Tecnomatix Plant Simulation 14 Online Tutorial

Welcome to Tecnomatix Plant Simulation 14 Online Tutorial. This online tutorial will introduce you to the basics of modeling and simulation of manufacturing and logistics processes using Tecnomatix Plant Simulation 14.

Objectives

This online tutorial is intended for self-study. It is aimed at first-time users from academia and industry, as well as students and young professionals who want to get started in discrete-event simulation and who would like to work with Plant Simulation. After completing this online tutorial, you will be able to build, run, and analyze material flow and logistic simulation models using Plant Simulation.

Content overview

- General understandings of simulation
- Introduction to the graphical user interface of Plant Simulation
- Understanding modeling techniques and object structures in Plant Simulation
- Modeling material flow simulations using standard library objects
- Analyzing and visualizing simulation results
- Introduction to the basics of 3D modeling in Plant Simulation

Quick navigation

Click on one of the buttons below to start with the first chapter or jump to a specific chapter.

Chapter 1: Introduction to Simulation

This chapter will cover theoretical aspects and general understandings of simulation, as well as introduce you to the key capabilities of Plant Simulation. After completing this chapter, you will be able to:

- Understand what simulation is and what benefit simulation provides
- Understand different types of simulation
- Understand key capabilities and use cases of Plant Simulation

Simulation

"Simulation is the representation of a system with its dynamic processes in an experimentable model to reach findings which are transferable to reality. In particular, the processes are developed over time." (VDI 3633-1, 2010)

The increasing complexity of products, processes, and systems is one of many reasons why simulation is increasingly used in the planning, implementation, and operation of technical systems. As an integral part of the digital factory concept, simulation helps to shorten time-to-market and supports decision-making processes. Simulation is particularly useful for analyzing a system, in which its behavior dynamically changes over time and a mathematical solution cannot be calculated at a reasonable effort.

During the planning phase, simulation can be used to:

- Determine the required amount of resources, transport systems, buffer sizes
- Validate and optimize planning parameters
- Evaluate planning alternatives based on different scenarios
- Analyze system behavior over long time periods and during ramp-ups
- Provide a proof-of-concept

During the operation phase, simulation can be used to:

- Optimize control strategies, scheduling, and order sequencing
- Verify resource and buffer utilization, batch sizes, and inventory levels
- Analyze what-if scenarios without interrupting an operating system
- Analyze the influence of failures and interruptions
- Train workers in the process and material flow

Types of simulation

Simulation can be used for physical phenomena, business processes, pedestrian and traffic flow, as well as manufacturing and logistics operations. Mainly, there are two different types of simulation:

- Continuous simulation
- Discrete-event simulation

Whereas continuous simulation view state changes in the simulation model over time in a continuous manner, discrete-event simulation only inspects certain points in time, at which a certain event takes places or a certain model component changes its states, e.g. when a station starts or finishes processing a part. As such, discrete-event simulation jumps from one event or one state change to the next and skips the time in between.

Simulation activities

The scope of simulation can range from a sub-area of a production facility to a globally distributed supply-chain network. In manufacturing and logistics, simulation helps to identify bottlenecks, high inventory levels, feasible throughput amount, and suitable control strategies. A typical process of a simulation project consists of (Rabe, Spieckermann, Wenzel, 2008):

- Problem and target description
- System analysis
- Data collection
- Model formalization and implementation
- Experiment and analysis of simulation results

Simulation is always based on a model, which is a simplified reproduction of a planned or an existing system. Thus, simulation results can only be as good as the model and the data on which the simulation experiment is based (VDI 3633-1, 2010). In order to get feasible results with reasonable efforts, the level of detail of a simulation model should be neither too detailed nor too abstracted.

Tecnomatix Plant Simulation

Plant Simulation is a discrete-event process simulation software and part of the Tecnomatix digital manufacturing solutions by Siemens PLM Software. Plant Simulation enables its user to create an object-oriented and hierarchical simulation model of production and logistics systems, including their complex control strategies. Based on these simulation models, you can perform an analysis of the system characteristics, detect potential vulnerabilities or bottlenecks, and optimize performance including resource utilization, throughput, and material handling operations.

Key capabilities of Plant Simulation

The main key capability of Plant Simulation is the object-oriented and hierarchical modeling approach. The object-oriented approach enables a child object to inherit its properties (attributes) and behaviors (methods) from a parent object. The modeling capabilities of Plant Simulation not only allows you to use standard library objects but also to develop reusable custom library elements. This allows you to define complex material flow rules and control strategies, check their performance in a simulation model, and reuse them in other simulation models.

According to Siemens PLM Software, other key capabilities of Plant Simulation are:

- Object libraries and management for standard and custom library elements
- Interactive 2D and 3D visualization with facility layout integration
- Integrated charts to visualize and analyze throughput, utilization, and bottlenecks
- Integrated neuronal networks, genetic algorithms, and experiment manager
- Open system architectures that support numerous interfaces (e.g. ActiveX, SQL, ODBC, Socket, etc.)

Before you begin

Make sure that Plant Simulation is installed and running on your computer. Plant Simulation is available with commercial and non-commercial licenses. Non-commercial licenses for academia include Educational License and Research License. If you are a student from an academic institution, you can download the free student version of Plant Simulation from Siemens PLM Software.

Note that the amount of objects that can be created in a simulation model is limited to 80 objects with Student License and 1000 objects with Educational License. You can still load a simulation model created using other license options in Student or Educational License. However, if the amount of objects exceeds the limit, you will not be able to save the model.

Also, note that simulation models created with a non-commercial license cannot be used for commercial purposes. Models created with Student or Research License can still be opened in Plant Simulation with a commercial license, e.g. Professional License. However, it is not possible to open a model created with Educational License in Plant Simulation with a commercial license.

This tutorial is created using Plant Simulation 14.0.1 with Professional License. If you use a previous version of Plant Simulation (e.g. Plant Simulation 13 and downwards), some functionalities may vary and you will not be able to open the example models provided in this tutorial due to the downward incompatibility of Plant Simulation.

Note that you can open and run simulation models created using a previous version with a newer version of Plant Simulation (e.g. from Plant Simulation 13 to Plant Simulation 14). However, if you save the model with a newer version of Plant Simulation, you will not be able to open it in the previous version of Plant Simulation anymore.

previous chapter | next chapter

Chapter 2: First Steps in Plant Simulation

This chapter will introduce you to first steps in Plant Simulation and help you to get familiar with the graphical user interface of Plant Simulation. After completing this chapter, you will be able to:

- Launch and set preferences of Plant Simulation
- Understand the working environment, toolbars and ribbon menus in Plant Simulation

Getting started

After you have installed Plant Simulation, double-click on the Plant Simulation desktop shortcut or select the program from Windows start menu to launch Plant Simulation. The start page of Plant Simulation will open. From the start page, you can choose to create a new model, open an existing model, as well as to view example models, demo videos and tutorials. The start page also includes a link to the Plant Simulation community forum where you can discuss modeling questions with other Plant Simulation users.

