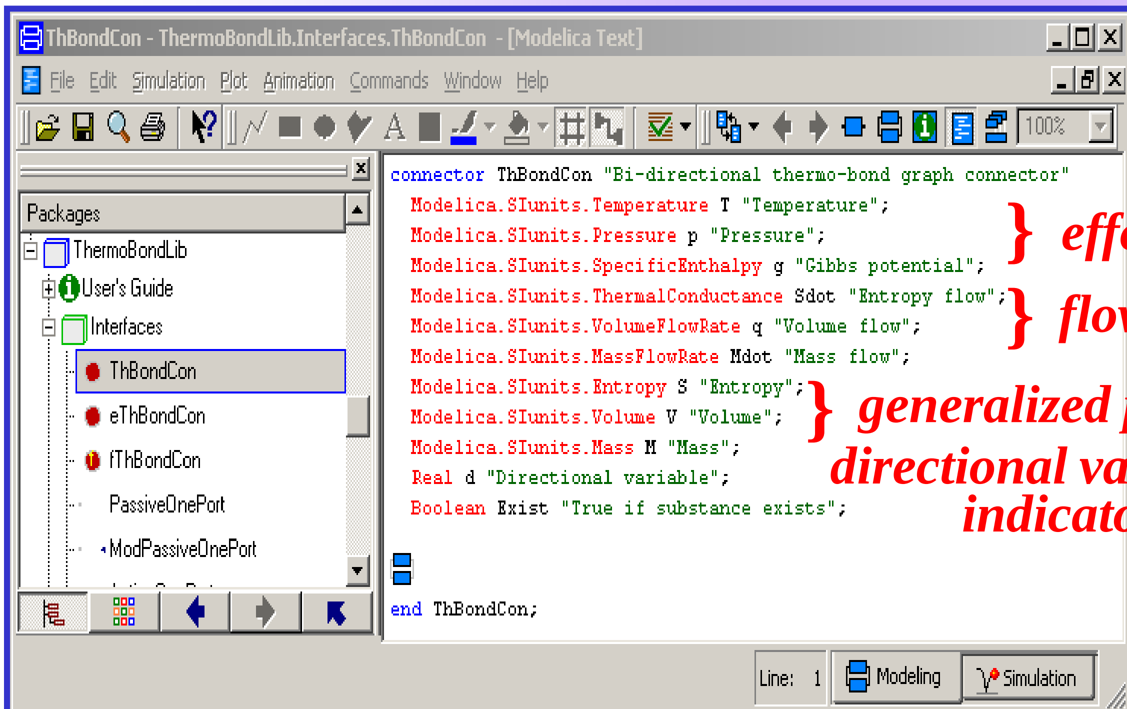
The *Dymola* Thermo-Bond-Graph Library

- In this lecture, we shall introduce a third bond-graph library, one designed explicitly to deal with ***convective flows***.
- To this end, we shall need to introduce a new type of bonds, bonds carrying in parallel three distinct, yet inseparable, power flows: a ***heat flow***, a ***volume flow***, and a ***mass flow***.
- These new ***bus-bonds***, together with their corresponding ***bus-0-junctions***, enable the modeler to describe convective flows at a high abstraction level.
- The example of a ***pressure cooker*** model completes the presentation.

Table of Contents

The Thermo-Bond Graph Connectors I

- We shall need to introduce new ***thermo-bond graph connectors*** to carry the six variables associated with the three flows. They are designed as an ***11-tuple***.



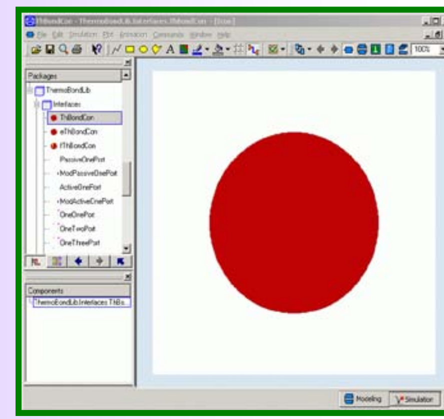
} *efforts, e*

} *flows, f*

} *generalized positions, q*

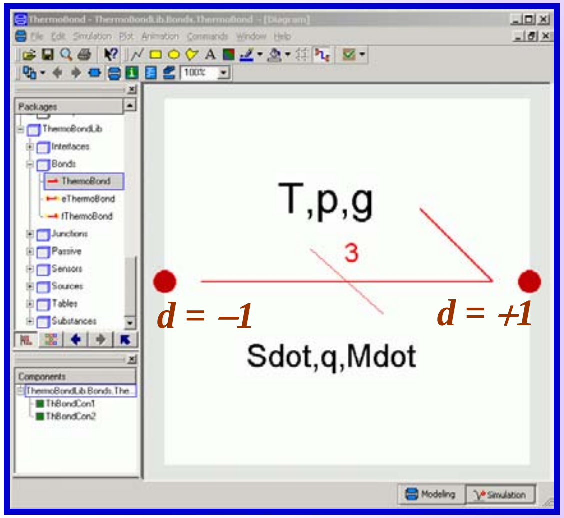
directional variable, d

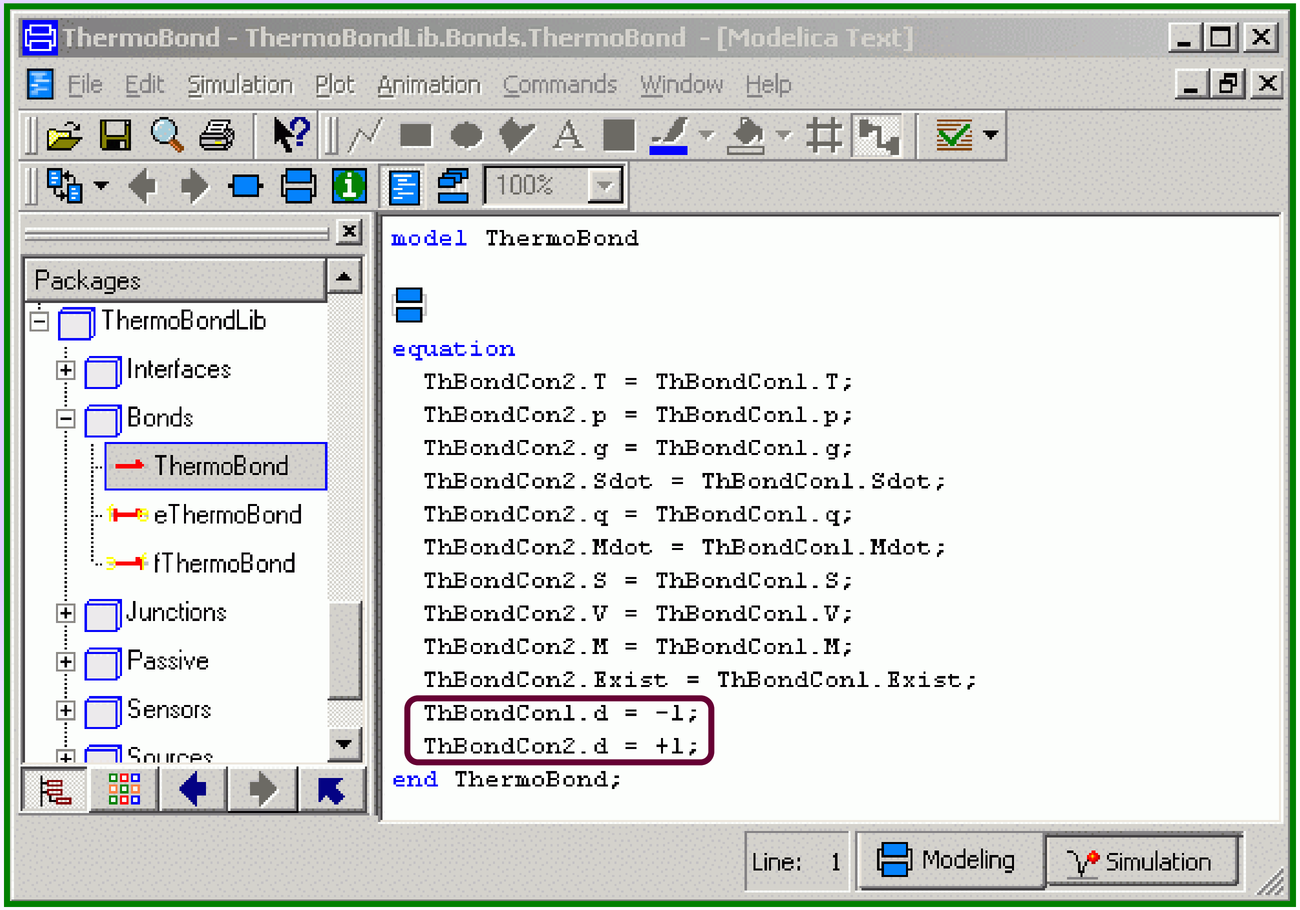
indicator variable



The thermo-bond connector icon is a red dot.

The A-Causal Thermo-Bond Model

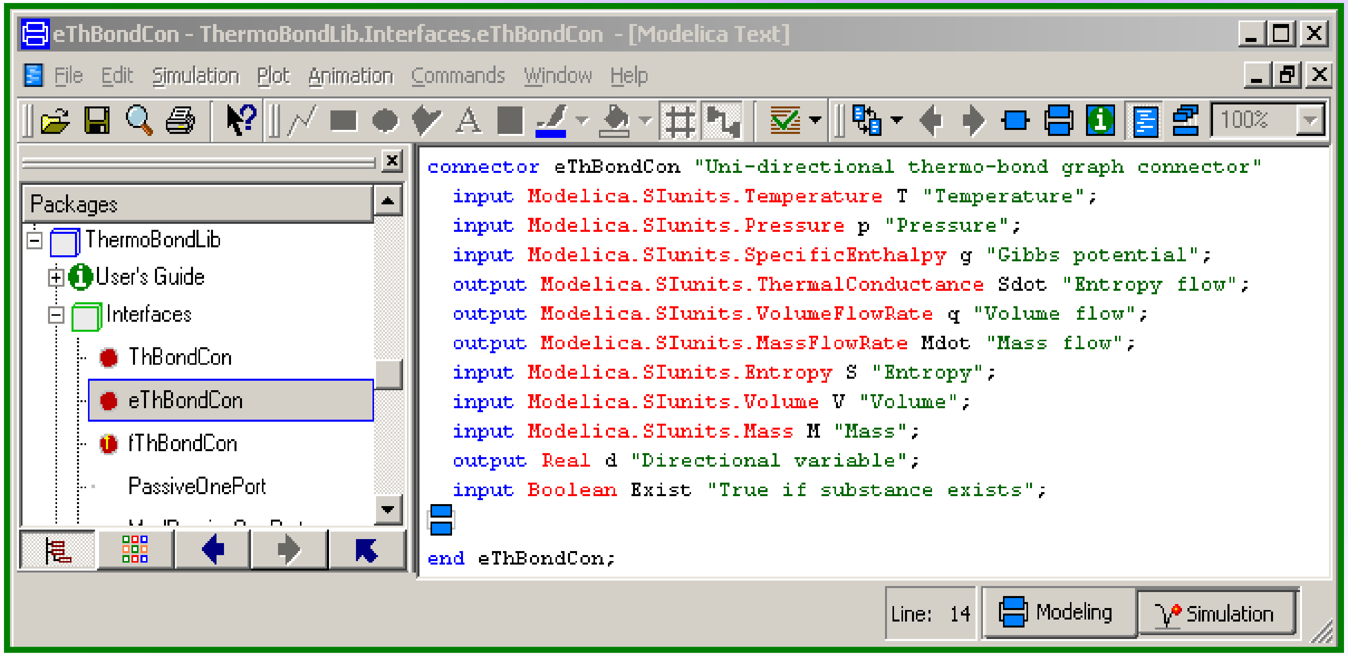


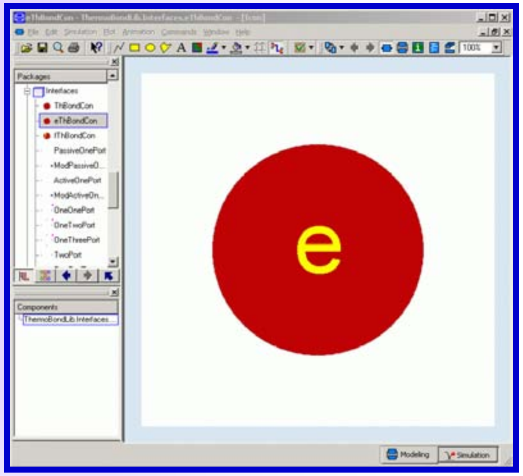


The thermo-bond is equivalent to the regular bond, except that ten variables need to be connected across, instead of only two.

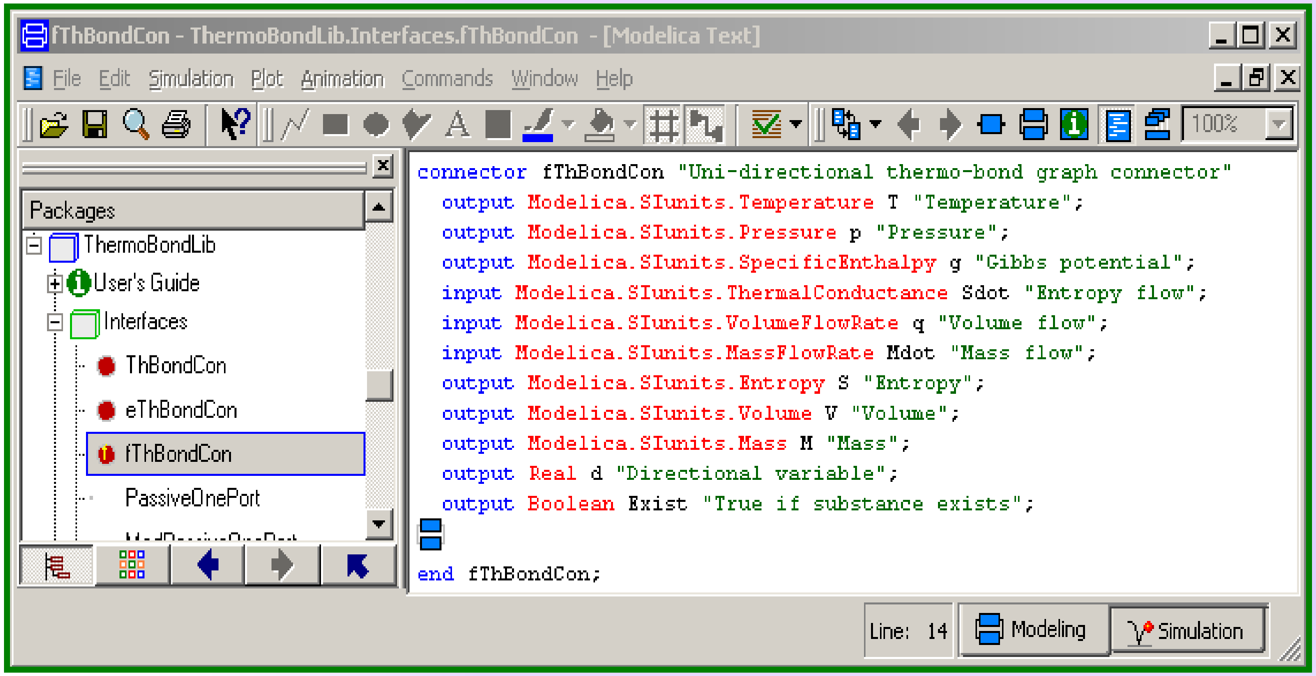
The Thermo-Bond Graph Connectors II

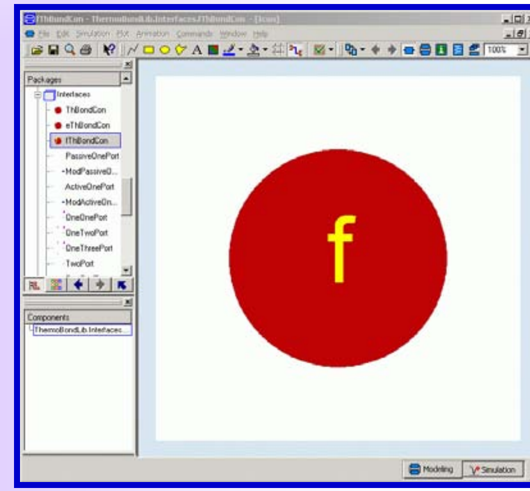
- Like in the case of the general bond-graph library, also the thermo-bond-graph library offers ***causal*** next to ***a-causal*** bonds.





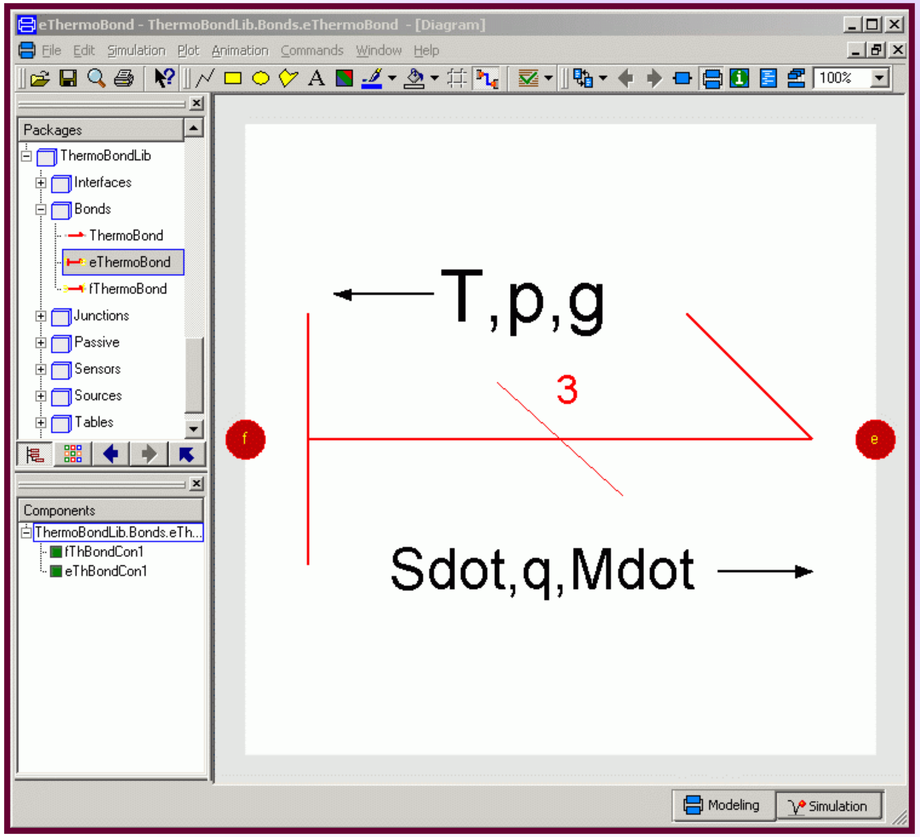
The Thermo-Bond Graph Connectors III





- Either the three efforts or the three flows are treated as input variables.

