Modeling in Thermoptim of the quench of a gas (water quench)

The quenching of a synthesis gas (water quench) is a form of gas humidification by washing which allows both to clean ash-type impurity and tars by cooling it and saturating it with water.

This operation has the distinction of bringing into play two separate streams: the synthesis gas and water, which exchange matter and energy through an interface. It behaves like a quadrupole receiving two input fluids, and out of which come two others.

This poses a slight difficulty to achieve a model in Thermoptim, since the only components available are either processes or nodes. The solution is to train the quadrupole by combining an inlet mixer and an outlet divider, the two being connected by a process-point playing a passive role.

For the model to be consistent, it synchronizes the calculations made by the two nodes. Specifically the outlet divider takes control of the mixer, whose role is limited to conduct an update of the coupling variables associated with the inlet stream. The model structure is given in Figure 1.

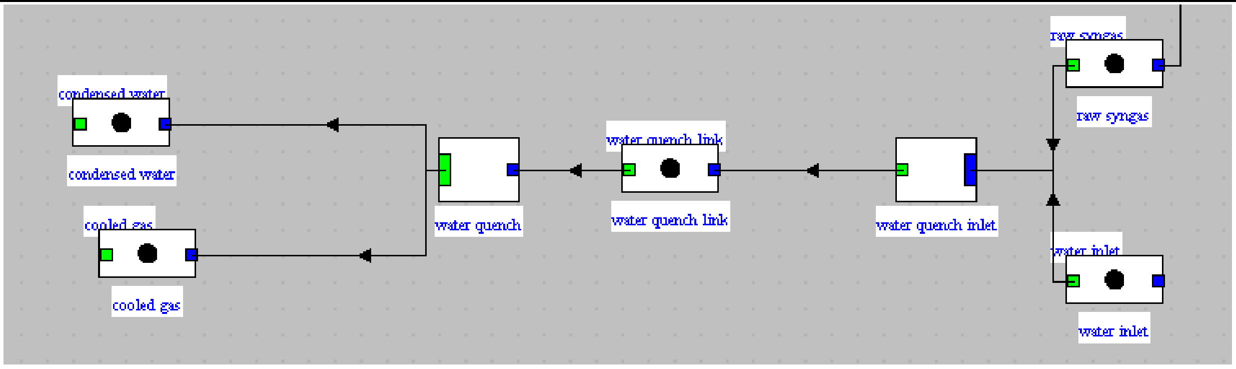


Figure 1: Diagram of water quench component

Model of the synthesis gas quench

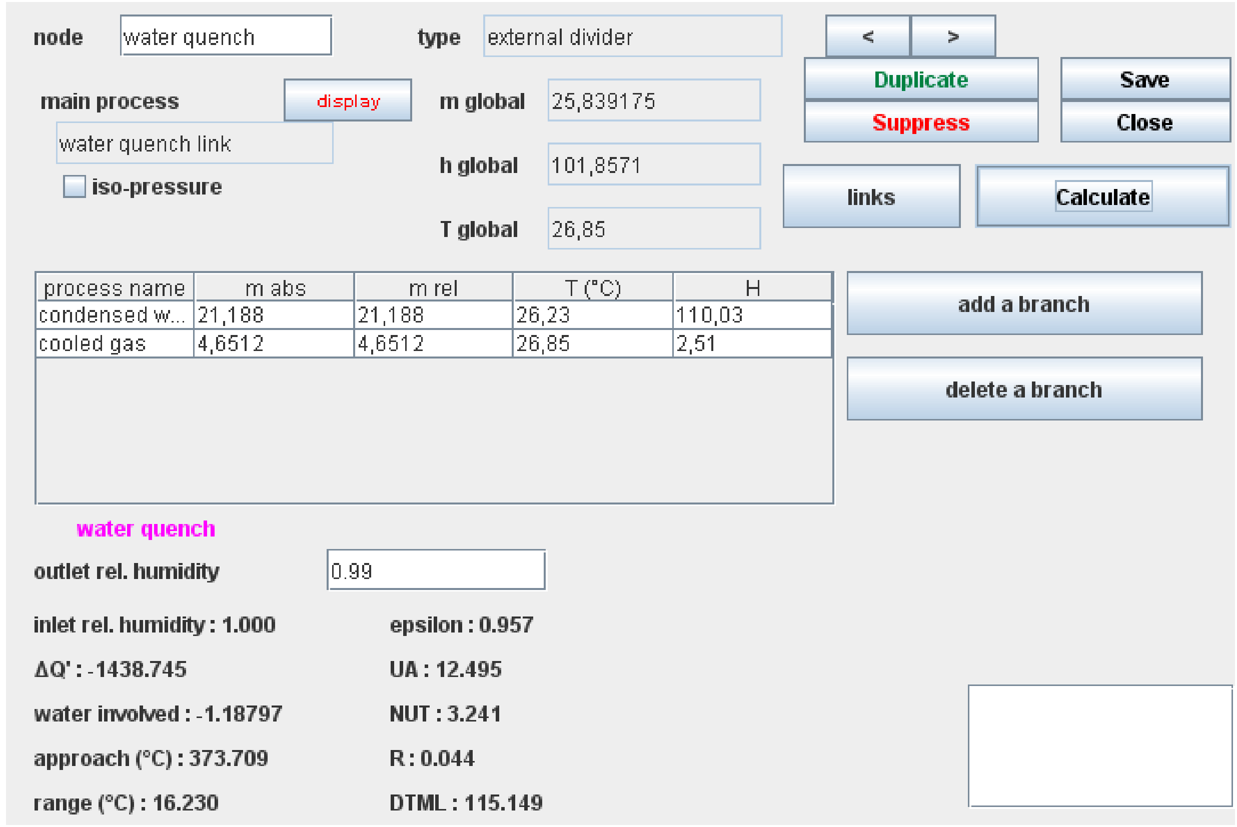
The model developed here is particularly simple: it merely sets the mass and energy balances, assuming known the gas temperature and relative humidity in the outlet component. The flow rates of both inflows are set by conditions upstream of the component and not recalculated. If the water flow is insufficient for the heating (up to the temperature of the gas exiting) to match the required enthalpy of the gas, a message notifies the user.

The functions for calculating properties of Thermoptim wet gas and points were made accessible from the external classes. We advise you to refer to the note: "Calculations of wet gas from external classes "for a detailed presentation of the available methods.

The model is the following:

1) the only parameter the is value of the relative humidity ε in outlet component, read on the screen (a priori very close to 1);
2) we first calculate the absolute humidity of the incoming gas, and determine the mass flow of dry gas from the wet gas;
3) the outlet relative humidity ε is set, and the properties of the outlet wet gas are calculated, which gives the specific and total enthalpies which must be extracted from the gas
4) the flow of water supplied by the gas is determined and the composition of the wet gas is changed at the outlet
5) the enthalpy balance on water provides its outlet temperature, which must be lower than that of the gas exiting
6) values downstream of the node are updated

In Thermoptim, the component is as was said represented by an external mixer connected to an external divider, the calculations being made by the latter. Classes are called WaterQuenchInlet and WaterQuench. The component screen is given in figure 2.



node water quench **type** external divider

main process water quench link

☐ **iso-pressure**

m global 25,839175

h global 101,8571

T global 26,85

process name	m abs	m rel	T (°C)	H
condensed w...	21,188	21,188	26,23	110,03
cooled gas	4,6512	4,6512	26,85	2,51