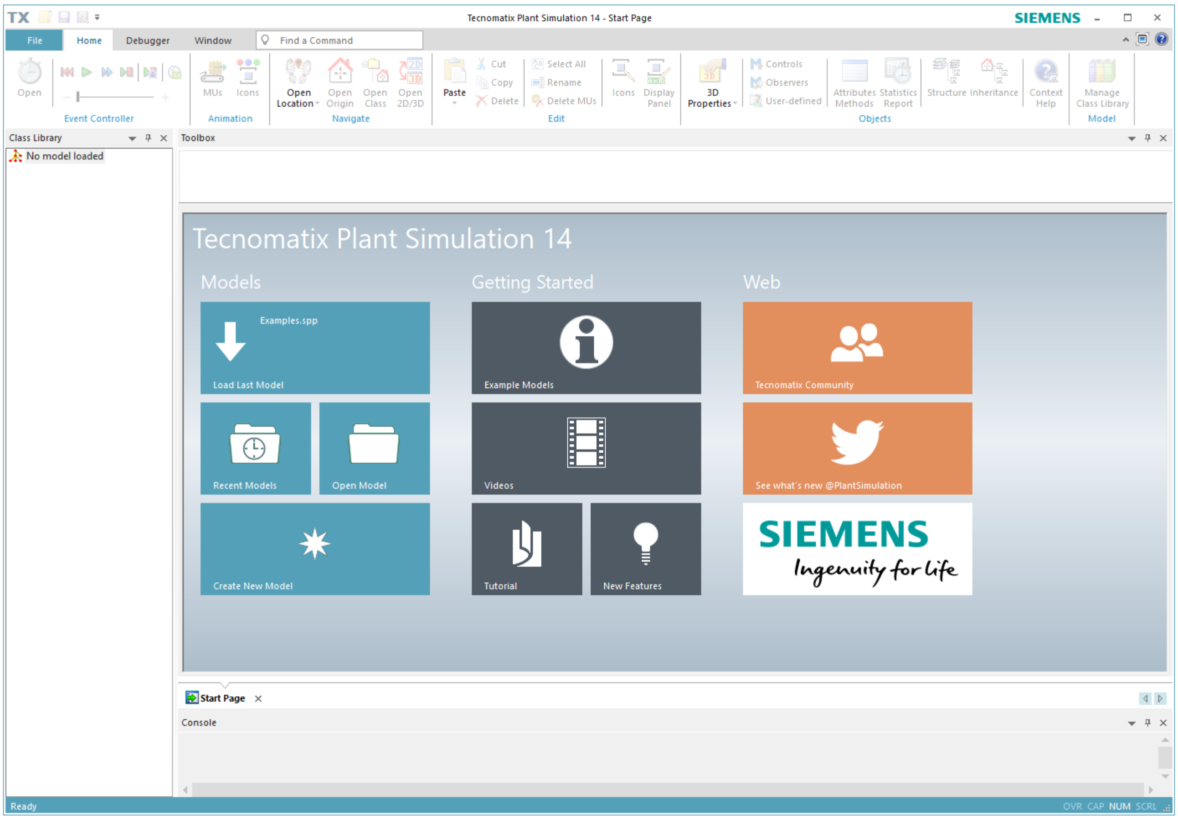

Figure 1: Plant Simulation start page

Tips: Plant Simulation is available in different languages. By default, Plant Simulation will run in the same language as your system language. You can configure the program language using the start options of Plant Simulation.

1. Create a desktop shortcut for Plant Simulation
2. Right-click on the shortcut and select **Properties** from the context menu
3. Extend the path to Plant Simulation in the target box with desired start options, for example:
    - If you want to change the program language of Plant Simulation, enter */UILanguage:<Language>* (e.g. */UILanguage:DEU* for German and */UILanguage:ENU* for English)
    - If you want Plant Simulation to save your model every defined time intervals, enter *-a <TimeIntervalInMinutes>* (e.g. *-a 10*)
4. You can find all possible start options in Tecnomatix Plant Simulation Help > Setting-Up and Starting > Starting Plant Simulation > Entering Start Options.

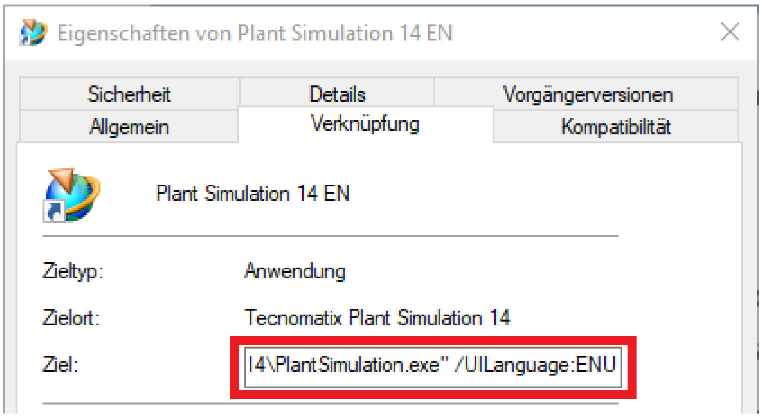

Figure 2: Changing language in the start options

Setting preferences

Before you start working with Plant Simulation, you might want to change some of the default settings that affect your simulation model.

- If you want to change the setting for a specific model, select **File** → **Model Settings**. This option is only available after you open a model in Plant Simulation and will only apply to the active model.
- If you want to change general settings which will apply to new models, select **File** → **Preferences**. This is also required to enter the license information before you create any new model in Plant Simulation.

In the Preference window, you can change the model language, date and time format, simulation units, and timescale among other things. To enter license information, navigate to the License tab. Select the correct *License type* from the drop-down menu. If you have a License file, click on the folder icon to navigate to the license file's path. If you get your license from a license server, enter the address and port of your license server. Click **OK** and restart the program for the change to take effect.

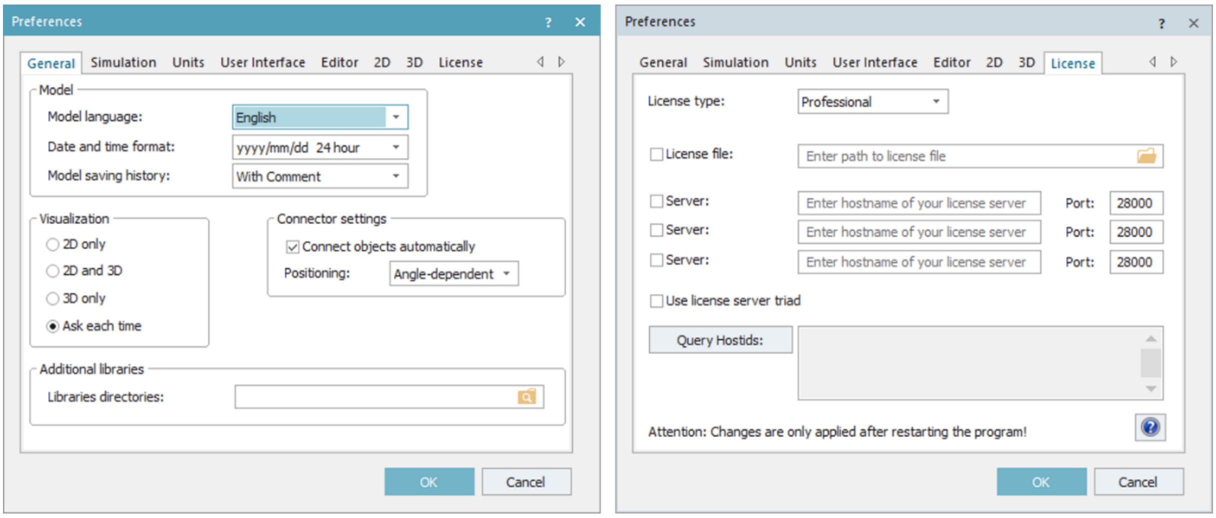

Figure 3: Preferences window – General tab (left) and License tab (right)

Working environment

Click on **Create New Model** on Plant Simulation start page or select **File** → **New** to create a new model. Plant Simulation will ask whether you want to create the model in 2D or in 3D. If you select 2D for example, you can always activate the 3D visualization later and vice versa. Choose one of the option and Plant Simulation will automatically create a new model frame and load available object classes.

The graphical user interface (GUI) of Plant Simulation consists of different sections, as shown in the figure below. The numbered sections are:

1) Ribbon menu
2) Class library window
3) Toolbox with shortcut symbols
4) Model frame window
5) Console window

