, as the SOW tells that the error appeared at initialization?

For comparison with the correct programmed model one could plot the PHF temperature and the pipe temperature and see that they are the same (Figure 13)

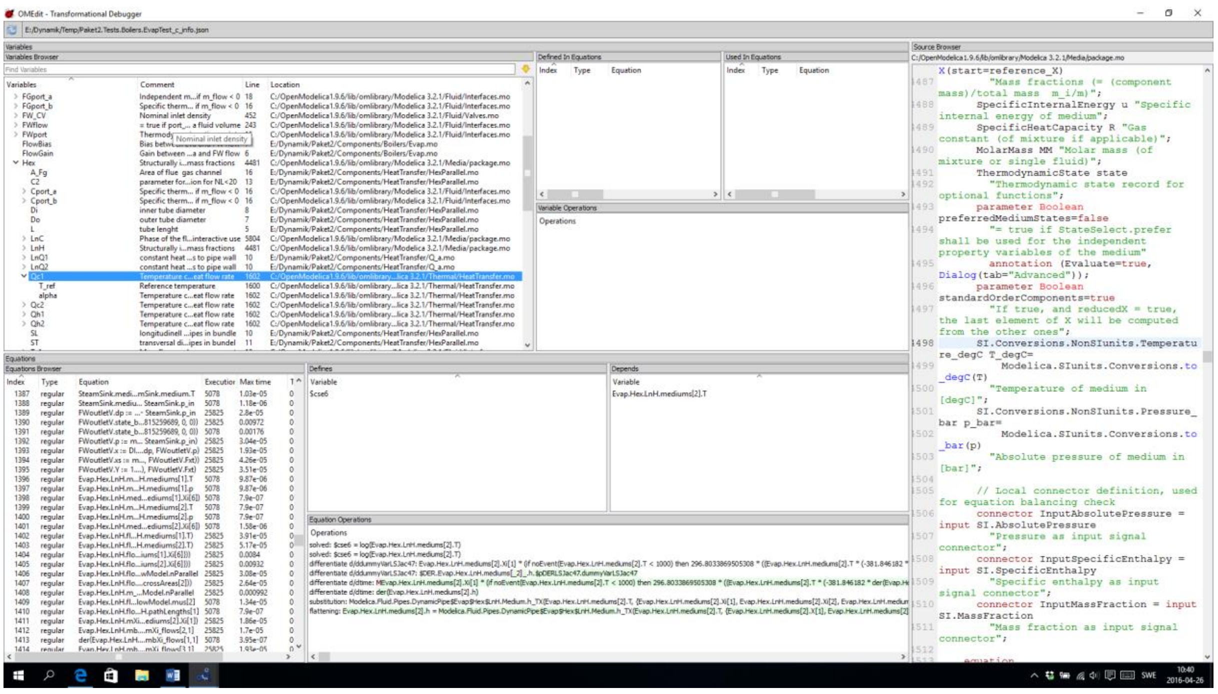


| Variables | Value |
|---|---|
| H_flows[3] | 1.13031e+08 |
| Hb_f...s[1] | -2.31845e+07 |
| Hb_f...s[2] | -1.07547e+07 |
| Qb_f...s[1] | 2.33567e+07 |
| Qb_f...s[2] | 1.07807e+07 |
| T_start | 342.001 |
| Us[1] | 4.34162e+08 |
| Us[2] | 1.48656e+08 |
| V | 8.43214 |
| Wb_f...s[1] | 0 |
| Wb_f...s[2] | 0 |
| X_start[1] | 1.0 |
| allo...rsal | 1 |
| crossArea | 0.00156228 |
| cros...M[1] | 0.00156228 |
| cros...M[2] | 0.00156228 |
| cros...s[1] | 0.00156228 |
| cros...s[2] | 0.00156228 |
| der(Us[1]) | 172209 |
| der(Us[2]) | 26024 |
| der(ms[1]) | 0.193401 |
| der(ms[2]) | 0.0346946 |
| dhei...M[1] | 0 |
| dheights[1] | 0 |
| dheights[2] | 0 |
| diameter | 0.0446 |
| dim...[1] | 0.0446 |
| dim...[2] | 0.0446 |
| dime...s[1] | 0.0446 |
| dime...s[2] | 0.0446 |
| dxs[1] | 0.5 |
| dxs[2] | 0.5 |
| ener...mics | 1 |
| flowModel | |
| fluid...es[1] | 4.21607 |
| fluid...es[2] | 4.21607 |
| h_start | 289000 |
| heatPorts[1] | |
| Q_flow | 2.33567e+07 |
| ☑ T | 479.893 |
| heatPorts[2] | |

```
      Evap.Hex.LnH.mediums[1].Xi[1]:VARIABLE(min = max(0.0, max(0.0, max(0.0, max(0.0, max(0.0, 0.0))))) max = min(1.0, min(1.0, min(1.0, min(1.0, min(1.0, min(1.0, 1.0)))))) start = 0.768 unit = "1" nominal = 0.1 stateSelect=StateSelect.prefer )  "Structurally independent mass fractions".Paket2.Tests.Boilers.EvapTest_c, .Paket2.Components.Boilers.Evap$Evap, .Paket2.Components.HeatTransfer.HexParallel$Evap$Hex, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnH, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnH.Medium.BaseProperties$Evap$Hex$LnH$mediums, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnH.Medium.BaseProperties$Evap$Hex$LnH$mediums.InputMassFraction$Evap$Hex$LnH$mediums$Xi type: Real [6]
