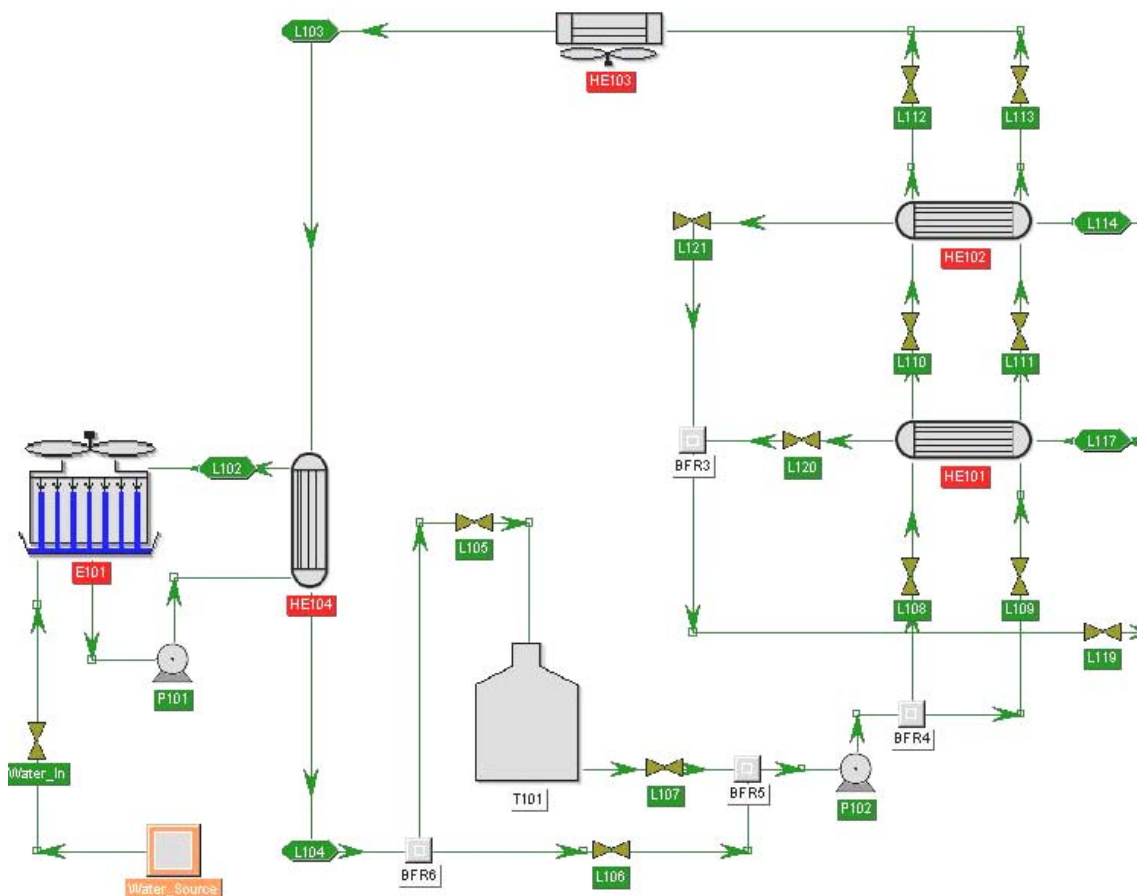


# Mobatec Modeller

## *Getting Started* *step-by-step tutorial for beginning users*



---

<b>1</b>	<b>Introduction</b> .....	<b>2</b>
1.1	Modelling Concepts .....	3
<b>2</b>	<b>Development and Simulation of a Simple Model</b> .....	<b>5</b>
2.1	Process Description .....	5
2.2	Physical Topology .....	6
2.3	Construction of the Physical Topology .....	8
2.3.1	Starting a New Model .....	8
2.3.2	Inserting New Systems .....	9
2.3.3	Inserting New Connections .....	10
2.3.4	Reshaping the Physical Topology .....	10
2.3.5	Adding Graphics to the Model .....	11
2.4	Equation Topology .....	12
2.5	Construction of the Equation Topology .....	13
2.5.1	Searching for Improper Equations and Uninitialized Variables .....	13
2.5.2	Adding a Pre-Defined Equation to an Object .....	13
2.5.3	Defining Parameter and Variable Values .....	15
2.5.4	Copying and pasting object equations .....	17
2.5.5	Defining New Equations and Variables .....	17
2.5.6	Initial Value Calculations .....	19
2.5.7	Code Generation .....	21
2.6	Using the Run-Time Simulation Environment .....	22
2.6.1	General Simulation Settings .....	22
2.6.2	Plotting, displaying and monitoring variables .....	24
2.6.3	Configuring a Simulation Run: Snapshots, Scripts and Sequences .....	28
2.6.4	Control Settings .....	30
2.7	Summary .....	31
<b>3</b>	<b>Introducing Species</b> .....	<b>32</b>
3.1	New Physical Topology .....	33
3.1.1	Group Systems .....	34
3.1.2	Reconnecting a Connection .....	36
3.1.3	Composite or Vector Connections .....	37
3.2	Species Distribution .....	37
3.2.1	Creating New Species and Adding Species Parameters .....	38
3.3	Defining Heat and Mass Connections of the New Physical Topology .....	39
3.3.1	Equation Class: mass or mole based balances system .....	39
3.3.2	Flexibility to define Mass Connections .....	40
3.3.3	Information Connections .....	41
3.3.4	Reference Species .....	42
3.3.5	Heat Connection of the New Physical Topology .....	43
3.4	Systems Definition .....	43
3.5	Initializing the New Physical Topology .....	44
3.5.1	Import State From Current Simulation .....	45
3.6	Graphical User Interface .....	45
3.6.1	Display Objects .....	45
3.6.2	Control .....	45
3.7	Summary .....	46
<b>4</b>	<b>Introducing Reactions</b> .....	<b>47</b>
4.1	Physical Property Database .....	47
4.2	New Topology .....	48
4.2.1	Connections and Systems Definition .....	49
4.2.2	Injecting Species .....	50
4.3	Inserting Chemical Reactions .....	50
4.4	Controller .....	52
4.5	Summary .....	52

# 1 Introduction

---

## Mobatec Modeller

*"Rapid development of consistent dynamic process models"*

*Mobatec Modeller is designed to effectively assist a model designer in building and maintaining consistent models. Using **Mobatec Modeller** significantly reduces the modelling time and overall effort.*

*Mobatec Modeller is a software instrument following a systematic modelling method (developed by prof. dr. dipl-ingH.A. Preisig and [dr. ir. M.R. Westerweele](#)) for the design of first principles (i.e. physical insight) based dynamic process models for physical, chemical and biological processes.*

This document introduces most of the elementary operations that are needed to build and test (dynamic) process models with **Mobatec Modeller**. A simple model of a central heating unit will be used to guide a beginning user through the required steps of developing and simulating a process model. Although the model is quite simple, a certain basic knowledge on thermodynamics is a prerequisite to understand all the steps taken in this tutorial.

We have purposely chosen to build this model completely "from scratch" (i.e. not using predefined objects from a library), such that you get a feel for all the possibilities of **Mobatec Modeller** for building and editing process models of any size. Several of the low level operations, such as defining (or even choosing) equations, will not be used by model designers during "normal" model design, since numerous library objects are available.

Many of the steps taken to build the tutorial example can be performed in several ways. It is left to the user if he prefers to use the toolbar, the context-sensitive popup menus (invoked by a right mouse-click) or the keyboard shortcuts (a summary of these can be found in the menu [Help][Keyboard Shortcuts]) to perform the required actions. Several "Show-Me-Movies" have been added to show the beginning user one way of getting to the desired results.

The **Mobatec Modeller** Help Files are divided into several documents. First time users, who wish to have a bit more in-depth knowledge about the used modelling methodology are recommended to first read though the document on "Concepts and Modelling Methodology", since **Mobatec Modeller** builds on several concepts that might be new to you. A simple description of the used modelling methodology will be outlined in the next paragraph.

### 1.1 Modelling Concepts

**Mobatec Modeller** is a computer-aided modelling tool which is designed to assist a model designer to map a process into a mathematical model, using a systematic modelling methodology. The modelling methodology implemented in **Mobatec Modeller** is based on the hierarchical decomposition of processes (in which material and energy exchange are playing a predominant role during normal operation) into networks of capacities (volume elements) and physical connections between the capacities. This means that a process is divided into a network of interconnected volume elements. Each volume element of such a network represents a capacity, which is able to store mass and energy (i.e. enthalpy) and consists of a single phase that is uniformly distributed (and hence displays uniform properties over its volume). The connections have no capacity and represent the transfer of mass and energy between the capacities.

The construction of a process model with this methodology consists of the following steps:

- 1) Break the process down into volume elements (also called systems or capacities) that exchange mass and energy through physical connections. The resulting network represents the *physical topology* of the process.

The way of breaking down the process into capacities and connections determines largely the level of detail included in the model (more capacities → more detail). It is consequently also one of the main factors for determining the accuracy of the description the model provides.

- 2) Describe the distribution of all involved chemical and/or biological species as well as all reactions in the various parts of the process. This represents the *species topology*.

This species topology is superimposed on the physical topology and defines which species and what reactions are present in each part of the physical topology.

- 3) **Mobatec Modeller** automatically generates all the needed balance equations for component mass and enthalpy of each system, since these balances can trivially be formed from the model designers definition of the physical and species topology of the process. The user cannot edit the generated balance equations!

- 4) Add algebraic equations (the so-called Constitutive Relations) to the model definition, such as transfer laws, kinetic rate expression, geometric relations, state variable transformations, etc.

The dynamic balance equations (step 3) and the algebraic equations, which are placed on top of the physical topology and species topology, represent the *equation topology*.

- 5) Add the (dynamic) behaviour of the information processing units, such as transmitters, adjusters and controllers.

These steps for building a model do not have to be done strictly in this sequence - at least not for the overall model. It is left to the model designer when the details are being specified in each part of the model.

The next chapter will introduce a tutorial example from which you can learn to setup a model from scratch. It is suggested, though, that before you start with the tutorial, you open a few example models and "click around" in these models, in order to get a feel for the program. You will notice that the Properties Dialog is context sensitive and directly shows the properties of the selected object. When you press the right mouse button when the mouse is over a button, text field, radio button, checkbox or list box in the Properties Dialog, a tooltip will appear that will provide some online help. This tooltip will also appear when you hold your mouse still for a few seconds on such an object (if this option has not been disabled in the Preferences Tab).

The tutorial example which is explored in the next chapters introduces the following concepts:

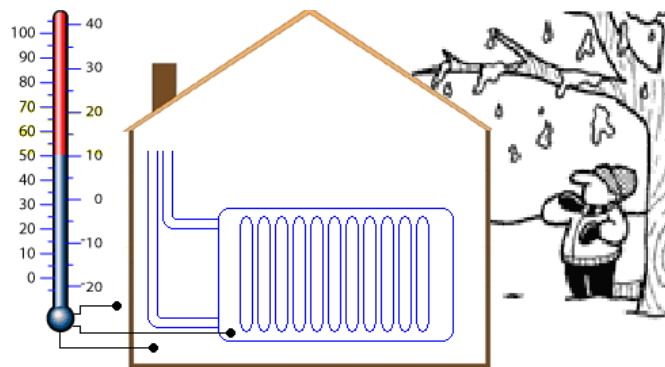
1. Setting up a model from scratch
2. Modelling objects (systems, connections, species, reactions)
3. Heat and Component Mass balances
4. Heat and Mass flows
5. Making new equations, species and reactions
6. Global variables
7. Information flows
8. Species (i.e. components) parameters
9. Reactions equations
10. Equation objects

## 2 Development and Simulation of a Simple Model

A model for the central heating of a room in house can be anywhere from very simple through very complex, depending on the amount of detail the model designer wants to take into account for the model. In this case, we are going to start out with a very simple model and will gradually work this out into a more detailed model. Step-by-step several concepts of **Mobatec Modeller** will be introduced and worked out.

### 2.1 Process Description

To get started, we keep our first model very simple. The “Process” only consists of a room, a heating device and the environment of the room. The objective is to get a model that can (dynamically) describe the temperature in the room.



1. A room with a heating device and a “cold” environment.

The room is considered to be a fixed volume  $V_{\text{Room}}$  ( $50 \text{ m}^3$ ) filled with air and is ideally mixed (i.e. uniform temperature). The initial room temperature is  $T_{\text{Room}}$  ( $293 \text{ K}$  ( $\approx 20 \text{ }^\circ\text{C}$ )). A heat leak to the surroundings is always present and occurs through a given fixed area  $A_{\text{Leak}}$  ( $20 \text{ m}^2$ ) with a given heat transfer coefficient  $U_{\text{Leak}}$  ( $2 \text{ J/m}^2/\text{K/s}$ ). The heat capacity of the wall is neglected. The heating element that heats up the room has a fixed heat transfer area  $A_{\text{Heating}}$  ( $4 \text{ m}^2$ ), with a fixed (i.e. temperature independent) heat transfer coefficient (from the heating element to the air in the room)  $U_{\text{Heating}}$  ( $5 \text{ J/m}^2/\text{K/s}$ ). The heating element can have any user-defined temperature  $T_{\text{HeatSource}}$  between  $283 \text{ K}$  ( $\approx 10 \text{ }^\circ\text{C}$ ) and  $363 \text{ K}$  ( $\approx 90 \text{ }^\circ\text{C}$ ). All heat transfer is assumed to be linearly driven by a

temperature difference. Heat by radiation is neglected. The (initial) ambient temperature (i.e. the temperature of the surroundings of the room) is  $T_{\text{Ambient}}$  (270 K ( $\approx -3$  °C)).

### 2.2 Physical Topology

In order to make a model of any process, we need to break it down into smaller parts, rendering an interconnected network of systems (capacities). The network of systems describes, so-to-speak, the physical structure of the process and is referred to as the *physical topology* of that process. This physical topology contains, once established, the maximum of information about the dynamic phenomena captured in the model. Any modification on the topology changes the dynamic information contents.

For the sake of modelling, some of the systems included within a physical topology can be defined as “non-dynamical systems”, because the dynamics of these systems is either outside the dynamic window, or is not of interest for the model (for example, a source or sink). The systems can be divided into several “sub-classes”:

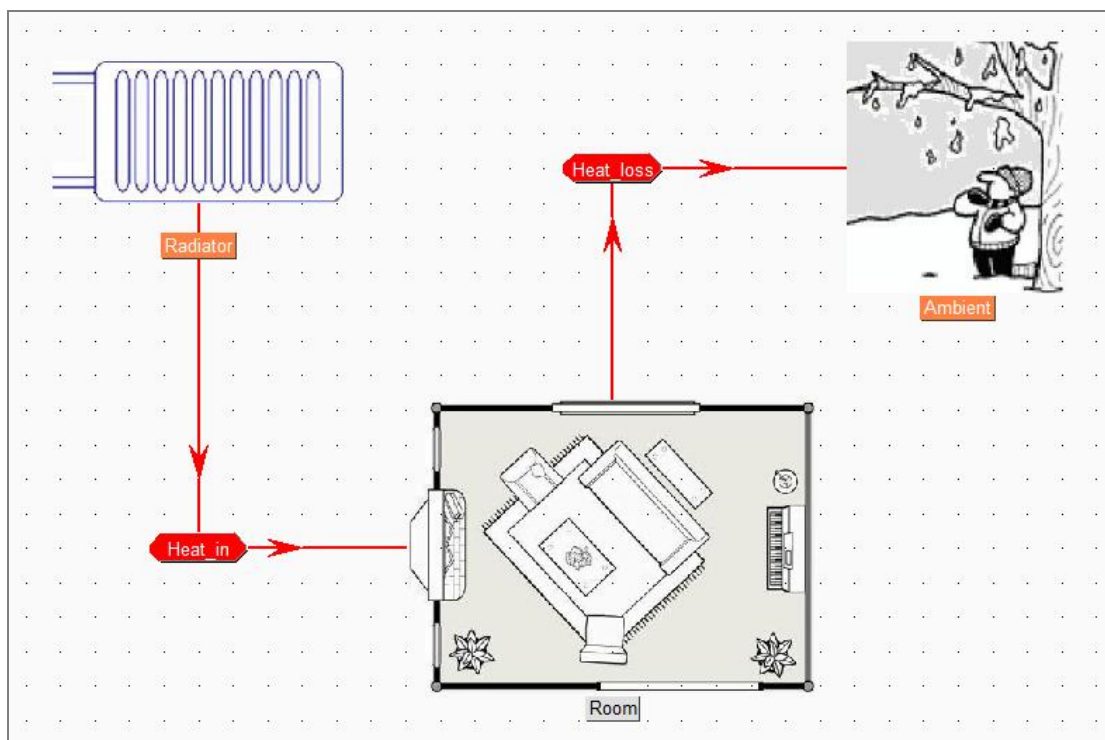
- **Capacity:** The system represents a lumped capacity for which the component mass balances and heat (i.e. enthalpy) balance will be automatically setup. All other variables will have to be directly or indirectly linked to the balanced variables.
- **Battery Limit:** A battery limit system is an infinitely large resource of which a user can infinitely fast change the physical properties (i.e. a source or a sink). Both outgoing and incoming connections can be connected to this type of system.
- **Steady State:** A steady-state systems represents a system with infinitely fast dynamics. The balance equations for the fundamental extensive variables are set up, but the differential terms are set to zero.
- **Information or Control:** Used to pass and process information. “Non-physical” equations (such as controller equations) are typically calculated in these systems. Only information connections (passing information to and from systems and connections) can be connected to information systems.
- **Interface:** Used to interface to other models. The other models can be solver specific or other Mobatec Modeller models.
- **Composite or Grouped:** With group systems the user can group several systems to increase the survey ability of the model. These group systems are especially handy, and even indispensable, when larger and more complex models are being built.
- **Repetitive Structures:** Used to build repetitive structures. To interconnect repetitive structures the "Set Repetitive System" feature of connections has to be used.