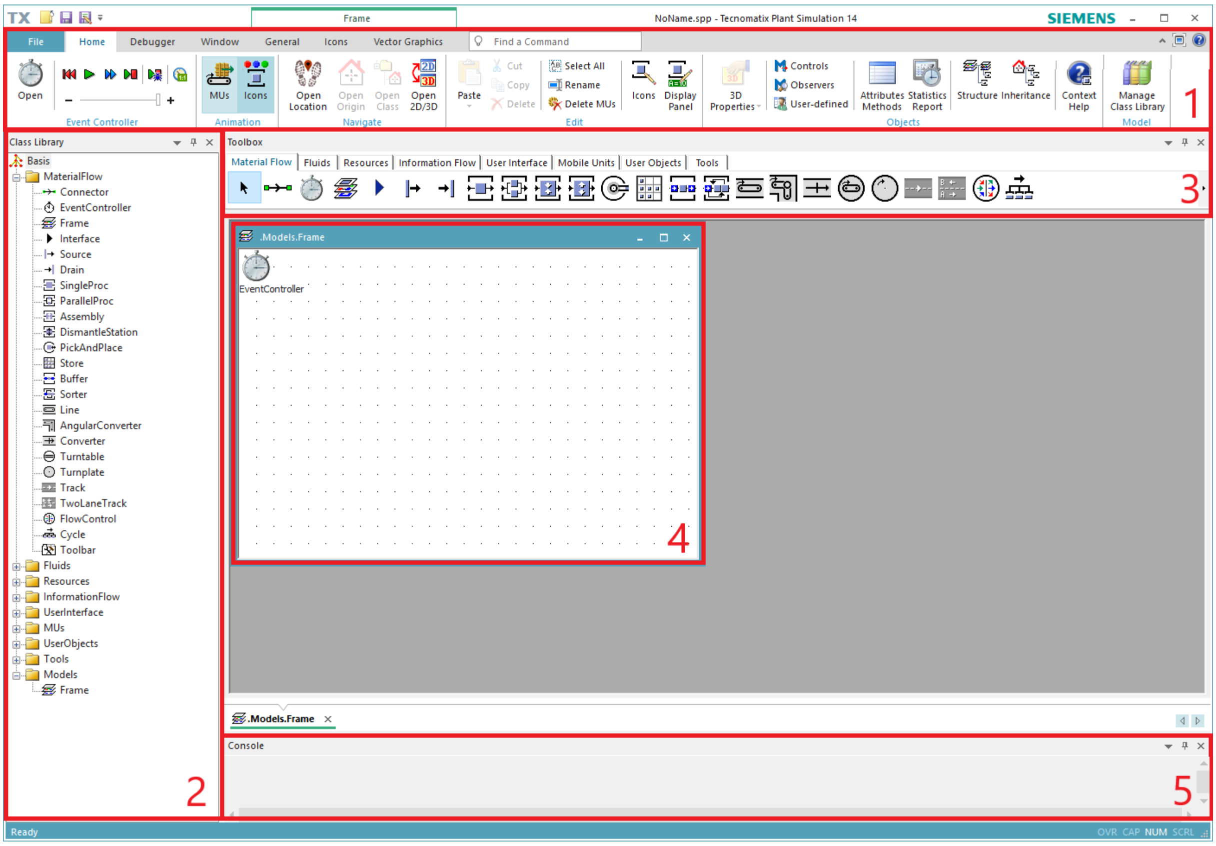

Figure 4: Working environment in Plant Simulation

Ribbon menu

Plant Simulation uses the ribbon interface to provide quick access to general functions and toolbars. The toolbars are grouped in a number of ribbon tabs. There are default ribbon tabs and object-specific ribbon tabs. You can also modify and customize the ribbon by right-clicking on an empty ribbon area and select **Customize the Ribbon…** from the context menu as shown in the figure below.

The default ribbon tabs are always displayed in the Plant Simulation window. These are *Home*, *Debugger*, and *Window* ribbon tab. The *Window* ribbon tab, as shown in the figure below, contains toolbars to toggle the display for each GUI section. For example, you can display or hide the Class Library or the Toolbox by clicking on the corresponding icon. If you create a simulation model in 2D, you can also activate the 3D model using the Activate 3D icon in the *Window* ribbon tab.

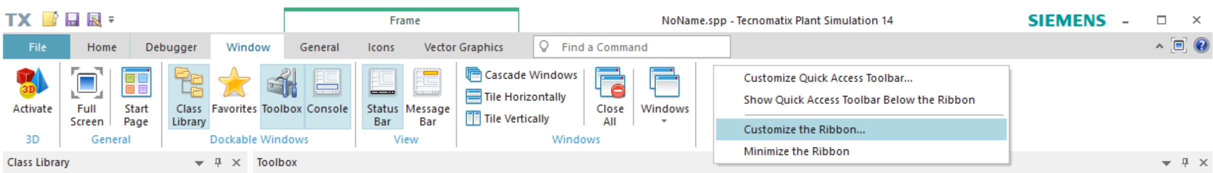

Figure 5: Window ribbon tab

Object-specific ribbon tabs are only displayed if the specific object is selected or activated in the simulation model. For example, if you are currently working within a frame, the *Frame* ribbon group consisting of *General*, *Icons*, and *Vector Graphics* ribbon tab, will be displayed beside the default ribbon tabs. Other objects which provide object-specific ribbon tabs are for example *3D Viewer*, *Method*, *Icon Editor*, and *TableFile*.

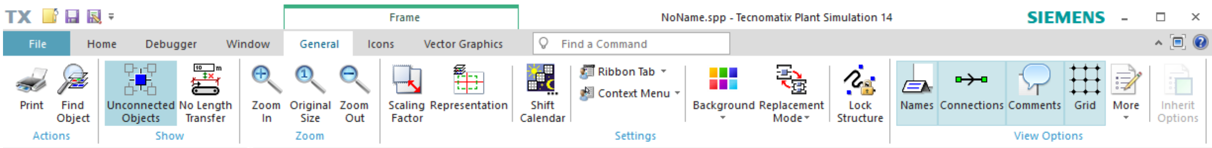

Figure 6: Frame ribbon group – General ribbon tab

Class library and toolbox

If you create a new model or open an existing model in Plant Simulation, the model file will load the *Class Library* with its available object classes. By default, the *Class Library* contains a folder with *MaterialFlow, Resource, InformationFlow*, *MUs,* and *UserInterface* objects, as well as *Models* and *Tools* folder. The structure of the *Class Library* looks similar to the one of Windows Explorer. You can modify or extend this structure as needed.

If you use Plant Simulation with a license that limits the number of objects in a simulation model, the *Basis* tree in the *Class Library* will show the number of used and maximum possible objects in the model. You can add or remove an object class from the class library by clicking on **Manage Class Library** from the *Home* ribbon tab.

- The Basic Objects tab contains built-in object classes that can be loaded or removed from the simulation model. It also shows which license type is required in order to be able to load the object.
- In the Libraries tab, you can select additional libraries. These can be either built-in libraries or self-developed libraries. The currently used version is shown for each loaded library.

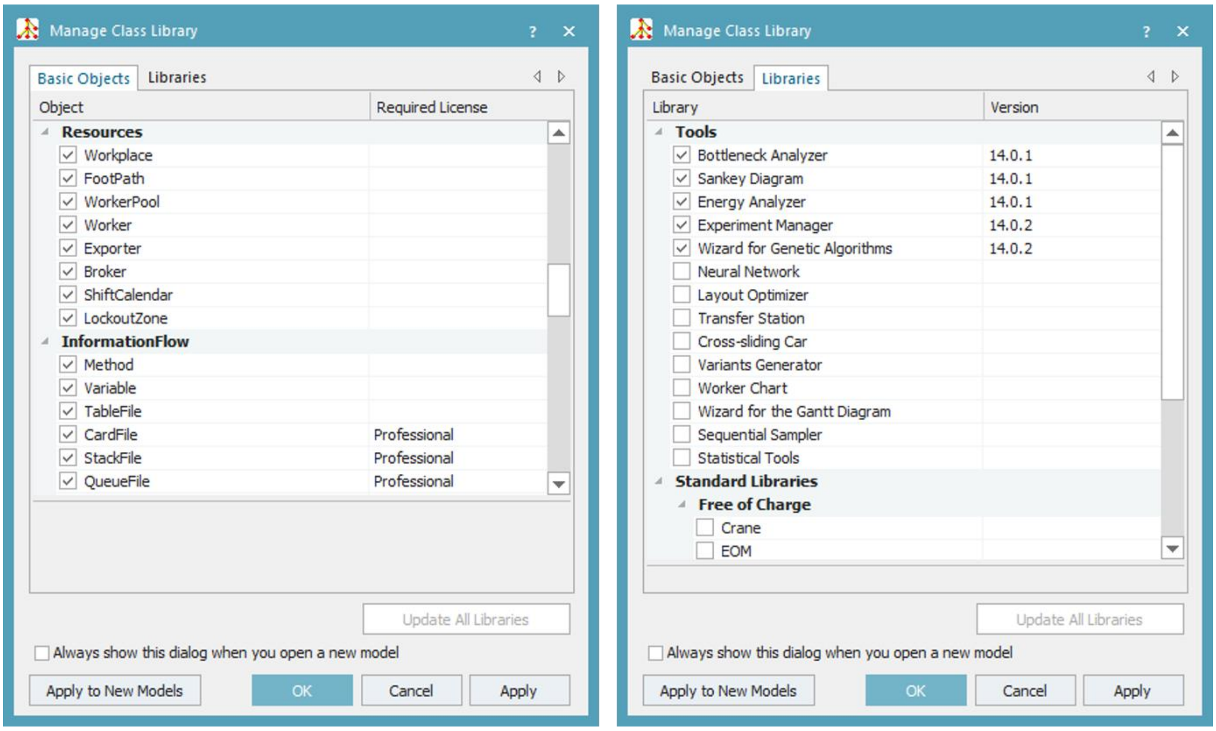

Figure 7: Manage Class Library dialog

The *Toolbox* is a container for the different Plant Simulation toolbars that hold the objects of the *Class Library*. The Toolbox allows you for easy and fast access to object classes during the modeling.