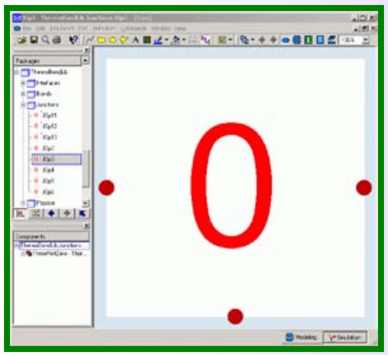
The Causal Thermo-Bond Blocks

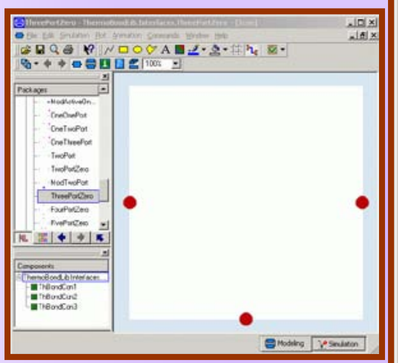


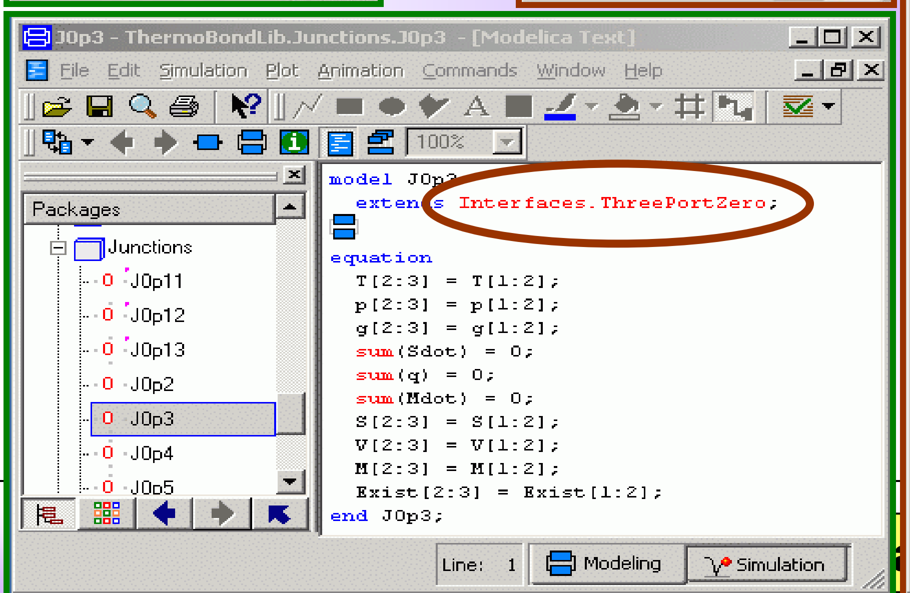
- Using these connectors, causal thermo-bond blocks can be defined.
- The ***f-connector*** is used at the side of the causality stroke.
- The ***e-connector*** is used at the other side.
- The causal thermo-bond-graph connectors are only used in the context of the thermo-bond blocks. Everywhere else, the a-causal thermo-bond-graph connectors are to be used.

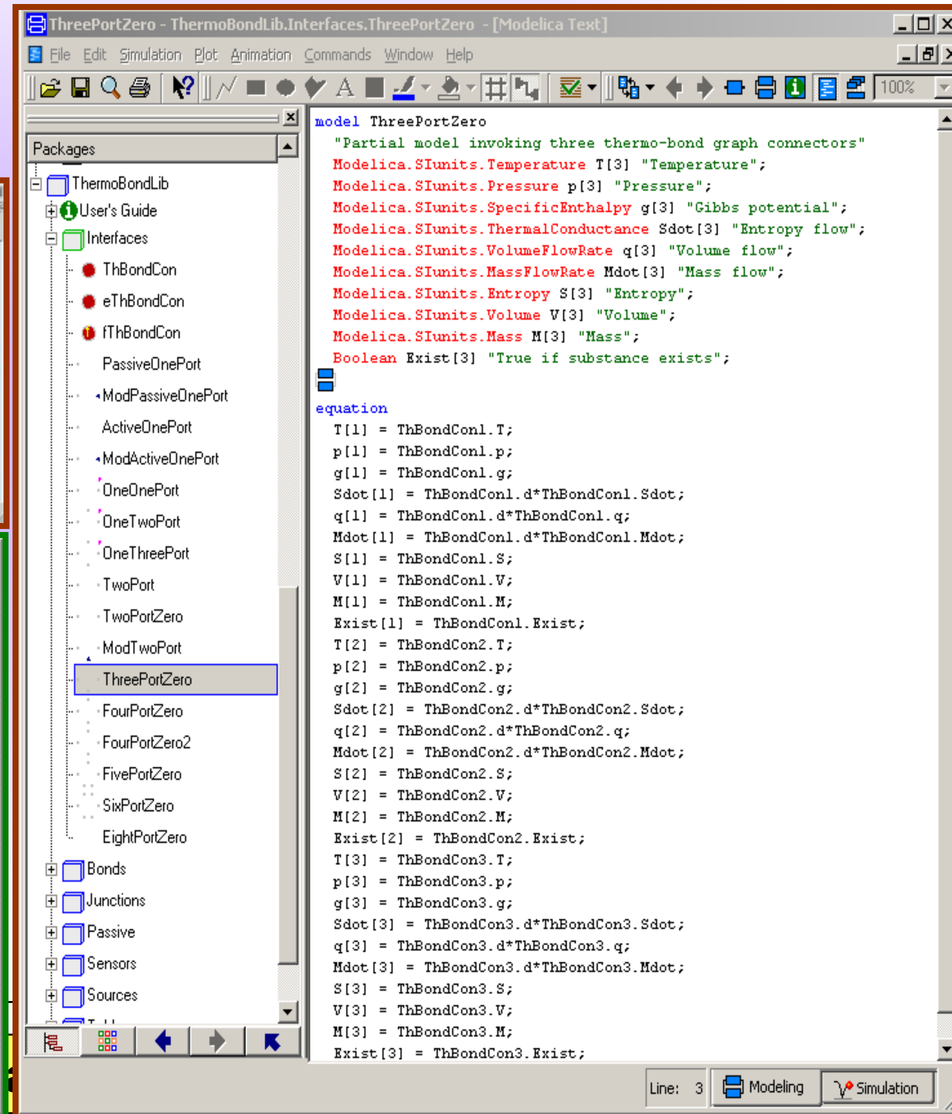
The Bus-0-Junctions

The junctions can now be programmed. Let us look at a ***bus-0-junction with three bond attachments***.