      Evap.Hex.LnH.mediums[1].p:VARIABLE(min = max(0.0, max(0.0, max(0.0, max(0.0, max(0.0, max(0.0, max(0.0, max(0.0, max(0.0, max(0.0, 0.0)))))))))) max = min(100000000.0, min(100000000.0, min(100000000.0, min(100000000.0, min(100000000.0, min(100000000.0, min(100000000.0, min(100000000.0, min(100000000.0, min(100000000.0, 100000000.0)))))))))) start = 101325.0 unit = "Pa" nominal = 1000000.0 stateSelect=StateSelect.prefer )  "Absolute pressure of medium".Paket2.Tests.Boilers.EvapTest_c, .Paket2.Components.Boilers.Evap$Evap, .Paket2.Components.HeatTransfer.HexParallel$Evap$Hex, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnH, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnH.Medium.BaseProperties$Evap$Hex$LnH$mediums, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnH.Medium.BaseProperties$Evap$Hex$LnH$mediums.InputAbsolutePressure$Evap$Hex$LnH$mediums$p type: Real [2]

[32] 09:20:50 Translation Warning
Iteration variables with default zero start attribute in torn nonlinear equation system:
      $DER.Evap.Hex.LnC.statesFM[2].d:DUMMY_DER(fixed = false )  "Density".Paket2.Tests.Boilers.EvapTest_c, .Paket2.Components.Boilers.Evap$Evap, .Paket2.Components.HeatTransfer.HexParallel$Evap$Hex, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnC, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnC.Medium.ThermodynamicState, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnC.Medium.Density$Evap$Hex$LnC$statesFM$d type: Real [2]
```

**Figure 13 Plotting of the temperatures when model is correct**

Checking the variables browser now reveals that the port temperature is no longer to be found (Figure 14)