In this tutorial we will only use the capacity systems and battery limits systems to describe the dynamics of the example process. *Capacities* are the “normal” systems, which are used to describe the dynamic contents (i.e. the potential for energy accumulation in this example) of the process. For these systems the balance equations (for component mass and/or enthalpy) are automatically generated.

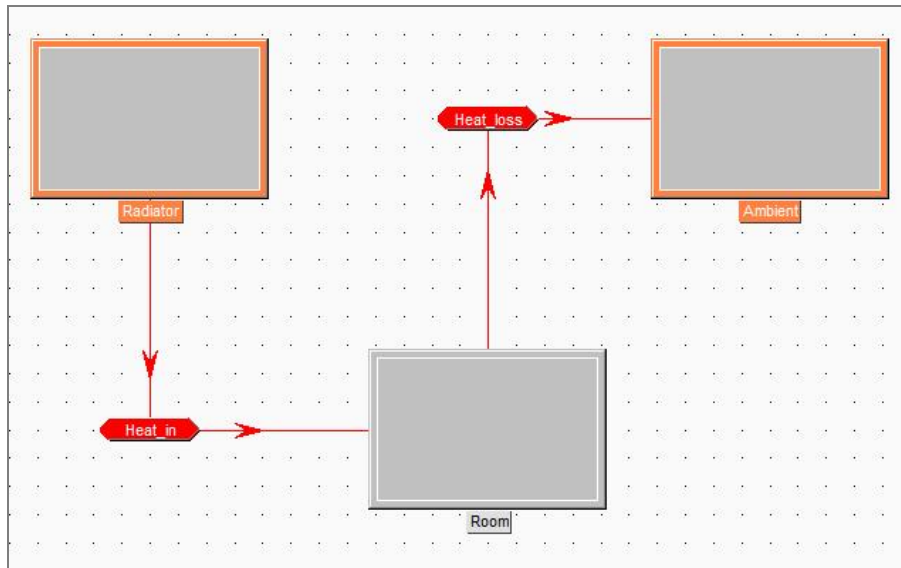
*Battery Limits* systems represent “resource” systems, the dynamics of which are not of interest for the model. These systems are usually infinitely large and only the intensive properties (e.g. temperature, pressure, component fractions, etc.) are defined, since the dynamics (which involve the extensive quantities (i.e. component mass and/or enthalpy)) of these systems are ignored. Another property of battery limits is that the values of the defined intensive properties can be altered infinitely fast.

***So, how would this physical topology look for the simple process we are considering?***

We want to model the influence of the ambient temperature and the temperature of the heating unit on the room temperature. In this example process, we are only interested in heat streams and assume that mass flows are of no interest (i.e. can be neglected) for our model.



2. Physical Topology for the room with a heating device.



3. Physical Topology for the room with a heating device, without pictures.


Figures 2 and 3 show a possible physical topology for our process (with and without pictures). There are three systems defined in this topology: *Radiator* (Battery Limit), *Room* (Capacity) and *Ambient* (Battery Limit). There are two heat flows: *HeatIn* and *HeatLoss*, which represent the heat fluxes from the heat source to room and from the room to the environment, respectively. Heat fluxes are represented by red connection objects. The arrows of the connection point from the origin of the connection to the target.

## 2.3 Construction of the Physical Topology

This paragraph will guide you through the required steps for building the physical topology as presented in figure 2. A preview of how this can be done can be watched by clicking the following link:


[Room 1 - Construct Physical Topology](#)

### 2.3.1 Starting a New Model

Select [File][New Model] from the menu or press [  ]. The **Properties Dialog** will show a **Name** field in which you can enter the name of the model. This is not the file name, but just the name of the top most system in the tree of systems you can define for a model. The name you enter cannot hold any "illegal" characters (such as #, !, @, \$, ^, \*, +, -, etc), since the name of a system can be used in the generated code files. The name can therefore also not start with a number.

Press the **OK** button and a new, empty model window will appear on the screen.

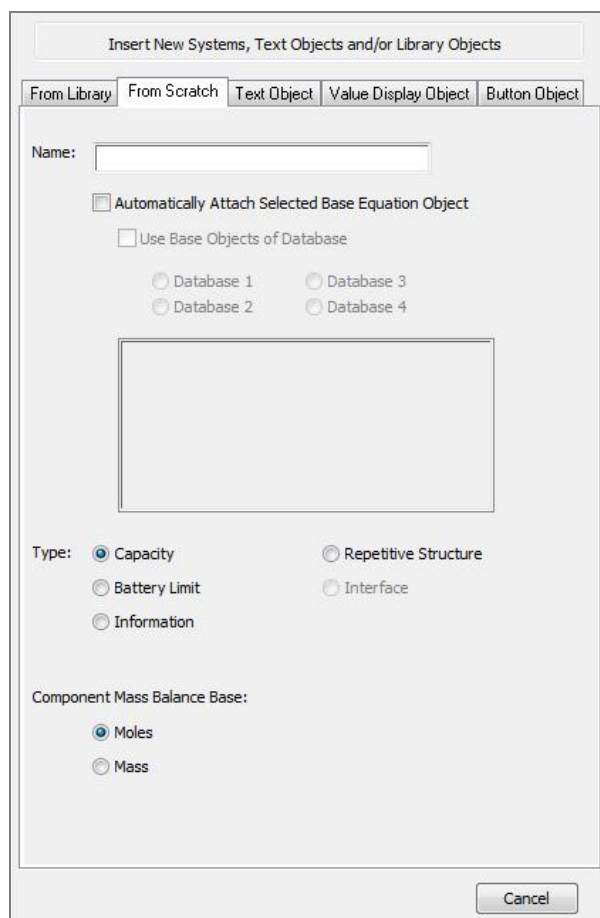
### 2.3.2 Inserting New Systems

Select [Edit][Insert New Systems] to be able to insert new systems. Alternatively, press [  ] in the **General** tab of the **Properties Dialog** or simply press "n". The **Properties Dialog** will now show several tabs. We are going to build a model completely from scratch in this tutorial and will not use predefined library objects. So, select the **From Scratch** tab and uncheck the "Automatically Attach Base Equation Object" checkbox (see figure 4).

A grey preview of where the system can be inserted is shown when you move your mouse over the model window. A left mouse click will insert a system. Select the appropriate system type (Capacity or Battery Limit) in the **Properties Dialog** (or press the appropriate keys on the keyboard to change the system type) and insert the three systems. Optionally, you can directly specify the system names in the **Name** field in the **Properties Dialog** (or by pressing F2). When no name is specified, **Mobatec Modeller** will generate a name from the systems unique object identifier.


A right mouse click (or pressing **ESC**, or selecting the **Cancel** button) will stop the "Insert New Systems"-mode.

Insert a Battery Limit system called "Radiator", a Capacity called "Room" and another Battery Limit called "Ambient".



4. **Properties Dialog** for inserting new systems (from scratch).

### 2.3.3 Inserting New Connections

Select [Edit][Insert New Connections] to add the two heat connections. Alternatively, press [  ] in the **General** tab of the **Properties Dialog** or simply press "m". First select **Connection Type** "Heat" on the **Properties Dialog**. To make the first connection, move the mouse to the *Radiator*. The system will light up when the mouse pointer is over it. Click on the *Radiator* (left mouse) and then select the *Room* system. The *Radiator* and *Room* are now connected via a heat connection. Next, click on the *Room* system and connect it with *Ambient*.

Optionally, you can directly specify the connection names in the **Name** field in the **Properties Dialog** (or by pressing F2). When no name is specified, **Mobatec Modeller** will generate a name from the connections unique object identifier.

To finish inserting connections, either click the right mouse button somewhere in the model window, press **ESC** or select **Cancel** in the **Properties Dialog**.

Name the connection from the *Radiator* to the *Room* "HeatIn" and the one from the *Room* to *Ambient* "HeatLoss".

### 2.3.4 Reshaping the Physical Topology

Inserting new systems and connections is done with default object sizes and connections are always aligned straight from the origin of the connection to the target, when they are newly defined. In general, however, people want to reshape, resize and realign objects.

Select the *Room* system and note that the **Properties Dialog** directly shows the object properties of *Room*. For user-convenience the **Properties Dialog** is fully context sensitive and directly reacts to mouse clicks in the model window. When a single object is selected, it can be resized and moved. When moving the object, you get a preview (in grey) of the new position of the moved object and all objects that are connected to it.

Connections can be graphically connected from their origin to their target in three ways:

1. **Auto Positioning** - Automatically assigns a position to the connection object. The connection object will be automatically positioned in a straight line in between its origin and target. The connection object can be selected, but it cannot be moved. The object will only move if its origin and/or target move.
2. **One Fixed Position** (default when inserting new connections) - The connection object can be freely moved over the screen. A straight line will be drawn from the origin of the connection to the connection object and another straight line will be drawn from the connection object to the target object. Moving the origin or target will not affect the position of the connection object.

3. **Three Fixed Positions** - The connection object and its two arrows can be freely moved over the screen. As with the one fixed position, the connection object can be freely moved anywhere on the screen. Additionally the two arrows that are drawn to and from the connection will become selectable and movable. Four straight lines will be drawn: from the origin object to the first arrow object, from the first arrow object to the connection object, from the connection object to the second arrow object and from the second arrow object to the target object. Moving the origin or target will not affect the positions of the connection and arrow objects.

Select one of the heat connections and on the **Properties Dialog** change the “**One Fixed Position**” radio button to “**Auto Positioning**” (this can also be done by pressing “1” on the keyboard when the object is selected). You will notice that the connection will be directly aligned between its origin and target. Now select the heat connection and on the **Properties Dialog** change the “**Auto Positioning**” radio button back to “**One Fixed Position**” (this can also be done by pressing “2” on the keyboard or by double clicking on the connection object). You can now freely move the connection object. When the option “**Three Fixed Position**” is chosen (by pressing “3”), also the arrows become active points that are freely movable.

As mentioned before, a selected object can be renamed by either changing the name in the Name field on the Physical Topology tab of the **Properties Dialog** or by pressing “F2” (this will open a small **Rename Box**). Keep in mind that object names may not contain illegal characters, since these names can be used in the generated code when compiling the model.

### 2.3.5 Adding Graphics to the Model

To make the model a bit more visually attractive, pictures or icons can be added to systems and connections. An icon first needs to be added to the “Plant Icons”, such that the same graphic can be used multiple times. Select the **Icons** tab on the **Properties Dialog** or click the **Define Plant Icons** button on the **Graphics** tab of a system or connection. On this **Icons** tab you can add or remove icons to and from the defined **Plant Icons**. A small preview of the selected icon will be available when an icon is selected in either the Plant Icons list or Available Icons list.

In order to insert icons from another location, just press the browse button to change the Icon Directory.

For extra online help, you can press the right mouse button on any *List Box*, button or field that has a tooltip associated with it. This online help will also appear if you hold your mouse still above such an object (when this option has not been disabled on the [Preferences][General] tab).

## 2.4 Equation Topology

As mentioned in the introduction chapter, **Mobatec Modeller** automatically generates the balance equations of the fundamental extensive quantities (component mass and enthalpy, for now), using the conservation principles. The resulting differential equations consist of flow rates and production rates, which are not further specified at this point. In order to fully describe the behaviour of the process, all the necessary remaining information has to be added to the symbolic model of the process. So, in addition to the balance equations, other relationships (i.e. algebraic equations) are needed to express transport rates for mass and heat, reaction rates, definitions of intensive variables, and so on. The resulting set of differential and algebraic equations (DAEs) is called the *Equation Topology*.

The equation topology forms a very important part of the modelling process, for with the information of this topology the complete model of the process is generated. The objective of the equation topology is the generation of a mathematically consistent representation of the process under the view of the model designer (who mainly judges the relative dynamics of the various parts, thus fixes intrinsically the dynamic window to which the model applies). In order to efficiently produce dynamic process models, **Mobatec Modeller** must, of course, appropriately deal with variables and equations.

For our simple heated house example, the equations are quite simple:

$$\text{Room Enthalpy Balance: } \frac{dH_{\text{Room}}}{dt} = Q_{\text{HeatIn}} - Q_{\text{HeatLoss}} \quad (\text{generated by Mobatec Modeller})$$

$$\text{Room Temperature: } H_{\text{Room}} = nt_{\text{Room}} * C_{p,\text{Room}} * T_{\text{Room}}$$

$$\text{Heat flux h0001: } Q_{\text{HeatIn}} = U_{\text{HeatIn}} * A_{\text{HeatIn}} * (T_{\text{Radiator}} - T_{\text{Room}})$$

$$\text{Heat flux h0002: } Q_{\text{HeatLoss}} = U_{\text{HeatLoss}} * A_{\text{HeatLoss}} * (T_{\text{Room}} - T_{\text{Ambient}})$$

in which,

H	=	Enthalpy, in J
Q	=	Heat flow, in J/s
U	=	Overall Heat Transfer Coefficient, in J/m <sup>2</sup> /K/s
T	=	Temperature, in K
nt	=	Total molar hold-up, in mol
C <sub>p</sub>	=	Specific Heat, in J/mol/K

## 2.5 Construction of the Equation Topology

We will now add the needed equations to complete our first model. A demonstration of the steps that are outlined in this paragraph can be viewed by clicking on the following link:

[Room 2 - Construct Equation Topology](#)

### 2.5.1 Searching for Improper Equations and Uninitialized Variables

**Mobatec Modeller** has a very powerful search feature, especially for larger and more complex models. For beginning users, however, it is also useful to use the search function to find objects where information about equations and variables is lacking.

Select [Edit][Find/Search..] from the menu (or press CTRL+F) to focus the **Search** tab in the **Properties Dialog**. Select **Equations and Variables** and choose **Improper Equations**. The **Result List Box** will now display all objects that do not have proper equations yet. Also, all objects with improper equations will be highlighted in the model window. Notice that the connections objects *HeatIn* and *HeatLoss* are highlighted, in our room model.

In order for a model to compile correctly, all objects need to be properly defined. When the user clicks on one of the objects in the **Result List Box** of the Search tab, **Mobatec Modeller** will automatically zoom in to this selected object (and highlight it). When the user double-clicks, the properties of the selected objects will be shown in the **Properties Dialog**.

If the checkbox **Automatically Update Results** is checked (default), **Mobatec Modeller** will update the search results whenever a change is made to the model. For extremely large models it is recommended that this checkbox is unchecked because it could get time consuming to check the entire model each time a modification has been done.

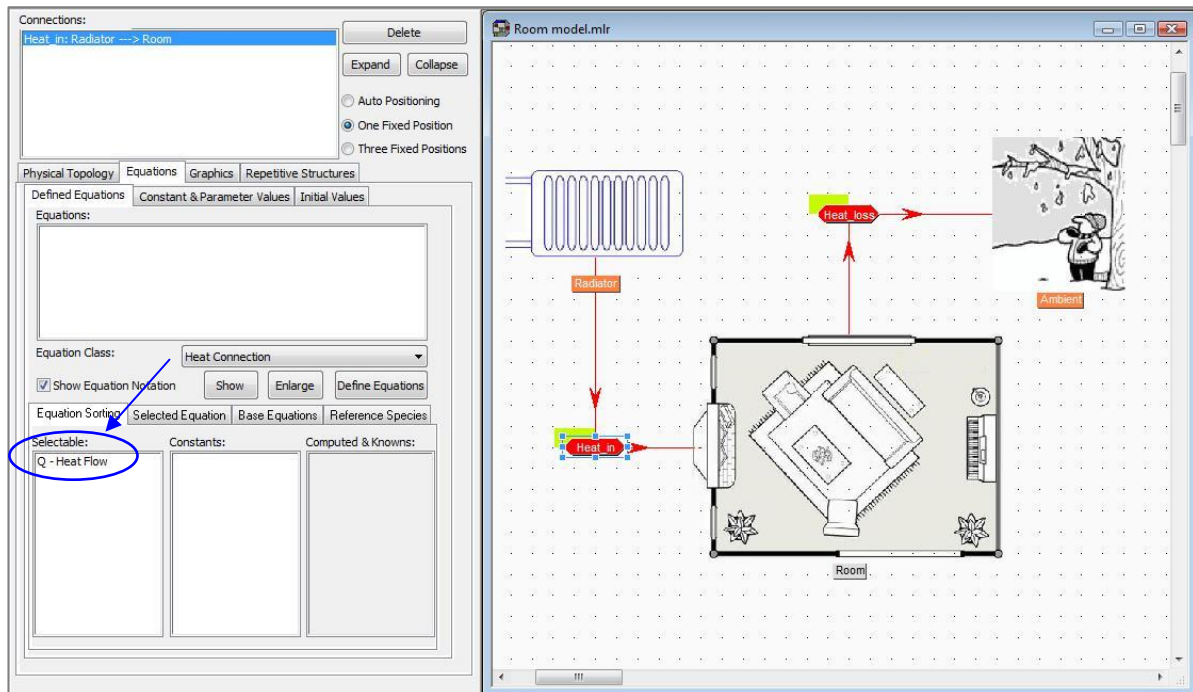
Now activate a second search by selecting the **Search 2** tab. On this tab select **Equations and Variables** and choose **Uninitialized Variables**. All objects with uninitialized variables and/or parameters will be highlighted.