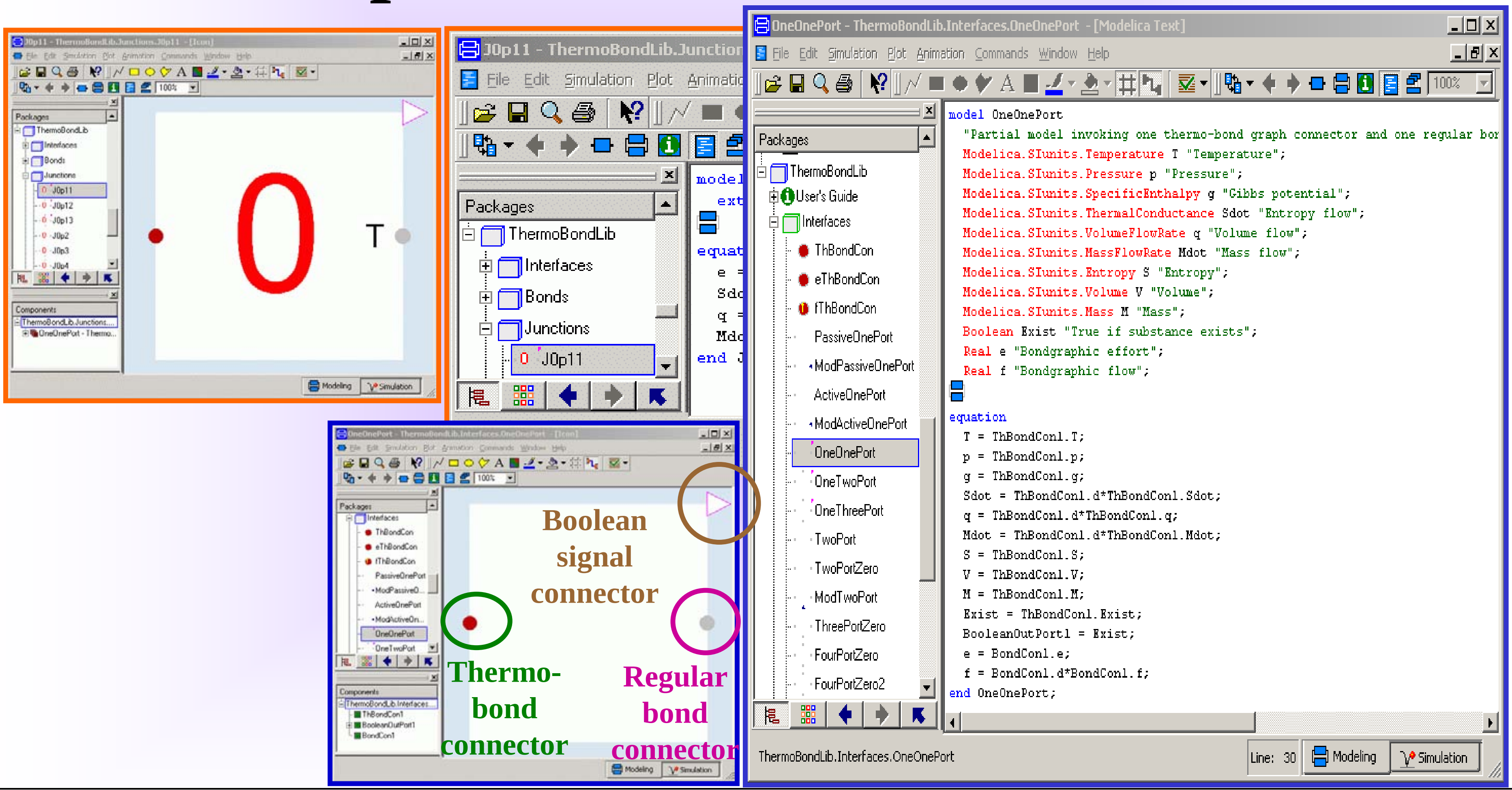




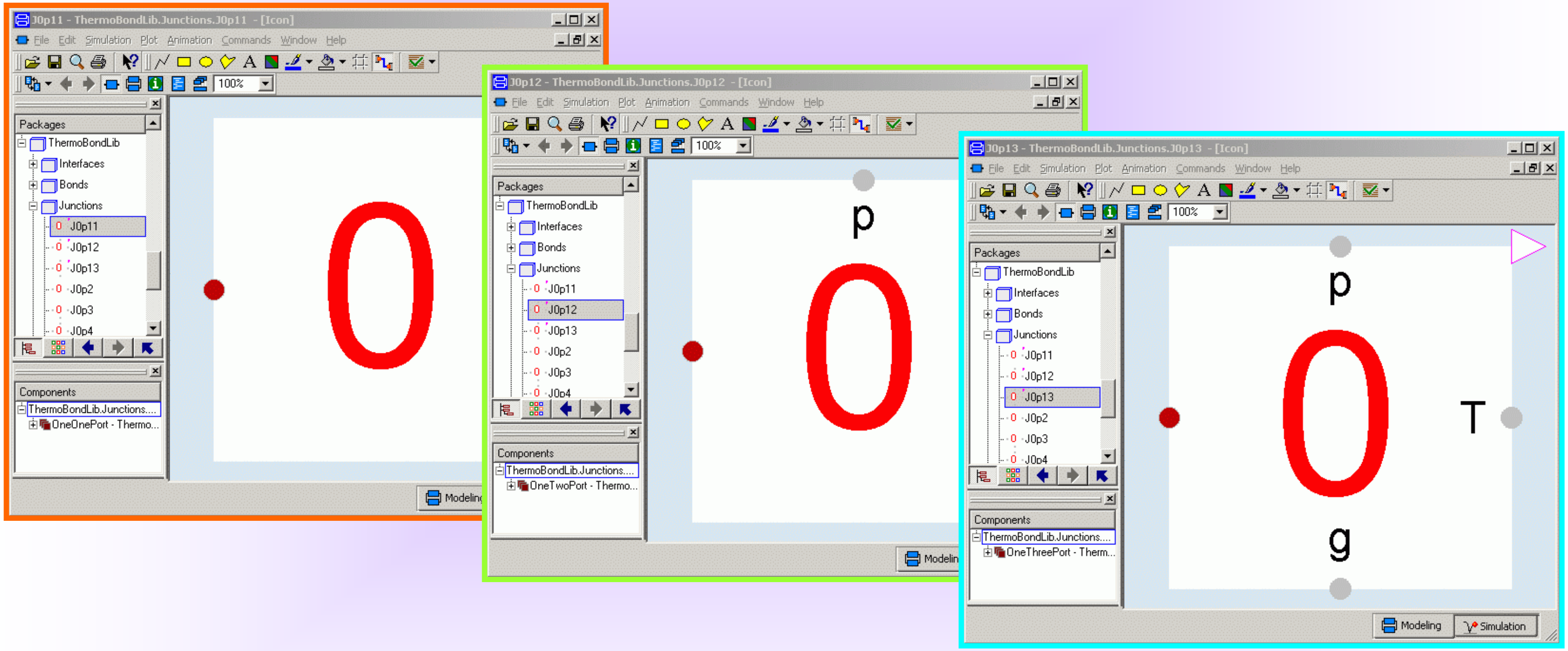




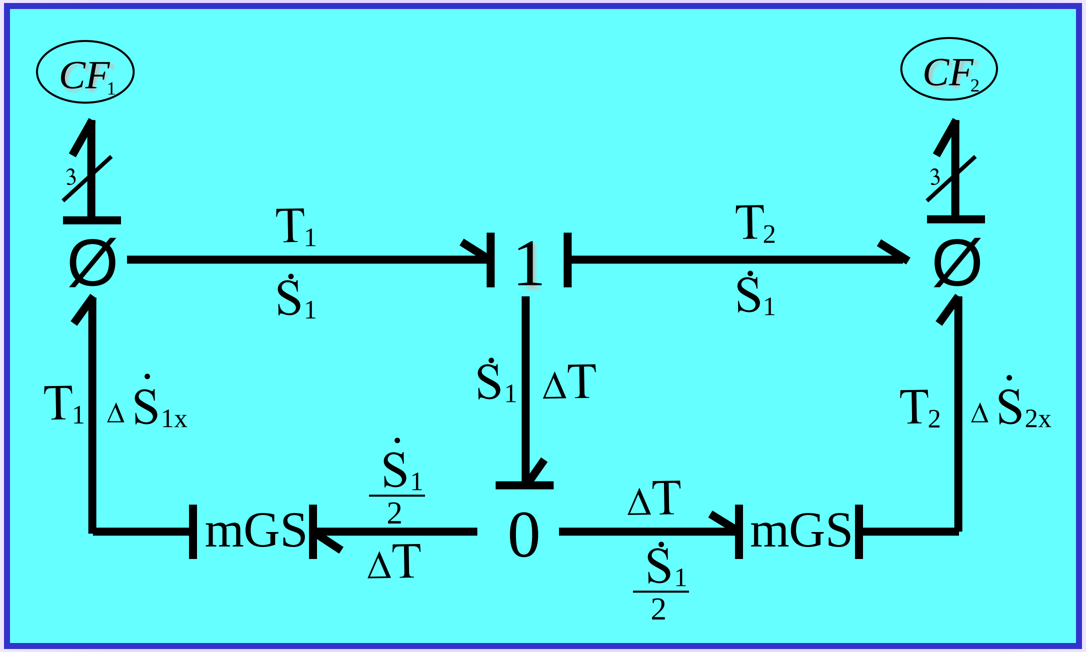
Special Bus-0-Junctions I

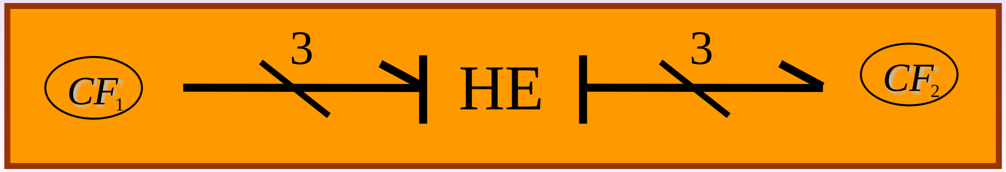


Special Bus-0-Junctions II

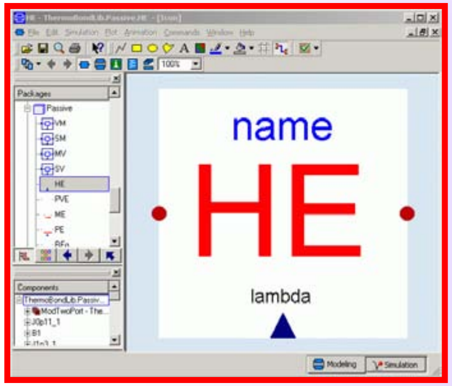


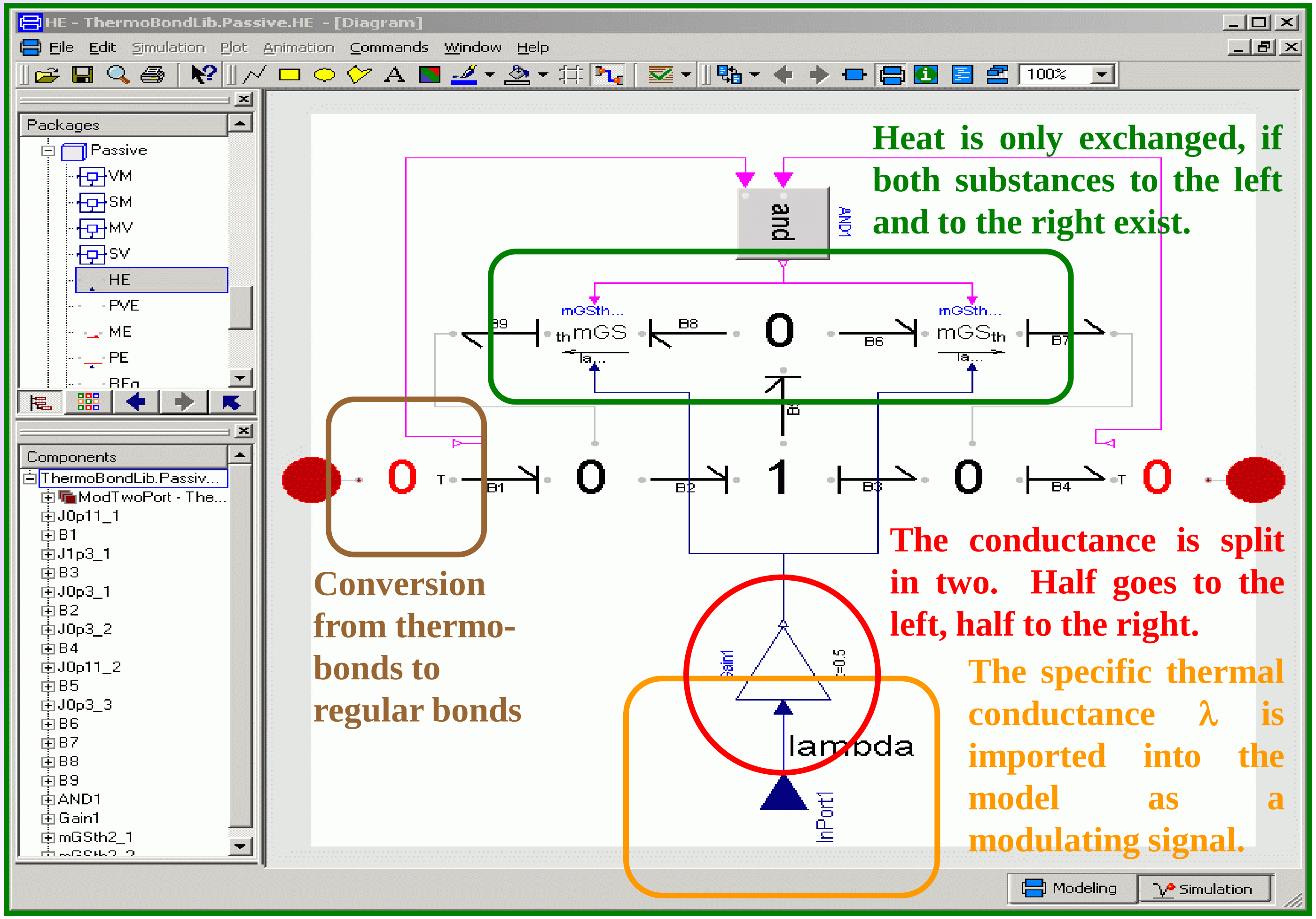
The Heat Exchanger



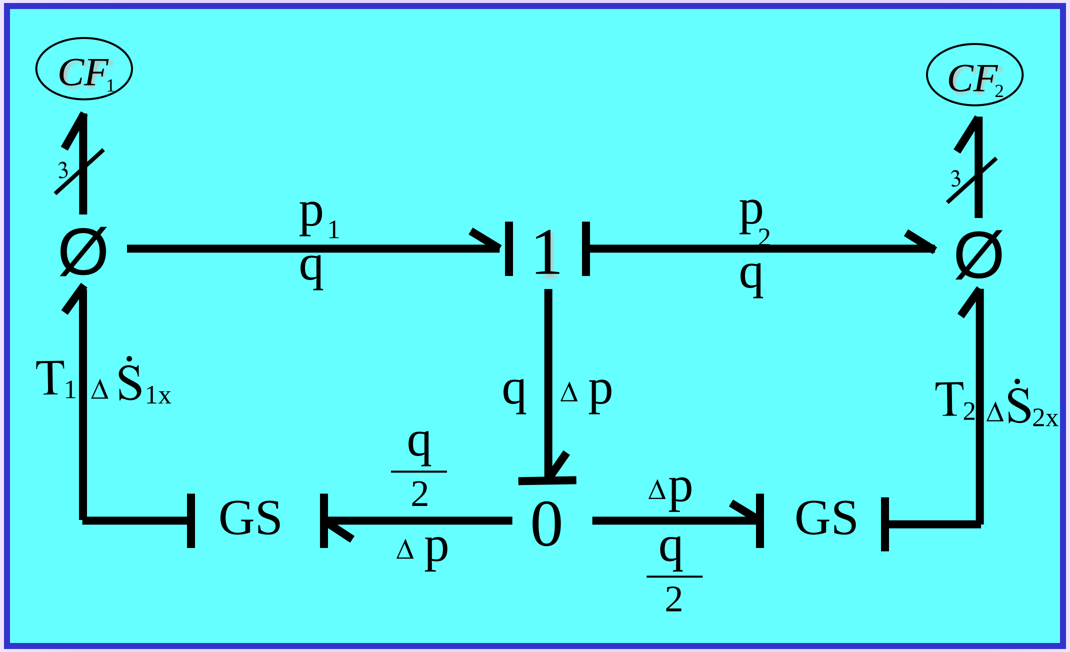


The Thermo-Bond Heat Exchange Element

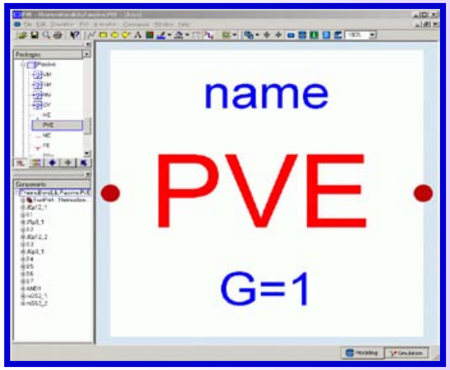




The Volume Work



The Pressure Volume Exchange Element

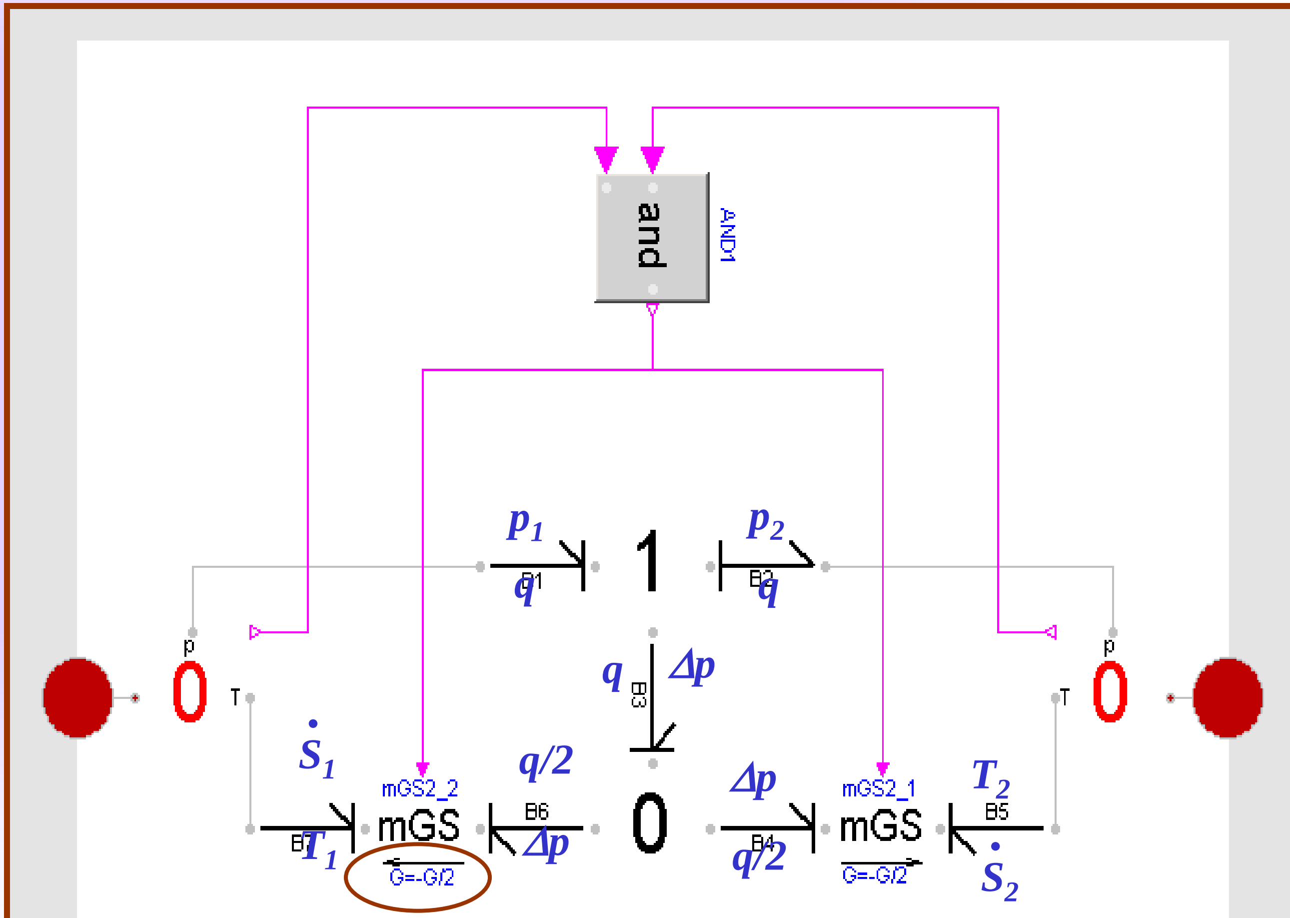


We pick up the two pressures.

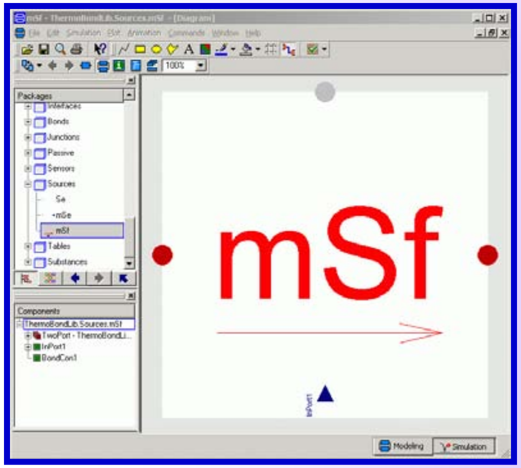
The difference between the two pressures causes a negative volume flow.

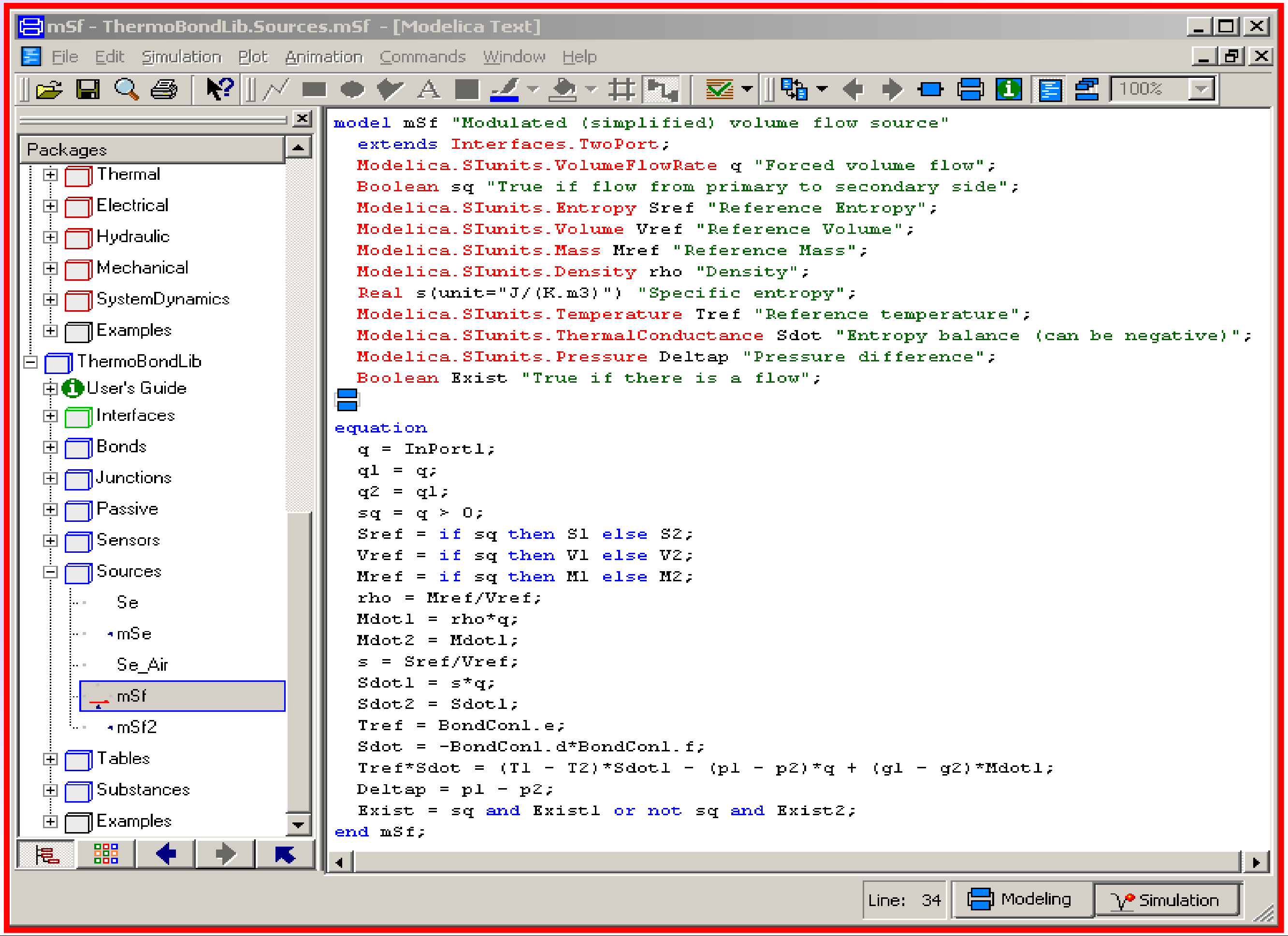
The surplus power is split in two to be converted into additional entropy.

The generated entropy is delivered back to the thermal ports.



Forced Volume Flow I





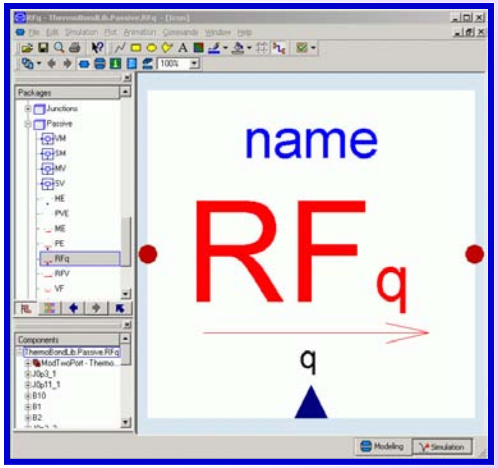

```
model mSf "Modulated (simplified) volume flow source"
  extends Interfaces.TwoPort;
  Modelica.SIunits.VolumeFlowRate q "Forced volume flow";
  Boolean sq "True if flow from primary to secondary side";
  Modelica.SIunits.Entropy Sref "Reference Entropy";
  Modelica.SIunits.Volume Vref "Reference Volume";
  Modelica.SIunits.Mass Mref "Reference Mass";
  Modelica.SIunits.Density rho "Density";
  Real s(unit="J/(K.m3)") "Specific entropy";
  Modelica.SIunits.Temperature Tref "Reference temperature";
  Modelica.SIunits.ThermalConductance Sdot "Entropy balance (can be negative)";
  Modelica.SIunits.Pressure Deltap "Pressure difference";
  Boolean Exist "True if there is a flow";

equation
  q = InPort1;
  q1 = q;
  q2 = q1;
  sq = q > 0;
  Sref = if sq then S1 else S2;
  Vref = if sq then V1 else V2;
  Mref = if sq then M1 else M2;
  rho = Mref/Vref;
  Mdot1 = rho*q;
  Mdot2 = Mdot1;
  s = Sref/Vref;
  Sdot1 = s*q;
  Sdot2 = Sdot1;
  Tref = BondCon1.e;
  Sdot = -BondCon1.d*BondCon1.f;
  Tref*Sdot = (T1 - T2)*Sdot1 - (p1 - p2)*q + (g1 - g2)*Mdot1;
  Deltap = p1 - p2;
  Exist = sq and Exist1 or not sq and Exist2;
end mSf;
```

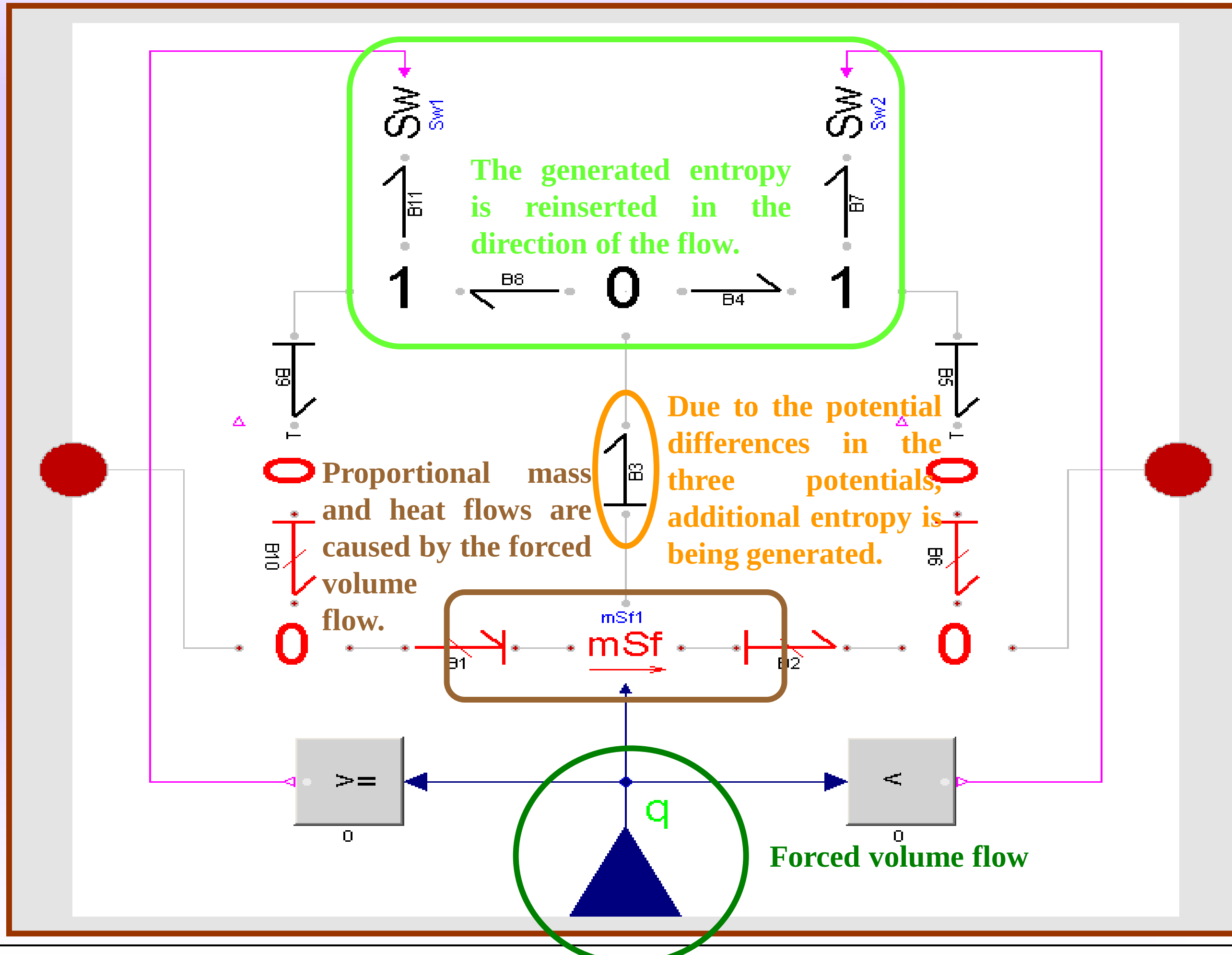
The forced volume flow causes a proportional forced mass flow and an also proportional forced entropy flow.

Forced Volume Flow II

- The model presented here cannot yet be used to represent e.g. a ***pump*** or a ***compressor***, because it doesn't consider the power needed to move the fluid around.
- The model is acceptable to describe small mass movements such as pressure equilibrations between the bulk and a (mathematical) boundary layer.
- An improved forced volume flow model shall be discussed later in this lecture.

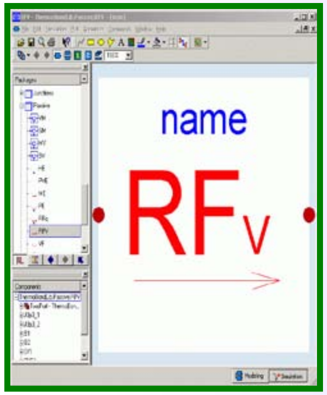
The Resistive Field

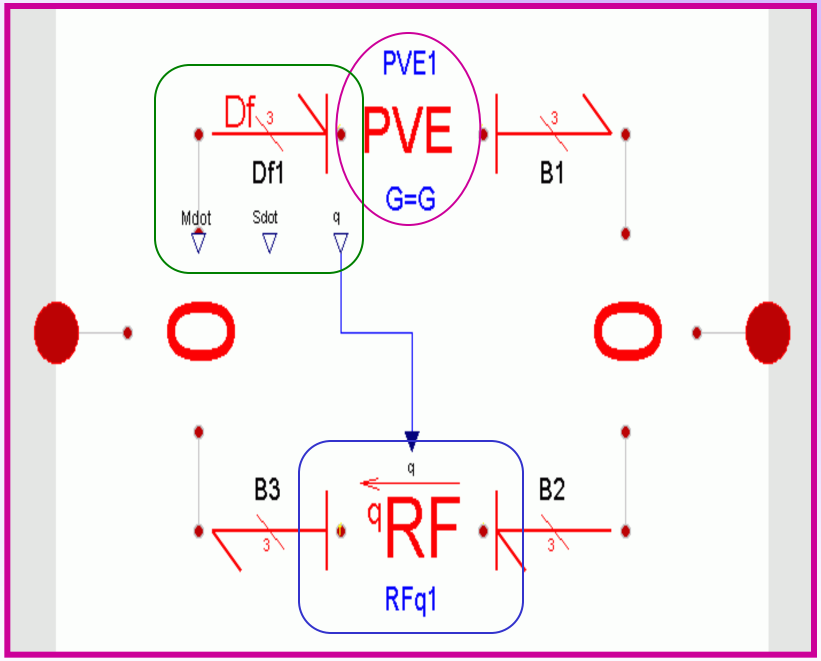




Pressure Equilibration With Constant Volume

- Sometimes it is useful to allow a mass flow to take place, while the volume doesn't change (remember the gas cartridge).