OMEdit - Transformational Debugger

E:/Dynamik/Temp/Paket2.Tests.Boilers.EvapTest_c_info.json

Variables Browser

| Variables | Comment | Line | Location |
|---|---|---|---|
| FGport_a | Independent m...if m_flow < 0 | 18 | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Fluid/Interfaces.mo |
| FGport_b | Specific therm... if m_flow < 0 | 16 | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Fluid/Interfaces.mo |
| FW_CV | Nominal inlet density | 452 | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Fluid/Valves.mo |
| FWflow | = true if port_... a fluid volume | 243 | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Fluid/Interfaces.mo |
| FWport | Thermody | | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Fluid/Interfaces.mo |
| FlowBias | Bias betw | | E:/Dynamik/Paket2/Components/Boilers/Evap.mo |
| FlowGain | Gain between ...a and FW flow | 6 | E:/Dynamik/Paket2/Components/Boilers/Evap.mo |
| Hex | Structurally i...mass fractions | 4481 | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Media/package.mo |
| A_Fg | Area of flue gas channel | 16 | E:/Dynamik/Paket2/Components/HeatTransfer/HexParallel.mo |
| C2 | parameter for...ion for NL<20 | 13 | E:/Dynamik/Paket2/Components/HeatTransfer/HexParallel.mo |
| Cport_a | Specific therm... if m_flow < 0 | 16 | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Fluid/Interfaces.mo |
| Cport_b | Specific therm... if m_flow < 0 | 16 | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Fluid/Interfaces.mo |
| Di | inner tube diameter | 8 | E:/Dynamik/Paket2/Components/HeatTransfer/HexParallel.mo |
| Do | outer tube diameter | 7 | E:/Dynamik/Paket2/Components/HeatTransfer/HexParallel.mo |
| L | tube lenght | 5 | E:/Dynamik/Paket2/Components/HeatTransfer/HexParallel.mo |
| LnC | Phase of the fl...interactive use | 5804 | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Media/package.mo |
| LnH | Structurally i...mass fractions | 4481 | C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Media/package.mo |
| LnQ1 | constant heat ...s to pipe wall | 10 | E:/Dynamik/Paket2/Components/HeatTransfer/Q_a.mo |
| LnQ2 | constant heat ...s to pipe wall | 10 | E:/Dynamik/Paket2/Components/HeatTransfer/Q_a.mo |
| Qc1 | Temperature c...eat flow rate | 1602 | C:/OpenModelica1.9.6/lib/omlibrary...lica 3.2.1/Thermal/HeatTransfer.mo |
| T_ref | Reference temperature | 1600 | C:/OpenModelica1.9.6/lib/omlibrary...lica 3.2.1/Thermal/HeatTransfer.mo |
| alpha | Temperature c...eat flow rate | 1602 | C:/OpenModelica1.9.6/lib/omlibrary...lica 3.2.1/Thermal/HeatTransfer.mo |
| Qc2 | Temperature c...eat flow rate | 1602 | C:/OpenModelica1.9.6/lib/omlibrary...lica 3.2.1/Thermal/HeatTransfer.mo |
| Qh1 | Temperature c...eat flow rate | 1602 | C:/OpenModelica1.9.6/lib/omlibrary...lica 3.2.1/Thermal/HeatTransfer.mo |
| Qh2 | Temperature c...eat flow rate | 1602 | C:/OpenModelica1.9.6/lib/omlibrary...lica 3.2.1/Thermal/HeatTransfer.mo |
| SL | longitudinell ...ipes in bundle | 10 | E:/Dynamik/Paket2/Components/HeatTransfer/HexParallel.mo |
| ST | transversal di...ipes in bundel | 11 | E:/Dynamik/Paket2/Components/HeatTransfer/HexParallel.mo |