We will keep these searches active while completing our model.

### 2.5.2 Adding a Pre-Defined Equation to an Object

Select the heat connection *HeatIn* and on the **Properties Dialog** select the **Equations** tab. The **Equations** tab is divided in 3 sub-tabs (for heat connections): **Defined Equations**, **Parameters** and **Initial Values**. The first tab, **Defined Equations**, shows which equations are defined in the currently selected object and holds 4 sub-tabs: **Equation Sorting**, **Selected Equation**, **Base Equations** and **Reference Species**. On the first sub-tab, **Equation Sorting**, you should notice

that one variable is selectable, namely “Q – Heat Flow”. This variable is there, because **Mobatec Modeller** automatically derives the balance equations for all capacity systems and the heat flow  $Q_{HeatIn}$  is present in the heat balance of system Room.



5. Screenshot of Room model. The properties of Heat Connection  $HeatIn$  are focussed. The Heat Flow (Q) is still selectable.

The appearance of Q in the **Selectable List Box** means that the user should either select this variable and consequently define it as a constant (or parameter) or he should define an equation from which Q can be calculated.

Press the **Define Equations** button and a database will open with all the predefined heat flow equations. In our case, there will only be one equation listed, since we only have access to the demonstration database. Select the database equation ( $Q = U * A * (or.T - tar.T)$ ) and add it to the **Object Equations**. Press **OK** to confirm your decision.

Notice that, after you have pressed **OK**, the “Search Lights” for **Improper Equations** will also burn on the systems  $HeatSource$  and  $Room$ , the origin and target of connection  $HeatIn$ . This is because the newly inserted equation refers to the temperatures in the origin and target (written as:  $or.T$  and  $tar.T$ ) and, therefore, the origin and target objects “know” that the variable “T – Temperature” should be defined. In these objects, “T – Temperature” is added as a selectable variable, such that the user can decide whether this temperature should be a constant or should be calculated from an equation. In the connection object  $HeatIn$ , the origin and target temperatures will be treated as known variables. The three other variables (U, A and Q) are selectable.

Select the variables U and A by clicking on them in the **Selectable List Box**. By doing this, the selected variables will be considered as “constants” (i.e. known) and will move to the Constants List

Box. The variable  $Q$  can now be calculated from the inserted equation ( $U$ ,  $A$ , or  $T$  and  $tar.T$  are known) and therefore  $Q$  will automatically be moved to the **Computed & Known List Box**.

To denote that  $Q$  can be calculated from the equation, the notation in the Equations List Box changes from "\*\*\*  $Q = U * A * (or.T - tar.T)$ " to " $Q: Q = U * A * (or.T - tar.T)$ ". The three stars (\*\*\*) before an equation in the **Equations List Box** tell the user that this equation cannot be used (yet) to solve a variable.

After selecting the two constants ( $U$  and  $A$ ), the connection object *HeatIn* will no longer have improper equations. There are some uninitialized variables, however.

### 2.5.3 Defining Parameter and Variable Values

Model variables that appear in equations can be divided into three "classes" in **Mobatec Modeller**: *Constants*, *Parameters* and *Variables*.

- **Constants** remain constant during simulation. They have a fixed value.
- **Parameters** are "adjustable" constants. The user can manually (or via a controller) adjust the value of a parameter. Typical examples of parameters are valve positions and battery limit conditions.
- **Variables** have a state-dependent value. The values are calculated each time step by an equation solver.

Select the **Constant & Parameter Values** tab and notice that the constants " $U$ " and " $A$ " are displayed, but there are no values defined yet. The Constant & Parameter Values tab holds a table that has 8 columns:

**Variable:** This column cannot be edited and displays the names of the variables (i.e. constants and parameters) that are currently defined by the equations of the currently selected object.

**Value:** In this column you can specify values for the variables of the current object. A value of "-" means the value is undefined.

**Unit:** Non-editable fields that display the engineering units of the variables. If you wish to change the engineering unit (or the default values of the minimum and maximum) of a specific variable, this should be done in the variables and equations database of the focused model (see paragraph 2.5.5).

**Minimum:** Minimum value for a variable. The following will always apply:  
Minimum  $\leq$  Value  $\leq$  Maximum

**Maximum:** Maximum value for a variable.

- M:** This checkbox determines whether the default values for the minimum and maximum of a variable are used or whether user specified values are used
- S:** Marking this checkbox will substitute the value of constant directly in the generated code.
- P:** Check this checkbox if you want the constant to be a parameter. For some solvers this means that an extra equation is added, such that the constant can be treated as a variable and is placed in the correct list.

Now enter the appropriate values for the overall heat transfer coefficient "U", the area "A", in the **Value** field (see section 2.1 for the suggested values). Next, select the Initial Values tab and notice that the variable Q is listed here. We will not enter a value here yet.

The Initial Values tab holds a table that has 8 columns:

- Variable:** This column cannot be edited and displays the names of the variables that have been already defined by the equations of the currently selected object.
- Value:** In this column you can specify values for the variables of the current object. A value of "-" means the value is undefined.
- Unit:** Non-editable fields that display the engineering units of the variables. If you wish to change the engineering unit (or the default values of the minimum and maximum) of a specific variable, this should be done in the variables and equations database of the focused model.
- Minimum:** Minimum value for a variable. The following will always apply:  
Minimum  $\leq$  Value  $\leq$  Maximum
- Maximum:** Maximum value for a variable.
- M:** This checkbox determines whether the default values for the minimum and maximum of a variable are used or whether user specified values are used
- S:** When the selected variable can be written explicitly by its defining equation, marking this checkbox will substitute every occurrence of this variable with the right-hand-side of the explicit equation.
- I:** (Only for system objects). If a checkbox in this column is checked, the corresponding variable(s) will be used for the initial value calculation. The initial value calculation will be performed in two steps. In the first step the variables marked with "X" in this column will be treated as known and the fundamental extensive quantities (component mass and enthalpy) are calculated (if possible) from the equation sorting that results from this. In the second step, the initial values are calculated

from the fundamental extensive quantities. In this way a consistent set of initial values is guaranteed.

Select the *Radiator* system and focus on the **Defined Equations** tab. T- Temperature should be listed in the Selectable List Box. Since the heat source is to be “freely” chosen in our simple model, we will define the temperature as a parameter. We will insert, for instance, a value for the temperature of the radiator of 60 °C.

First, click on “T – Temperature” in the **Selectable List Box**, such that it becomes a constant. Then select the **Constant & Parameter Values** tab and enter the appropriate start value for the heater temperature. Next, select the “P”-column, such that an “X” appears in this column for the *Radiator* temperature. T is now a user-adjustable parameter. You can also check the “M”-column to adjust the minimum and maximum temperature of the heat source.

Select the *Ambient* system, define the temperature as a constant and enter a value for this temperature.

### 2.5.4 Copying and pasting object equations

When an object needs the same equations as an already defined object, you can use the **Copy/Paste Equations** feature to quickly copy the equations of the defined object and paste it to other objects (of the same class).


Select connection *HeatIn* with the right mouse button, such that the context menu appears. In this menu, choose **Copy Equations**. Alternatively, you could select *HeatIn* and press “E” on the keyboard (a list of all shortcut keys can be found in the menu [Help][Keyboard Shortcuts]). Now, select connection *HeatLoss* and choose **Paste Equations** from the context menu (or press “e”). All defined equations, including the equation sorting, are now copied to connection *HeatLoss*. Since no variables were defined yet in connection *HeatLoss*, the values of the parameters and variables are copied as well.

### 2.5.5 Defining New Equations and Variables

Sometimes, an equation you wish to use is not yet defined in your available databases. **Mobatec Modeller** offers you ample abilities to define any kind of equation and add it to any database.

We are now going to add the Enthalpy equation for the room:  $H = nt * Cp * T$

This equation is not available in the demonstration database, so we will have to add it to the database before we are able to use it in our model.

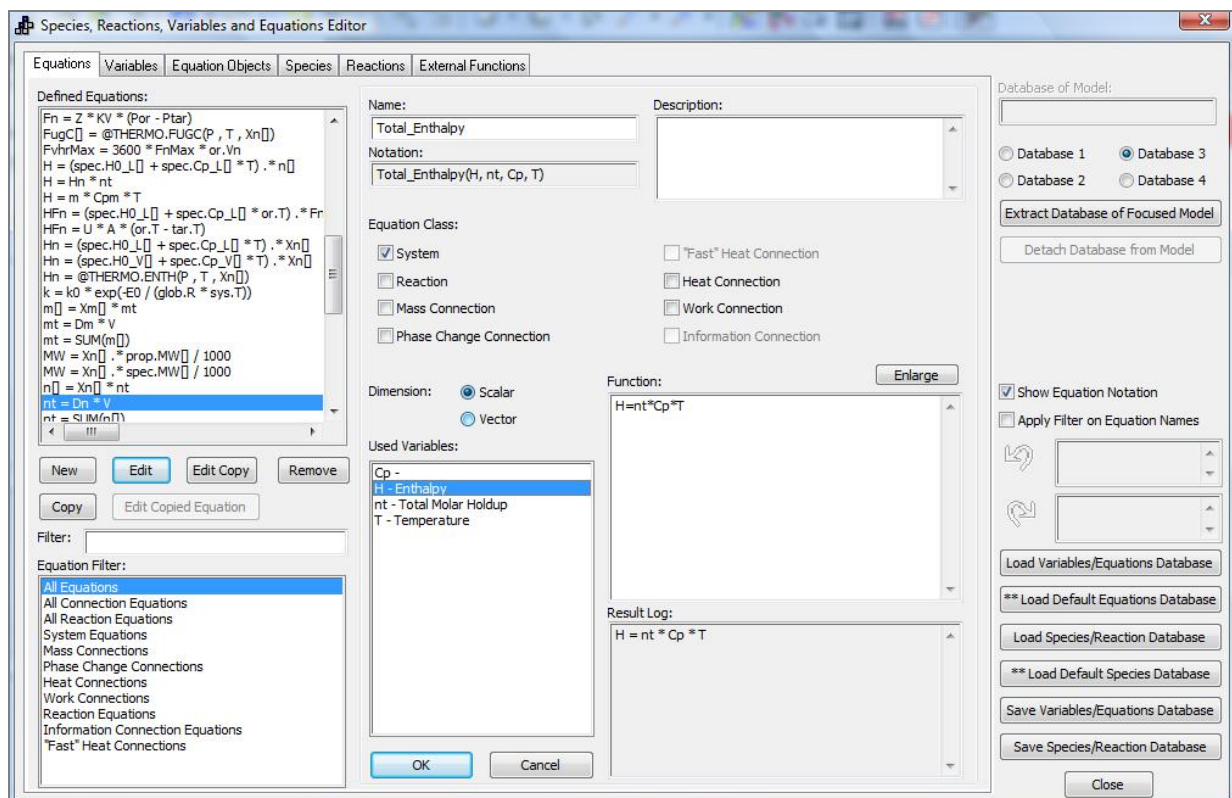
Select [Tools][Database Editor] from the menu or click on [  ]. The **Database Editor** will open, showing the default database (in our case, the demonstration database). Select the **Equations** tab and click the **New Equation** button to define a new equation.

All equations in a database (and also in a model) must have a unique name. This name can be used by some of the supported solvers for code generation. Therefore it cannot contain any "illegal" character, such as \*, #, %, \$, @, etc.

Enter a valid name for the Enthalpy equation in the **Name** field. We want to use this equation in the Room system, so we are dealing with a System equation. Check the **System** checkbox in the list of available **Equation Classes**.

Select the Function field and enter the equation in symbolic form:  $H = nt * Cp * T$

Notice that, while typing, the equation will be parsed (i.e. checked) and **Mobatec Modeller** will automatically derive which variables are used in the equation. When the equation contains any errors, it will not be possible to click the **OK** button.



6. Database Editor, focused on the Equations tab. Entering a new equation.

Click the **OK** button and select the Variables tab. By entering the new equation, a new variable, **Cp**, has been introduced. Find this variable in the **Defined Variables List** and double click on this variable. The fields of this variable can now be edited. Give the variable the following specifications:

Symbol:	Cp
Name:	Specific Heat Constant
Engineering Unit:	J/mol/K
Upper bound:	1.7E+308
Lower bound:	0

Click the **OK** button to accept the changes.

Click the **Close** button to exit the **Database Editor**. The new equation will now be available for use in our model.

Select the Room system and on the **Defined Equations** tab (of the **Equations** tab) click on the **Define Equations** button. The **Set Equations Dialog** that appears now, contains a **Filter** field. This filter is especially handy for larger equation databases. All equations that contain the specified filter string are displayed in the **Database Equations List Box**.

Type "T" in the **Filter** field and notice that all equations without this letter disappear. Now select the **H = nt \* Cp \* T** equation and add it to the **Object Equations**. Click **OK** to confirm your changes.

Define **nt** and **Cp** as constants and give them appropriate values (The Cp of air is roughly 29.1 J/mol/K, and the molar hold-up of 50 m<sup>3</sup> room at 1 bar and 20 °C is about 2050 mol (calculated via the Ideal Gas Law:  $P * V = nt * R * T$ ))

### 2.5.6 Initial Value Calculations

When you want to perform a dynamic simulation run of your model, you need a certain start condition, the so-called the **initial values** for all defined variables. In order for the solver to find a correct solution, you need to provide a consistent set of initial values.

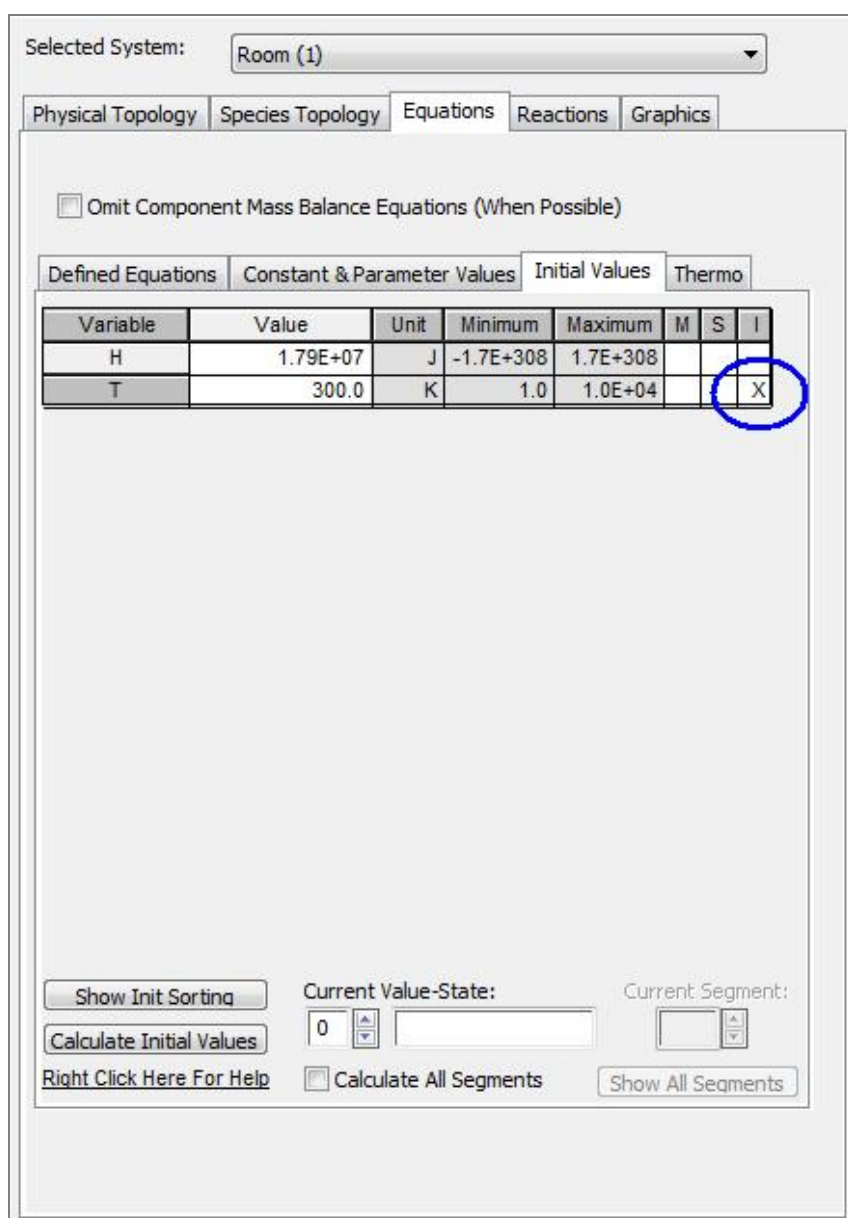
The variables of an object (either a system or a connection) are related to each other via the defined algebraic equations of the object. Therefore their values can, in principle, not be chosen independently. When you think about it, all (initial) values are actually related to the fundamental state variables (component mass and enthalpy) of the defined capacities. (The document *Concept and Modelling Methodology* explains this issue in more detail). So, providing values for the component hold-up (i.e. component mass) and enthalpy would, in principle, be enough to let the computer compute (i.e. calculate) a consistent set of initial values, using the defined algebraic relations.

In general, however, a modeller has no direct knowledge about the component mass and enthalpy in a system. Usually, he does have some information about, for example, component fraction, pressure and temperature in systems. Therefore it is also possible with **Mobatec Modeller** to select a set of variables (in a system) from which first the values of the fundamental state variables can be

calculated. From these values, the other values will then be computed to get a consistent initial value set (for this system).