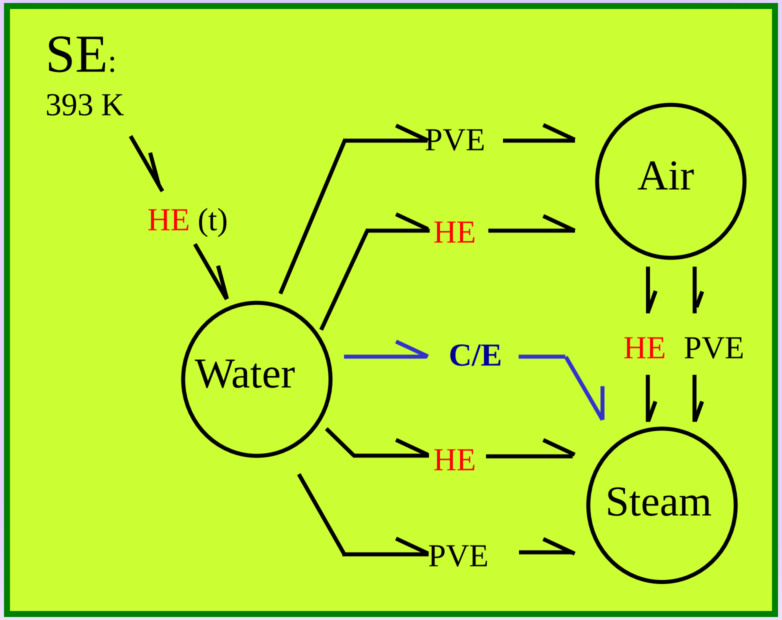
Pressure equilibration causes a volume flow to occur.

A flow sensor element measures the volume flow taking place.

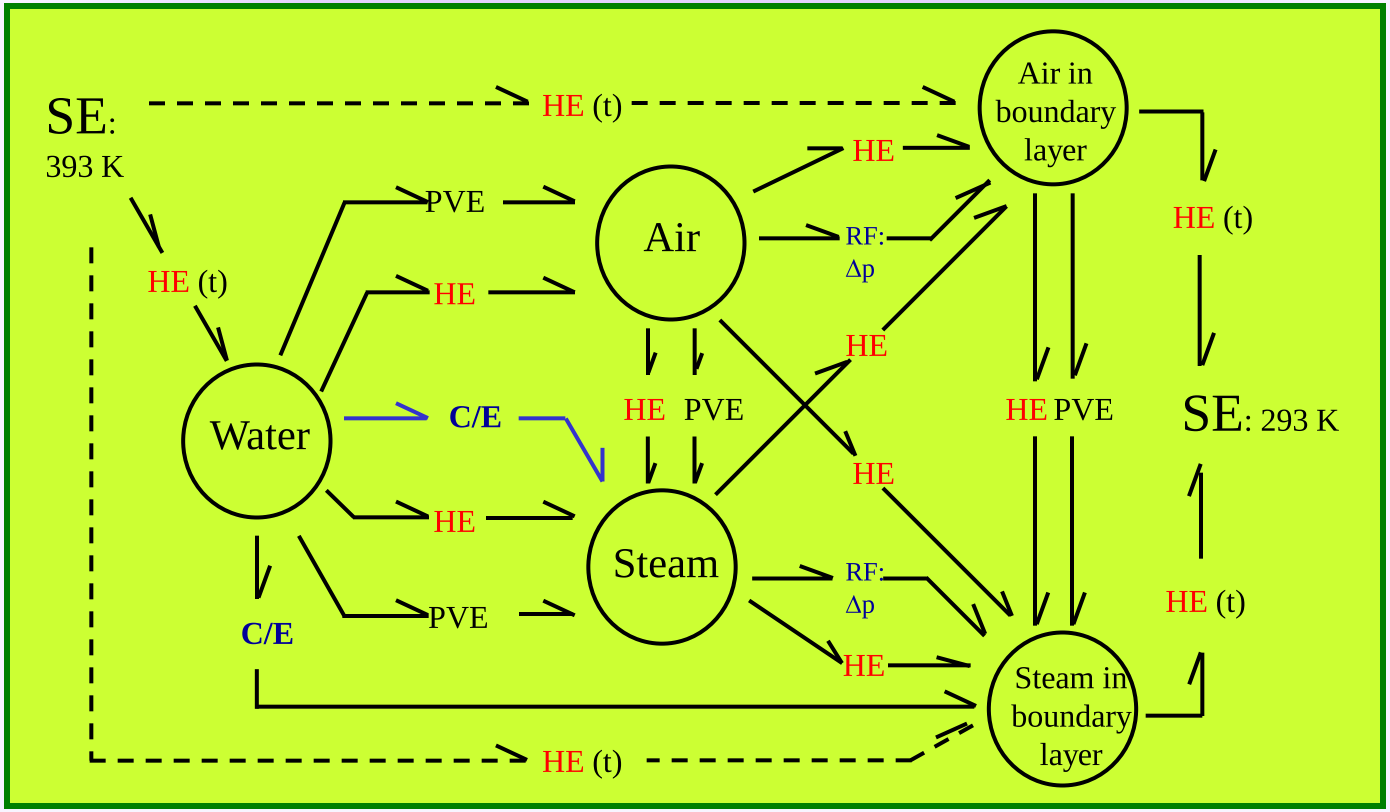
A counter volume flow of equal size is forced.

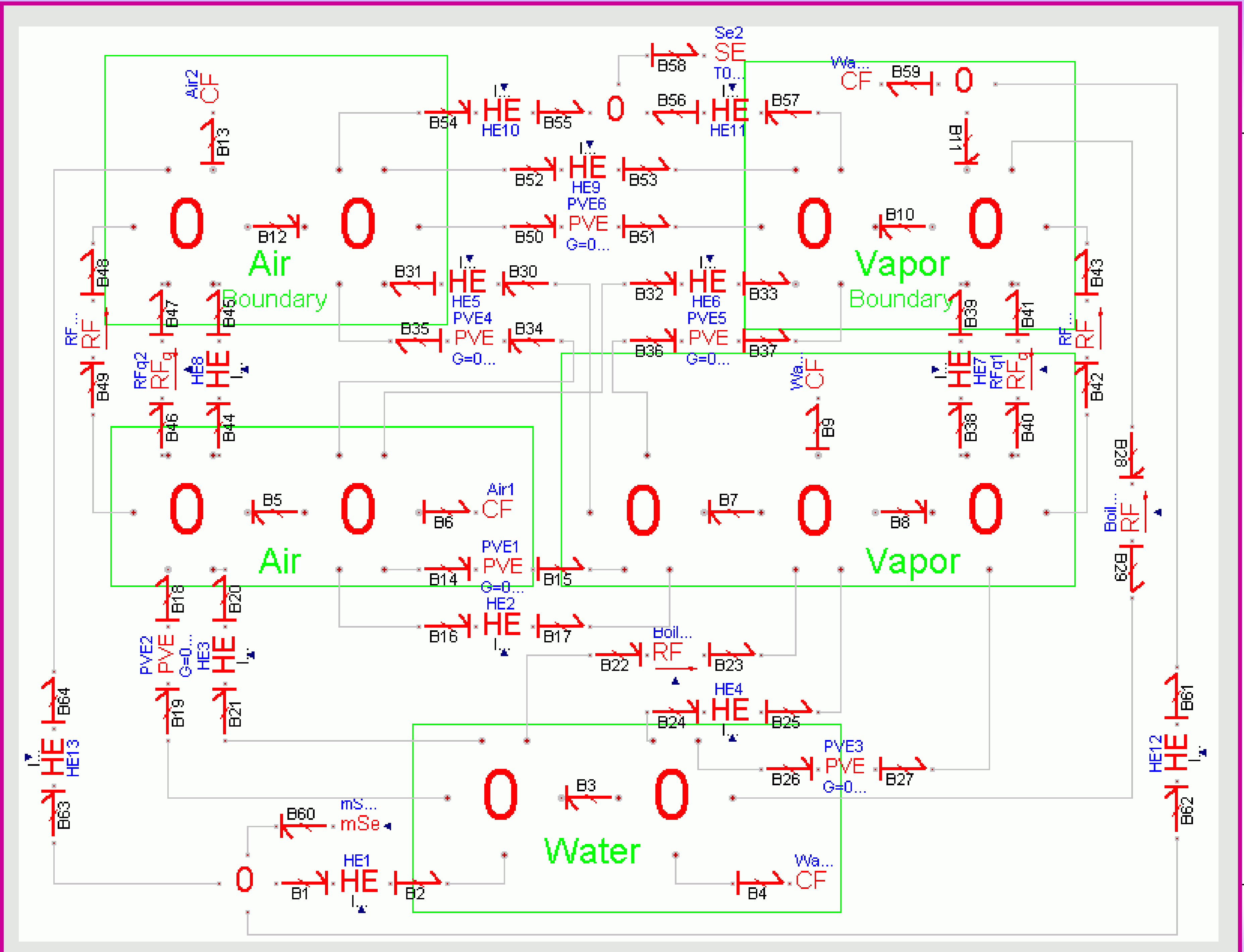
The pressure gets equilibrated, but the volume remains the same.

The Pressure Cooker I



The Pressure Cooker II




Air
Boundary
Vapor
Boundary
Air
Vapor
Water
Se2
SE
HE10
HE11
HE9
PVE6
PVE
HE5
PVE4
HE6
PVE5
Air1
CF
PVE1
HE2
HE4
PVE3
HE1
mSe
HE3
PVE2
HE8
HE7
HE12
HE13
RF
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
B26
B27
B28
B29
B30
B31
B32
B33
B34
B35
B36
B37
B38
B39
B40
B41
B42
B43
B44
B45
B46
B47
B48
B49
B50
B51
B52
B53
B54
B55
B56
B57
B58
B59
B60
B61
B62
B63
B64

Capacitive Fields

- Let us look at the capacitive field for air.

Linear capacitive field:

$$\mathrm{der}(e) = C^{-1} \cdot f$$

By integration:

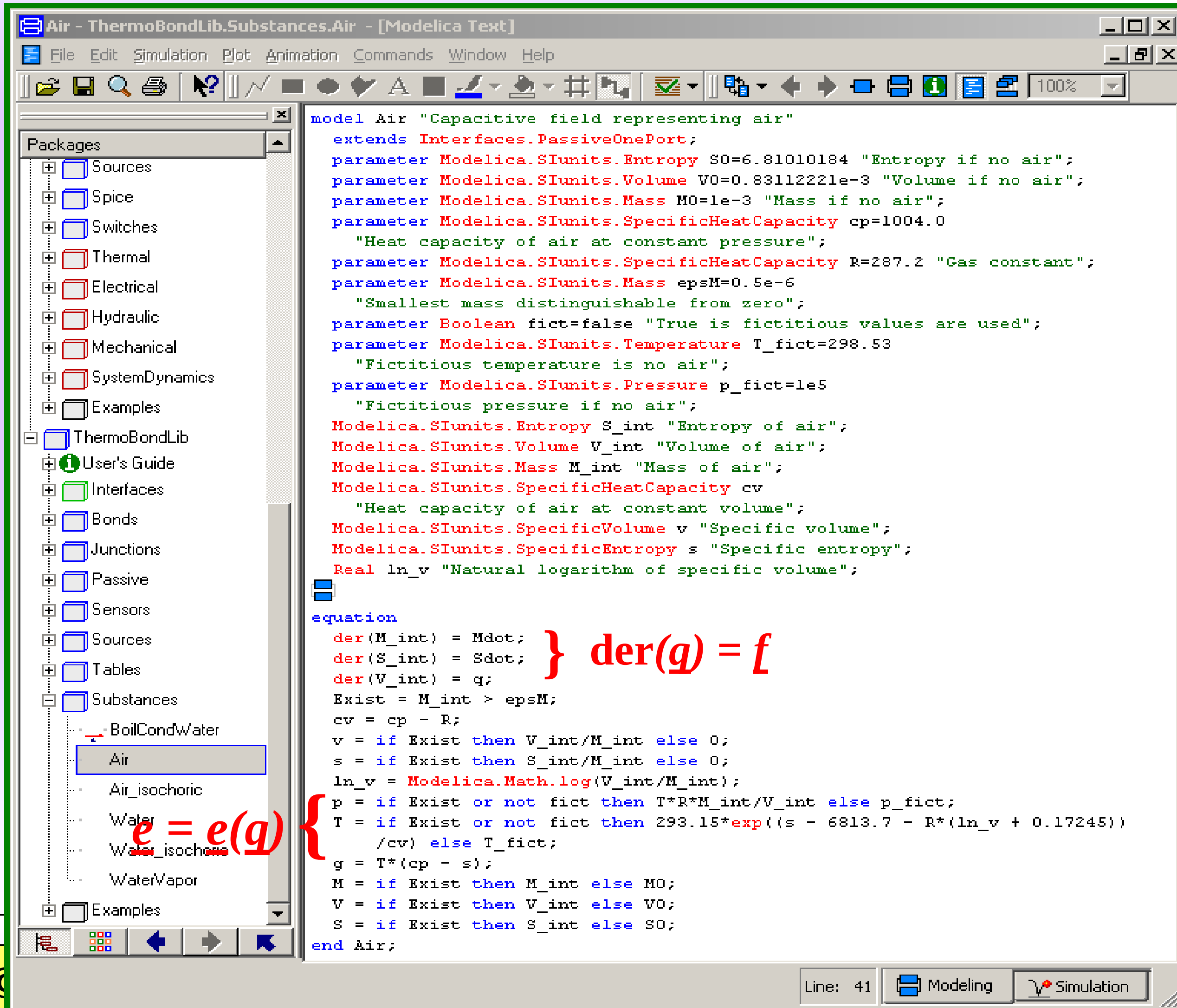
$$\mathrm{der}(q) = f$$

$$e = C^{-1} \cdot q$$

Non-linear capacitive field:

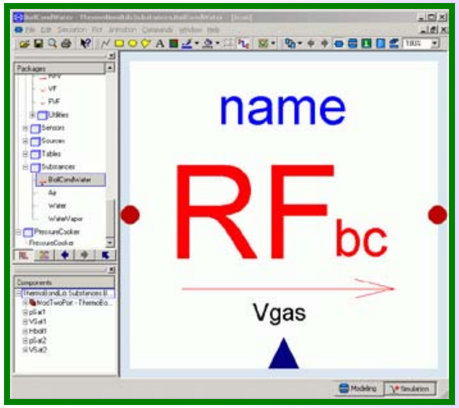
$$\mathrm{der}(q) = f$$

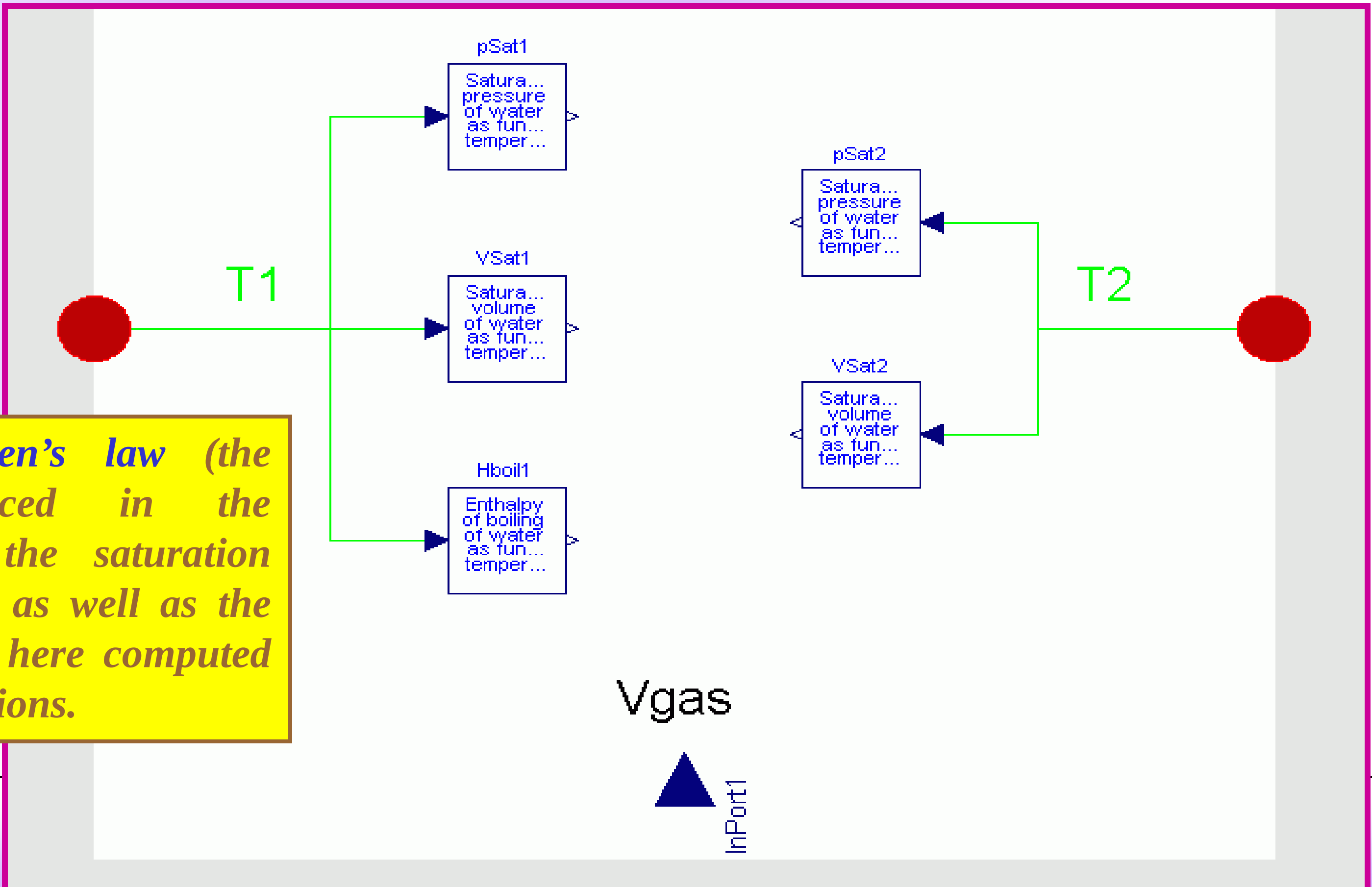
$$e = e(q)$$



Evaporation and Condensation I

- The models describing ***evaporation*** and ***condensation*** are constructed by interpolation from steam tables.





Instead of using Teten's law (the approximation embraced in the Biosphere 2 model), the saturation pressures and volumes, as well as the enthalpy of boiling are here computed using table-lookup functions.