Nominal inlet density

Equations Browser

| Index | Type | Equation | Executior | Max time | T |
|---|---|---|---|---|---|
| 1387 | regular | SteamSink.medi...mSink.medium.T | 5078 | 1.03e-05 | 0 |
| 1388 | regular | SteamSink.mediu... SteamSink.p_in | 5078 | 1.18e-06 | 0 |
| 1389 | regular | FWoutletV.dp := ...- SteamSink.p_in | 25825 | 2.8e-05 | 0 |
| 1390 | regular | FWoutletV.state_b...815259689, 0, 0)) | 25825 | 0.00972 | 0 |
| 1391 | regular | FWoutletV.state_b...815259689, 0, 0)) | 5078 | 0.00176 | 0 |
| 1392 | regular | FWoutletV.p := m... SteamSink.p_in) | 25825 | 3.04e-05 | 0 |
| 1393 | regular | FWoutletV.x := DI....dp, FWoutletV.p) | 25825 | 1.93e-05 | 0 |
| 1394 | regular | FWoutletV.xs := m..., FWoutletV.Fxt)) | 25825 | 4.26e-05 | 0 |
| 1395 | regular | FWoutletV.Y := 1....), FWoutletV.Fxt) | 25825 | 3.51e-05 | 0 |
| 1396 | regular | Evap.Hex.LnH.m...H.mediums[1].T | 5078 | 9.87e-06 | 0 |
| 1397 | regular | Evap.Hex.LnH.m...H.mediums[1].p | 5078 | 9.87e-06 | 0 |
| 1398 | regular | Evap.Hex.LnH.med...ediums[1].Xi[6]) | 5078 | 7.9e-07 | 0 |
| 1399 | regular | Evap.Hex.LnH.m...H.mediums[2].T | 5078 | 7.9e-07 | 0 |
| 1400 | regular | Evap.Hex.LnH.m...H.mediums[2].p | 5078 | 7.9e-07 | 0 |
| 1401 | regular | Evap.Hex.LnH.med...ediums[2].Xi[6]) | 5078 | 1.58e-06 | 0 |
| 1402 | regular | Evap.Hex.LnH.fl...H.mediums[1].T) | 25825 | 3.91e-05 | 0 |
| 1403 | regular | Evap.Hex.LnH.fl...H.mediums[2].T) | 25825 | 5.17e-05 | 0 |
| 1404 | regular | Evap.Hex.LnH.flo...iums[1].Xi[6]})) | 25825 | 0.0084 | 0 |
| 1405 | regular | Evap.Hex.LnH.flo...iums[2].Xi[6]})) | 25825 | 0.00932 | 0 |
| 1406 | regular | Evap.Hex.LnH.flo...wModel.nParallel | 25825 | 3.08e-05 | 0 |
| 1407 | regular | Evap.Hex.LnH.flo....crossAreas[2])) | 25825 | 2.64e-05 | 0 |
| 1408 | regular | Evap.Hex.LnH.m_...Model.nParallel | 25825 | 0.000992 | 0 |
| 1409 | regular | Evap.Hex.LnH.fl...lowModel.mus[2] | 5078 | 1.34e-05 | 0 |
| 1410 | regular | Evap.Hex.LnH.flo...H.pathLengths[1] | 5078 | 7.9e-07 | 0 |
| 1411 | regular | Evap.Hex.LnH.mXi...ediums[2].Xi[1]) | 25825 | 1.86e-05 | 0 |
| 1412 | regular | Evap.Hex.LnH.mb....mXi_flows[2,1] | 25825 | 1.7e-05 | 0 |
| 1413 | regular | der(Evap.Hex.LnH....mbXi_flows[1,1] | 5078 | 3.95e-07 | 0 |
| 1414 | regular | Evap.Hex.LnH.mb....mXi_flows[3,1] | 25825 | 1.93e-05 | 0 |

Defines

| Variable |
|---|
| $cse6 |

Depends

| Variable |
|---|
| Evap.Hex.LnH.mediums[2].T |

Equation Operations

```
solved: $cse6 = log(Evap.Hex.LnH.mediums[2].T)
solved: $cse6 = log(Evap.Hex.LnH.mediums[2].T)
differentiate d/ddummyVarLSJac47: Evap.Hex.LnH.mediums[2].Xi[1] * (if noEvent(Evap.Hex.LnH.mediums[2].T < 1000) then 296.8033869505308 * ((Evap.Hex.LnH.mediums[2].T * (-381.846182 *
differentiate d/ddummyVarLSJac47: $DER.Evap.Hex.LnH.mediums[_2]_.h.$pDERLSJac47.dummyVarLSJac47
differentiate d/dtime: MEvap.Hex.LnH.mediums[2].Xi[1] * (if noEvent(Evap.Hex.LnH.mediums[2].T < 1000) then 296.8033869505308 * ((Evap.Hex.LnH.mediums[2].T * (-381.846182 * der(Evap.He
differentiate d/dtime: der(Evap.Hex.LnH.mediums[2].h)
substitution: Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnH.Medium.h_TX(Evap.Hex.LnH.mediums[2].T, {Evap.Hex.LnH.mediums[2].Xi[1], Evap.Hex.LnH.mediums[2].Xi[2], Evap.Hex.LnH.medium
flattening: Evap.Hex.LnH.mediums[2].h = Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnH.Medium.h_TX(Evap.Hex.LnH.mediums[2].T, {Evap.Hex.LnH.mediums[2].X[1], Evap.Hex.LnH.mediums[2]
```