Select the *Room* system and on the **Equations** tab select the **Initial Values** sub-tab. Enthalpy (H) and Temperature (T) should be list of variables. Enter a value of 300 K for T and afterwards check the "I"-column for T.

Optionally, press the **Show Init Sorting** button to check if the equation sorting, for performing the initial value calculation, is correct and that all variables can be calculated from the current selection (there should be no variables listed in the **Selectable** Listbox of the dialog window that appears).



Selected System: Room (1)

Physical Topology | Species Topology | Equations | Reactions | Graphics

Omit Component Mass Balance Equations (When Possible)

Defined Equations | Constant & Parameter Values | Initial Values | Thermo

Variable	Value	Unit	Minimum	Maximum	M	S	I
H	1.79E+07	J	-1.7E+308	1.7E+308			
T	300.0	K	1.0	1.0E+04			X

Show Init Sorting | Current Value-State: 0 | Current Segment: [ ]

Calculate Initial Values |  Calculate All Segments | Show All Segments

[Right Click Here For Help](#)

- Initial Values tab of Room system, with T as variable from which the initial values are calculated.

Next, click the **Calculate Initial Values** button. A value for the enthalpy **H** should appear. The initial value for enthalpy was calculated via the user-defined relation  $H = nt \cdot Cp \cdot T$ . During a simulation run this equation will be used to calculate the temperature **T**, since the enthalpy will be calculated from the (automatically generated) enthalpy balance.

The initial values of connection objects are, in most cases, (indirectly) coupled to the fundamental state variables of the interconnected systems. The values of these variables are treated as “known” values for the connection object, which makes the initial value calculation for connection objects a trivial operation.

Select the *HeatIn* heat connection and on the **Initial Values** sub-tab of the **Equations** tab click the **Calculate Initial Value** button. A value for the heat transfer **Q** is now calculated via the relation  $Q = U \cdot A \cdot (or.T - tar.T)$ . Do the same for connection *HeatLoss*.

### 2.5.7 Code Generation

After completing all the modelling steps, **Mobatec Modeller** is ready to generate a code file with a consistent set of model equations. Since this code consists of mathematical equations, an interface to any problem solving package could, in principle, be established. Several **Differential Algebraic Equation** solvers are currently supported by **Mobatec Modeller** (e.g. Mobatec LauTrane Solver, Process Studio’s e-Modeler, Aspen Custom Modeller, Matlab, ...). Check your license ([Tools][License]) to see which solvers are available to you.

**Mobatec Modeller** has an integrated **Run-Time Simulation Environment** for **Mobatec LauTrane Solver**, such that you can test and tune your model (or parts of it).

Click anywhere in the model window to display the general properties in the **Properties Dialog**. Select the **Model Generation** tab and focus the **Mobatec Solver** sub-tab. Click on the **Show Code** button to display the code that is generated by **Mobatec Modeller**. This code can be compiled by our **Mobatec Built-In** compiler or by the **Mobatec LauTrane** compiler (which requires either an installed Fortran or C compiler). The compiled code can be used for doing dynamic simulation runs for testing and tuning. A (tuned and tested) model can be easily connected to other software and, for example, form the basis for an Operator Training Simulator.



## 2.6 Using the Run-Time Simulation Environment

A demonstration of the steps that are outlined in this paragraph can be viewed by clicking on the following link:

[Room 3 – Using Run-Time Simulation Environment](#)

Once all the equations are properly defined and all the variables initialized, the model can be compiled and then tuned and tested in the Run-Time Simulation Environment (click on "*Compile and go to Simulation*").

Click anywhere in the model window to display the general properties in the **Properties Dialog**. Select the **General** tab and click on **Compile and go to Simulation** tab. A dialog box will appear informing us that "*The model was successfully loaded into the Simulation Environment*".

The user has the option to move from **Simulation** to **Modelling Environment** by clicking on [  ] or vice versa [  ].

### 2.6.1 General Simulation Settings

We will notice first that the tabs of the **Properties Dialog** have changed with respect to the **Modelling Environment**. The **General** tab contains all the simulation settings:

- **Run the current Simulation Script [F6]:** The currently loaded Simulation Script (if any) will be executed.

With Simulation Scripts a user can perform a sequence of actions automatically. If a script contains one or more errors, the script cannot be executed. For help on scripts, browse to the Scripts tab and select the "Script Help" button.

- **Load a Simulation Script:** A browser will open from which you can load a Simulation Script (.ssf) File. If a simulation script file should be loaded automatically upon loading the simulation model, the file should be located in the same directory than the model (.mlr) file.
- **Perform a Simulation Run [F7]:** A Simulation Run performs calculations for a defined run period (for dynamic simulation) or a maximum number of steps (for steady-state simulations). During a Simulation Run there will be (almost) no screen updates of variable tables and plot windows to make the calculation time as short as possible.

A Simulation Run can be aborted by pressing the Stop button or by clicking the right mouse button in the model window.

- **Start calculations [F9]:** Calculations will start and will only stop when:
  - 1) the stop button has been pressed.

- 2) a calculation error has been encountered.
- 3) a Steady-State simulation has reached its solution.
- 4) the user switches to the Modelling Environment.

After each calculation step all variable tables and plot windows will be updated. Also, all state dependent information that is displayed in the model (Value Objects) is updated after each step.

- **Single step [F8]:** A single calculation step with the user defined step size will be performed. After this step, all displayed information will be updated (if no calculation error has occurred).
- **(Re) Load Model:** In case the Built-in Solver is used, the last compiled model will be reloaded into the Simulation Environment.

If a Fortran or C-compiled model is used, the model that is specified by the Model Folder (on the Solver Options tab) will be loaded.

In both cases, all plots will be cleaned and any connections to OPC servers will be broken.

- **Clean All Plots:** All the graphs plotting the desired variables will be cleaned.
- **Stop Calculations:** This will stop the currently active calculations, simulation run or simulation script. In case the *Adaptive Step Size* integration method is used, the step size of the next calculation will be reset to the given step size.
- **Rewind Simulation:** All values of the parameters and variables of the currently active simulation will be reset to the value they had when the model was loaded into the Simulation Environment. Also, by default, all plot windows will be cleaned and any active sequences will be reset.
- **Number of steps:** When a Steady-State simulation is performed, this number represents the maximum number of steps that will be calculated before a Simulation Run will be stopped. The calculations are also stopped, if the simulation run converges before this number is reached.
- **Run Period:** When a Dynamic Simulation Run is performed, this value represents the simulation time interval (in seconds) after which the calculations should stop. In case of a Fixed Step size, the number of calculation steps can be calculated by dividing the Run Period by the Step Size. In case of an Adaptive Step, the number of calculations depends on how quick the model converges.
- **Step Size:** This value represents the step size (in seconds) of the calculations for the **Fixed Step Size** integration method. When the **Adaptive Step Size** integration method is chosen, it represents the initial step size. If the model is converging the step size will increase otherwise it will decrease (to find a more accurate solution).
- **Simulation Time:** This value represents the calculated simulation time (not the time it takes to do the calculations). It can be chosen randomly.

When a simulation is rewound, this time is set to 0, by default.

- **Calculation Delay (in ms):** To mimic Real-Time behaviour a calculation delay can be introduced. E.g. a calculation delay of 1000 ms and a step size of 1 sec, will give a simulation step result 1000 ms after the previous results were available (unless a calculation step takes more than 1000 ms).

- **Simulation type:** Either a **Steady-State** or a **Dynamic** simulation can be performed.

When trying to find a **Steady-State** solution, the solver will try to converge as quick as possible to the final values. The accuracy of the intermediate results is of less interest.

During **Dynamic** runs, more attention is paid to the accuracy of the results of each time step.

- **Integration method:** The user can determine if the problem should be solved with a **Fixed Step Size** or if an **Adaptive Step Size** can be used.

In general, the **Adaptive Step Size** integration method is more accurate (when no sudden changes are made, if the step size has grown big!) and gives fast results if the model converges quickly. If the relative unbalances remain big during calculations, the adaptive step size method will reduce the calculation step to its minimum, resulting in long (but more accurate) calculations.


The **Fixed Step Size** integration method gives solutions at fixed time interval, independent of the accuracy.


The rest of simulation settings, i.e. tolerance or number of iterations, can be stated in the **Solver Options** tab.

In the **Settings** tab a simulation desktop can be loaded and the user is able to attach and to activate the desired features for the current simulation.

### 2.6.2 Plotting, displaying and monitoring variables

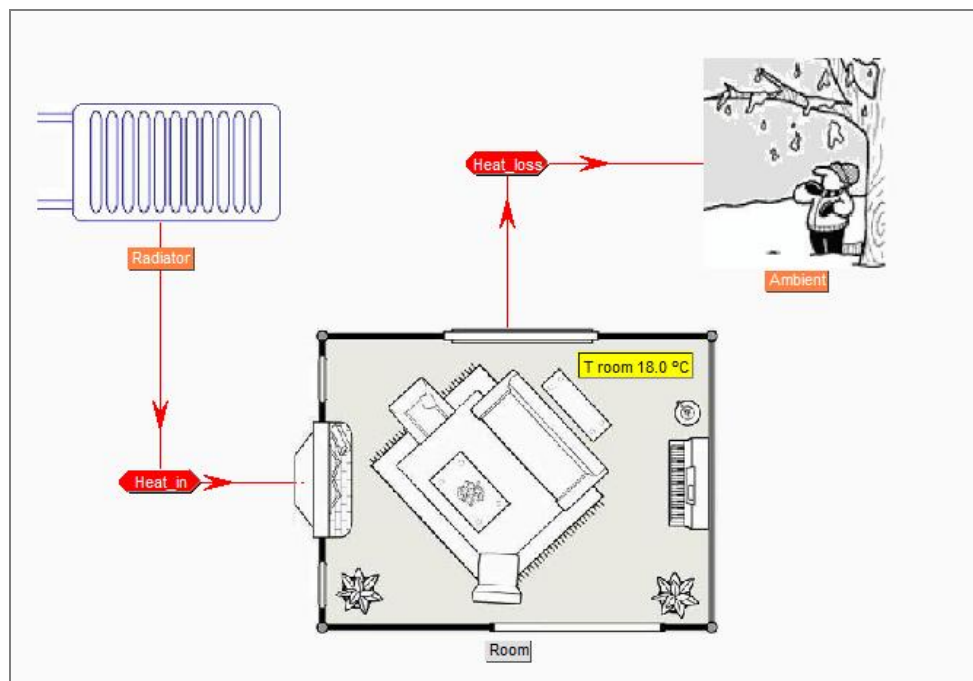
**Mobatec Modeller** allows the user to visualize all desired variables and parameters in the **Simulation Environment**. By clicking the right mouse button when the pointer is over a system or connection, a table with either the variables or the parameters defined in the object will be displayed. The table will also appear by clicking on "*Show Variable Table*" or "*Show Parameter Table*" in the **Variable Table** tab or by pressing "v" or "p" respectively when the model window is focused. The table contains a **Filter** field which is especially handy when many variables or parameters are defined in a system. All variables or parameters that contain the specified filter string are displayed in the table.

Very often the user is interested in having a monitoring scheme attached to the physical topology of the model which displays the most relevant variables (i.e. the temperature of the room). In **Mobatec Modeller** display objects can be added to the physical topology with the aim of monitoring variables of the system. First, we have to switch from **Simulation** to **Modelling Environment** (e.g. by clicking on [  ]). In the **General** tab, click on **Insert New Objects** and select the sub-tab **Value Display Object**. The next step is to locate the display object in the flow sheet by clicking with the left button mouse on the desired location. One click on the object will open the *Value Object* window in which the user can define its configuration by modifying several settings:

- **Display Type:** The user can choose between *Value Display* (numerical display) or *Filled Rectangle* (graphical display). In our example, we will select *Value Display*.
- **Description:** A brief description or the name of the variable to be displayed can be added (i.e. "T room").
- **Simulation Variable:** The name of the variable to be displayed. It must be the same name defined in the *Species, Reactions, Variables and Equations Editor* [  ] (i.e. "T"). Alternatively, a full variable name (as used in the simulation environment) can be entered (e.g. "T\_Room"), but this could make copy/pasting this object less convenient.
- **Linked Object:** The user has to select the object that the variable belongs to (i.e. "Room").
- **Engineering Unit String:** The user can add the units of the variable to be displayed (i.e. "°C").
- **Number of Decimals:** Number of decimals to be displayed in the object (i.e. "1").
- **Display Value Offset:** The user can include a value offset to the variable. For instance, if we want to display the temperature in Celsius an offset of -273.15 has to be inserted since the temperature is defined in Kelvin (i.e. "-273.15").
- **Display Value Multiplication Factor:** In order to convert the units defined in the variable editor a multiplication factor has to be inserted in this option (in our case "1"). For instance, if the mole flow is defined in the Editor in mol/s and the user wants to display it in kmol/h, a value of 3.6 would have to be inserted.
- **Max. Value:** Maximum value of the variable to be displayed. Important when the variable is being displayed as a filled rectangle (i.e. "100").
- **Min. Value:** Minimum value of the variable to be displayed. Important when the variable is being displayed as a filled rectangle (i.e. "-20").
- **Allow Value Access in Simulation Environment:** The user has the option to either monitor the variable just to display it or to modify the value of the variable during a simulation run (normally when parameters are the variables being displayed, i.e. valve position). In the example, the object will display a variable, so we will not select this option.
- **Draw outline:** This will draw an outline around the object.

- **Draw in Background:** Option to send the object to the front or to the back in the flow sheet.
- **Orientation:** Select the way the rectangle is being filled in case of a Filled Rectangle.
- **Value Object Color:** The user can choose the color of the object.
- **Value Object Font or Fill Color:** The user can select the font color or, in case of a rectangle, the fill color.
- **Width & height:** The user can define the size of the object.

In figure 8, the physical topology of our Room model including the temperature display is presented.



8. Physical topology with a display object monitoring the temperature of the room.

Now we will switch back to **Simulation Environment** by clicking on [  ] or by pressing F11.

In the **Plotting** tab the user has the option to create 25 plot windows to plot the desired variables defined in the model. The user only has to select first the variables in the table, then click on "Add Selected Variables From Variable Table" and finally press on "Show/Update Plot Window" (or just right click mouse on the selected variable/s in the **Variable Table** and the options "Add Selected Variables to Current Plot" and "Add Selected Variables to New Plot" will show up). Alternatively, a variable name can be typed at the "New Plot Variable" field and can be added by pressing "Add Plot Variable". If the variable does not exist in the model, nothing will happen. Otherwise the variable will be added to the Defined Plot Variables.

The plots, whose features can be modified in the **Plot Properties** sub-tab (i.e. scale Y-Axis, Time Axis, etc), are being drawn while the simulation is running. Besides having a dynamic visualization of the simulation run, this allows the user, for instance, to visualize the influence of modifying parameters in the variables the user is interested in.

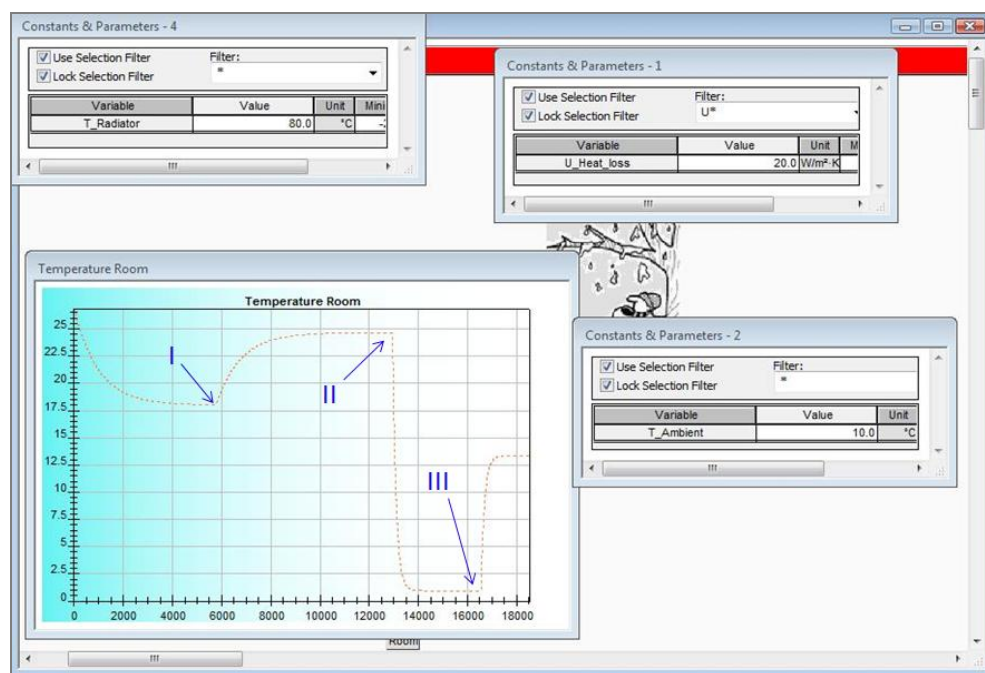
In our *Room* model the temperature of the room is the variable we are interested in. For instance, we could analyze the influence of both the ambient temperature and the heating system temperature on the temperature of the room. Also it could be interesting to modify the properties of the wall (i.e. the heat transfer coefficient of the wall of the room).

For performing a real time dynamic simulation of our *Room* model, the settings to state are the following: in the **General** tab we will select "*Dynamic*" as **Simulation Type**, "*Fixed Step Size*" as **Integration Method** and *1000 ms* as **Calculation delay** (i.e. real time simulation).