Evaporation and Condensation II

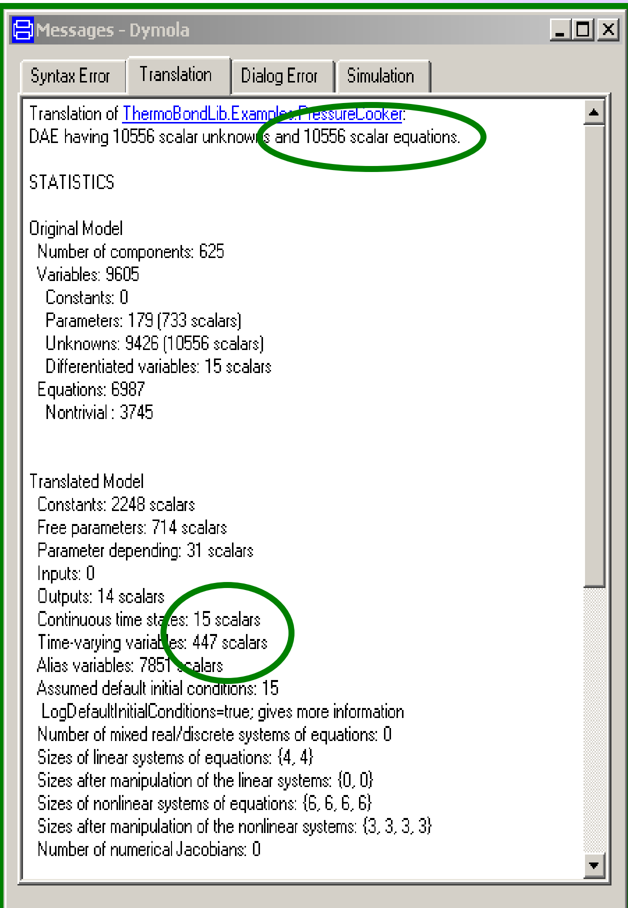
```
model BoilCondWater "Boiling and condensation of water"
  extends Interfaces.ModTwoPort(InPort
          Modelica.SIunits.Volume));
  parameter Real Rb(unit="m.s") = 0.01
  parameter Real Rc(unit="m.s") = 0.01
  Modelica.SIunits.Pressure pSat_liq "
  Modelica.SIunits.Pressure pSat_gas "
  Modelica.SIunits.SpecificVolume VSat
  Modelica.SIunits.SpecificVolume VSat
  Modelica.SIunits.SpecificEnthalpy hb
  Modelica.SIunits.Volume Vgas "Gas vo
  Modelica.SIunits.Pressure pgas "Part
  Modelica.SIunits.MassFlowRate Mboil
  Modelica.SIunits.MassFlowRate Mcond
  Boolean fd "True if forward flow";
  Modelica.SIunits.HeatFlowRate DeltaQ
    "Heat generated by potential equil
  Boolean Exist "True if flow exists";
  Modelica.SIunits.Mass Mref "Referenc
  Modelica.SIunits.AngularFrequency xi
  Modelica.SIunits.ThermalConductance
  Modelica.SIunits.ThermalConductance
  Modelica.SIunits.Volume Vsteam "Maxi
  Real hum1 "Relative humidity compute
  Real hum2 "Relative humidity compute
```

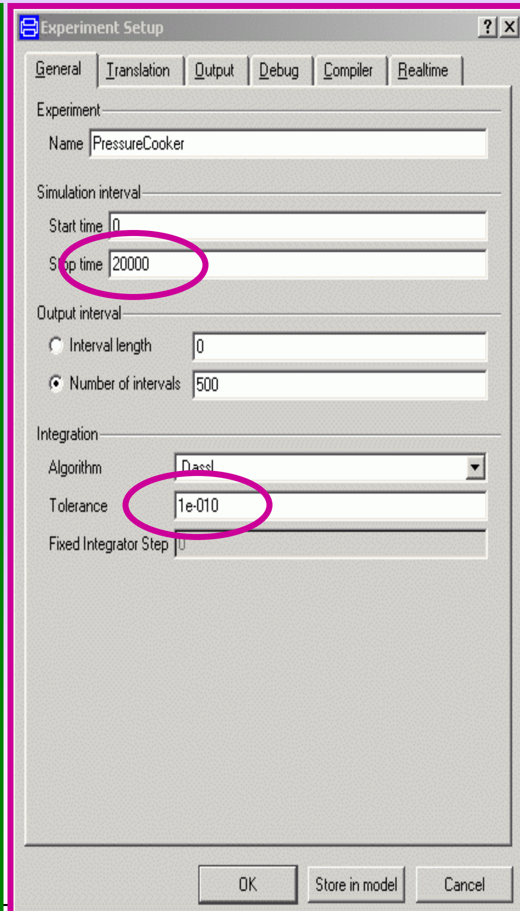
A bit messy!

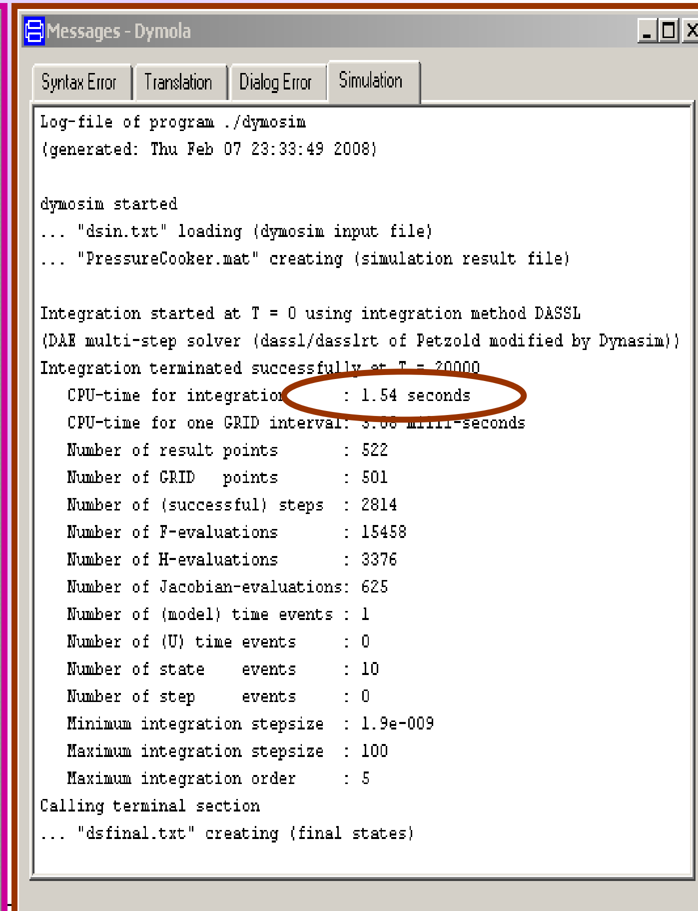
```
equation
  pSat1.u = T1;
  pSat2.u = T2;
  VSat1.u = T1;
  VSat2.u = T2;
  Hboil1.u = T1;
  pSat_liq = pSat1.y;
  pSat_gas = pSat2.y;
  VSat_liq = VSat1.y;
  VSat_gas = VSat2.y;
  hboil = Hboil1.y;
  Vgas = s;
  pgas = p2*V2/Vgas;
  Mboil = if ((pSat_liq > p1) and Exist1) then Rb*(pSat_liq - p1) else 0;
  Mcond = if ((pSat_gas < pgas) and Exist2) then Rc*(pgas - pSat_gas) else
          0;
  Mdot1 = Mboil - Mcond;
  Mdot2 = Mdot1;
  q1 = Mboil*VSat_liq - Mcond*VSat_gas;
  q2 = q1;
  fd = Mdot1 > 0;
  DeltaQdot = (g1 - g2)*Mdot1 - (p1 - p2)*q1;
  Exist = (fd and Exist1) or (not fd and Exist2);
  Mref = if fd then M1 else M2;
  xi = if Exist then Mdot1/Mref else 0;
  Sdot2_aux = if Exist2 then Sdot1 + (DeltaQdot + (T1 - T2)*Sdot1)/T1 else
          Sdot1;
  Sdot1_aux = if Exist1 then Sdot2 + (DeltaQdot + (T1 - T2)*Sdot2)/T2 else
          Sdot2;
  Sdot1 = if fd then xi*S1 + hboil*Mdot1/T1 else Sdot1_aux;
  Sdot2 = if fd then Sdot2_aux else xi*S2 - hboil*Mdot2/T2;
  Vsteam = 1/(p2/pSat_gas - 1)*Vgas;
  hum1 = pgas/pSat_gas;
  hum2 = 1 - V2/Vsteam;
end BoilCondWater;
```

Simulation of Pressure Cooker

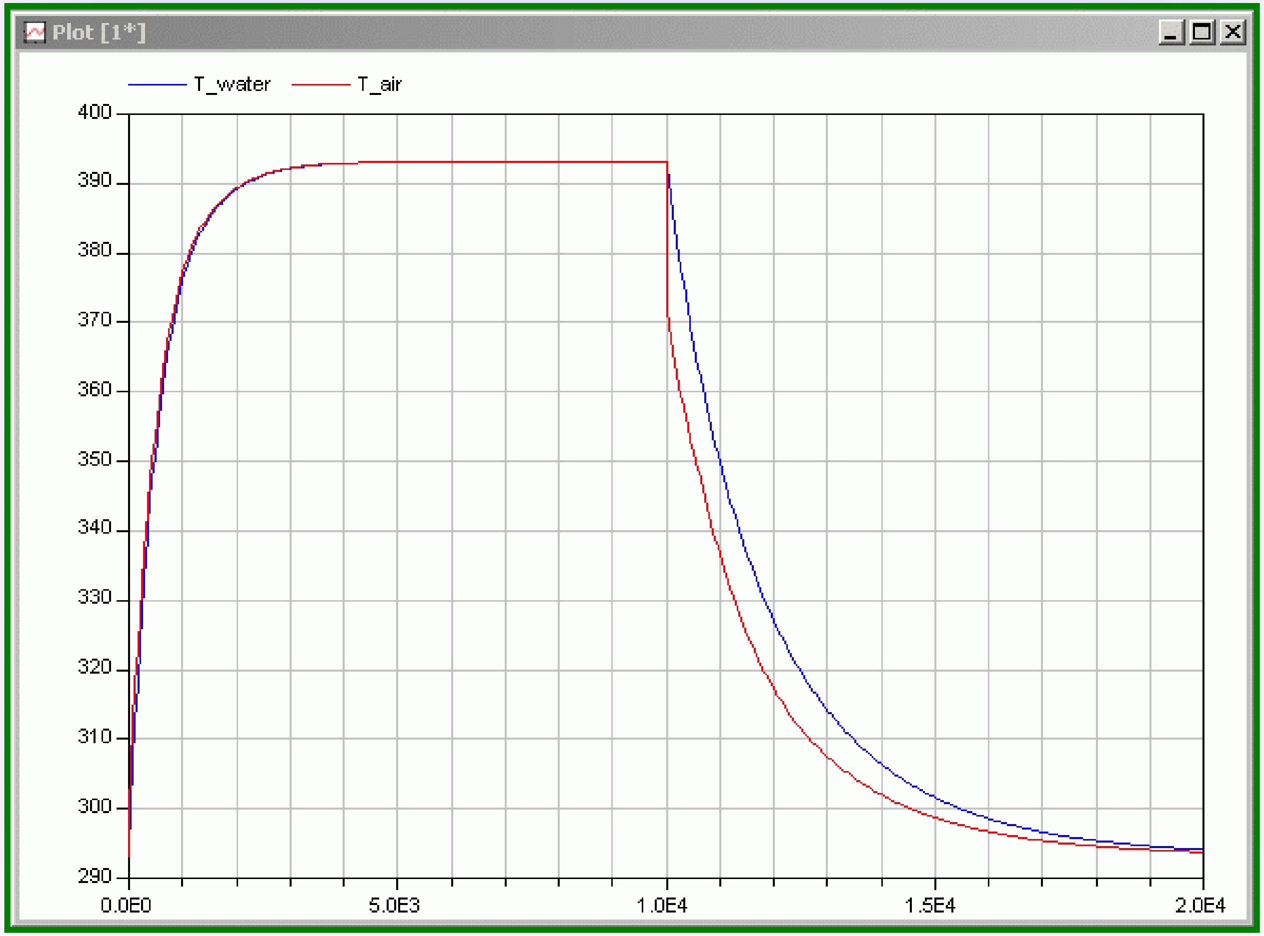
- We are now ready to compile and simulate the model.



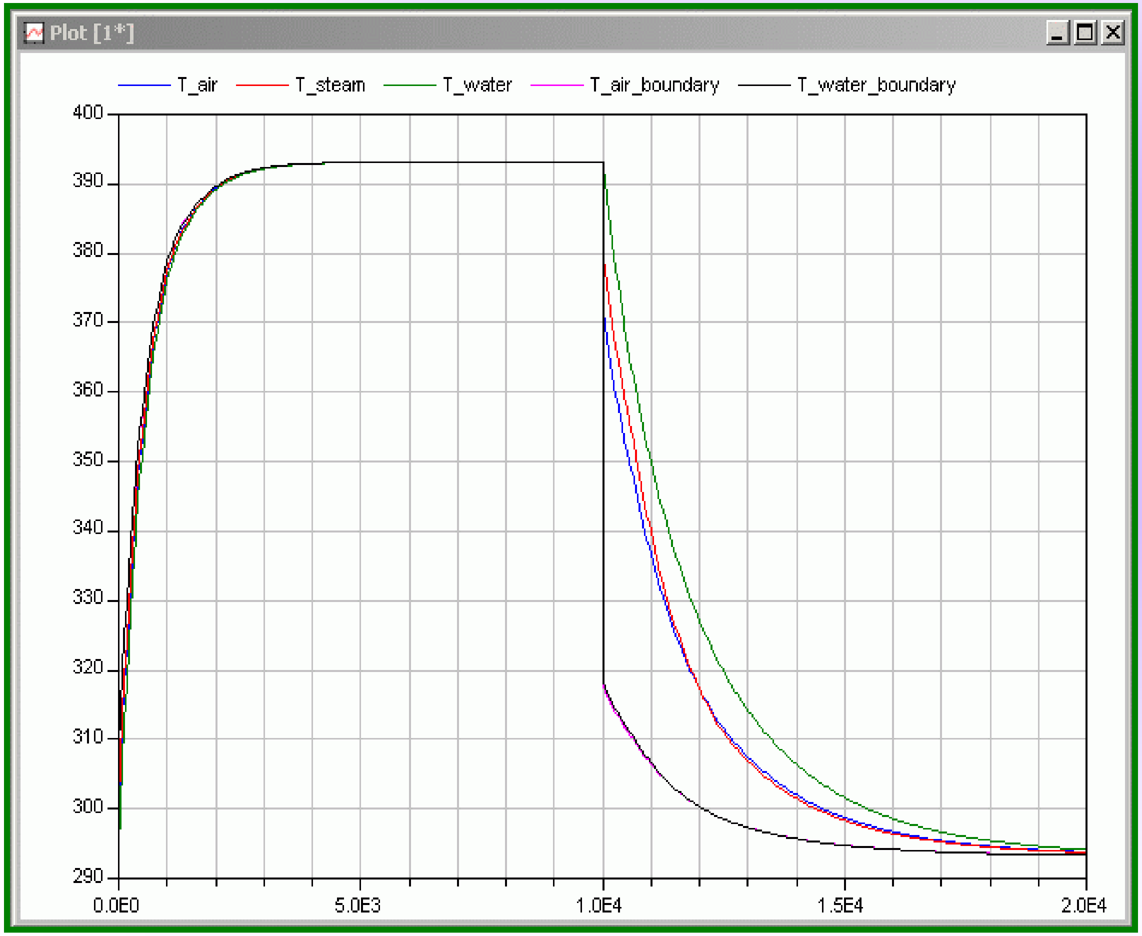




Simulation Results



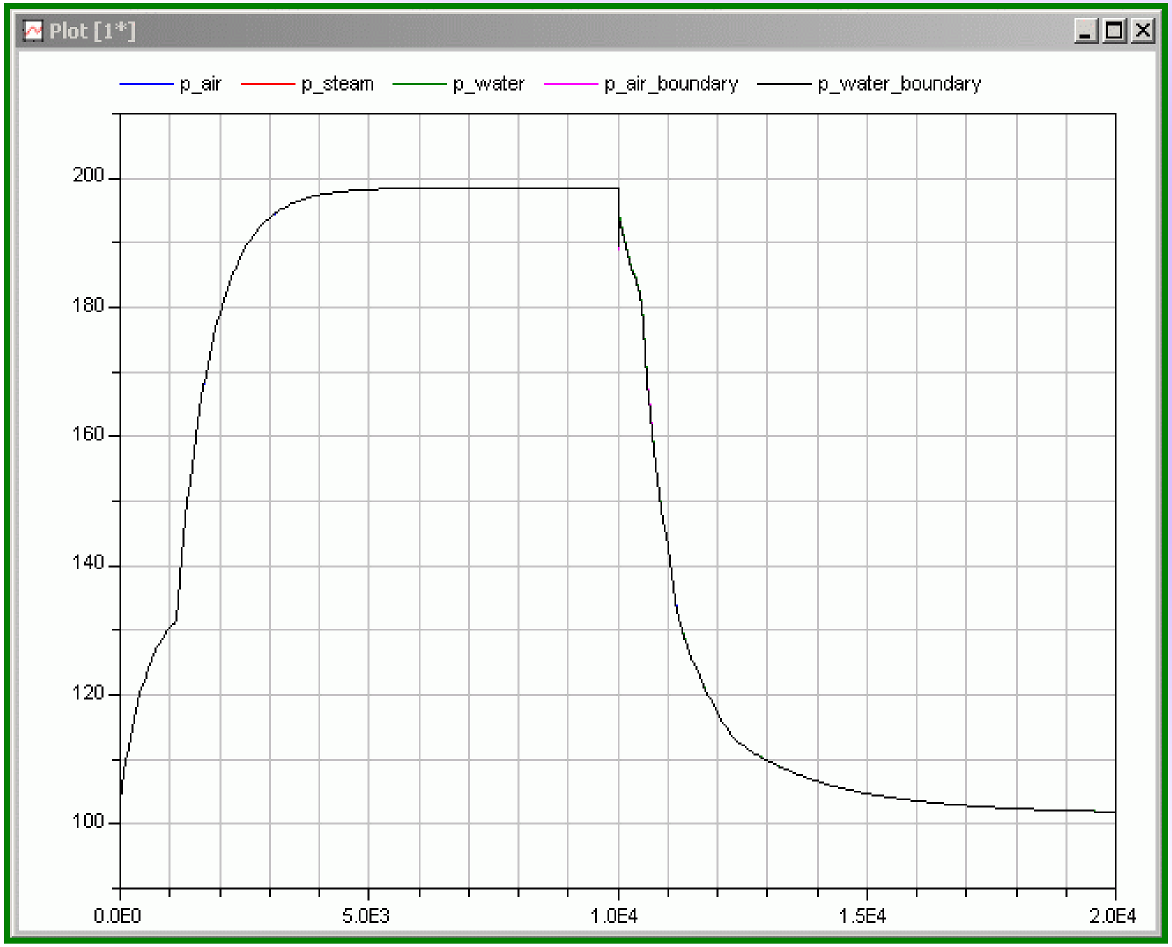
Simulation Results II



Heating is sufficiently slow that the temperature values of the different media are essentially indistinguishable. The heat exchangers have a smaller time constant than the heating.

During the cooling phase, the picture is very different. When cold water is poured over the pressure cooker, air and steam in the small boundary layer cool down almost instantly. Air and steam in the bulk cool down more slowly, and the liquid water cools down last.