water quench

outlet rel. humidity 0.99

inlet rel. humidity : 1.000

epsilon : 0.957

ΔQ' : -1438.745

UA : 12.495

water involved : -1.18797

NUT : 3.241

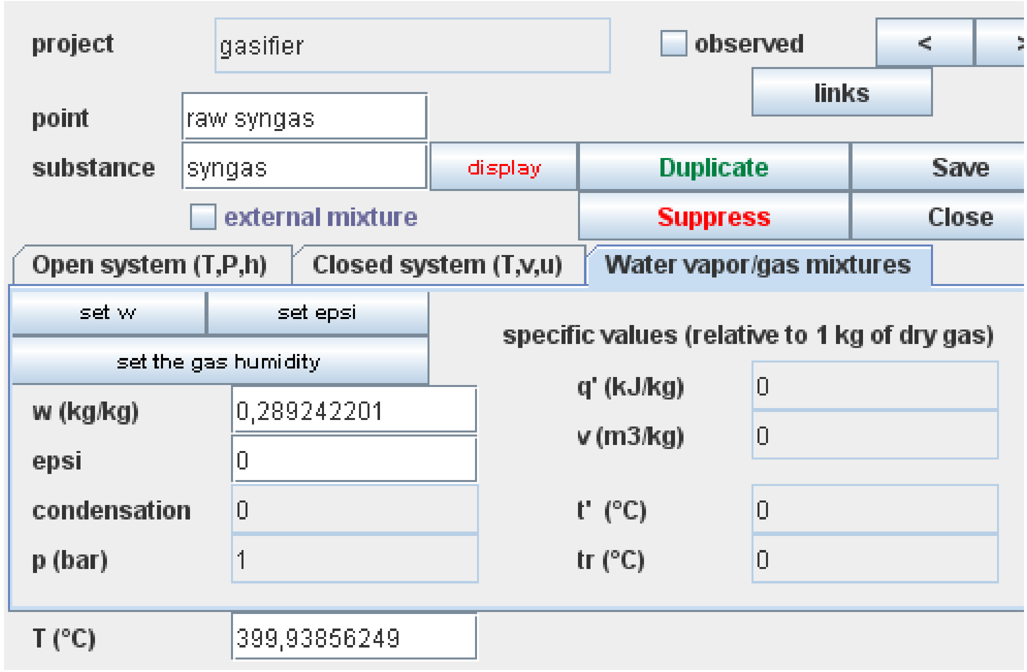
approach (°C) : 373.709

R : 0.044

range (°C) : 16.230

DTML : 115.149

Figure 2: Component "water quench" screen



project gasifier

☐ **observed**

point raw syngas

substance syngas

☐ external mixture

specific values (relative to 1 kg of dry gas)

w (kg/kg) 0,289242201

epsi 0

condensation 0

p (bar) 1

q' (kJ/kg) 0

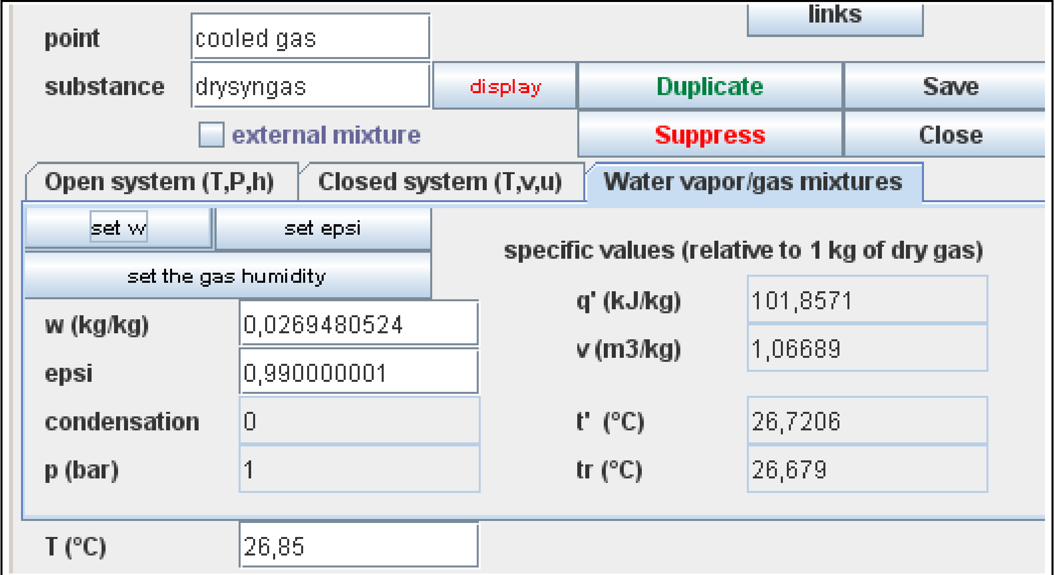
v (m3/kg) 0

t' (°C) 0

tr (°C) 0

T (°C) 399,93856249

Figure 3: Inlet point screen



point: cooled gas
substance: drysyngas

specific values (relative to 1 kg of dry gas)

w (kg/kg): 0,0269480524
epsi: 0,990000001
condensation: 0
p (bar): 1
q' (kJ/kg): 101,8571
v (m3/kg): 1,06689
t' (°C): 26,7206
tr (°C): 26,679
T (°C): 26,85