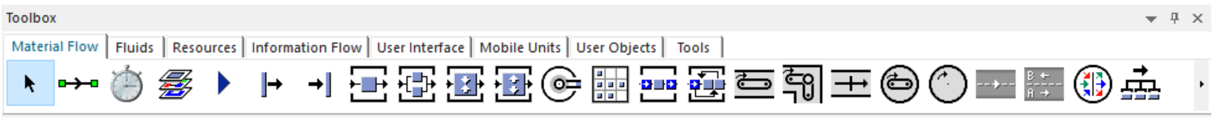

Figure 8: Material Flow Toolbox

Model frame and console window

Your simulation model is always created within the **Frame** . By default, Plant Simulation will place a *Frame* under the *Models* folder when you create a new model. You can use this as your main frame.

In the course of this tutorial, we will refer to objects in the *Class Library* and *Toolbox* as class objects and objects inserted in the frame as instance objects. We will cover more about object inheritance in the next chapter.

There are several ways to insert objects from the *Class Library* into the frame, as shown in the figure below.

- Drag & drop a class object from the *Class Library* to the *Frame* window.

- Drag & drop a class object from the *Toolbox* to the *Frame* window.
- Left-click on a class object in the *Toolbox* and left-click again on the desired position in the *Frame* window.

Tips: You can also insert multiple instances of the same class object by pressing and holding down the Ctrl-key while you left-click in the *Frame* window. This way, you do not have to return to the *Toolbox* each time after inserting an object.

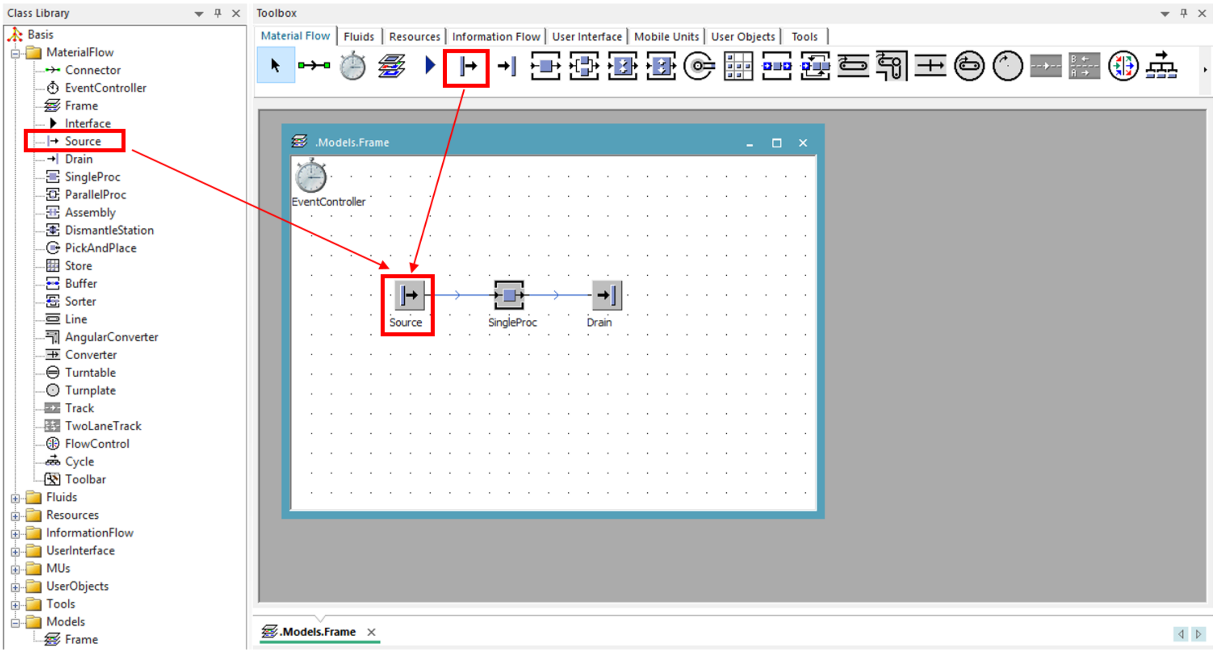

Figure 9: Modeling in Frame

In the *Class Library*, you can create any number of frames at any hierarchical levels. This is especially useful if you want to divide your simulation model into sub-models. This way, you can model parts or area of a larger simulation model independently using sub-frames. You can then insert any number of sub-frames into another frame or into your main frame to create a hierarchical model structure.

previous chapter | next chapter

Chapter 3: Modeling in Plant Simulation – Material Flow (I)

This chapter will introduce you to the modeling of material flow objects in Plant Simulation. After completing this chapter, you will be able to:

- Understand the modeling approach of Plant Simulation
- Know the difference between point-oriented and length-oriented modeling
- Create simulation models using material flow objects in Plant Simulation
- Run your simulation model

Modeling approach

One of the main features of Plant Simulation is the object-oriented modeling approach. By using this approach, you can define the properties of an object in a class, create multiple instances of this class, and make use of the inheritance between the class and its instances. You can imagine a class as a template, whereas instances are individual objects that are created using this template. When you define the attributes of a class, they will be passed on to all instances created from this class. This means that if you modify the value of an attribute in the class object, the change will also be applied in all instance objects derived from this class. This is more efficient and less error-prone than having to change the same property for each individual instance object every time.

However, you can also assign the value of an attribute individually for an instance object. This will overwrite the derived attribute in this instance object and deactivate the inheritance from the class object. The overwritten attribute in the instance object influences neither the attribute in the class object nor the attribute in other instances of the class. On the other hand, any change you make in the class object will not be applied anymore to the instance object, in which the inheritance is deactivated. Plant Simulation provides a toggle button to activate and deactivate the inheritances for each attribute. An active inheritance is shown using this icon, while inactive inheritance is shown using this icon.

As mentioned in the previous chapter, we will refer to objects in the *Class Library* and the *Toolbox* as class objects. Objects that are inserted from the Class Library or the Toolbox into a *Frame* will be referred to as instance objects.

You can also define a new class object based on an existing class object in the Class Library. The Class Library provides the following methods to create a new class object from an existing class object:

1) Right-click and select **Duplicate**
   Duplicating a class object will create a new independent class object. All attributes of the existing class will be copied to the new class object, but the inheritance relations are severed.
2) Right-click and select **Derive**
   Deriving a class object will create a new class object and keep the inheritance relation between the new and the existing class object. The new class object represents a subclass of the existing class object. Any changes made in the attribute of the class will be passed on to the subclass, as long as the inheritance for this attribute is still active.

To show all objects derived from a class object, right-click on the class object in the Class Library and select **Show Inheritance**. In the example shown in the figure below, you can see that the *WorkerBeginner* class was derived from the *Worker* class and the inheritance relations between them are still active. On the contrary, the *WorkerAdvanced* class was duplicated from the *Worker* class and therefore is not shown in the inheritance structure.

You can also see which properties are inherited in the subclass. In this example, the value for the attributes *Priority* and *Capacity* in the *WorkerBeginner* class are inherited from the *Worker* class, while the value for the attributes *Efficiency* and *Speed* are overwritten.