Simulation Results III

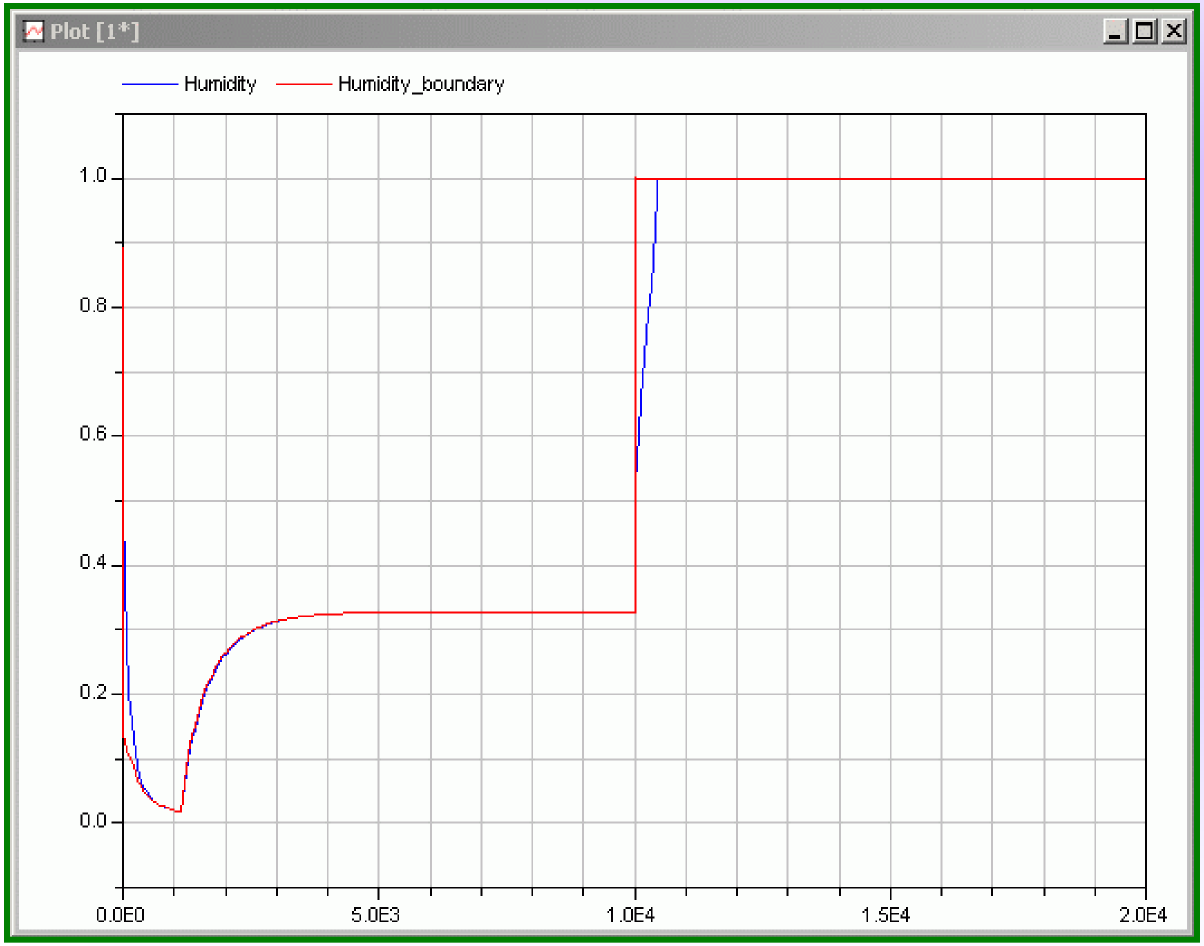


The pressure values are essentially indistinguishable throughout the simulation.

During the heating phase, the pressures rise first due to rising temperature. After about 150 seconds, the liquid water begins to boil, after which the pressure rises faster, because more steam is produced (water vapor occupies more space at the same temperature than liquid water).

The difference between boundary layer and bulk pressure values in the cooling phase is a numerical artifact.

Simulation Results IV

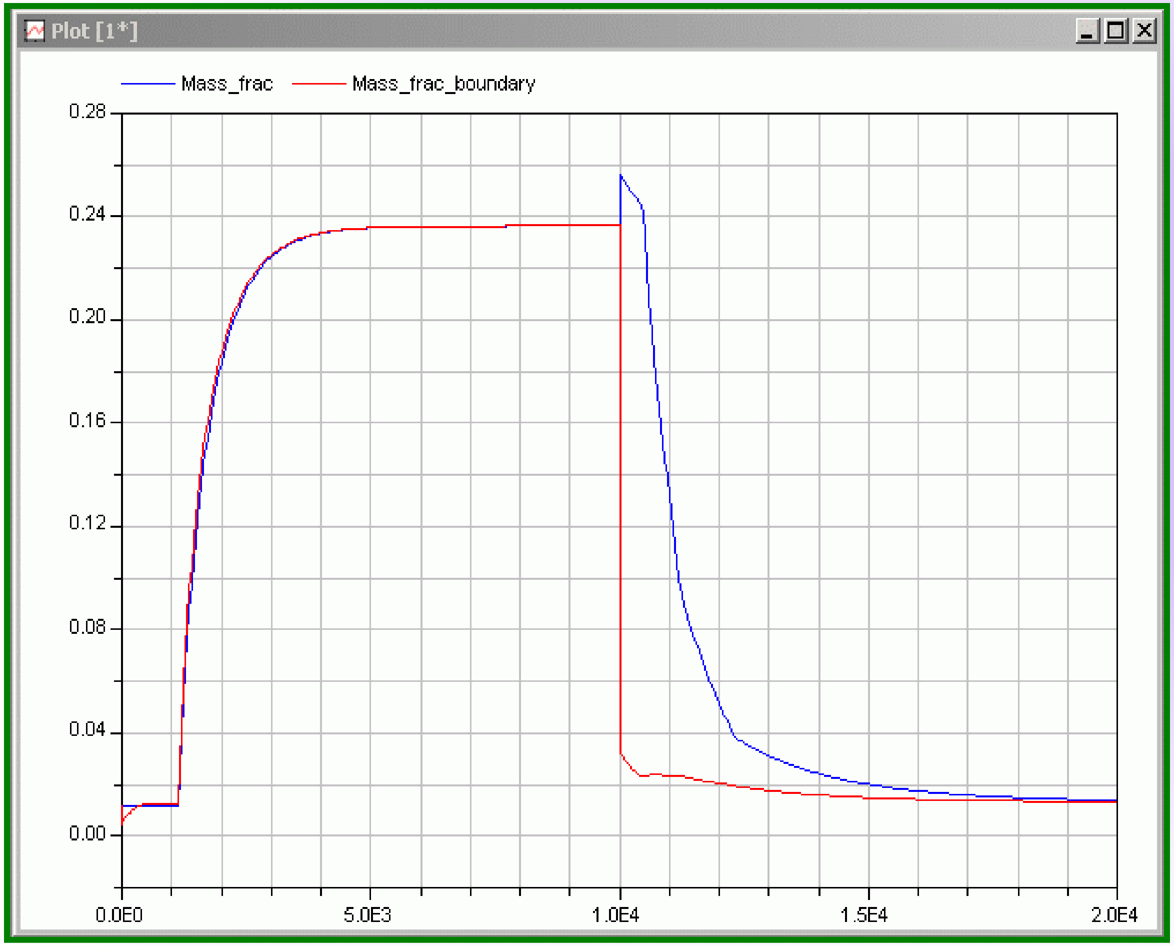


The relative humidity decreases at first, because the saturation pressure rises with temperature, i.e., more humidity can be stored at higher temperatures.

As boiling begins, the humidity rises sharply, since additional vapor is produced.

In the cooling phase, the humidity quickly goes into saturation, and stays there, because the only way to ever get out of saturation again would be by reheating the water.

Simulation Results V



The ***mass fraction*** defines the percentage of water vapor contained in the air/steam mixture.

Until the water begins to boil, the mass fraction is constant. It then rises rapidly until it reaches a new equilibrium, where evaporation and condensation balance out.

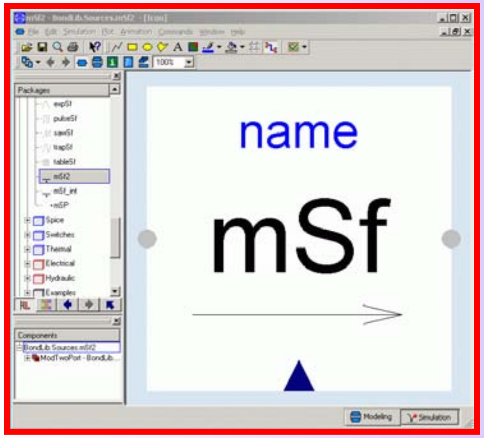
During the cooling phase, the boundary layer cools down quickly, and can no longer hold the water vapor contained. Some falls out as water, whereas other steam gets pushed into the bulk, temporarily increasing the mass fraction there even further.

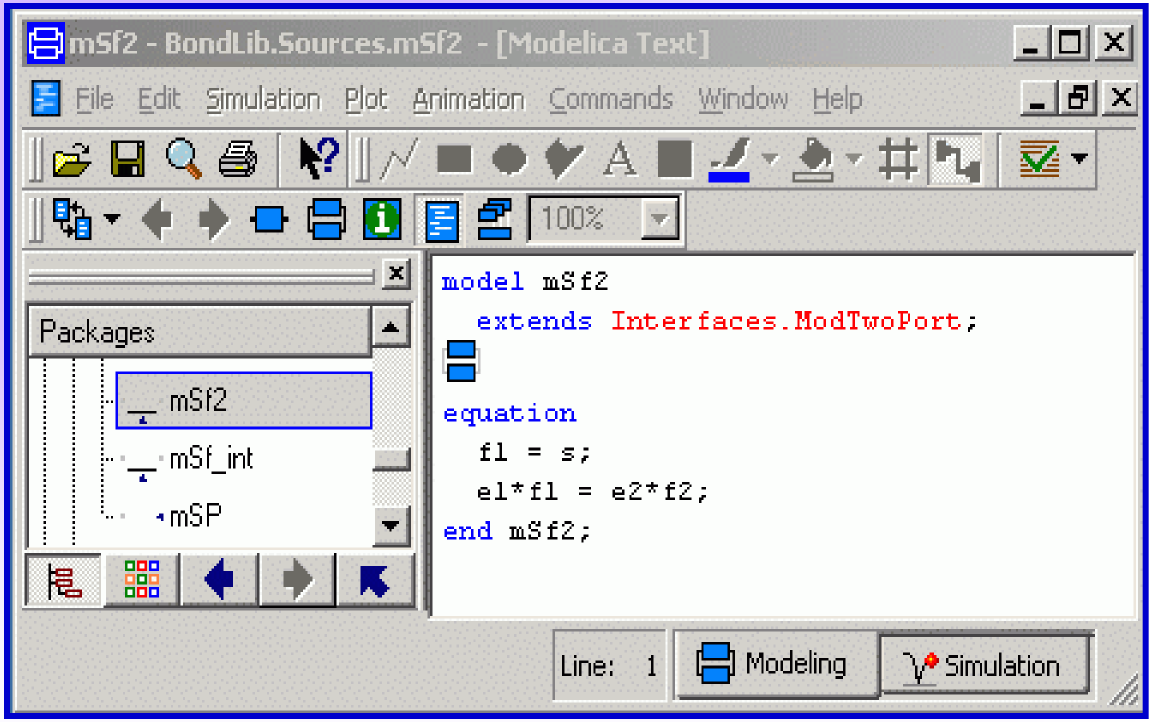
Free Convective Mass Flow

- We are now ready to discuss ***free convective mass flow***, such as mass flow occurring in a segment of a pipe.
- The convective mass flow occurs because more mass is pushed in from one end, pushing the mass currently inside the pipe segment out by the other end.
- To this end, we need to introduce some more models.

The Forced Flow Source

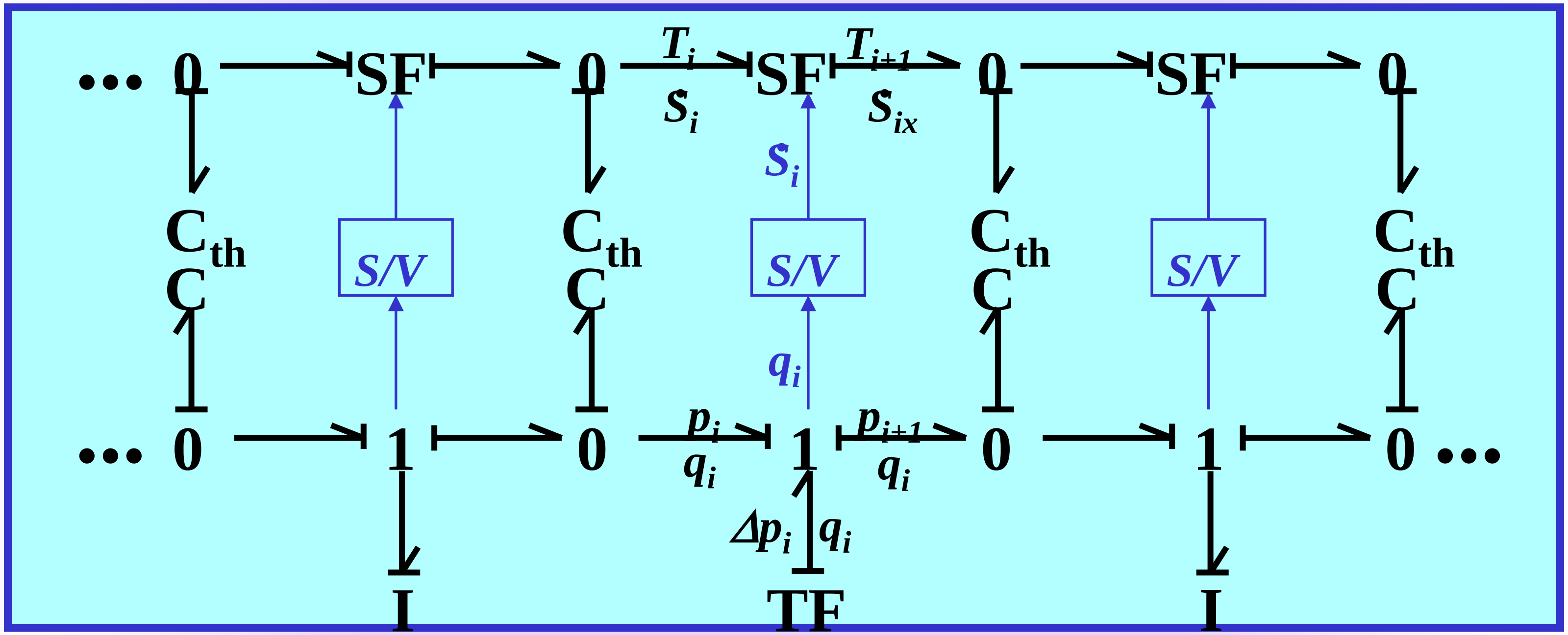
- This model describes an element of the regular bond-graph library.





The primary side is a flow source, the secondary side can be either a flow or an effort source. Its equation is defined to satisfy power continuity across the element.

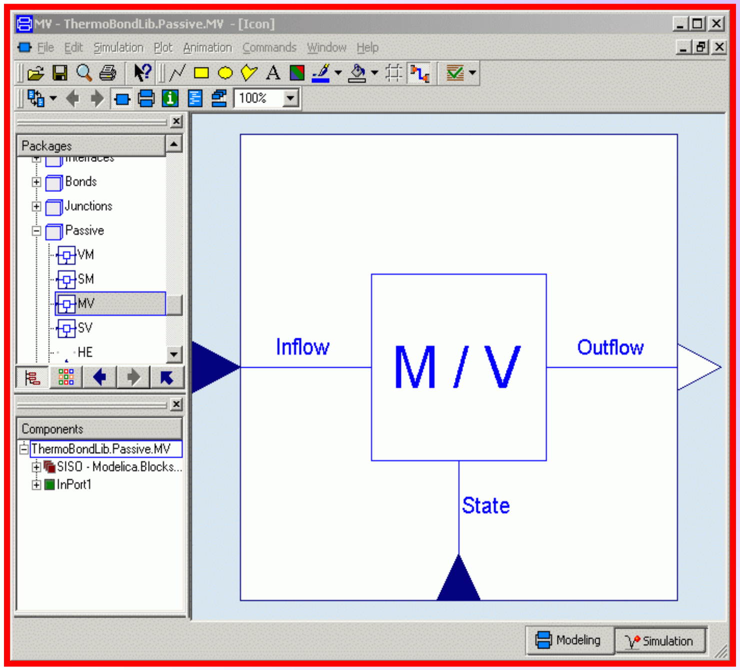
Density and Specific Entropy I

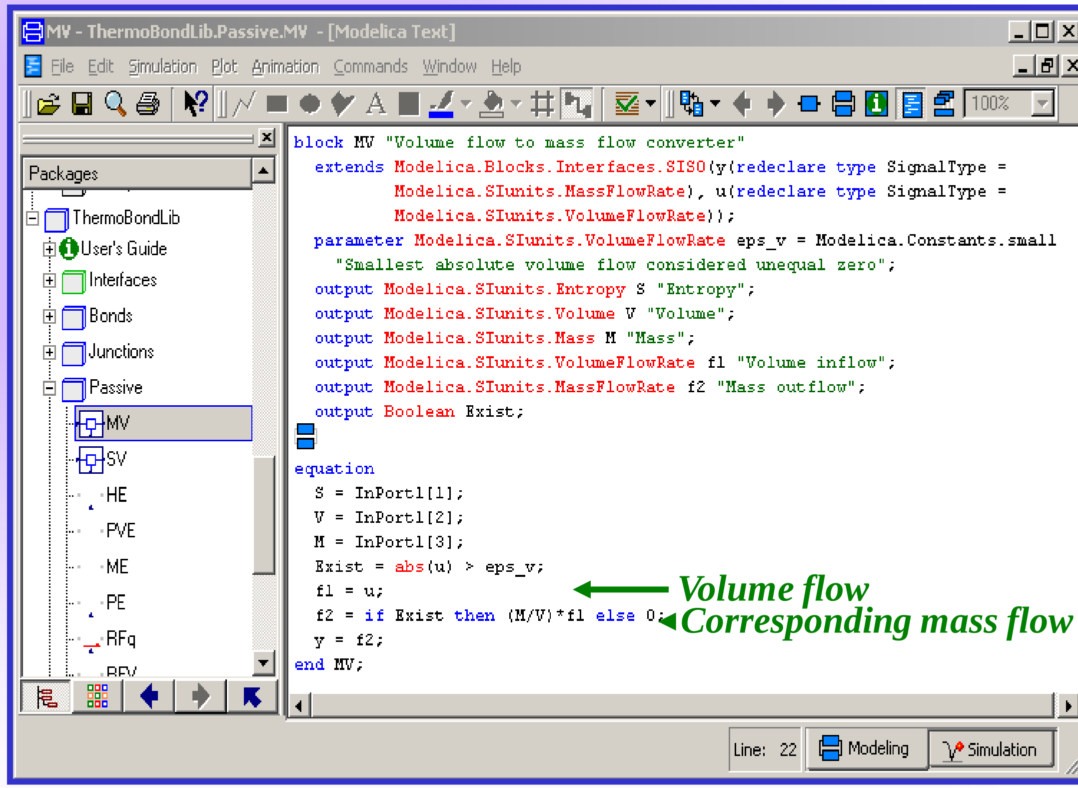


- As mentioned some lectures ago, we shall need modulated flow sources (as introduced one slide ago) that are modulated by the specific entropy and/or the specific mass (i.e., the density).