Figure 4: Outlet point screen

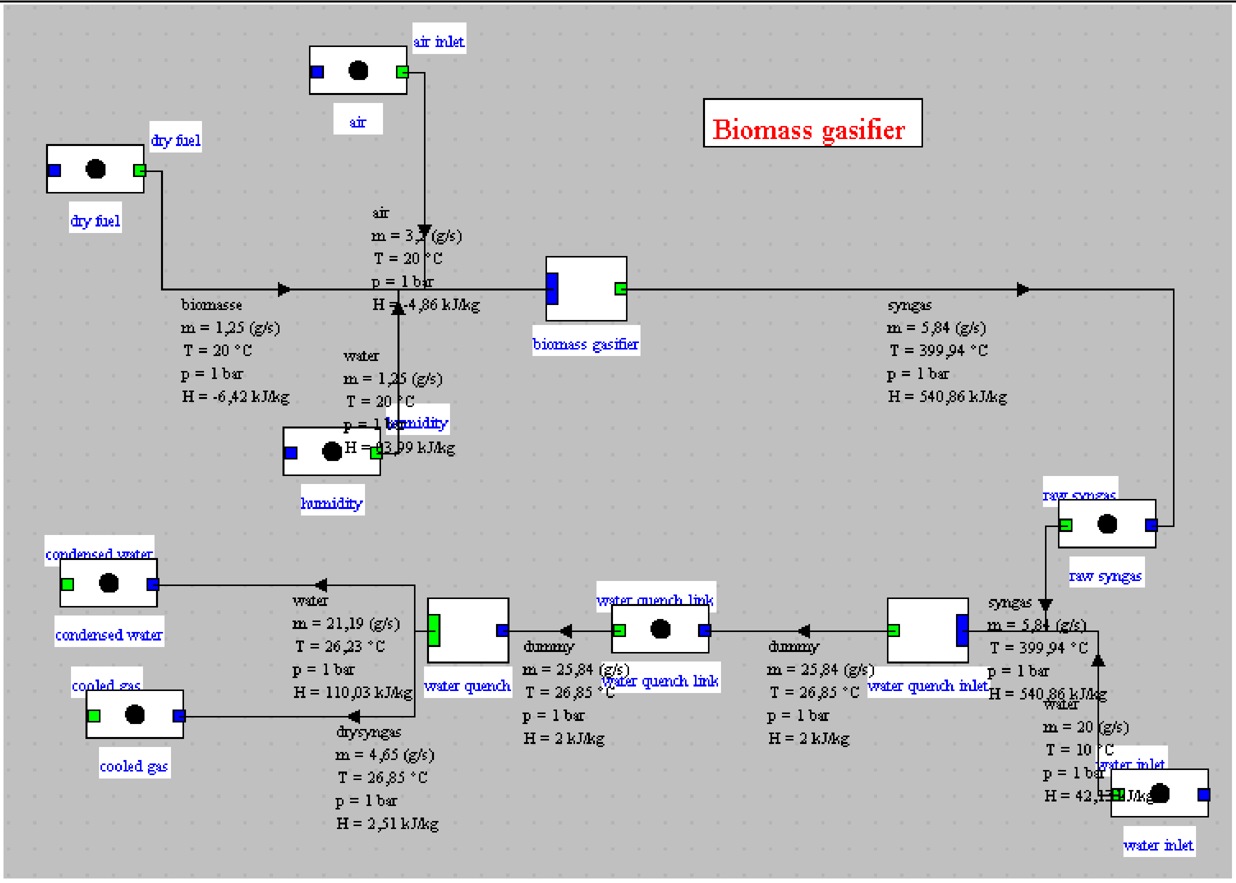
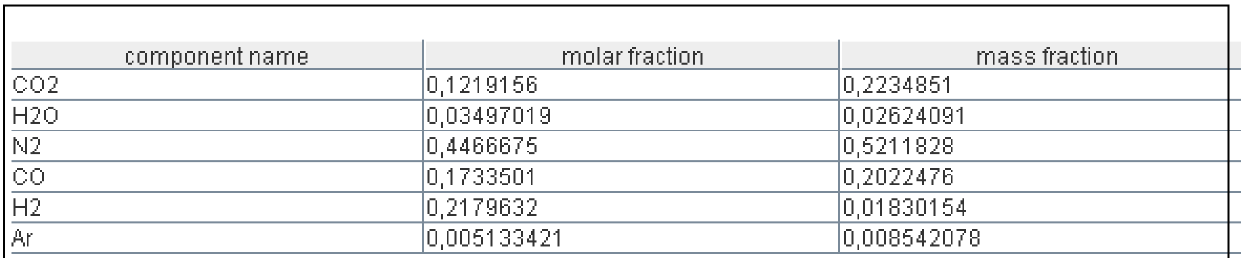