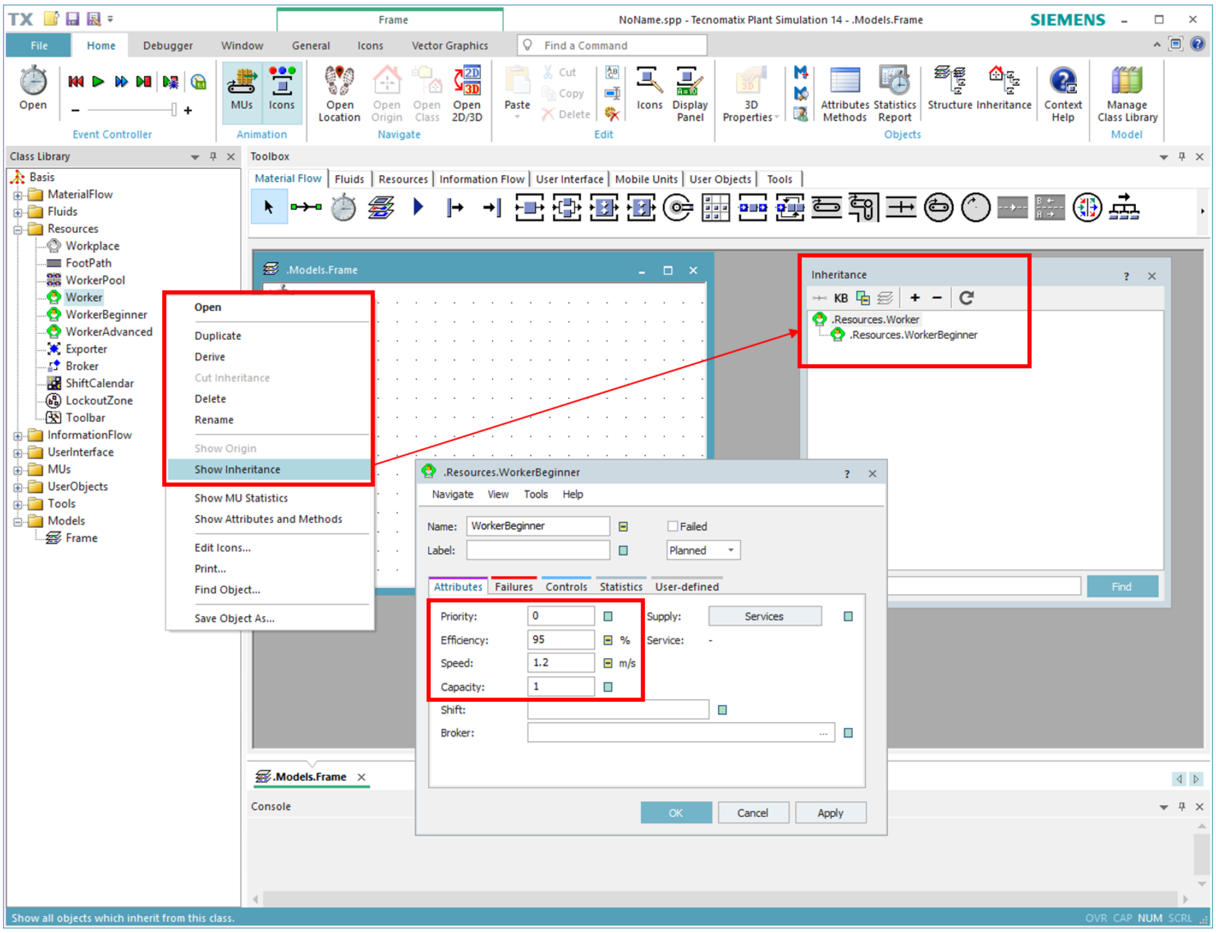

Figure 10: Class, subclass, and inheritances

By making use of object inheritances in **Frame** you can create a hierarchical structure of your simulation model. This means that you can model parts or area of your overall plant independently in different frames, make sure that these frames work by themselves, and then insert them into the main frame to build the entire plant.

Interface connects the material flow into and out of the frame. You can only connect a frame to another object if the frame contains at least one interface. If an interface is connected to another object, Plant Simulation will automatically recognize based on the predecessor and successor relationship information whether the interface is an entrance or an exit of the corresponding frame.

Material flow objects within the same frame are connected by **Connector** . This connection is unidirectional and establishes the predecessor-successor relationship between material flow objects. Each predecessor and successor is numbered in the connector. You can see this relationship information if you hover with your mouse over a connector. Alternatively, you can always show them by right-clicking on the frame and selecting **View Options** → **Show Predecessors** or **Show Successors**.

Exercise: Create your model structure

In this tutorial, we will model an example factory which produces radiators. The factory is divided into a manufacturing area and assembly area. In the manufacturing area, the raw material is processed through various manufacturing steps. In the assembly area, the processed part is assembled together with other purchased parts to form a finished product. The finished products are packaged and transported to shipment. An overview of the whole production process is shown in the figure below.

In this exercise, we will first create the structure of the factory using a top-down approach with frames. During the course of the tutorial, you will extend the model with further objects from the Class Library.

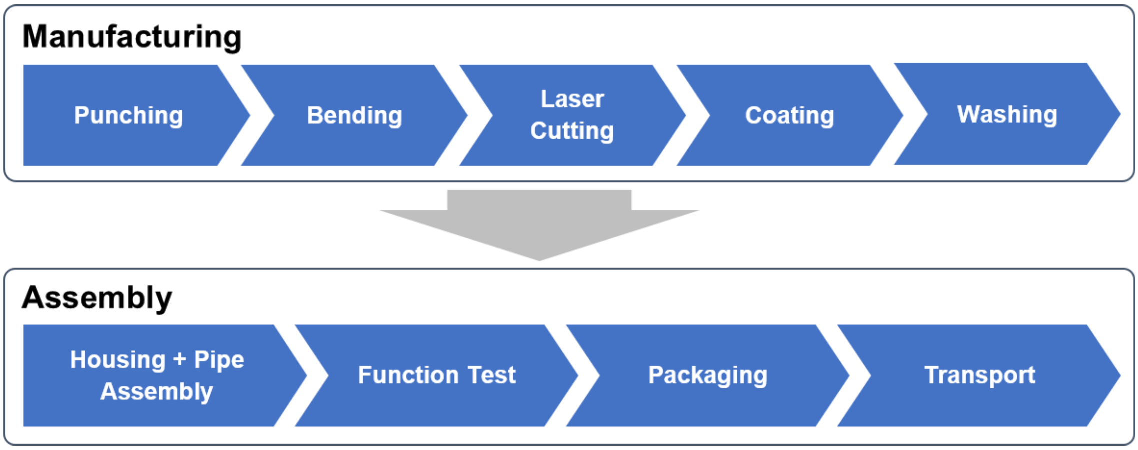

Figure 11: Overview of the production process

Step-by-Step:

1. Select **Create New Model** from the start page and select **2D** at the dialog prompt.
2. Plant Simulation will create a default frame under the *Models* folder in the Class Library. Select this frame and click **F2** to rename it to *"Plant"*. This will be our main simulation frame.
3. Right-click on the *Models* folder and select from the **New** → **Folder** from the context menu to create a new folder in the Class Library and nest it under the *Models* folder. Rename the folder to *"FacilityArea"*.
4. Right-click on the *FacilityArea* folder and select **New** → **Frame** from the context menu to create a new frame under this folder. Rename the frame to *"Manufacturing"*.
5. Repeat the previous step and rename the frame to *"Assembly"*.
6. Open the *Manufacturing* frame. Insert two **Interfaces** from the Toolbox into this frame. Rename the first one to *"Entry"* and the second one *"Exit"*. Try to adjust the

position of the interfaces in the frame. Entry interface should be arranged on the left side of the frame and exit interface on the right side of the frame.

7. Repeat the previous step in the *Assembly* frame.
8. Open the *Plant* frame. Drag & drop the *Manufacturing* frame and the *Assembly* frame from the Class Library onto the *Plant* frame. This step will create an instance of the **Manufacturing** and **Assembly** frame each in the *Plant* frame.
9. Connect these two frame instances in the Plant frame: Select the **Connector** from the Toolbox, left-click on the Manufacturing frame and select the Exit interface, left-click again on the Assembly frame and select the Entry interface.
10. Save the model as **Tutorial_Model_01.spp**.