To start the simulation click on **Start Calculations**. The temperature of the room will start to be plotted. At any time the user has the option to modify the parameters (i.e.  $T_{\text{radiator}}$ ,  $T_{\text{ambient}}$  or  $U_{\text{wall}}$ ) and to see how the temperature of the room is affected by those changes. In figure 9 some simulation results are presented. In this simulation run some parameters were tuned to analyze how the temperature in the room responds to those changes.

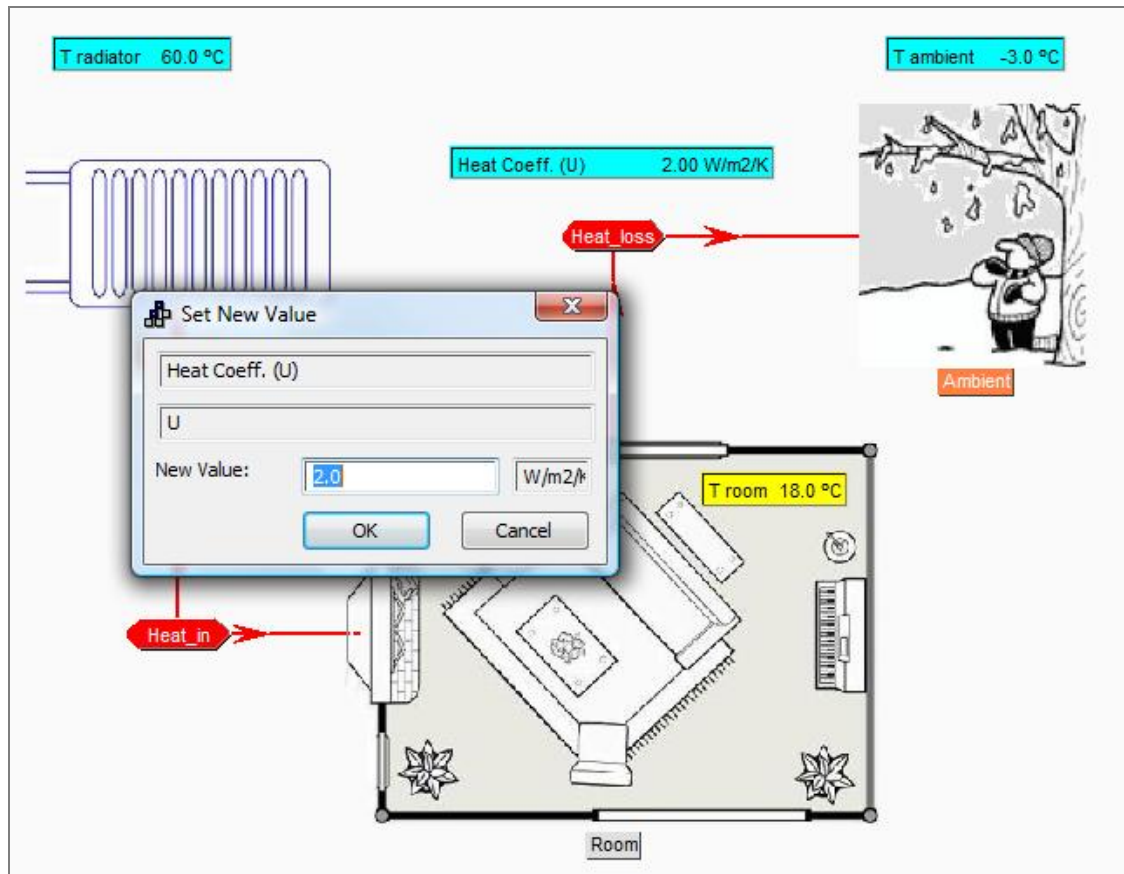
Sequence example:

- I.  $T_{\text{Radiator}} \rightarrow 80 \text{ }^\circ\text{C}$        $U_{\text{Heat\_loss}} = 2 \text{ W/m}^2/\text{K}$        $T_{\text{Ambient}} = -3 \text{ }^\circ\text{C}$
- II.  $T_{\text{Radiator}} = 80 \text{ }^\circ\text{C}$        $U_{\text{Heat\_loss}} \rightarrow 20 \text{ W/m}^2/\text{K}$        $T_{\text{Ambient}} = -3 \text{ }^\circ\text{C}$
- III.  $T_{\text{Radiator}} = 80 \text{ }^\circ\text{C}$        $U_{\text{Heat\_loss}} = 20 \text{ W/m}^2/\text{K}$        $T_{\text{Ambient}} \rightarrow 10 \text{ }^\circ\text{C}$



9. Tuning parameters during a Simulation run of the room model.

Another option to tune parameters would be by inserting *Display Objects* which are accessible in **Simulation Environment** (by selecting the option *Allow Value Access in Simulation Environment*). The user will have just to click on the object in **Simulation Environment** to change its value (see figure 10).



10. Display Objects accessible in a Simulation run.

The results being plotted can be exported as a *Comma Separated Value (.csv)* file just by clicking on the sub-tab "Extra" on the option "Export Time Table" of the **Plotting** tab.

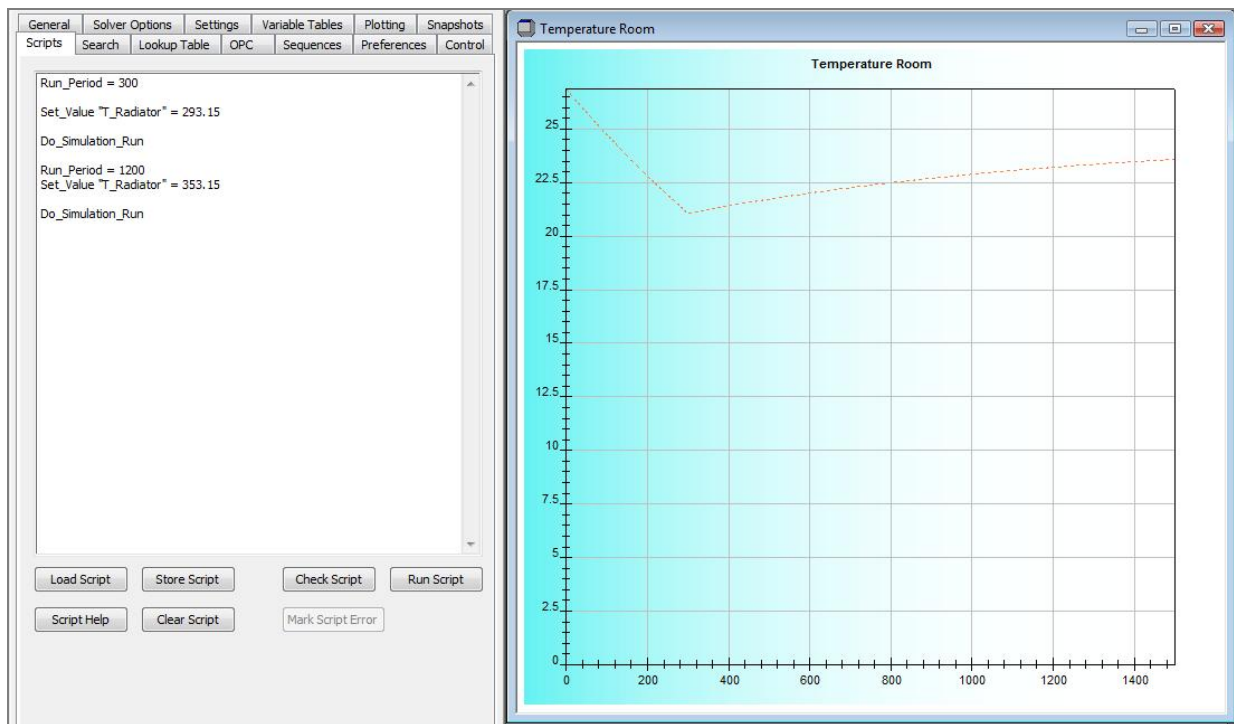
### 2.6.3 Configuring a Simulation Run: Snapshots, Scripts and Sequences

Within Simulation Environment the user can also load and store so called **Snapshots**. For instance, a certain state of the simulation can be stored and loaded in other simulation run. These Snapshots can be added and set while the simulation is running. These options can be found in the Snapshots tab.

The tab **Scripts** allows the user to perform a simulation run by introducing statements, i.e. *actions*, *time terms* and *conditions*. As a very simple example of a script for our *Room* model we could think

of just running a simulation in which the temperature of the radiator will be 20 °C during the first 5 minutes of the simulation run. Then, it will change to 80 °C during the next 15 minutes. After those 20 minutes the simulation will stop.

The script for describing such a simulation scheme as well as the response of temperature of the room will be the following:



11. Inserting a script in a Simulation run.

A list of the available functions that can be called from a script can be found in *Simulation Scripts* document by pressing the "Script Help" button.

Another handy option for the user to configure the simulation run can be found in the tab **Sequences**. Here the user will be able to include a sequence of events which will describe the automatic settings of the system (i.e. opening and closing valves in a batch sequence, switching to an emergency mode based on an initiators scheme, etc). The sequence consists of certain number of *steps* which are defined by a *statement* (action to be performed) and a *condition* which defines the end of the action and the beginning of the next step.

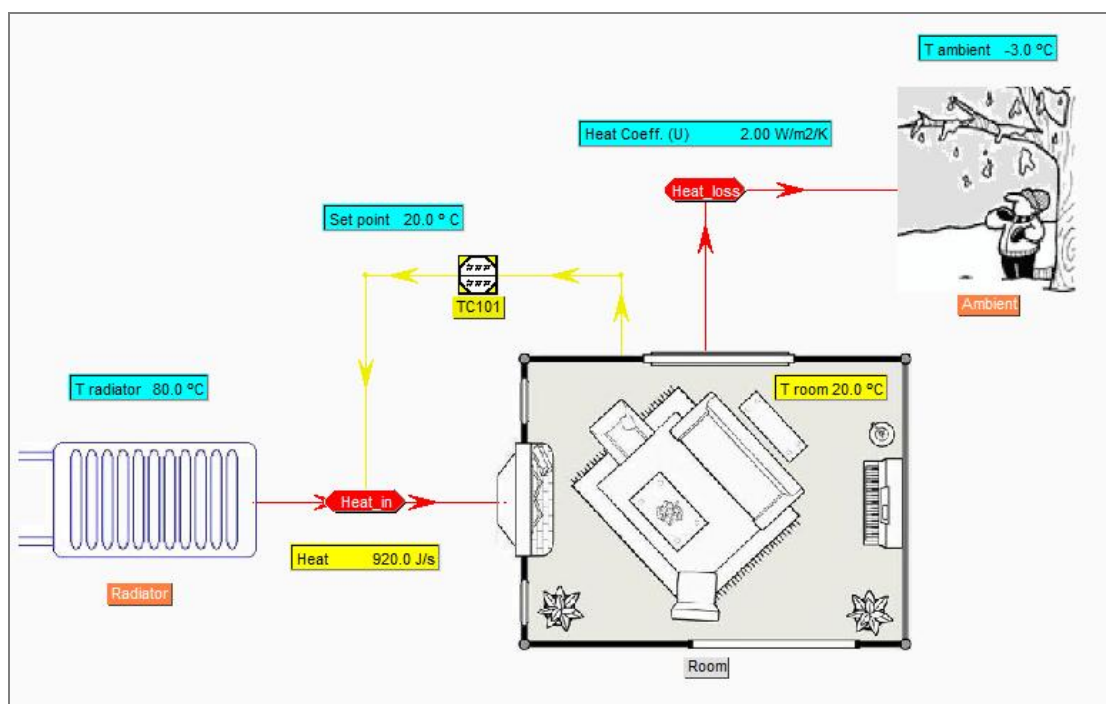
## 2.6.4 Control Settings

In order to introduce a simple control scheme in our *Room* model, we will include a temperature controller in our model. Based on our model scheme, we could think of the controlling the temperature of the room by varying the heat flux coming from the radiator.

First, a variable describing the “opening/closing” of the heating system has to be included in the equation. Therefore, in **Modelling Environment**, click on **Editor** and create the equation  $Q = Z \cdot U \cdot A \cdot (T_{\text{room}} - T_{\text{amb}})$ , where “Z” will be the manipulated variable and will vary from 0 (close) to 1 (maximum opening). The variable will appear as *Selectable* within the *Equation Sorting* tab, so the user will have to click on it.

The next step is to include the control setting in the physical topology of our model. Although the user has the option to define from scratch a control scheme, we will insert a controller available in the Library (*Insert New Object* → *From Library*). Select the “Temperature PI Controller” and locate it in the physical topology of our *Room* model. By clicking on the controller the user will see all the equations defined within the object and also will have the option to configure the controller (i.e. Gain controller, Integral Time Constant, etc.).

Then the controller has to be connected to the model. Therefore, we will connect first the *Room* with the controller ( $T_{\text{room}}$  is the measured variable) and secondly with the connection *Heat\_in* (Z is the manipulated variable). As shown in figure 12, two *Display Objects* were included in the physical topology: the *Set point* of the controller (accessible in Simulation Environment) and the heat flux of the connection *Heat\_in* (to see the control action).



12. Temperature control in the *Room* model.

Once the model is compiled, the user will be able to see how the temperature of the room responds to disturbances (i.e. ambient temperature) or to a change in the set point for a certain control configuration.

### 2.7 Summary

After this first two chapters of the beginning user tutorial, the reader should be able to transform the system to model into a *Physical Topology* (i.e. objects and connections); insert *equations*; initialize *variables*; include a *monitoring scheme*; add a *simple control scheme*; and perform a *simulation run* (plotting, extract results, scripts, etc).

In the next chapter the following features in Mobatec Modeller will be introduced:

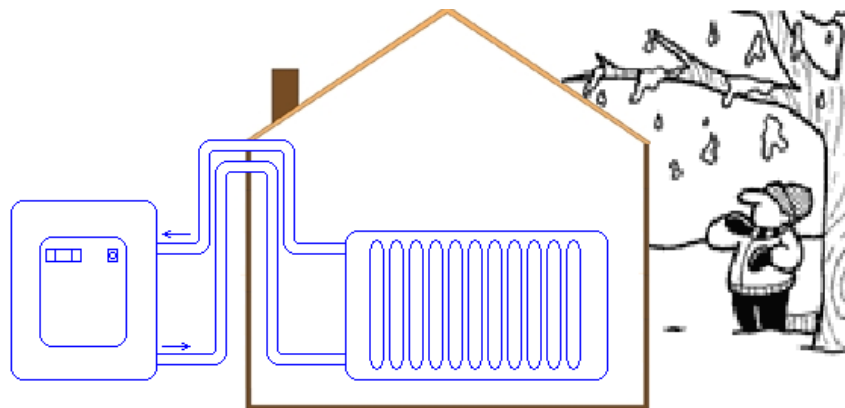
- Species Definition
- Hierarchical levels (groups)
- Mass Connections
- Information Connections

Moreover, some other features already introduced in the previous chapters (i.e. graphical user interface, systems definition, etc) will be further extended.

## 3 Introducing Species

---

We will now extend our model by refining the heating unit. The heating element with adjustable temperature is going to be replaced by a unit in which circulating heated water produces the required heat to warm up the room. Water flows from a reservoir to the heat exchanging element, from which it flows back again to the reservoir. The heat capacity of metal will also be taken into account now. The water in the heating element heats up the metal, which heats up the room. The metal of the reservoir is heated by a heating element from which a certain heat flux is outgoing.



13. Extended heating unit example. A water flow is pumped around from a reservoir.

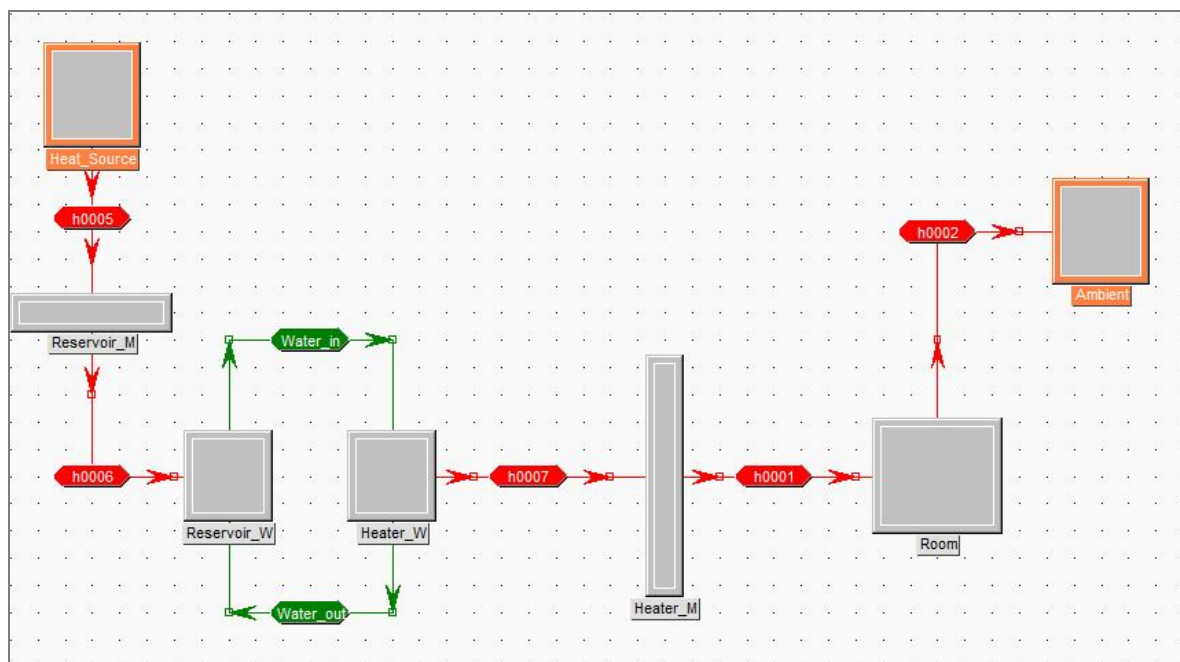
Although the extended model is still quite simple, we will run into several features of **Mobatec Modeller** that will be explained in this chapter.