Figure 5: Diagram of component "water quench"

Condensing a portion of the water has the effect of increasing the gas LHV (Figure 6).



component name	molar fraction	mass fraction
CO2	0,1219156	0,2234851
H2O	0,03497019	0,02624091
N2	0,4466675	0,5211828
CO	0,1733501	0,2022476
H2	0,2179632	0,01830154
Ar	0,005133421	0,008542078

Figure 6: composition of the cooled synthesis gas (LHV: 4.24 MJ /kg)

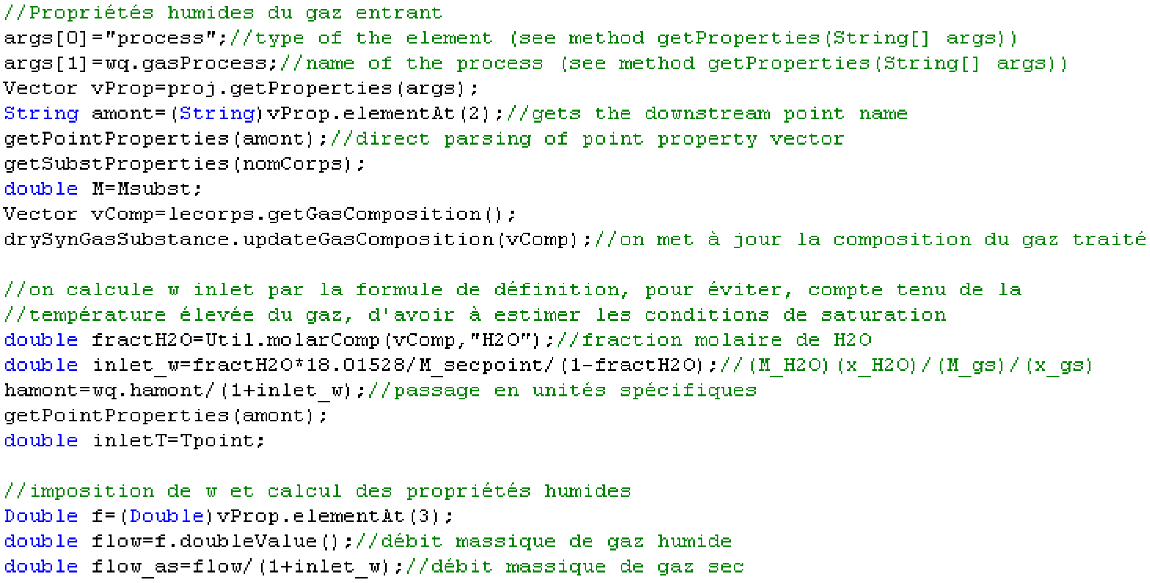
Study of the external class WaterQuench

To ensure consistency of the model (avoid that the inlet mixer connects to na inadequate outlet divider), each of the two nodes tries to instantiate the other seeking its class among the external components of the project, and verifies that both are well connected to the same connection process-point. If the operation fails, a message warns the user that the construction is incorrect. This is checked by the methods setupOutlet () and setupInlet ().