Source Browser

C:/OpenModelica1.9.6/lib/omlibrary/Modelica 3.2.1/Media/package.mo

```
X(start=reference_X)
        "Mass fractions (= (component mass)/total mass  m_i/m)";
      SpecificInternalEnergy u "Specific internal energy of medium";
      SpecificHeatCapacity R "Gas constant (of mixture if applicable)";
      MolarMass MM "Molar mass (of mixture or single fluid)";
      ThermodynamicState state
        "Thermodynamic state record for optional functions";
      parameter Boolean preferredMediumStates=false
        "= true if StateSelect.prefer shall be used for the independent property variables of the medium"
        annotation (Evaluate=true, Dialog(tab="Advanced"));
      parameter Boolean standardOrderComponents=true
        "If true, and reducedX = true, the last element of X will be computed from the other ones";
      SI.Conversions.NonSIunits.Temperature_degC T_degC=
          Modelica.SIunits.Conversions.to_degC(T)
        "Temperature of medium in [degC]";
      SI.Conversions.NonSIunits.Pressure_bar p_bar=
          Modelica.SIunits.Conversions.to_bar(p)
        "Absolute pressure of medium in [bar]";

      // Local connector definition, used for equation balancing check
      connector InputAbsolutePressure = input SI.AbsolutePressure
        "Pressure as input signal connector";
      connector InputSpecificEnthalpy = input SI.SpecificEnthalpy
        "Specific enthalpy as input signal connector";
      connector InputMassFraction = input SI.MassFraction
        "Mass fraction as input signal connector";

    equation
```

**Figure 14 Variables browser for the correct model**

This variable has been eliminated by the solver, but is still a part of the result file.

**Conclusion**

For this error the debugger gives not a clear pointer to the problem. It is helpful by pointing to an equation that should not be there, and giving the message that at initialization a division by zero occurred in that equation. It's an open question how many users that would have been helped by this information. But for sure the debugger puts focus on the heat transfer to the pipe, and then it is very likely that the users note that the parameter use_HeatTransfer is wrongly false. It is the authors recommendation that the OM is changed so that connection of none activated connectors becomes impossible.

## 3.3 Errors in the logarithmic temperature difference calculation

### 3.3.1 Motivation

Errors in the logarithmic temperature difference calculation should be treated the same way as the division by zero. Interesting is however, if the solver also for this types of errors manage to pass the critical point as their time duration could be expected to be very short. Basically this investigation is more an investigation of the solver and not the debugger but the debugger will be activated and therefore also this investigation is a part of the report.

### 3.3.2 Temperature differences passing zero

This test is achieved by removing the numerical fences that prevents zero passage. Unfortunately, it turns out that there are no passages that passes delT=0 for the case simulated, and the test needs further work to be carried out, and therefore postponed and not published in this issue of the report. This is unexpected and errors might have occurred at the transition from Dymola to OM.

### 3.3.3 Temperature differences at outlet and inlet passing equal values

This is happening without any numerical problems, i.e. the solver skips the critical time where they are equal or happens to avoid it without any actions. No further action for this issue of the report.

## 3.4 Boiling in evaporator

### 3.4.1 Motivation

This test is to see how debugger treats situations where solver gets stuck due to conditions in the process that forces solver to take very small steps in order to fulfil error tolerance requirement ($10^{-6}$).

### 3.4.2 Result

This was caused already in chapter 3.2.2 and no additional test is necessary.