### 3.1 New Physical Topology

A demonstration of some of the steps that are outlined in this paragraph can be viewed by clicking on the following link:

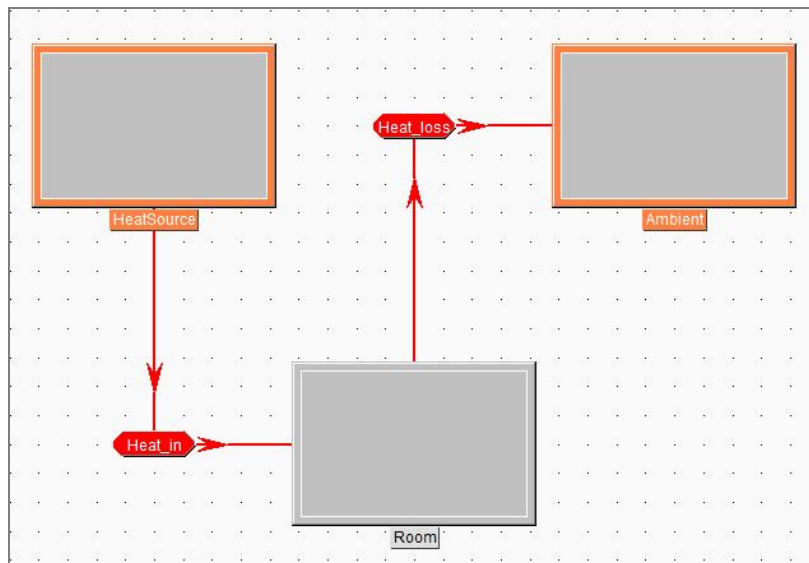
[Room 4 – Grouping and Reconnecting](#)

A simple, “flat” physical topology for this process could look as follows in **Mobatec Modeller**:



14. “Flat” topology of the extended heating unit example.

We want to use this example to introduce **Group** systems, with which the user can group several systems to increase the survey ability of the model. These group systems are especially handy, and even indispensable, when larger and more complex models are being built. Therefore, we will build the new physical topology based on our simple *Room* model consisting of only three systems (*Heat Source*, *Room* and *Ambient*) and two connections (*Heat\_in* and *Heat\_loss*) as shown in figure 15.




15. Starting Physical Topology for extending the heating unit example.

### 3.1.1 Group Systems

There are several ways to introduce group systems into your physical topology. You can select several systems and group these together or you can refine a system by placing systems "in" it. The refined system will be automatically transformed into a group system. Both possibilities will be discussed in this section.

Select the *HeatSource* system and choose [Edit][Group] from the menu (or right mouse click and "Group" or simply press "g" when the pointer is on the object). In the **Group Selected Systems** dialog that appears in the **Properties** dialog select the **Group Name** field and enter "Heating". Press the **Group Systems** button to confirm the grouping.

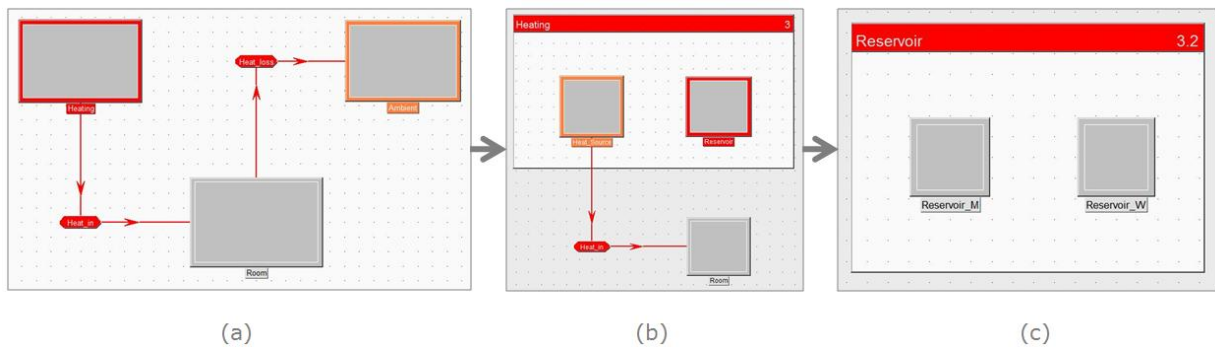
The *HeatSource* system will disappear and a new (group) system *Heating* will now appear in your model window. Double click on this system to zoom in on it and see the contents. The *Room* system will be drawn at a default position outside the *Heating* group system, which only consists of the *HeatSource* system for now. Drag the *Room* system to a preferred position and notice that it cannot be dragged into the *Heating* system, since it is not a part of *Heating*.

As mentioned before by double clicking on an object the user will zoom in and by clicking on "Hierarchy Level Up" [  ] we will zoom out.

Select [Edit][Insert New Systems] and insert a system called *Reservoir*. While still in the **Insert New Systems** mode, click on this new *Reservoir* system. An automatic zoom-in will occur. In the *Reservoir* system insert two more systems: "M" (representing the metal) and "W" (representing the water hold-up).

Quit the **Insert New Systems** mode and select the *Reservoir* system (by clicking on the blue bar). On the **Physical Topology** tab of the **Properties** dialog check the **Include Name in Sub Objects**

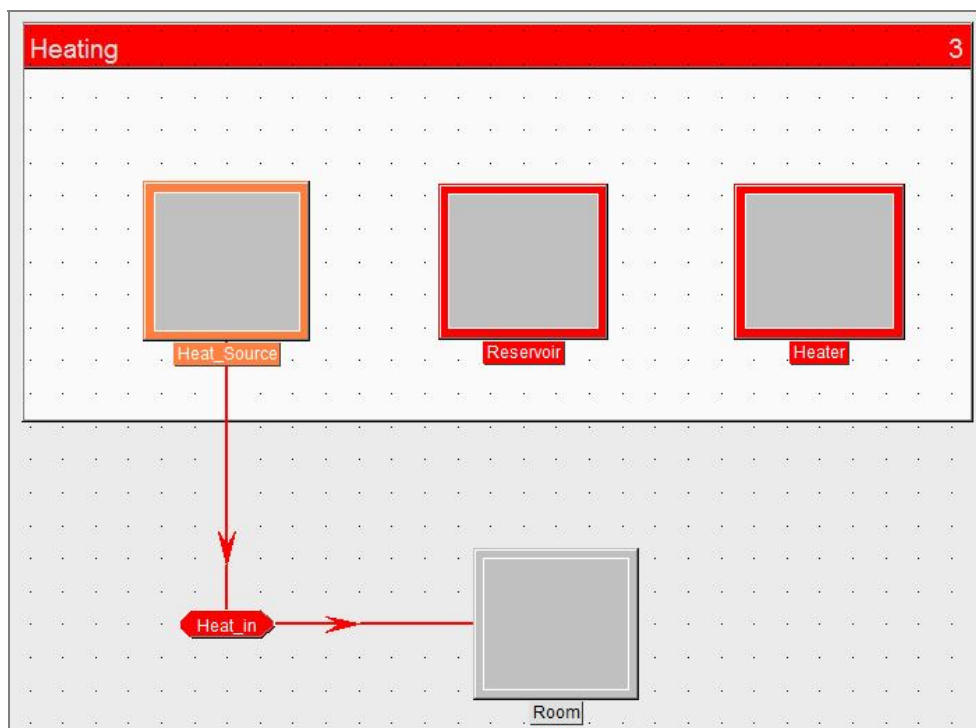
checkbox and notice that the systems "M" and "W" are directly prefixed with the name of the currently selected system (see figure 16).



16. Zoom in on the systems of the new physical topology.

This feature can be used to copies of the same structures (with the same internal names) globally unique names.


Zoom out from the *Reservoir* system (*Heating* should be the focused system). Copy the *Reservoir* system, paste it and rename the copy to *Heater*. The systems *Heater\_W* and *Heater\_M* should be displayed when zooming in on system *Heater*.



17. Zoom in on the systems of the new heating system.

### 3.1.2 Reconnecting a Connection

Now all the required connections will be inserted. First, we will reconnect the heat connection that connects the *HeatSource* with the *Room*, since the room will now be heated via the metal of the heating element.

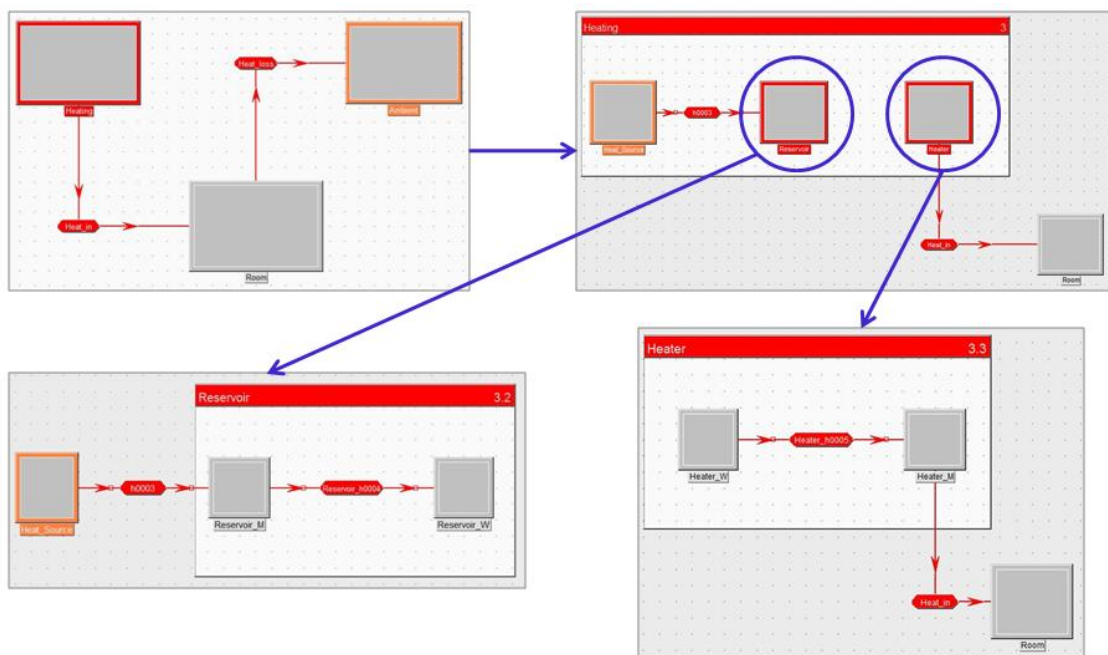
Set *Heating* as the focussed system and select the heat connection with the right mouse button. From the popup menu that appears select the **Reconnect Origin** item. You can now select a new origin for the heat connection. Every group system and every system that can serve as origin will light-up when you hover over it with the mouse. A single mouse click on a group system will invoke a zoom-in. A single mouse click on the active (blue) bar [  ] of the focussed system will invoke a zoom-out. Clicking anywhere in the model window will change the connection into a so-called **open connection**, with an undefined origin. Clicking on a capacity will reconnect the origin of the connection to the capacity.

Zoom in to the *Heater* system and reconnect the connection to *Heater\_M*.

Also install the following new heat connections:

- *HeatSource* → *Reservoir\_M*
- *Reservoir\_M* → *Reservoir\_W*
- *Heater\_W* → *Heater\_M*

The physical topology should look as follows:



18. Zoom in on the systems with the new neat connections.

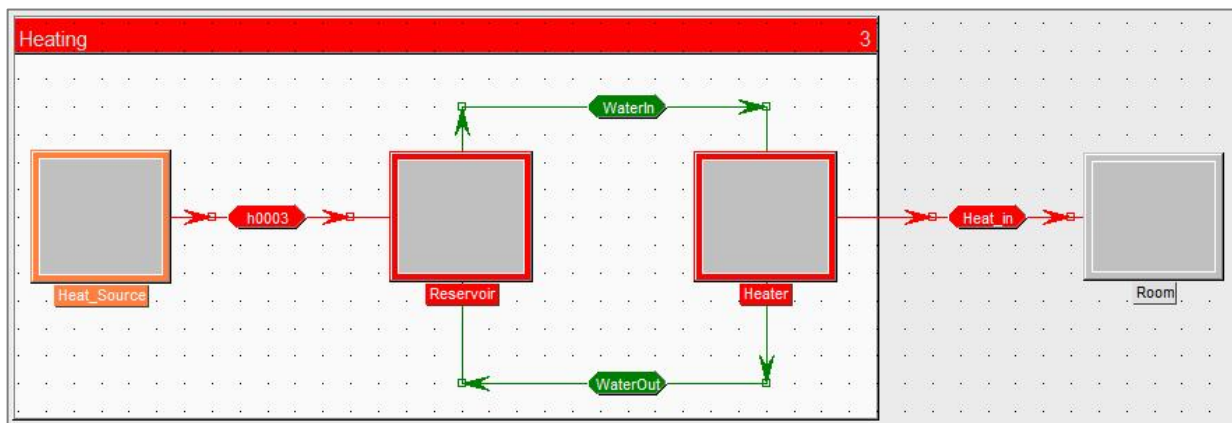
### 3.1.3 Composite or Vector Connections

To model the water that flows from the *Reservoir* to the *Heater* and back, we need to insert two mass connections. In this case, we will not consider **bi-directional** ( **elaborate more the mass connections type, what that means...**) flow through a connection and therefore we select the **Connection Is Uni-Directional** checkbox, when inserting these new connections (the directionality of mass connections can always be changed at a later time on the **Species Topology** tab or the **Equation Topology** tab).

Insert a uni-directional mass connection between *Reservoir\_W* and *Heater\_W* and also one from *Heater\_W* to *Reservoir\_W*. Notice that, when you insert the second connection, both connections will be packed together in a **“composite”** or **“vector”** connection. Composite connections are black and have an **“X”** in the active point of the connection object.

If you want both connections to be shown on the screen, select the connection and press the **Expand** button in the **Properties** dialog (alternatively, press **“x”** on your keyboard). To collapse the connections back into one vector connection, select one of the connections and press the **Collapse** button (or press **“c”** on your keyboard).

Expand the connections between *Reservoir\_W* and *Heater\_W* and name them *WaterIn* and *WaterOut* respectively. Your physical topology should now look something like this:



19. Physical Topology with Heating as focussed system.

## 3.2 Species Distribution

A demonstration of the steps that are outlined in this paragraph can be viewed by clicking on the following link:

[Room 5 - Species Insertion and Distribution](#)

To model the water that is being pumped around from the *Reservoir* to the *Heater* and back, we need to inject (i.e. insert) water in one of the systems that should hold water.

Select the *Reservoir\_W* system and select the **Species Topology** tab in the **Properties** dialog. Press the **Define Plant Species** button to define which species *can* be present in our “plant”. From the database that appears in the **Properties** dialog select the species “H2O – Water” and add this to the plant species set. Press **OK** (or **Apply**) to confirm your choice. Next, press the **Set Injected Species** button and add Water to the injected species of this system. Press **OK** (or **Apply**) to confirm.

The **Species in System** list box lists the available species for the selected system. The species that are present in the system are displayed with their formula, followed by their name (e.g. H2O - Water). When a species is injected by the user in the selected system, the notation is prefixed with three stars (for example: \*\*\* H2O - Water).

Species that are not prefixed with \*\*\* have propagated to the selected system through mass connections or phase transition connections or are produced via reactions that are defined within the system.

The species distribution is automatically calculated by **Mobatec Modeller** and defines which species can be present in which parts of the process. This enables the generation of the relevant balance equations of fundamental extensive quantities for the systems.

Select the system *Heater\_W* and notice that the species *Water* is also present here. It was distributed via the mass connection *WaterIn*. Select the *WaterIn* connection, focus on the **Species Topology** tab and notice that **Species Through Connection** list box contains the species Water.

The **Search** functionality of **Mobatec Modeller** can be used to find a species or sets of species. Press CTRL+F (or press the **Search** tab in the **Properties** dialog) and select [Species and reactions] [Species (Mass Domains)] and in the **Sub Search** select “H2O – Water”. All objects containing Water will light up and the **Result** list will show all objects (in this case sorted in domains) that contain water. Expand the Water domain by pressing the “+” and click on any object to zoom in on it.

### 3.2.1 Creating New Species and Adding Species Parameters

Sometimes the user is interested in defining a species that is not in the database. We just have to select the **Species** tab of the **Database Editor**, click on *New Species* and write the name, the code name and the formula.