The figure below shows the state of the model after completing this exercise. You will notice that there are two Manufacturing frame objects, and correspondingly two Assembly frame objects. The Manufacturing frame in the Class Library (2) is the class object. It is referenced using the path *.Models.FacilityArea.Manufacturing*. The Manufacturing frame inside the Plant frame (3) is an instance of the class and is referenced using the path *.Models.Plant.Manufacturing*. You will also notice that the Exit interface in the instance is already connected, while the Exit interface in the class frame is not connected. When you are inside an instance object, you can always navigate to the class object by selecting **Open Origin** or **Open Class** from the Home ribbon tab (4).

The content of the Manufacturing instance (3) inside the Plant frame (1) is automatically inherited from the Manufacturing class (2). This way, you will be able to reuse the Manufacturing class in various other frames. If you try to add objects into the Manufacturing instance, you will get an error message as the structure of an instance frame is locked by default. You can click on **Lock Structure** on the Frame ribbon tab General to lock/unlock the frame structure.

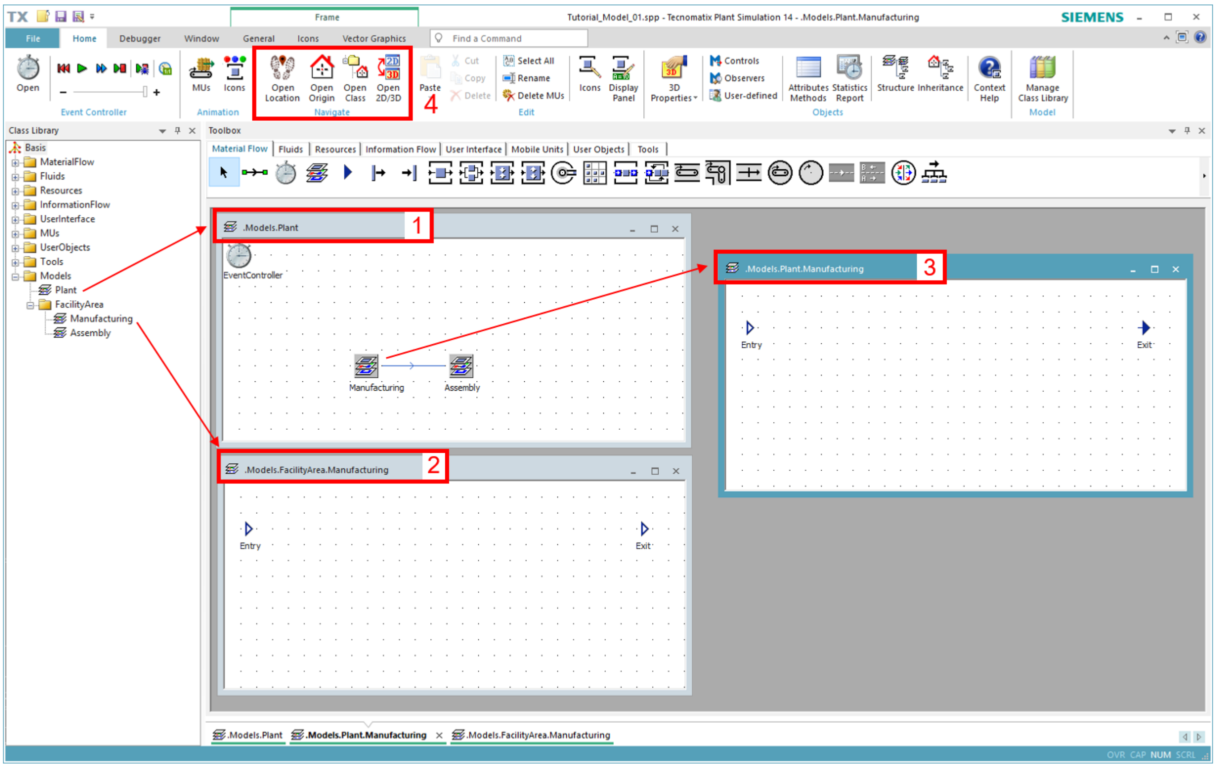

Figure 12: Model structure

Modeling material flow – Mobile Units

The **Mobile Units (MUs)** are movable objects that move through the simulation model. They represent the flow of materials and requires other stationary material flow objects to process and transport them. There are three kinds of MUs:

- **Entity** represents parts that are produced, processed, and transported in the simulation model.
- **Container** represents pallets, bins, boxes, etc. that can be used to load and transport one or multiple entities.
- **Transporter** represents vehicles that can transport entities, containers, and other transporters. Transporter can move on its own on a length-oriented object, such as tracks or lines.

Source introduces MUs into the simulation. It creates the parts that will be processed in the production facility according to given parameters. The Source can produce MUs by defined time intervals, amount of MUs to be created, as well as based on a delivery table or trigger.

Drain destroys the MUs from the simulation. It typically represents the end of a production line. The statistics of the deleted MUs are collected by the Drain and can be used to calculate KPIs, e.g. the throughput of the production facility.

Exercise: Mobile Units

In this exercise, we will extend the previous simulation model with a Source and a Drain. We will also define two new MU types in the simulation, one to represent the main part that is processed during manufacturing and one to represent the final assembled product.

Step-by-Step:

1. Expand the *MUs* folder in the Class Library. Right-click on the Entity class and select **Derive** from the context menu.
2. Double-click on the derived Entity class and rename it to *"SheetMetal"*. Change the *MU Size* in the Attributes tab as shown in the figure below.
3. Go to the Graphics tab. You will see that *vector graphics* are activated by default. In this case, Plant Simulation will show the animation of the parts with the correct size and correct distance to each other. If you wish to, you can change the color. Click on **Apply** or **OK** to apply the changes.

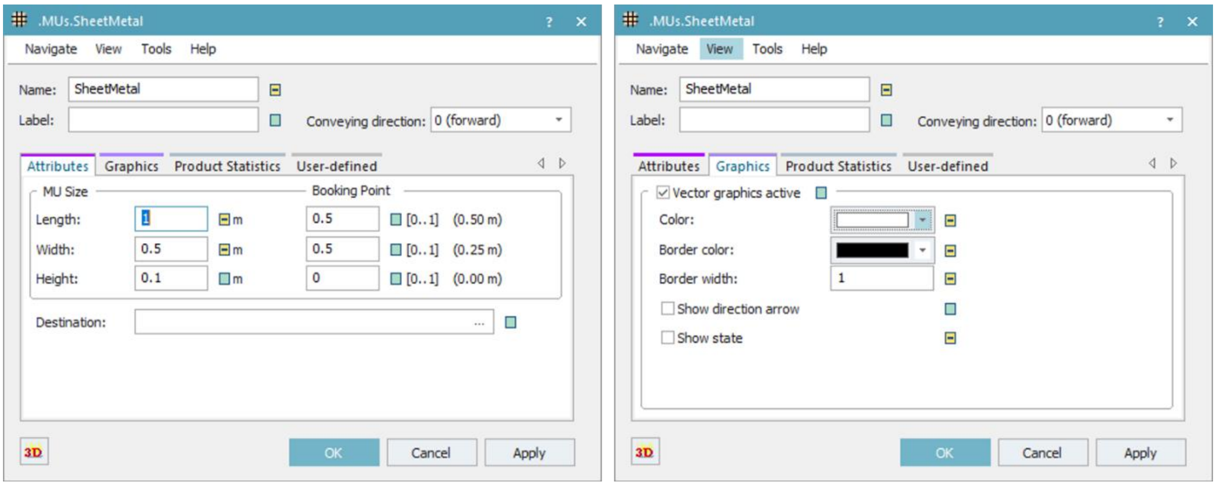

Figure 13: Entity dialog properties

4. Repeat the first step to derive another subclass of the Entity class.
5. Double-click on the derived Entity class and rename it to *"Radiator"*. This will be our final assembled product. Change the *MU Size* in the Attributes tab to Length *"1 m"*, Width *"0.5 m"*, Height *"1 m"*, and set all *Booking Point* to *"0.5"*.
6. Navigate to the Graphics tab and deactivate the vector graphics. In this case, Plant Simulation will show the default icons of the Entity during the animation.

7. Right-click on the Radiator class and select **Edit icons…** from the context menu. There are two icons of the Entity by default, one for the operational state and one for the waiting state. **Deactivate the Inheritance** using the icon in the *Edit* ribbon tab. You can then change the size of the icon and paint the icon as you like, for example, like in the figure shown below. Now we can easily differentiate between the SheetMetal entity and Radiator entity during the simulation.