Density and Specific Entropy II

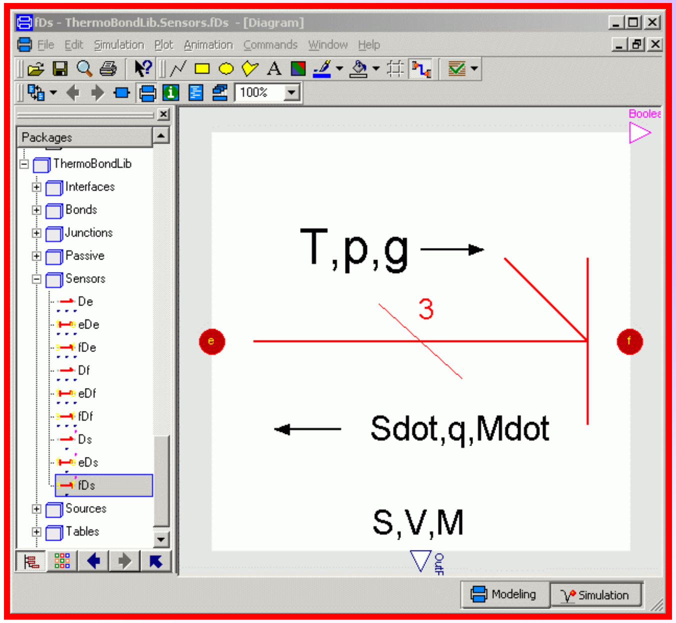
- These models are created as blocks:

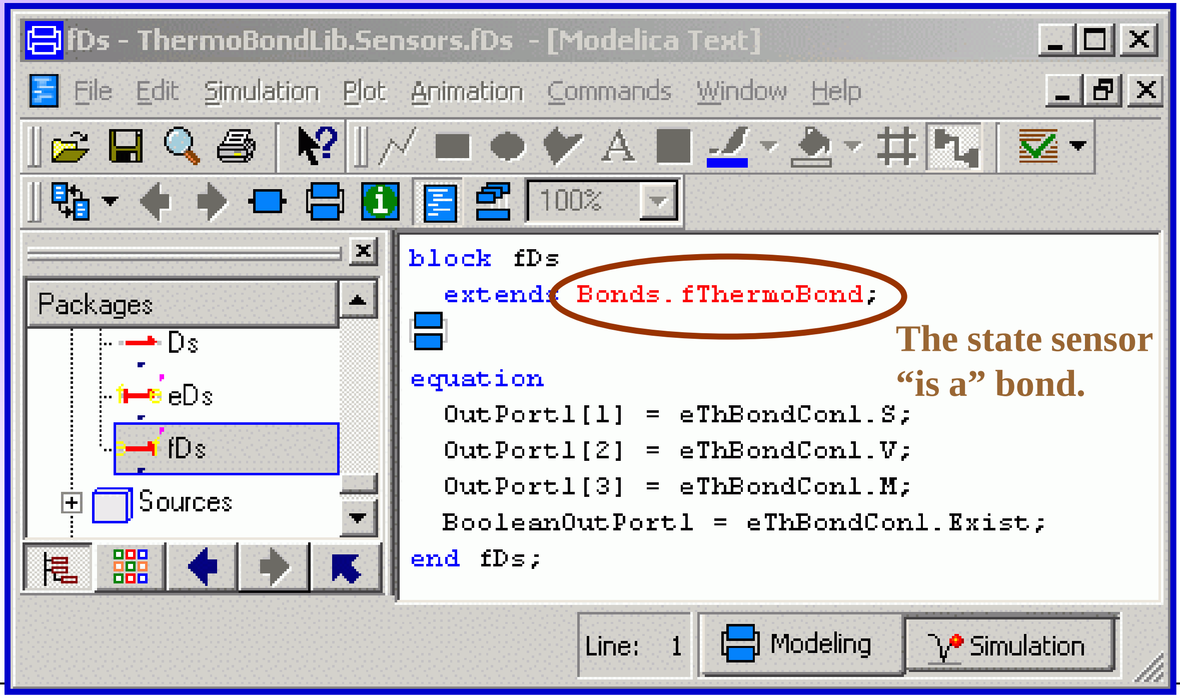




The State Sensor

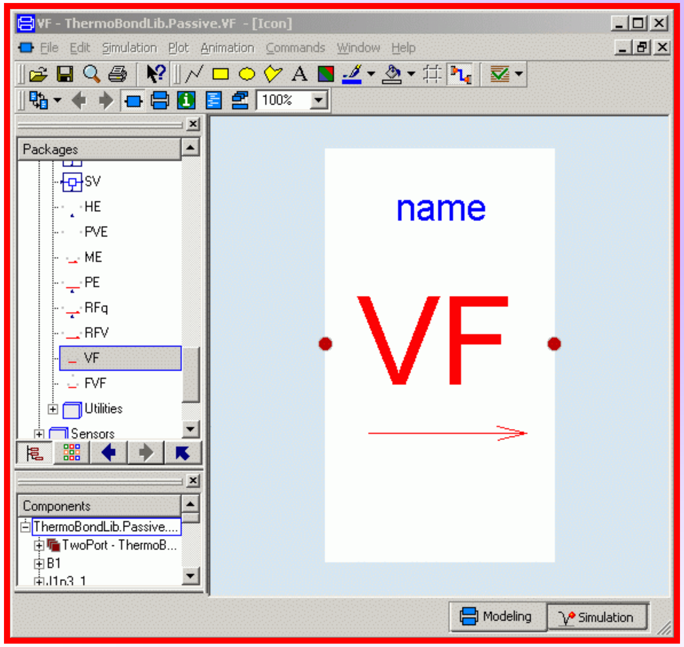
- Many elements that are related to substances require state information. This is generated by a specialized thermo-bond, the so-called ***state sensor element***.

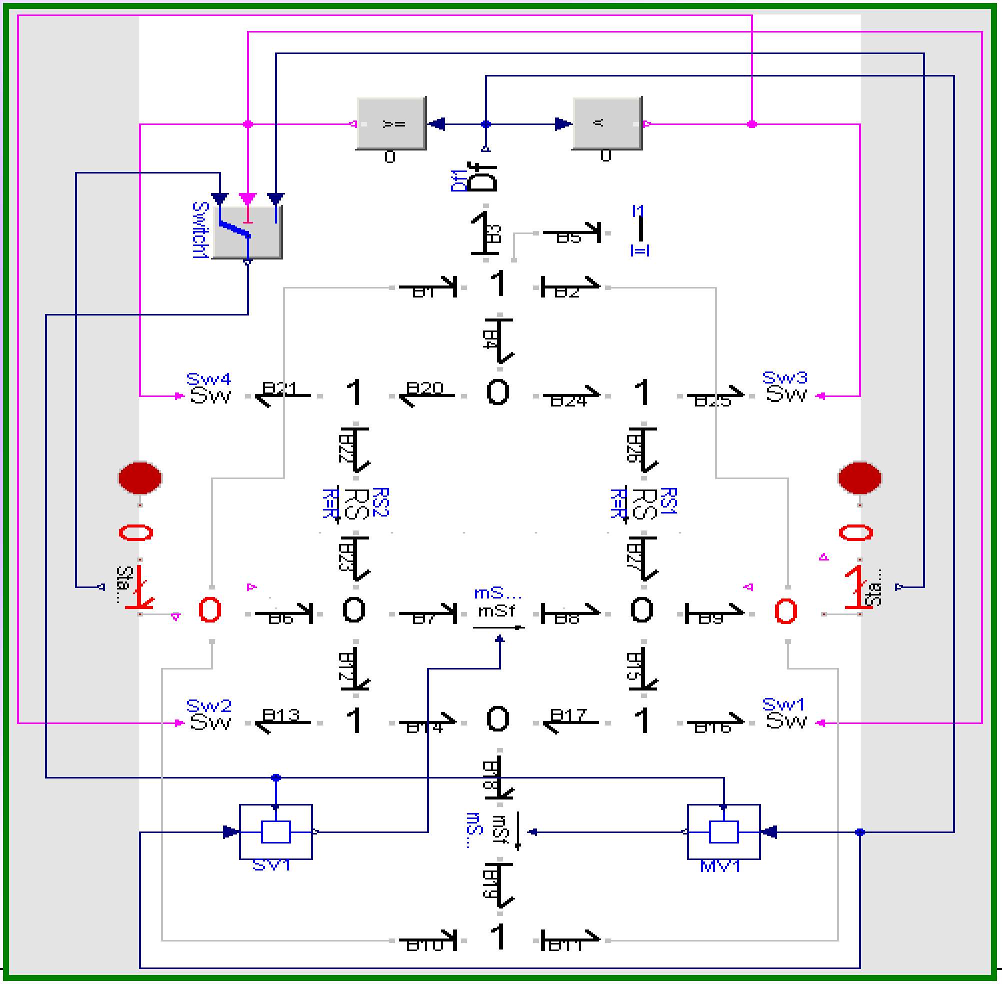


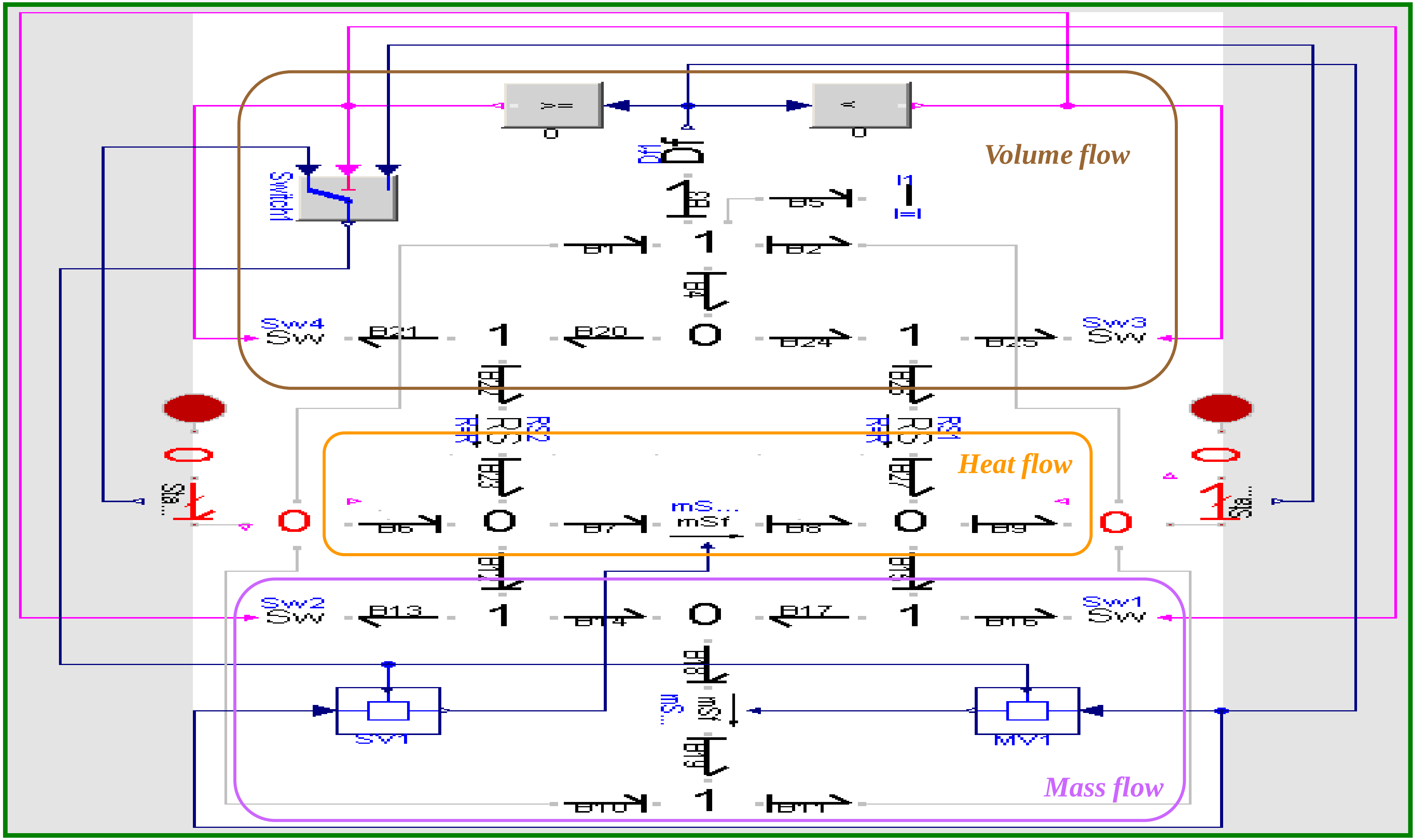


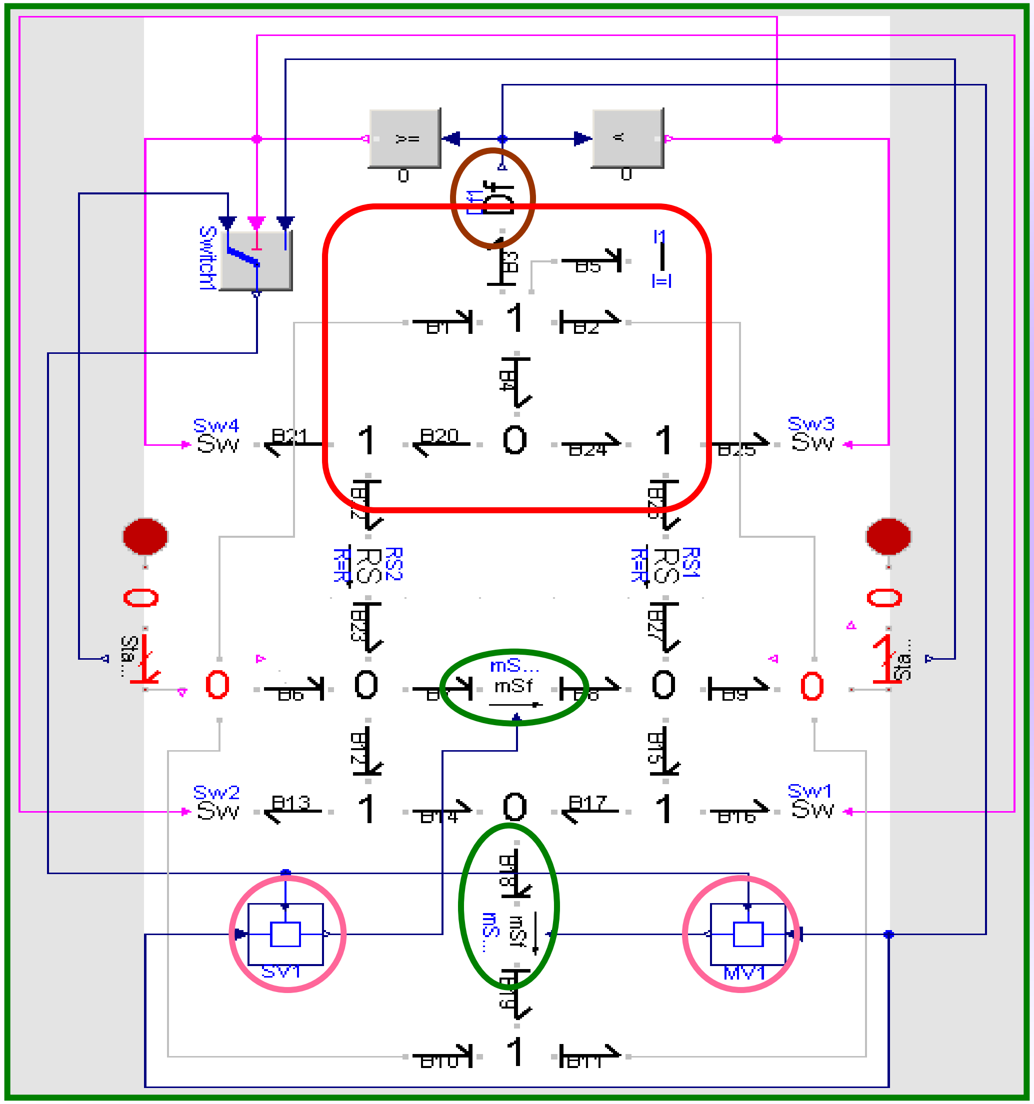
Free Convective Volume Flow

- We are now ready to describe the free convective volume flow.






Volume flow
Heat flow
Mass flow
Switch1
Sw4
Sw3
Sw2
Sw1
SV1
MV1
mSf

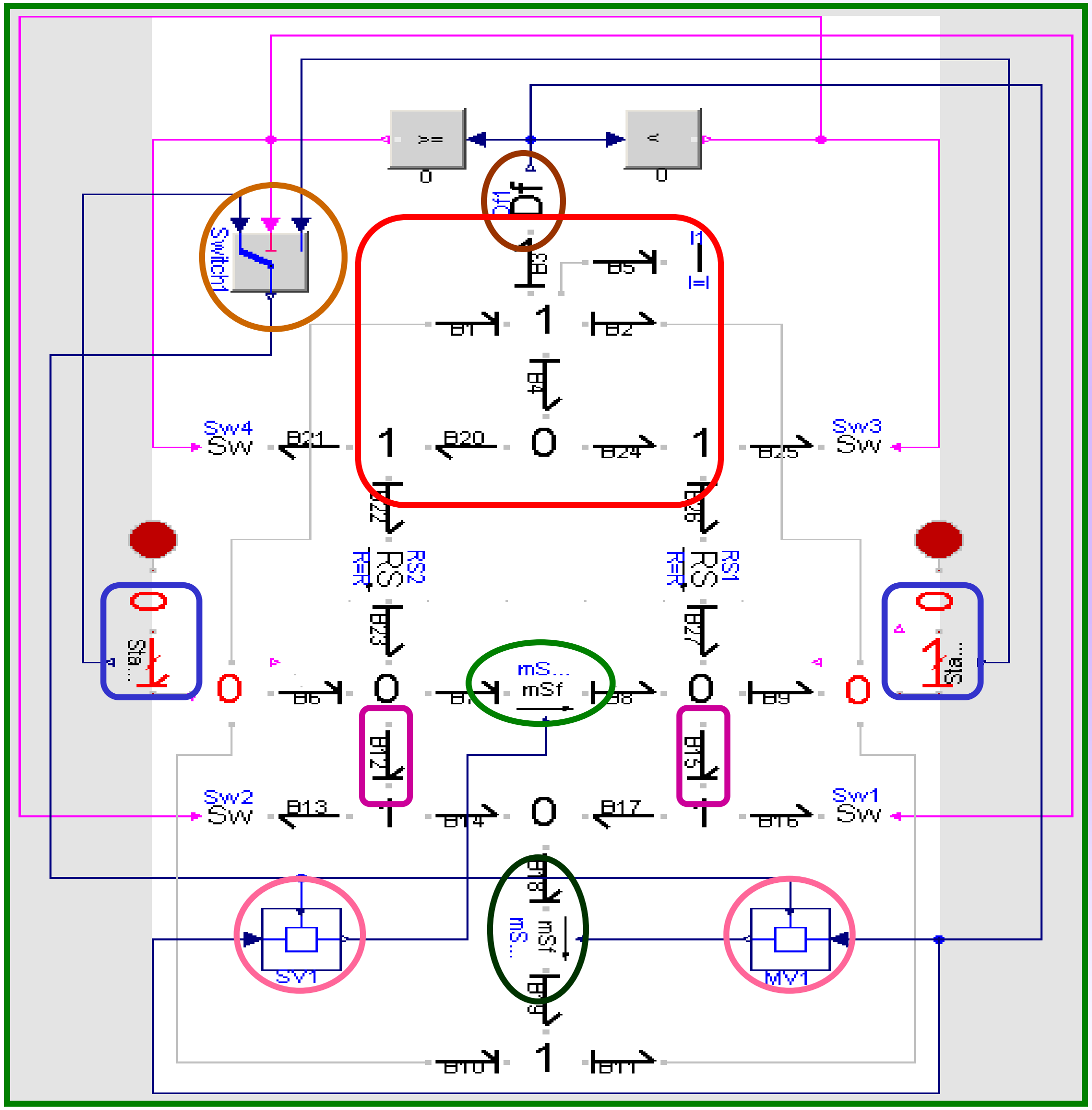


The volume flow is modeled as a wave equation with friction. The friction is in parallel with the inertia.

The flow is measured using a flow sensor element. The additional entropy generated by friction is reintroduced in the down-wind direction, i.e., in the direction of the flow. Switch elements are used to determine the reintroduction point.

Non-linear flow sources are used to model the parallel thermal and mass flows.

These are computed by converting the volume flows to consistent entropy and mass flows.



State sensor elements are used to determine the current values of volume, entropy, and mass.