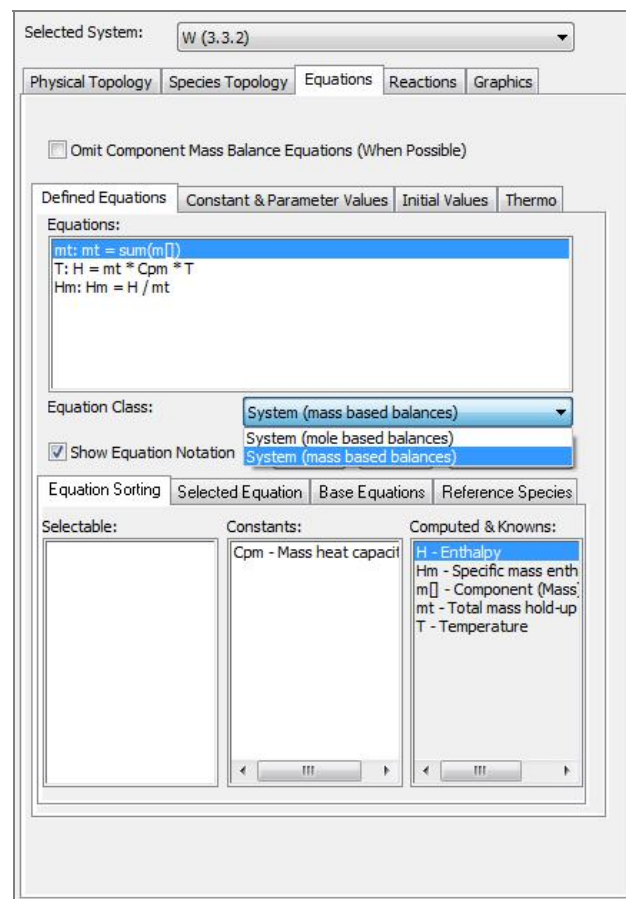
From the *Defined Variables* list, select the desired parameter and click on *Add Species Parameter*. To insert a species parameter in an equation, the function *spec* has to precede the name of the parameter. For instance, if the molecular weight (MW) has been defined as a species parameter the user will write *spec.MW* when editing an equation.

### 3.3 Defining Heat and Mass Connections of the New Physical Topology

Now the equations of the new heating system, consisting of the heat source and the loop of water going from the reservoir to the heater and back to the reservoir, will be defined.

#### 3.3.1 Equation Class: mass or mole based balances system

Since the loop consists of a water recirculation, defining this part of the model as *mass based* balances rather than *mole based* seems to be more convenient. The user can select the option that adapts better to the model by clicking on "Equation Class" which can be found in [Equations] [Defined Equations] in the **Properties Dialog** of a system (i.e. *Reservoir\_W* and *Heater\_W*).



20. Selecting the Equation Class: mass or mole based balances.

Note that the variables of both the systems and the mass connections change from mole based units (i.e. " $n[]$  - Component Molar Holdup" in the capacity systems and " $Fn[]$  - Component Molar Flow" in the connections) to mass based units (i.e. " $m[]$  - Component Mass Holdup" in the capacity systems and " $Fm[]$  - Component Mass Flow" in the connections).

### 3.3.2 Flexibility to define Mass Connections

To show the user the high flexibility of **Mobatec Modeller**, we will model those two mass connections differently, although the result will be the same.

After choosing mass based balances as **Equation Class**, the two selectable variables that the user will see in the two mass connections are  $Fm[]$  - *Component (Mass) Flow* and  $HF$  - *Enthalpy Flow*.

In the next table the equations inserted in both mass connections are presented. As mentioned before, modelling them differently aims at showing the high flexibility and modelling freedom that **Mobatec Modeller** offers.

Table 1 – Equations of the water loop: flexibility to define Mass Connections.

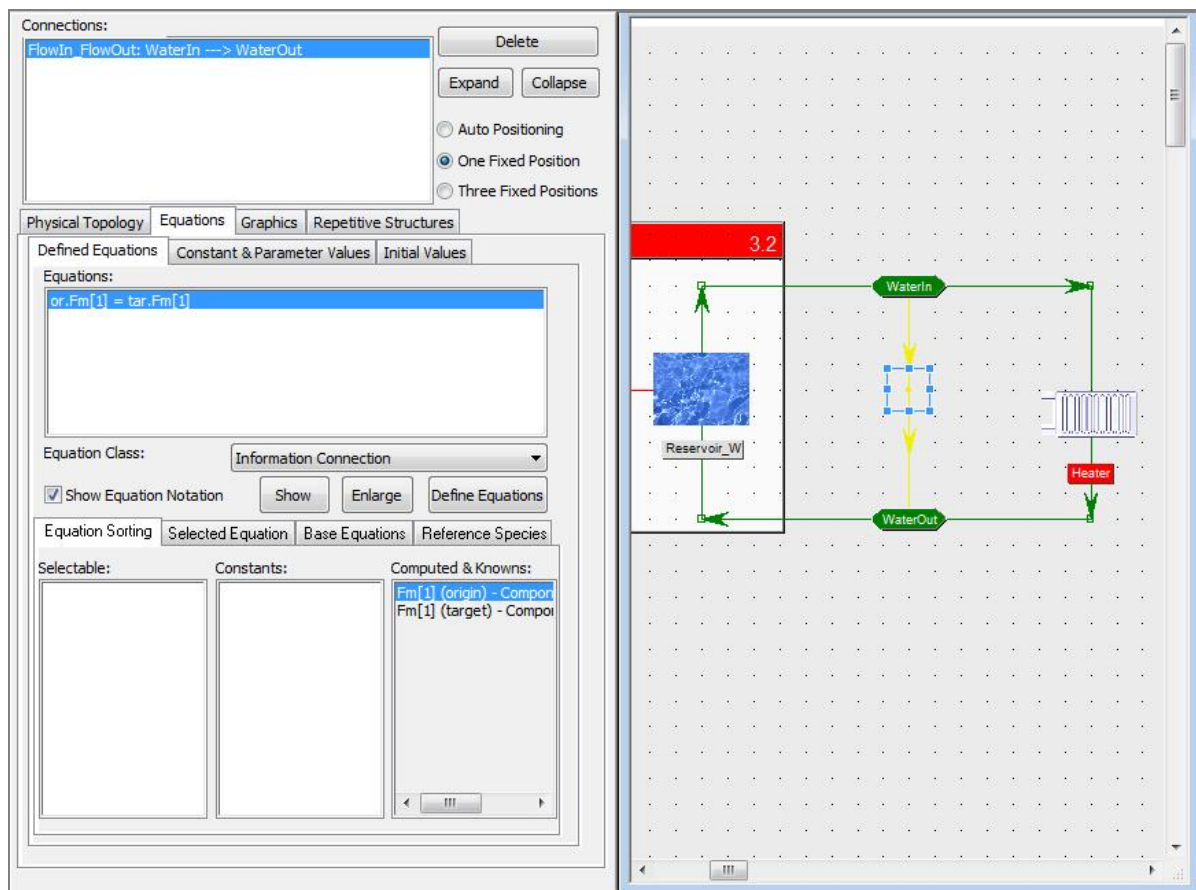
<b>Mass connection 1: <i>WaterIn</i></b>		
<b>Name</b>	<b>Equation</b>	<b>Comments</b>
Enthalpy_Flow	<pre>IF Fm[1] &gt; 0 THEN   HF = or.Hm * Fm[1] ELSE   HF = 0 END</pre>	Since there is only water, there will be enthalpy flow if there is water flow. Notice that the specific enthalpy defined in the equation has mass based units. So the variable $Hm$ (J/kg) was created. The user will have to connect the reference species[1] to a species existing in the system (in this case <i>Water</i> )
<b>Mass connection 2: <i>WaterOut</i></b>		
Total_Mass_Flow	$Fmt = \text{sum}(Fm[])$	Total mass flow is the summation of the component mass flows.
Enthalpy_Flow	<pre>IF Fmt &gt; 0 THEN   HF = or.Hm * Fmt ELSE   HF = 0 END</pre>	There will be enthalpy flow as long as there is mass flow.

After inserting these equations in the mass connections, we will have first to set the component mass flow as a constant in *WaterIn* (a reasonable value for the water flow in this kind of systems is 0.020 kg/s); and due to the fact that is a close-loop and there is only one species, the value of the water flow in one of the mass connections will be sent automatically to the other one via an *Information Connection*; then we will identify the *Reference Species* (in this case [1] → [Water]).

### 3.3.3 Information Connections

As mentioned before, the flow of water in the recirculation has been defined as a parameter. Therefore, in order to have a consistent loop, the value that the user defines in the connection from *Reservoir\_W* to *Heater\_W* should be the same that the one inserted in the connection from *Heater\_W* to *Reservoir\_W*. By introducing an *Information Connection* the user will have to insert the water flow in one connection and automatically the value will be sent to the other one.

In the **General** tab click on *Insert New Connections* (or just press "m") and select *Information* as the *Connection Type*. The origin of that connection will be the connection *WaterIn* and the target *WaterOut*. Once the two mass connections are connected by a yellow connection, the equation that sends the value of the water flow from the origin to the target has to be created. In the **Equations** tab of the **Database Editor** create a new equation whose class is *Information*, and its function is  $or.Fm[1] = tar.Fm[1]$ .

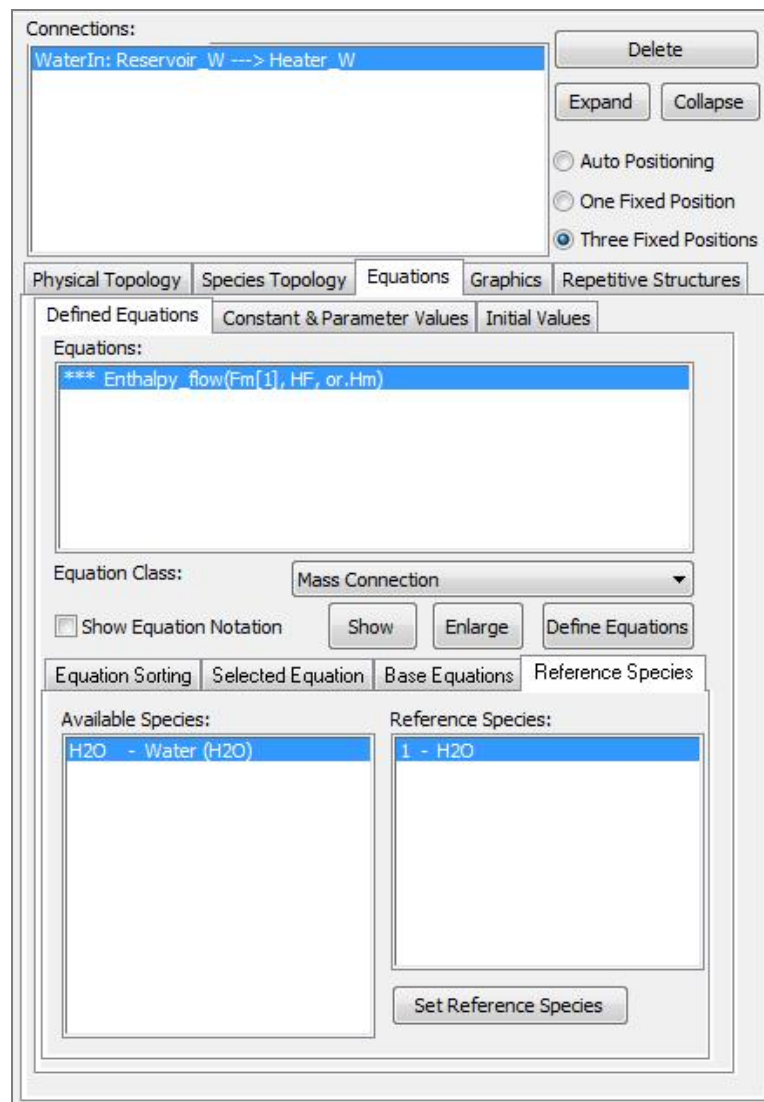


21. Inserting an *Information Connection*.

The next step is to add that equation to the information connection. Once the equation is inserted the user will see in the *Equation Sorting* of the *WaterOut* connection that all the variables are *Computed & Known*.

### 3.3.4 Reference Species

A reference species was inserted in the equation defined in the mass connection *WaterIn*. That reference species will have to be connected to an available species existing in the system. Click on the mass connection *WaterIn* and in the **Equations** tab select [Defined Equations] [Reference Species]. By double-clicking "H2O - Water (H2O)" we will set the connection between the reference and the species.



22. Reference Species in an equation.

### 3.3.5 Heat Connection of the New Physical Topology

The heat connections between *Reservoir\_M* – *Reservoir\_W* and *Heater\_W* – *Heater\_M* will be modelled as before (i.e. the heat transfer is assumed to be linearly driven by a temperature difference  $Q=U*A*(or.T - tar.T)$ ).

The heat connection between the *HeatSource* and *Reservoir\_M* has been modeled as  $Q = Z * Q_{max}$ , where the maximum heat flux is constant and the “Z” will be the manipulated variable in the control scheme.

The values assumed for the parameters of the heat connections are the following:

Table 2 – Parameters inserted in the Heat Connections of the heating system.

Heat connection	U (W/m <sup>2</sup> /K)	A (m <sup>2</sup> )
<i>Reservoir_M</i> – <i>Reservoir_W</i>	200	0.4
<i>Heater_W</i> – <i>Heater_M</i>	200	4
<i>Heater_M</i> – <i>Room</i>	5	4

For the parameter  $Q_{max}$  of the heat connection between *HeatSource* and *Reservoir\_M* a value of 3000 J/s was inserted.

### 3.4 Systems Definition

By clicking on the systems representing the walls (*Reservoir\_M* and *Heater\_M*) the user can see that based on the inserted heat connections, the temperature in those systems needs to be defined. The walls will be modelled in the same way we modelled the room (i.e. enthalpy equation for a capacity system). However, since now our system is a piece of metal, defining the enthalpy of the system in mass basis seems to be more convenient. Therefore, the equation will be  $H=mt * C_{pm} * T$ , where the variables “*mt* - Total Mass Holdup” and “*C<sub>pm</sub>* - Mass Heat Capacity” have to be created.

For instance, we can assume that the radiator has a mass of 45 kg, the reservoir is a 30 kg metal tank and both are made out of stainless steel ( $C_p = 800$  J/kg/K).

The next step will be to model the water systems, i.e. *Reservoir\_W* and *Heater\_W*. The user can see that there are two *Selectable* variables in those systems (Specific enthalpy and temperature).

The equations defined in the water holdup system are presented in the next table.

Table 3 – Equation defined in the water holdup systems.

Name	Equation	Comments
Specific_enthalpy	$Hm=H/nt$	The specific enthalpy is the enthalpy of the system divided by the total mass hold-up. The variable $mt$ (kg) was created.
Total_mass_hold_up	$mt=\text{sum}(m[.])$	The total mass hold up is the summation of the component mass hold up.
Enthalpy	$H=mt*Cpm*T$	The enthalpy of the system is defined as the total mass hold-up multiplied by the heat capacity and the temperature.

Once all those equations are defined and inserted in the water systems, the heat capacity will be selected as a constant ( $C_p = 4186 \text{ J/kg/K}$ ), and there will be a consistency between variables and equations.

### 3.5 Initializing the New Physical Topology

As mentioned before, by clicking on the **Search** tab the user can see the uninitialized variables and the improper equations (among other options). If we press on *Uninitialized Variables* several systems will light up.

First, an initial value for the temperature in the *Reservoir\_M* system has to be included (for instance,  $T_{\text{initial}} = 70 \text{ }^\circ\text{C}$ ) and the cell  should be checked for calculating the initial value of the enthalpy of the system.

For the water reservoir system (*Reservoir\_W*) once the water heat capacity value has been inserted, the user needs to initialize the variables. We will set as initial values the temperature ( $T_{\text{initial}} = 50 \text{ }^\circ\text{C}$ ) and a water hold-up of 50 kg. We will check the cell  of these variables for the initial calculation of the rest of the variables.

The two mass connections simulating the recirculation of water in the heating system also have to be initialized. As mentioned before, a constant value of 0.02 kg/s will be assumed.

Now we will zoom in on the *Heater* group. The *Heater\_W* system will be initialized in the same way the water reservoir was initialized. For instance, the initial temperature will be also 45  $^\circ\text{C}$  and the volume in the heater will be 6 liters (cell  checked as well).


In the system representing the metal of the heater an initial value for the temperature needs to be included. For instance a value of 30  $^\circ\text{C}$  could be inserted (also cell  checked).

And finally we will do the same for the Room system ( for instance,  $T_{\text{initial}} = 20 \text{ }^\circ\text{C}$ ).

### 3.5.1 Import State From Current Simulation

As mentioned in section 2.6.2 *Configuring a Simulation Run: Snapshots, Scripts and Sequences*, a certain state of a simulation can be saved and loaded.

Very often the user is interested in setting a certain simulation state as initial state for the next simulation run (for instance the steady state for certain conditions). There is a very handy feature in **Mobatec Modeller** that allows the user to import a current simulation state.

When working in **Simulation Environment**, the user has just to stop the simulation run in the desired state, switch back to **Modelling Environment** [] and select the **Model Generation** tab of the **General Property Dialog**. By pressing on "Import State From Current Simulation" within the **Mobatec Solver** sub-tab, the simulation state will be set as initial state for the next simulation run.

## 3.6 Graphical User Interface

Besides adding *Icons* to all the objects, both *Display Objects* and *Filled rectangles* will be inserted in our model. The new Room model will include two thermometers and a temperature control which the user will have the option to access to in **Simulation Environment**.