Moreover, consistency tests are performed on each node by the method checkConsistency () to check that the fluids are rightly connected: in this case, a wet gas and water in and out. Please refer to Volume 3 of the reference manual for explanations on this point, valid for all external nodes.

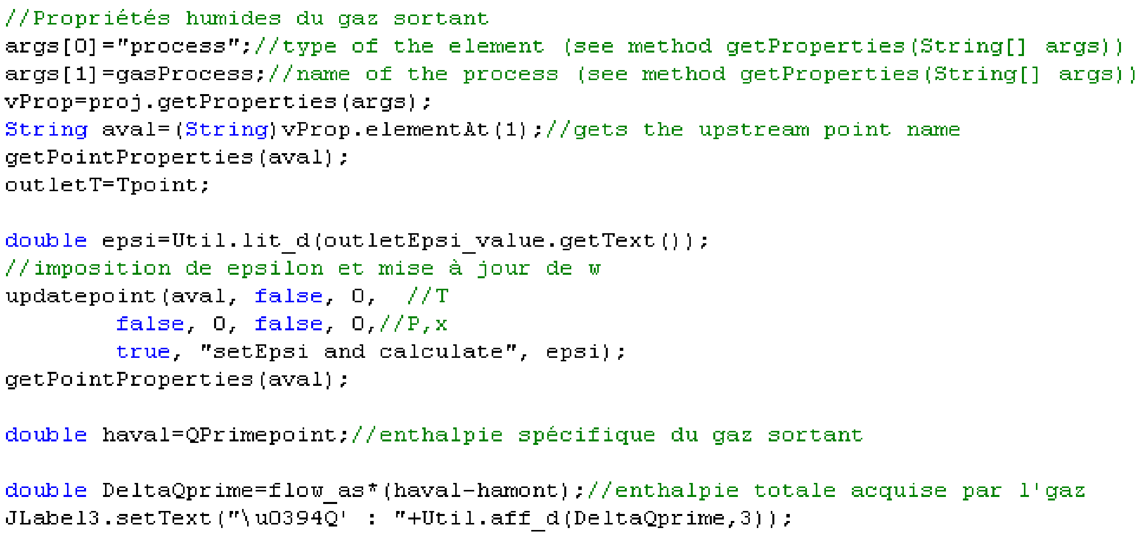
The study of the external class WaterQuench shows how the model was implemented. Six steps allow one to perform the calculations:

1) we first calculate the absolute humidity of the incoming gas, and determine the mass flow of dry gas from the wet gas then initialize the upstream specific enthalpy:



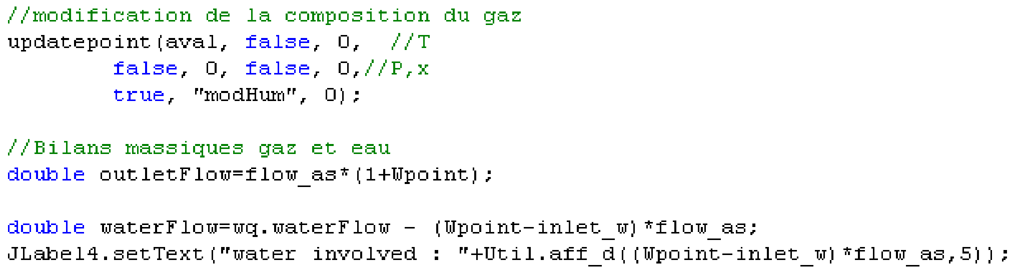
```
//Propriétés humides du gaz entrant
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=wq.gasProcess;//name of the process (see method getProperties(String[] args))
Vector vProp=proj.getProperties(args);
String amont=(String)vProp.elementAt(2);//gets the downstream point name
getPointProperties(amont);//direct parsing of point property vector
getSubstProperties(nomCorps);
double M=Msubst;
Vector vComp=lecorps.getGasComposition();
drySynGasSubstance.updateGasComposition(vComp);//on met à jour la composition du gaz traité

//on calcule w inlet par la formule de définition, pour éviter, compte tenu de la
//température élevée du gaz, d'avoir à estimer les conditions de saturation
double fractH2O=Util.molarComp(vComp,"H2O");//fraction molaire de H2O
double inlet_w=fractH2O*18.01528/M_secpoint/(1-fractH2O);//(M_H2O)(x_H2O)/(M_gs)/(x_gs)
hamont=wq.hamont/(1+inlet_w);//passage en unités spécifiques
getPointProperties(amont);
double inletT=Tpoint;

//imposition de w et calcul des propriétés humides
Double f=(Double)vProp.elementAt(3);
double flow=f.doubleValue();//débit massique de gaz humide
double flow_as=flow/(1+inlet_w);//débit massique de gaz sec
```