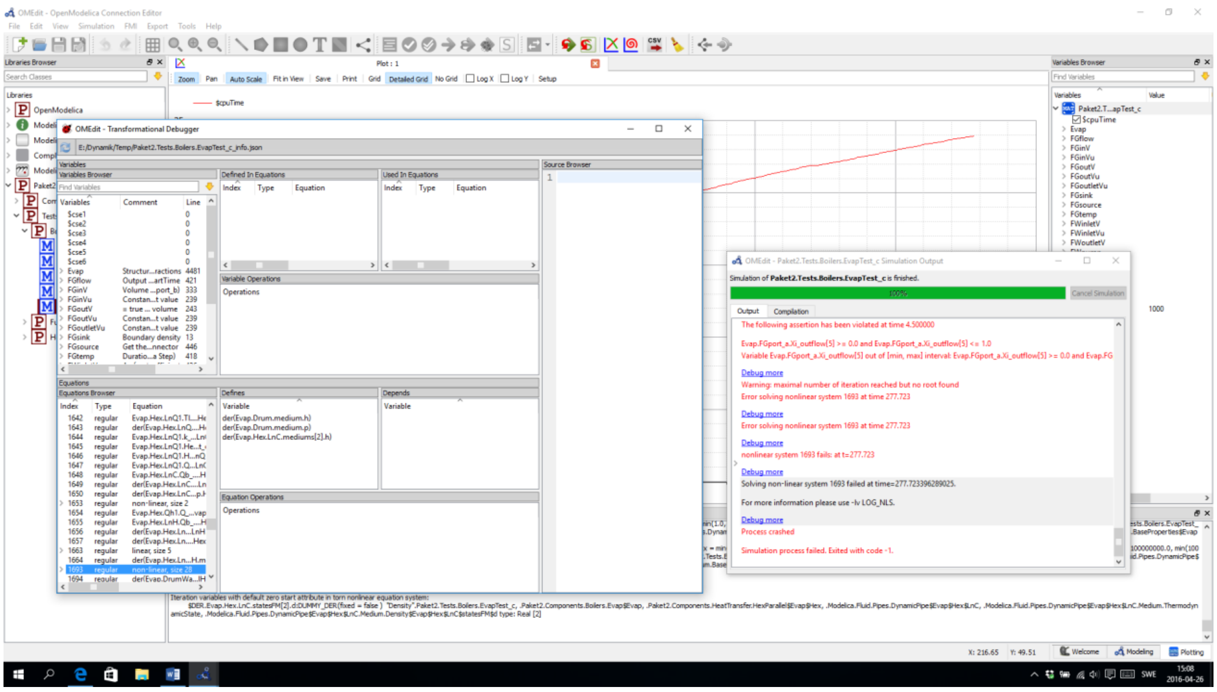
## 3.5 Bad initial values or bad simulation boundaries

### 3.5.1 Motivation

The debugger support, if any, is to be investigated for this type of problems where simulation runs into numerical problems.

### 3.5.2 Too high backpressure

By increasing the back pressure from the steam pipes to exceed drum pressure, and thus preventing steam flow out of the drum following output is received (Figure 15).