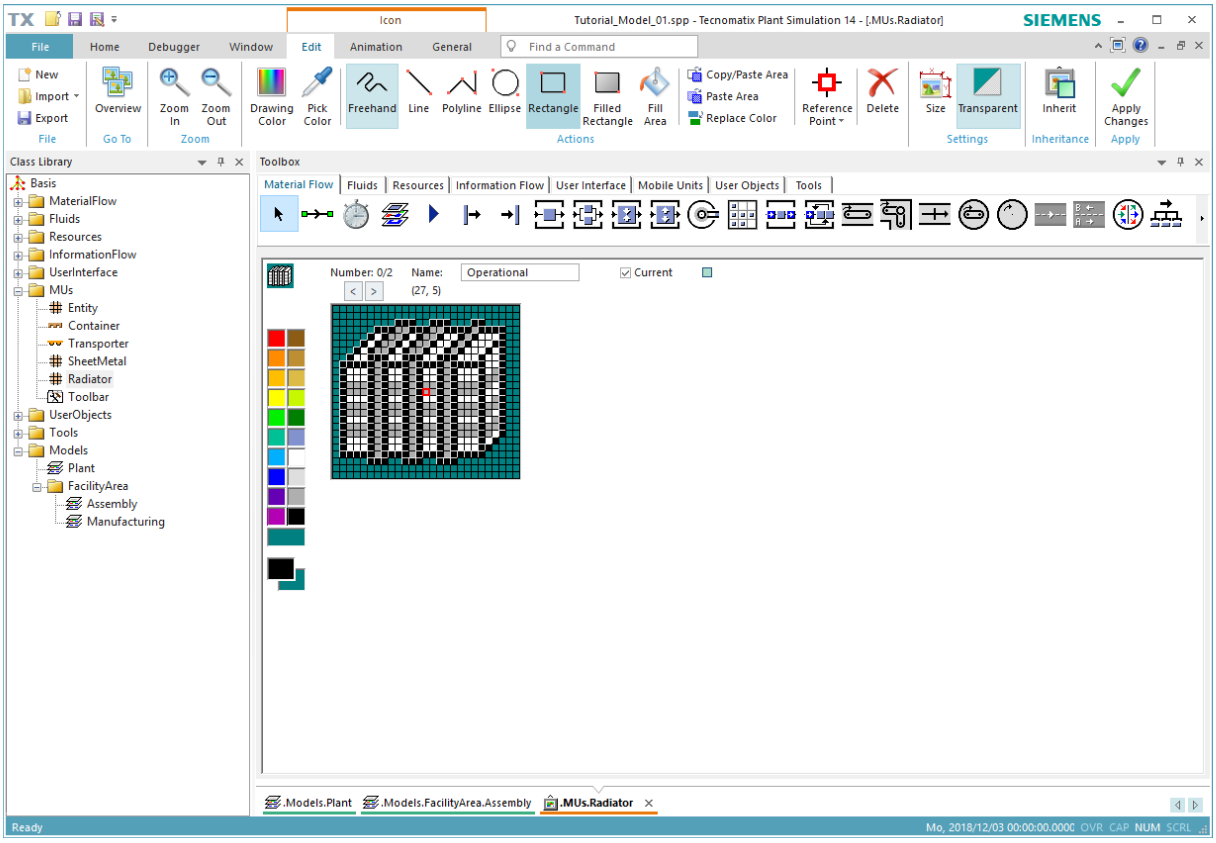

Figure 14: Edit icons

8. Open the Plant frame. Insert one **Source** and one **Drain** from the Toolbox. Position the Source on the left side of the Manufacturing frame and the Drain on the right side of the Assembly frame.

9. Select the **Connector** in the Toolbox and connect the objects as follows:

   - *Source* to the *Entry* interface of the *Manufacturing* frame
   - *Exit* interface of the *Assembly* frame to *Drain*

10. Double-click on the Source instance. Change the attributes as shown in the figure below. Click on the button under next to *MU* under *MU selection* and choose **Select Object**. In the Select Object window, choose the SheetMetal entity and click **OK**. The Source will now produce one instance of the SheetMetal entity every 30 seconds.

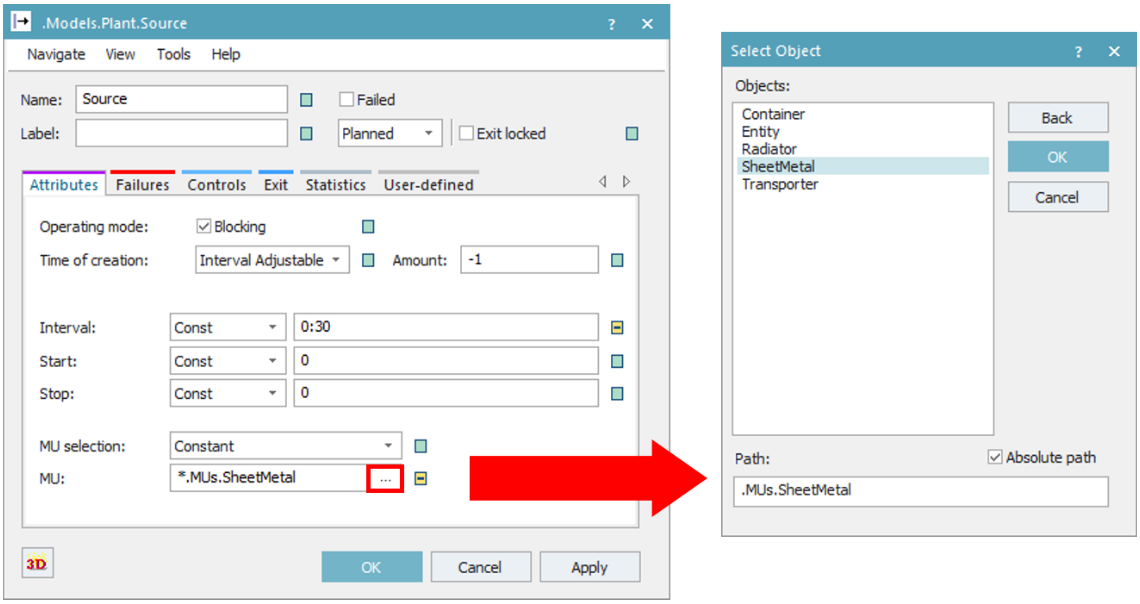

Figure 15: Source dialog properties

Download: The final state of the simulation model up to this exercise can be downloaded here Tutorial_Model_01.spp

previous chapter | next chapter

Chapter 3: Modeling in Plant Simulation – Material Flow (II)

Modeling material flow – Point-oriented and length-oriented objects

MUs are processed, transported and stored by material flow objects. Generally, we can differentiate between two categories of material flow objects: point-oriented and length-oriented material flow objects.

Point-oriented material flow objects do not take into account the physical length or dimension. MUs are processed or stored in these objects on a fixed location for a given time. Material flow objects that belong in this category are:

- **SingleProc** is a single station that can process one MU at a time. It receives an MU from its predecessor object and releases the MU after the defined set-up and processing times to its successor object.
- **ParallelProc** is a station that can process several MUs at a time. *ParallelProc* corresponds to two or more *SingleProc* stations that receive the same parts from the same predecessor.
- **AssemblyStation** assembles MUs received from multiple predecessor objects together. You can create an *Assembly Table* to define the number of parts required from each predecessor to create the final product.
- **DismantleStation** removes assembled parts from the main MUs.
- **Buffer** can temporarily hold MUs from the predecessor object when the successor object is unavailable to receive them. Buffer dimensioning is one of the main use cases for simulation study, as a sufficient buffer dimension is required to prevent production stop. A buffer can either be a queue (FIFO – First In First Out) or a stack (LIFO – Last In First Out).
- **FlowControl** is used to model different strategies for distributing and merging the flow of materials from one or multiple predecessor objects to one or multiple successor objects.

On the other hand, **length-oriented material flow objects** take into account their length and dimension as well as the length and dimension of the MUs that pass through them. Length-oriented material flow objects use the length, dimension, and speed information to determine the time required to pass on the MUs to the successor. Material flow objects that belong in this category are:

- **Line** is used to model a conveyor system that transports MUs along its entire length with a constant speed. An MU that is currently transported along the *Line* cannot pass another MU in front of it.
- **Angular converter** connects two *Line* objects and changes the conveying direction of the MUs accordingly. It can only move one MU at a time.
- **Turnplate** is used to model a rotating platform. It can connect two or more *Line* objects and ensure the uniform orientation of the MUs leaving it by rotating them.