2) we then calculate the properties of the outlet gas, by setting the relative humidity read on the screen, and we deduce the total enthalpy into play



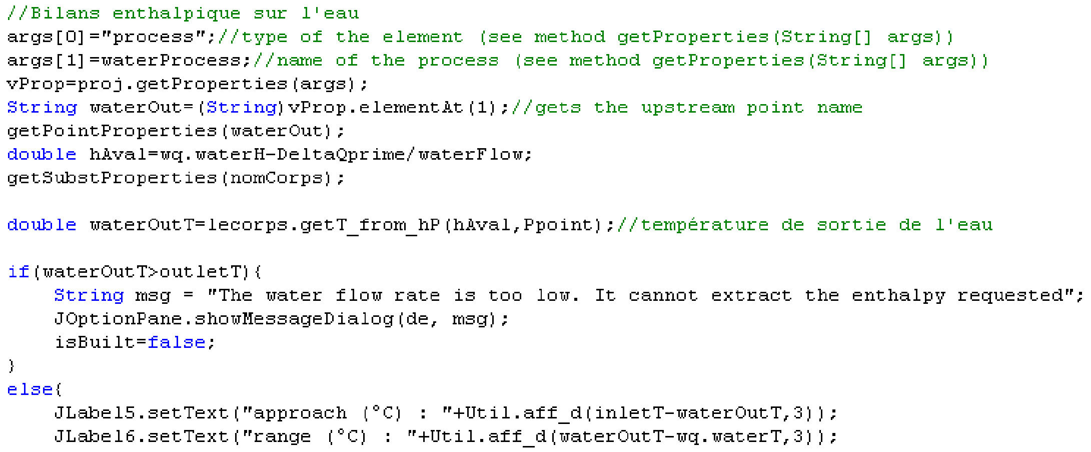
```
//Propriétés humides du gaz sortant
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=gasProcess;//name of the process (see method getProperties(String[] args))
vProp=proj.getProperties(args);
String aval=(String)vProp.elementAt(1);//gets the upstream point name
getPointProperties(aval);
outletT=Tpoint;

double epsi=Util.lit_d(outletEpsi_value.getText());
//imposition de epsilon et mise à jour de w
updatepoint(aval, false, 0,  //T
        false, 0, false, 0,//P,x
        true, "setEpsi and calculate", epsi);
getPointProperties(aval);

double haval=QPrimepoint;//enthalpie spécifique du gaz sortant

double DeltaQprime=flow_as*(haval-hamont);//enthalpie totale acquise par l'gaz
JLabel3.setText("\u0394Q' : "+Util.aff_d(DeltaQprime,3));
```

3) we modify the composition of the outlet gas, and calculate the mass flow rates out:



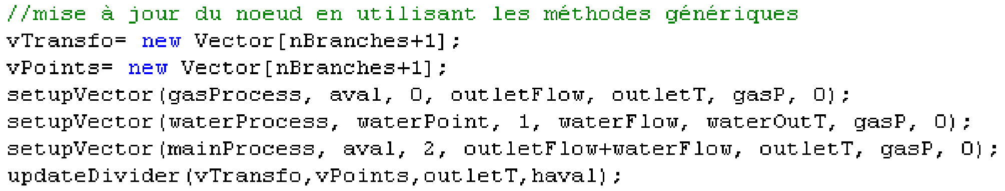
```
//modification de la composition du gaz
updatepoint(aval, false, 0,  //T
        false, 0, false, 0,//P,x
        true, "modHum", 0);

//Bilans massiques gaz et eau
double outletFlow=flow_as*(1+Wpoint);

double waterFlow=wq.waterFlow - (Wpoint-inlet_w)*flow_as;
JLabel4.setText("water involved : "+Util.aff_d((Wpoint-inlet_w)*flow_as,5));
```