### 3.6.1 Display Objects

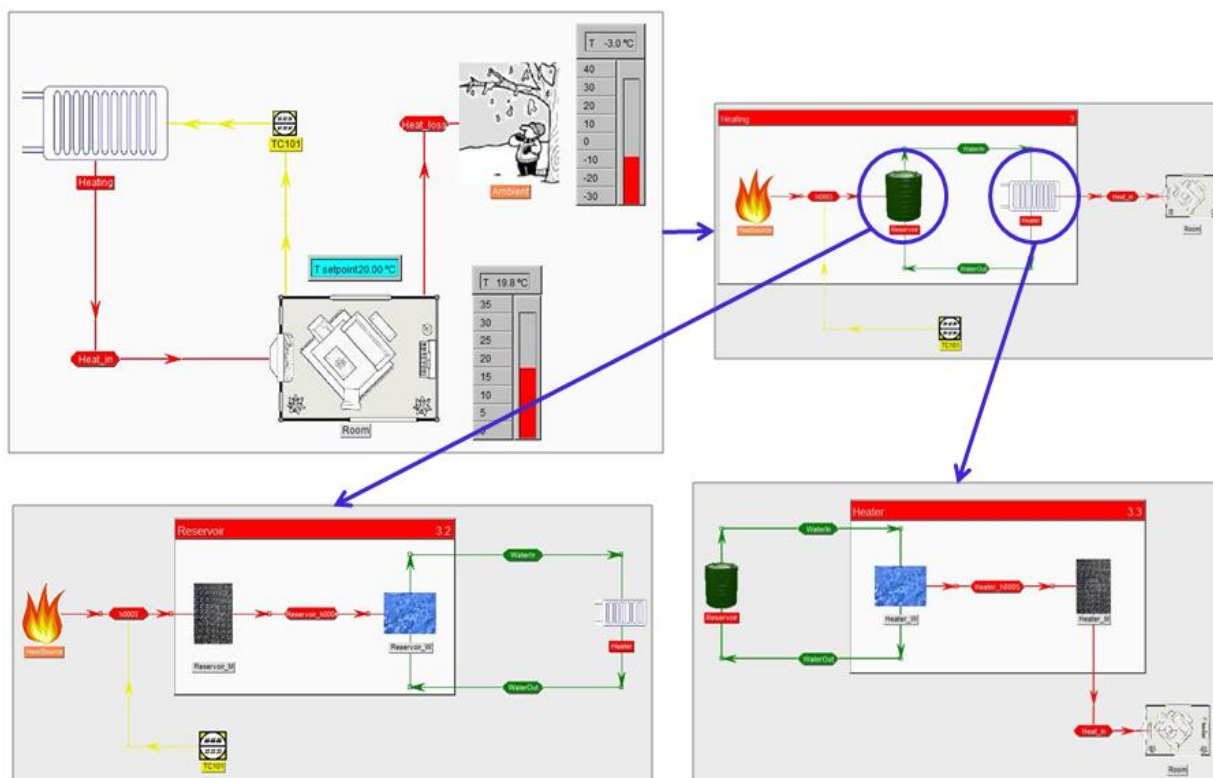
By inserting *Text Objects* and *Display Objects* the user has high flexibility and freedom to include in the Physical Topology of the model an attractive monitoring/display scheme. For instance, in our *Room* model, two mercury-in-glass thermometers (one linked to the *Room* system and the other one to the *Ambient* system) were inserted which are monitoring numerically and graphically the temperature in those systems (see figure 23).

### 3.6.2 Control

Moreover, a display object where the user can define the desired room temperature was inserted in the room (set point). Select [Insert New Objects (n)] → [Value Display Object] and introduce the settings for this object (i.e. "Setpoint" as *Simulation Variable*, "TC101" as *Linked Object*, *Allow Value Access in Simulation Environment*, etc). The temperature controller connects the measured variable (temperature of the room) and the manipulated variable (*Heat* from the *HeatSource*). Therefore, a variable "Z" will be included in the heat transfer equation defined in the heat connection going from the *HeatSource* to the *Reservoir\_M*.

As mentioned before the controller can be configured by tuning the gain and the time constant. The set point of the controller is displayed and accessible in **Simulation Environment**. The user will have the option to change it and to see how the temperature of the room responds for a certain control configuration.

In figure 23 the physical topology of our model with the icons and the objects is shown.



23. Inserting Species parameters in the Database Editor.

### 3.7 Summary

So far, several modelling and simulation features of **Mobatec Modeller** have been pointed out. At this point we are able to break down our system into *objects* (i.e. capacity or battery limits) and *connections* (heat, mass, information), insert *equations*, initialize *variables*, inject *species*, add a *control scheme*, work on the *graphical user interface* (i.e. monitoring, display objects, icons, etc) and perform a *simulation run* (plotting, extract results, scripts, etc)

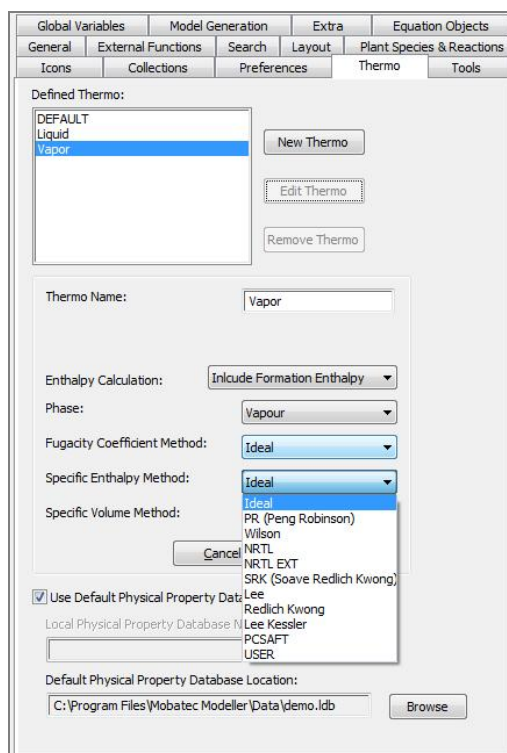
In the last chapter of this tutorial the user will learn basically how to insert chemical reactions and also how to work with the **Physical Property Database** when describing a system by a certain thermodynamic method is desired.

## 4 Introducing Reactions

In this new chapter the user will learn how to insert chemical reactions in the model and how to work with the **Physical Property Database**. The current *Room* model will be extended by including the modelling of the heat source, where a gas combustion reaction will be introduced. It was assumed that the system consists of a methane and an air inlets to a system where the combustion reaction takes place. The heat of reaction will be the heat source which will warm up the piece of metal of the reservoir.

### 4.1 Physical Property Database

**Mobatec Modeller** includes several thermodynamic property methods within the **Physical Property Database** which the user has the option to select among. In the **General Properties Dialog** select the tab called *Thermo*. The user can choose and define the thermodynamic property method (i.e. PR, Wilson, NRTL, SRK, Redlich Kwong, etc.) for the different phases existing in the model.



24. Physical Property Database.

The user can set either to *Use Default Physical Property Database* or to call a *Local Physical Property Database*.

The **Physical Property Database** of **Mobatec Modeller** can be accessed via the following calls (see *Operators and Functions in Mobatec* document in the *Help* tab):

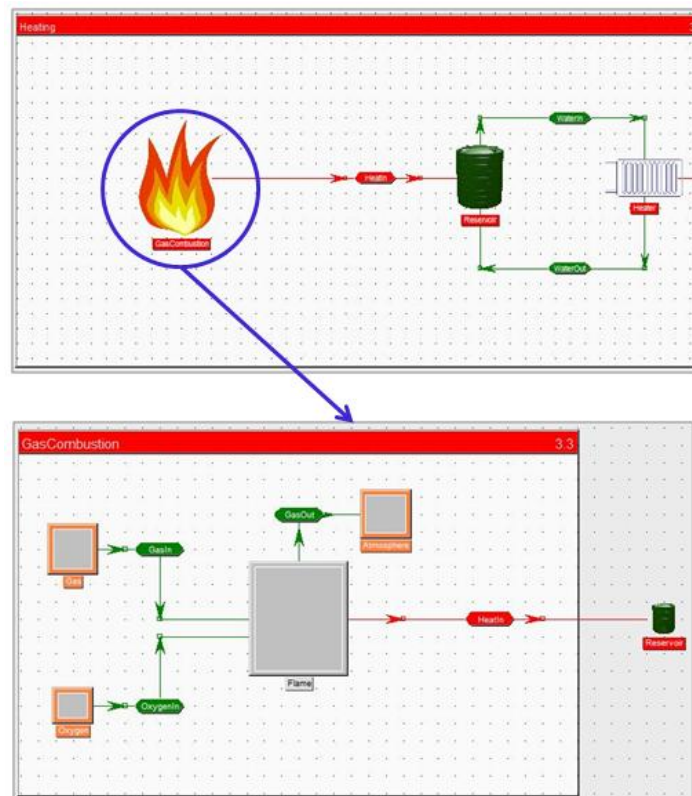
- @THERMO.ENTH( $P, T, Xn[]$ ) For the specific Enthalpy (in J/mol)
- @THERMO.ENTR( $P, T, Xn[]$ ) For the specific Entropy (in J/K/mol)
- @THERMO.FUGC( $P, T, Xn[]$ ) For the Fugacity Coefficients
- @THERMO.SPECV( $P, T, Xn[]$ ) For the specific Volume (in m<sup>3</sup>/mol)

The variables  $P$ ,  $T$  and  $Xn[]$  should represent the Pressure, Temperature and Component Fractions respectively.

In this new version of our *Room* model the specific enthalpy will be calculated via the **Physical Property Database**.

## 4.2 New Topology

Based on the description of the new heat source, the new physical topology of our heating system should look as follows:



25. New topology of the heating system.

As shown in the figure above, three connections (mass connections) and four systems (three battery limits and one capacity) have been added to the model.

#### 4.2.1 Connections and Systems Definition

Since now the **Physical Property Database** will be used for the specific enthalpy, the call function has to be inserted in the equation. All the equations defined in the new connections and systems are the following:

- **Three mass connections** → methane inlet, air inlet and gas outlet to the environment. The gas flow will be pressure difference driven. The equations to include are the followings:

$$\text{Total flow:} \quad Fnt = Z * KV * (or.P - tar.P) \quad \text{where the valve position (Z) was included in the equation}$$

$$\text{Component flow:} \quad Fn[] = or.Xn[] * Fnt$$

$$\begin{aligned} \text{Enthalpy flow:} \quad & \text{IF } Fnt > 0 \text{ THEN} \\ & HF = or.Hn * Fnt \\ & \text{ELSE} \\ & HF = 0 \\ & \text{END} \end{aligned}$$

- **Three Battery Limits** → representing the methane source, the air source and the atmosphere (where the products of the chemical reaction will end up). For the "Inlet" battery limits the specific enthalpy was defined as  $Hn=@THEMO.ENTH(P, T, Xn[])$ , and the temperature, pressure and molar fraction were set as parameters. For the "Atmosphere" only the pressure needs to be set as parameter.

- **One capacity system** → representing the gas system where the chemical reaction takes place (i.e. "Flame"). Since the system was defined as an ideal gas capacity system, the equations to define are the followings:

$$\text{Total molar hold-up:} \quad nt = \text{sum}(n[])$$

$$\text{Comp. Molar hold-up:} \quad n[] = Xn[] * nt$$

$$\text{Enthalpy:} \quad Hn = H / nt$$

$$\text{Specific enthalpy:} \quad Hn=@THEMO.ENTH(P, T, Xn[])$$

$$\text{Ideal gas law:} \quad P * V = nt * glob.R * T$$

The volume was set as a parameter.

## 4.2.2 Injecting Species

Based on our chemical reaction scheme (methane combustion), the species that need to be inserted in the system are methane, oxygen, nitrogen, carbon dioxide and water.

As explained in the previous chapter, for injecting species, select a system, click on [Species Topology] [Define Plant Species] and add the compounds (please note that the filter is a handy tool for searching).

Once the plant species have been defined we will inject the species into the different systems (i.e. CH<sub>4</sub> in the *Gas* battery limit, O<sub>2</sub> and N<sub>2</sub> in the *Air* battery limit, etc).

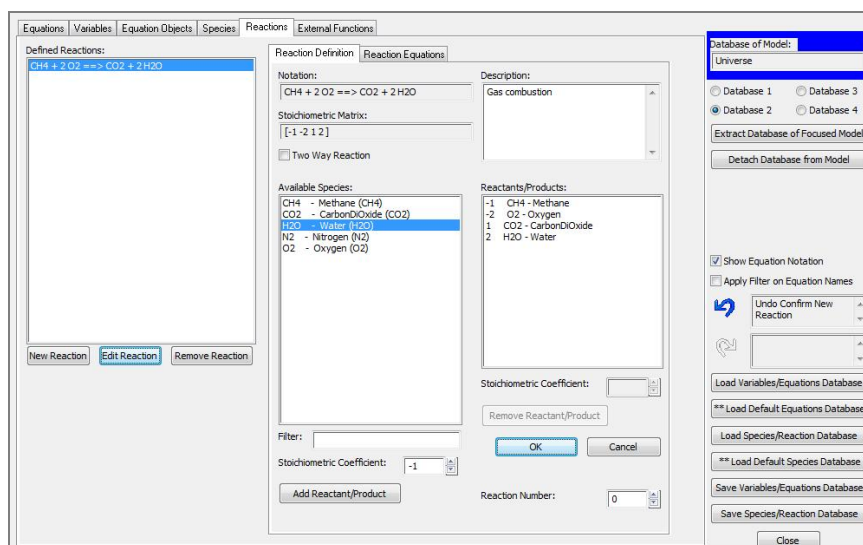
## 4.3 Inserting Chemical Reactions

A demonstration of the steps that are outlined in this paragraph can be viewed by clicking on the following link:

[Room 6 – Inserting Chemical Reactions](#)

In order to insert a chemical reaction in a system, the user has to proceed as follows:

- 1) **Edit the chemical reaction** → click on **Reactions** tab in the **Database Editor** and select **New Reaction**. In the **Reaction Definition** sub-tab the user will see all the available species within the system. We will select the species involved in the reaction and a stoichiometric coefficient will be selected for each of the compounds. Since we will define the methane combustion, the chemical reaction scheme should look as follows:



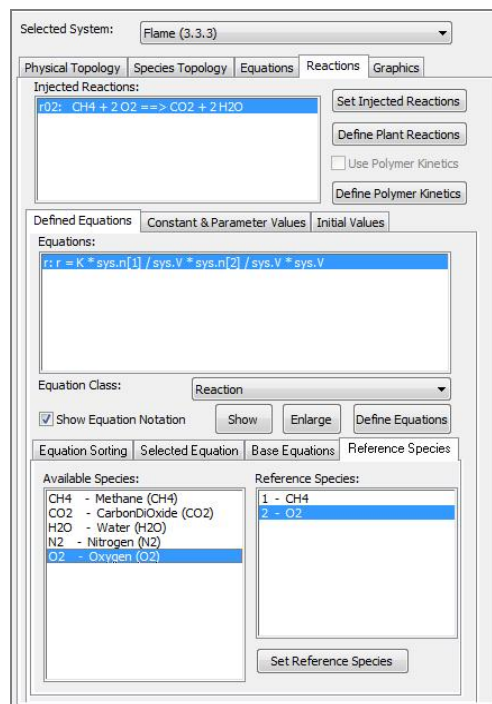
26. Editing a chemical reaction in the Database Editor.

- 2) **Define plant reactions and set injected reactions** → now the chemical reaction has to be injected in the “flame” system. Select the system and in the **Reactions** tab on the **Properties Dialog** click on *Define Plant Species*. We will add the chemical reaction describing the methane combustion to our model. The next step is to inject that reaction in the system. By clicking on *Set Injected Reactions* the reaction defined in the “plant” will be inserted in the system. The variable “*r* – Reaction Rate” should appear in the *Selectable* list box.
- 3) **Define reaction rate equation** → the kinetics of a chemical reaction is defined as a *Reaction Class Equation* in the **Database Editor**. Therefore, if the user wants to insert, for instance, a first order reaction rate equation, i.e.  $r = k \cdot [CH_4] \cdot [O_2]$ , the equation to edit in **Mobatec Modeller** will be the following (please note that the units for the reaction rate are mol/s!!):

$$r = k * sys.n[1] / sys.V * sys.n[2] / sys.V * sys.V$$

The prefix “sys” sets the connection between the variable defined in the system, i.e. volume or component molar hold-up, and the reaction rate equation. Then this reaction rate equation will be added to the “Flame” system, by clicking on *Define Equations* in the **Reactions** tab. The reaction rate constant “k” will be selected as a parameter.

The next step is to identify the species 1 and 2. Click on *Reference Species* and then connect the species with the reference number.



27. Setting the Reference Species in the chemical reaction.

- 4) **Initialized the reaction rate** → the reaction rate has to be initialized as the rest of variables of the system. In the **Reactions** tab of the *Flame* system, select the sub-tab **Initial Values** and by clicking on *Calculate Initial Values*, the reaction rate variable will be initialized.

#### 4.4 Controller



#### 4.5 Summary

The **Getting Started: step-by-step tutorial for beginning users** aims at introducing the basics of the modelling methodology of **Mobatec Modeller**. Although in these four chapters the reader has been running through the main features of **Mobatec Modeller**, there is a significant number of options that have not been introduced yet.

Nevertheless, after reading this document, watching the “Show-Me-Movies” and modelling the tutorial example, the user should have the capability to create a model and perform a simulation run of numerous processes of interest.

The reader should note as well that this document builds a model completely “from scratch”, with the objective of knowing all the possibilities that this modelling tool offers. However, during “normal” model design, the modeller has the option to use predefined objects from a library which reduces significantly the time spent on the modelling process.