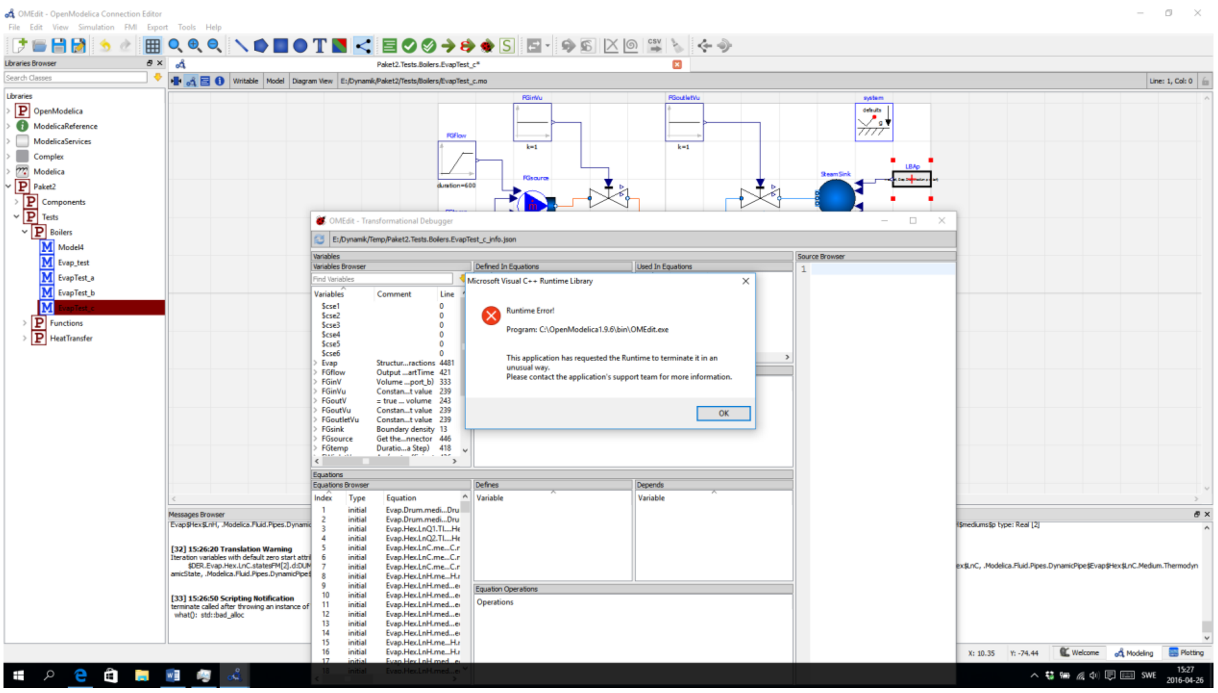


```
Simulation of Paket2.Tests.Boilers.EvapTest_c is finished.

The following assertion has been violated at time 4.500000

Evap.FGport_a.Xi_outflow[5] >= 0.0 and Evap.FGport_a.Xi_outflow[5] <= 1.0
Variable Evap.FGport_a.Xi_outflow[5] out of [min, max] interval: Evap.FGport_a.Xi_outflow[5] >= 0.0 and Evap.FG

Debug more
Warning: maximal number of iteration reached but no root found
Error solving nonlinear system 1693 at time 277.723

Debug more
Error solving nonlinear system 1693 at time 277.723

Debug more
nonlinear system 1693 fails: at t=277.723

Debug more
Solving non-linear system 1693 failed at time=277.723396289025.

For more information please use -lv LOG_NLS.

Debug more
Process crashed

Simulation process failed. Exited with code -1.
```

```
Iteration variables with default zero start attribute in torn nonlinear equation system:
       $DER.Evap.Hex.LnC.statesFM[2].d:DUMMY_DER(fixed = false )  "Density".Paket2.Tests.Boilers.EvapTest_c, .Paket2.Components.Boilers.Evap$Evap, .Paket2.Components.HeatTransfer.HexParallel$Evap$Hex, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnC, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnC.Medium.ThermodynamicState, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnC.Medium.Density$Evap$Hex$LnC$statesFM$d type: Real [2]
```