4) the enthalpy balance on the water then provides the outlet water temperature



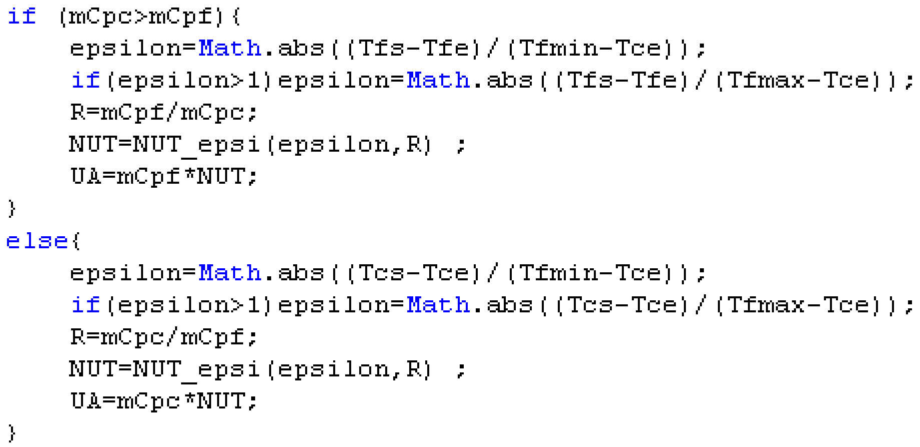
```
//Bilans enthalpique sur l'eau
args[0]="process";//type of the element (see method getProperties(String[] args))
args[1]=waterProcess;//name of the process (see method getProperties(String[] args))
vProp=proj.getProperties(args);
String waterOut=(String)vProp.elementAt(1);//gets the upstream point name
getPointProperties(waterOut);
double hAval=wq.waterH-DeltaQprime/waterFlow;
getSubstProperties(nomCorps);

double waterOutT=lecorps.getT_from_hP(hAval,Ppoint);//température de sortie de l'eau

if(waterOutT>outletT){
    String msg = "The water flow rate is too low. It cannot extract the enthalpy requested";
    JOptionPane.showMessageDialog(de, msg);
    isBuilt=false;
}
else{
    JLabel5.setText("approach (°C) : "+Util.aff_d(inletT-waterOutT,3));
    JLabel6.setText("range (°C) : "+Util.aff_d(waterOutT-wq.waterT,3));
```

5) the node is updated using the generic methods described in the reference manual



```
//mise à jour du noeud en utilisant les méthodes génériques
vTransfo= new Vector[nBranches+1];
vPoints= new Vector[nBranches+1];
setupVector(gasProcess, aval, 0, outletFlow, outletT, gasP, 0);
setupVector(waterProcess, waterPoint, 1, waterFlow, waterOutT, gasP, 0);
setupVector(mainProcess, aval, 2, outletFlow+waterFlow, outletT, gasP, 0);
updateDivider(vTransfo,vPoints,outletT,haval);
```

6) Finally, we estimate the overall characteristics of the heat exchanger



```
if (mCpc>mCpf){
    epsilon=Math.abs((Tfs-Tfe)/(Tfmin-Tce));
    if(epsilon>1)epsilon=Math.abs((Tfs-Tfe)/(Tfmax-Tce));
    R=mCpf/mCpc;
    NUT=NUT_epsi(epsilon,R) ;
    UA=mCpf*NUT;
}
else{
    epsilon=Math.abs((Tcs-Tce)/(Tfmin-Tce));
    if(epsilon>1)epsilon=Math.abs((Tcs-Tce)/(Tfmax-Tce));
    R=mCpc/mCpf;
    NUT=NUT_epsi(epsilon,R) ;
    UA=mCpc*NUT;
}
```