Up-wind state information is being used to convert the volume flow into consistent entropy and mass flows.

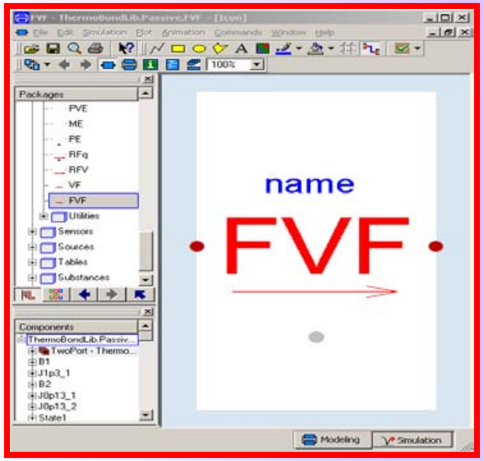
Since entropy doesn't need to be preserved, the non-linear flow source is inserted directly into the thermal branch.

Since mass flow must be preserved, the non-linear flow source is inserted under a 1-junction in the mass flow branch.

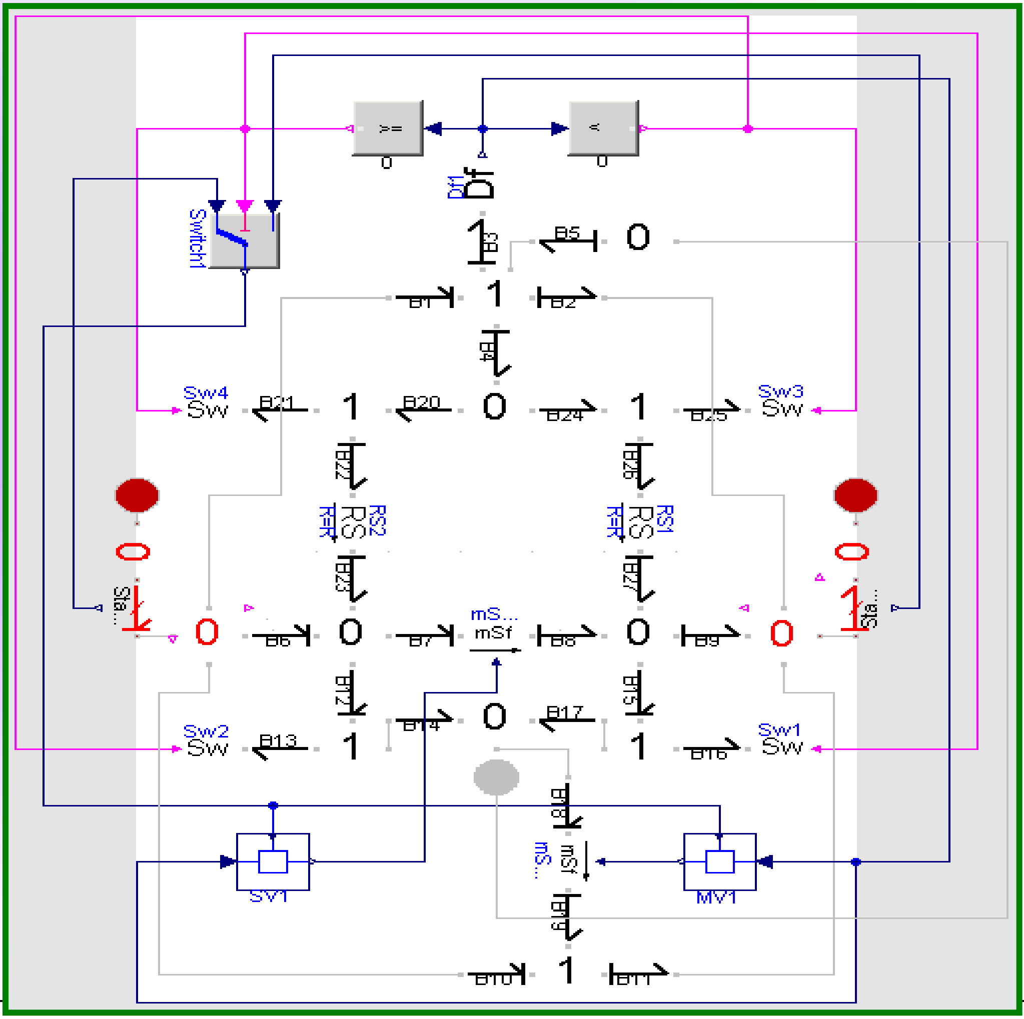
To move mass with the volume, additional energy is needed that is taken from the thermal domain.

Forced Convective Volume Flow

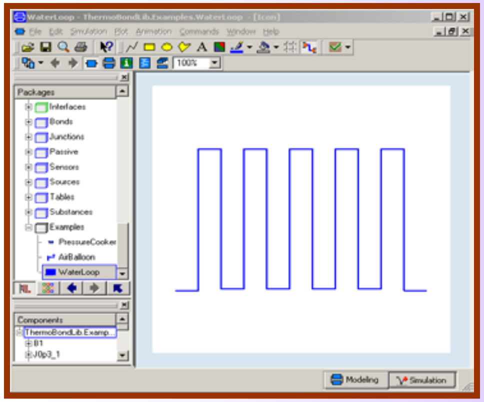
- We are now ready to describe the forced convective mass flow.

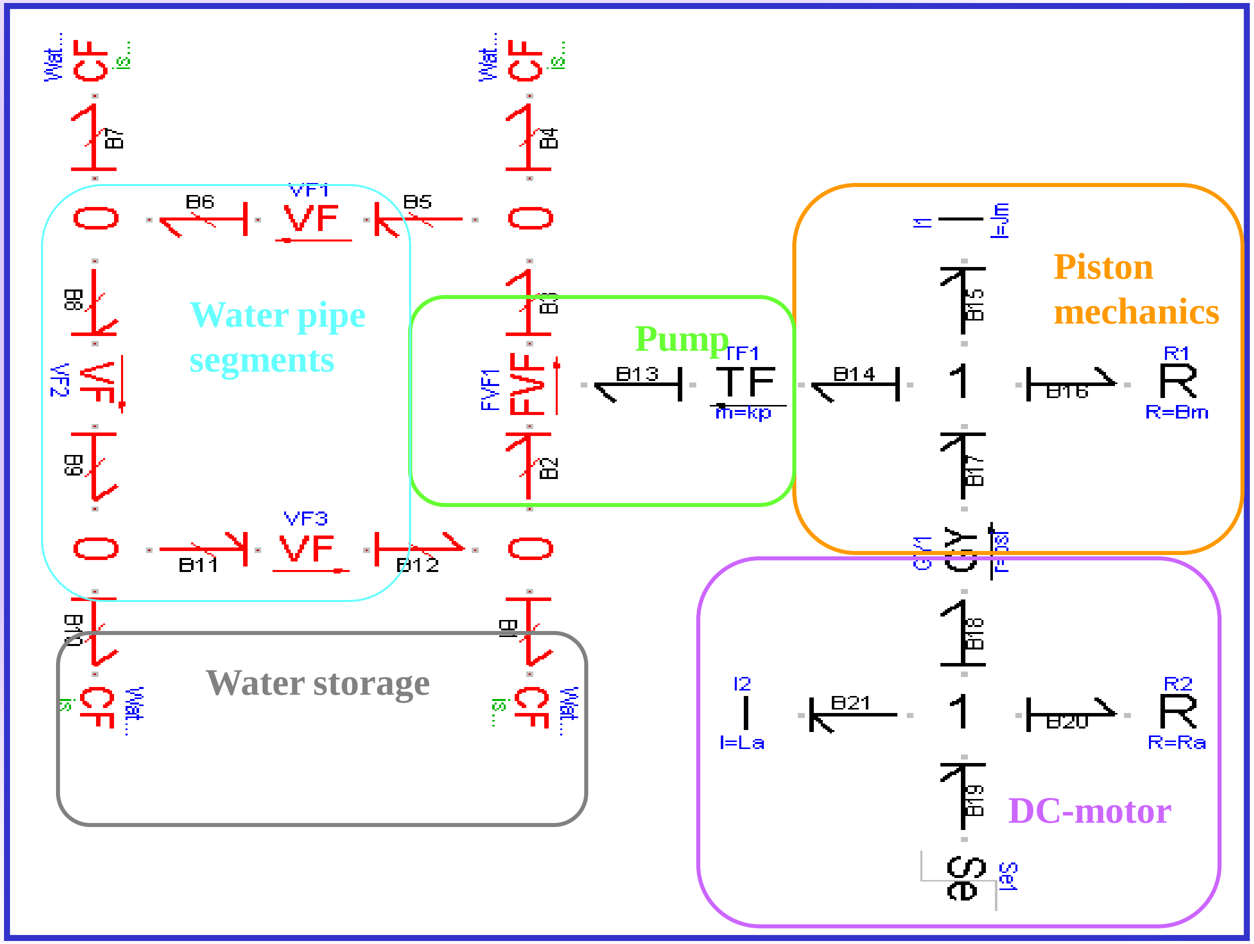


The model is almost the same as the free convective flow model, except that a volume flow is forced on the system through the regular bond connector at the top, replacing the inductor.



The Water Serpentine I



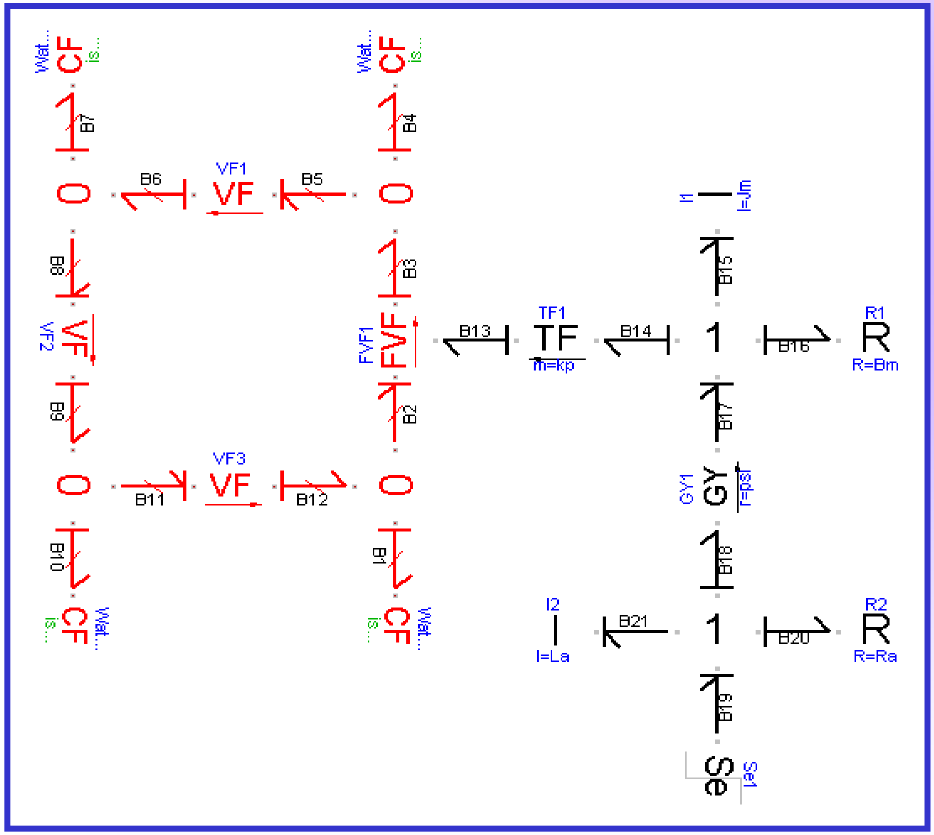


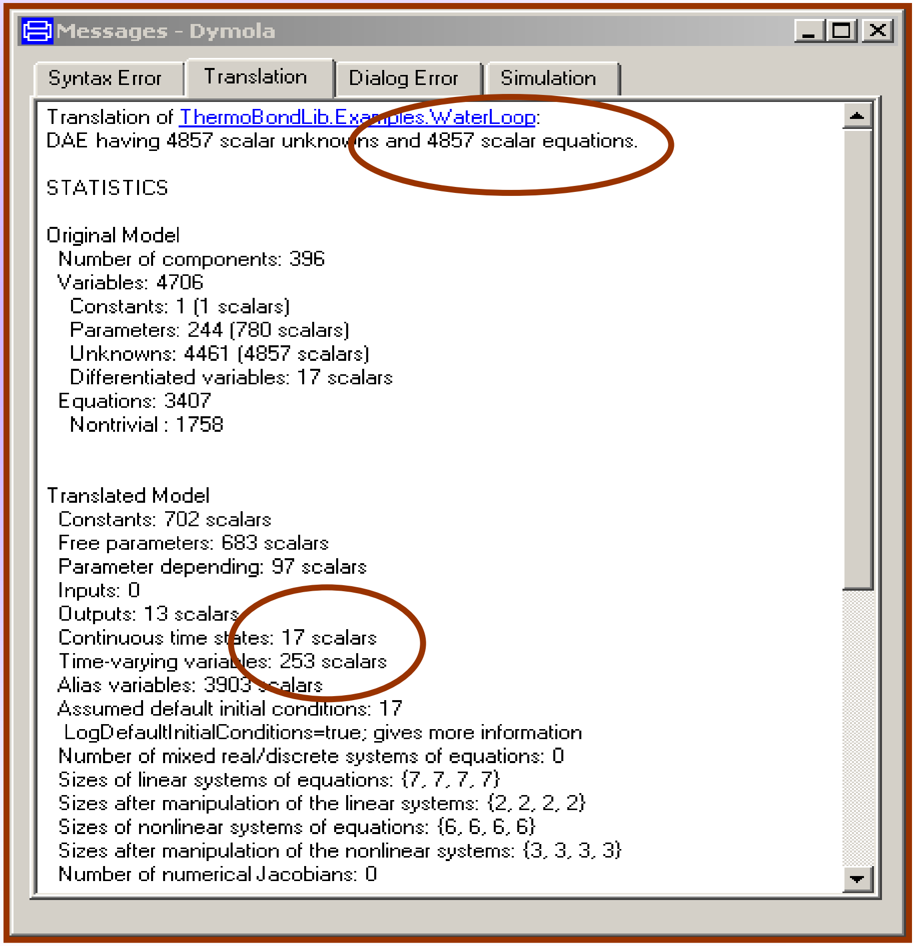
The Water Serpentine II

- The pump forces a flow, thereby creating a higher pressure at the outflow, while creating a lower pressure at the inflow.
- Mass is transported through the pump with the volume. Since the mass is getting condensed, it occupies less space. Thus, there is "surplus" volume that gets used to "finance" the mass transport in the ***Gibbs equation***.
- In the pipe segments, the pressure is gradually reduced again, thus each pipe segment has a higher pressure at the inflow than at the outflow. The mass thus expands, and the volume consumed in the pump is gradually given back, so that the overall volume in the water serpentine is being preserved.

The Water Serpentine III

- We are now ready to simulate this system.



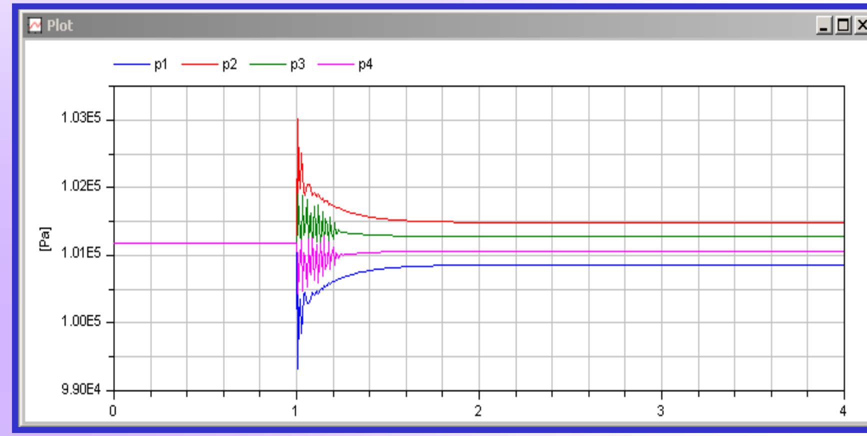


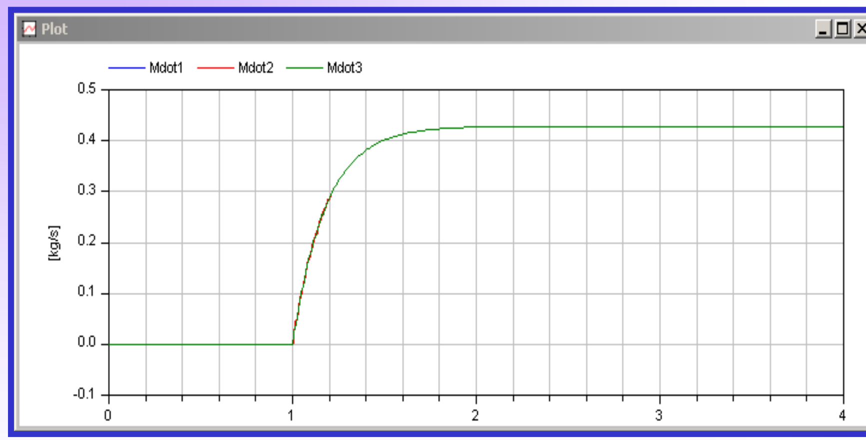
The Water Serpentine IV

Messages - Dymola

Syntax Error | Translation | Dialog Error | Simulation

```
Log-file of program ./dymosim
(generated: Fri Feb 08 16:13:54 2008)

dymosim started
... "dsin.txt" loading (dymosim input file)
... "WaterLoop.mat" creating (simulation result file)

Integration started at T = 0 using integration method DASSL
(DAE multi-step solver (dassl/dasslrt of Petzold modified by Dynasim))
Integration terminated successfully at T = 4
   CPU-time for integration      : 0.4 seconds
   CPU-time for one GRID interval: 0.8 milli-seconds
   Number of result points       : 514
   Number of GRID   points       : 501
   Number of (successful) steps  : 499
   Number of F-evaluations       : 3336
   Number of H-evaluations       : 1053
   Number of Jacobian-evaluations: 138
   Number of (model) time events : 1
   Number of (U) time events     : 0
   Number of state    events     : 6
   Number of step     events     : 0
   Minimum integration stepsize  : 6.17e-009
   Maximum integration stepsize  : 0.476
   Maximum integration order     : 5
Calling terminal section
... "dsfinal.txt" creating (final states)
```





Comparison With *Biosphere 2*

- In the ***Biosphere 2*** model, only the (sensible and latent) heat were modeled. The mass flows were not considered.
- Consequently, you never know in the ***Biosphere 2*** model, how much water is available where. It is always assumed that the pond never dries out, and that the plants always have enough water to be able to evaporate in accordance with their temperature and saturation pressure.
- In the case of the ***pressure cooker*** model, both the mass flows and the heat flows were modeled and simulated. Consequently, the case is caught, where all the water has evaporated, while the air/steam mixture is still not fully saturated.

References I

- Cellier, F.E. and J. Greifeneder (2003), "Object-oriented modeling of convective flows using the Dymola thermo-bond-graph library," *Proc. ICBGM'03, Intl. Conference on Bond Graph Modeling and Simulation*, Orlando, FL, pp. 198 – 204.

- Greifeneder, J. and F.E. Cellier (2001), "Modeling multi-element systems using bond graphs," *Proc. ESS'01, European Simulation Symposium*, Marseille, France, pp. 758 – 766.

- Cellier, F.E. and J. Greifeneder (2008), "ThermoBondLib – A New Modelica Library for Modeling Convective Flows," *Proc. Modelica'08*, Bielefeld, Germany, pp. 163 – 172.

References II

- Cellier, F.E. (2007), *The Dymola Bond-Graph Library*, Version 2.3.
- Cellier, F.E. (2007), *The Dymola Thermo-Bond-Graph Library*, Version 2.0.