**Figure 15 Output windows at too high back pressure**

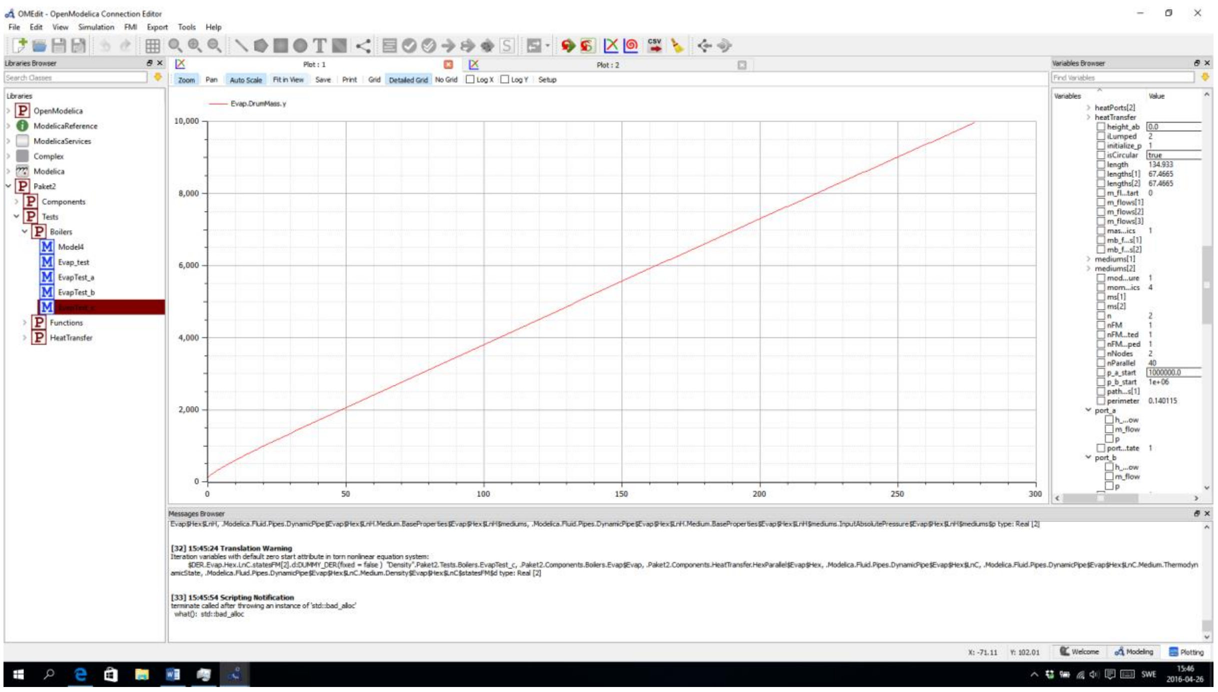
The simulation terminates at 277.7 s. The result file is written, i.e. it is possible to plot. The plotting reveals that the simulation crash is probably due to that the drum gets filled with water. The TDW points at the drum. The SOW recommends to log NLS. Doing this gives a not responding OM. Restart gives a runtime error (Figure 16 Unexpected outcome at repeat of simulation with log_NLS activated)



```
Microsoft Visual C++ Runtime Library

Runtime Error!

Program: C:\OpenModelica1.9.6\bin\OMEdit.exe

This application has requested the Runtime to terminate it in an
unusual way.
Please contact the application's support team for more information.
```

**Figure 16 Unexpected outcome at repeat of simulation with log_NLS activated**

A restart and simulation again without LOG_NLS activated gives the same result as in Figure 15. The plotting of the Drum parameter mass shows that drum gets filled as it has no outlet (Figure 17).



```
[32] 15:45:24 Translation Warning
Iteration variables with default zero start attribute in torn nonlinear equation system:
       $DER.Evap.Hex.LnC.statesFM[2].d:DUMMY_DER(fixed = false )  "Density".Paket2.Tests.Boilers.EvapTest_c, .Paket2.Components.Boilers.Evap$Evap, .Paket2.Components.HeatTransfer.HexParallel$Evap$Hex, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnC, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnC.Medium.ThermodynamicState, .Modelica.Fluid.Pipes.DynamicPipe$Evap$Hex$LnC.Medium.Density$Evap$Hex$LnC$statesFM$d type: Real [2]

[33] 15:45:54 Scripting Notification
terminate called after throwing an instance of 'std::bad_alloc'
  what():  std::bad_alloc
```

**Figure 17 Plot of drum mass at blocked drum outlet**

**Conclusion**

The debugger test failed here on OM problem with handling LOG_NLS. However, at this error the result file was generated and provides a tool for debugging. On the other hand, LOG_NLS is not a part of the debugger. The debugger information for this type of failure is not sufficient to remedy the problem directly, although it points at the drum as a probable cause. Eventually the recommended logging of NLS could have given the direct cause of crash. From OM user point of view, the plotting after crash is very valuable, and it reveals that the drum gets filled from the LBA system (no check valves in this test model), which calls for corrective actions regardless of the what caused the actual solver crash.

# 4 Constraint

The basic property of the debugger is at numerical problems and violation of assertions like numerical ranges that a certain variable is expected never to exceed. Then the debugger points out the equation causing the problem. In this report only small models have been tested, that shows that the debugger works. To really evaluate the benefits of the debugger, but also its functionality, it should be applied to larger models.