Exercise: Point-oriented objects

In this exercise, we will extend the previous simulation and model the manufacturing area using point-oriented material flow objects. It is highly recommended that you only change the Manufacturing frame in the folder FacilityArea in the Class Library.

Step-by-Step:

1. In the frame *.Models.FacilityArea.Manufacturing*, insert the material flow objects and rename them as shown in the figure below. Hold down the Ctrl-key to insert the same object several times without returning to the Toolbox each time.
2. Connect the objects using **Connector**. Hold down the Ctrl-key and left-click on the objects to insert the Connector several times and create multiple connections without returning to the Toolbox each time.

Tips: To create a non-straight connection, click on the first object you want to connect, set one or more anchor points by clicking on the free area in the frame, and finally click on the second object you want to connect. This is especially useful if the connectors would overlap each other. You can select and move the anchor points afterward.

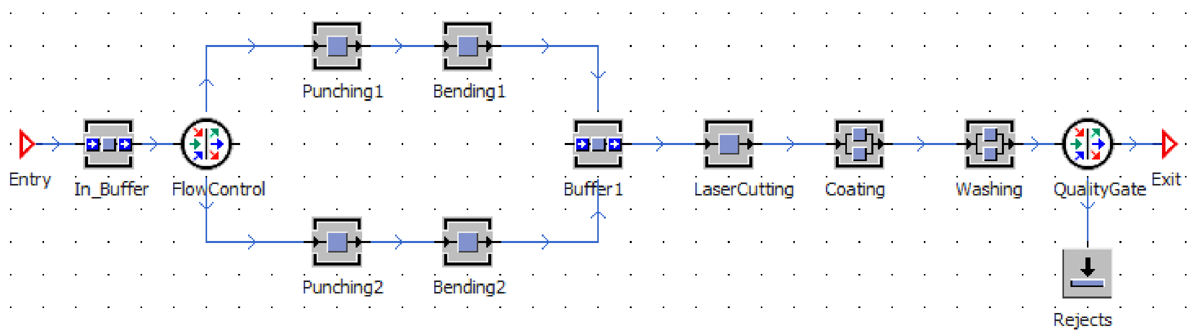

Figure 16: Objects in the Manufacturing frame

3. Double-click on each object to open the dialog properties. Rename the objects as shown in the figure and modify the attributes according to the following parameters:

    - Buffer (all) Capacity: *8*, Buffer type: *Queue*
    - Punching Processing time: *3:00*
    - Bending Processing time: *3:00*
    - LaserCutting Processing time: *1:00*
    - Coating X-dimension: *2*, Y-dimension: *2*, Processing time: *6:00*
    - Washing X-dimension: *1*, Y-dimension: *2*, Processing time: *2:00*

4. The default exit strategy of the **FlowControl** is cyclic, which means that the MUs will be passed on cyclically to all successors. However, the *QualityGate* should only pass on the good parts and reject the bad parts, e.g. randomly based on a distribution function.

5. Double-click the QualityGate and select “*Percentage*” from the dropdown menu. Click on **Apply**. The button *Open List* should appear. Click on this button and change the distribution percentage according to the figure below. Make sure that the successor 1 is the *Exit* interface and successor 2 is the *Reject* drain. Click on **Apply** and **OK**.

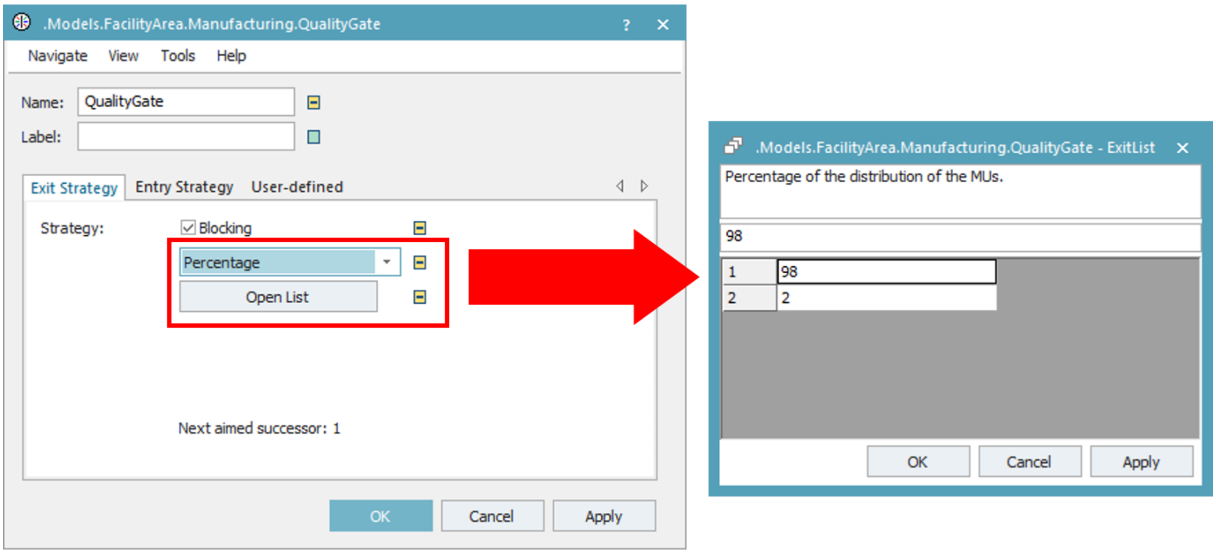

Figure 17: Exit strategy of the FlowControl

Download: The final state of the simulation model up to this exercise can be downloaded here Tutorial_Model_02.spp.

Exercise: Length-oriented objects

In this exercise, we will extend the previous simulation and model the assembly area using point-oriented and length-oriented material flow objects. Similar to the previous exercise, it is highly recommended that you only change the Assembly frame in the folder FacilityArea in the Class Library.

Step-by-Step:

1. In the *.Models.FacilityArea.Assembly* frame, insert the material flow objects and rename them as shown in the figure below. Except for the Pallet_Source, all Source objects produce the standard Entities. The MUs for the **Pallet_Source** should be of the class *Container* MUs.

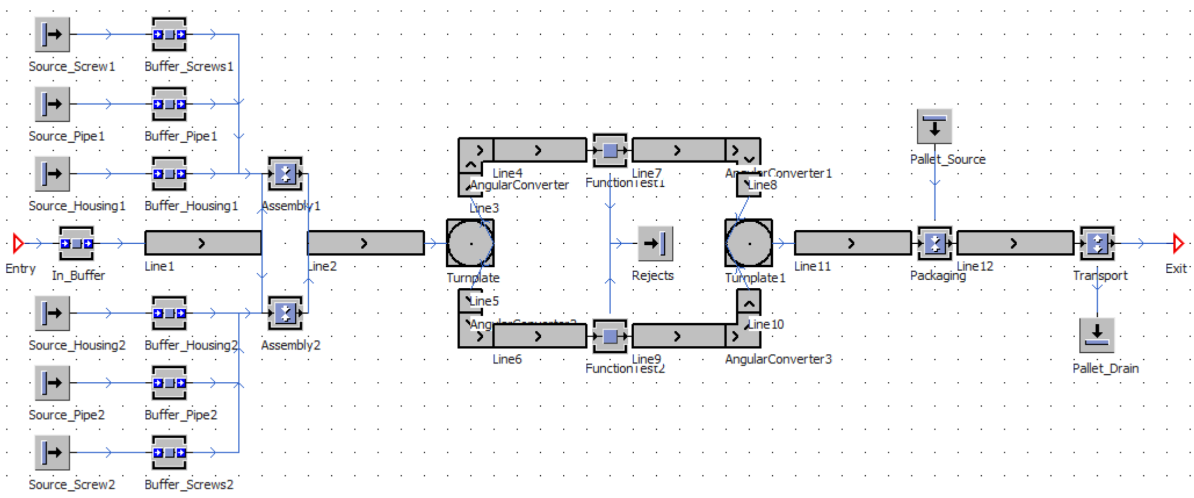

Figure 18: Assembly frame

2. There are two ways to insert the length-oriented objects (*Line*, *AngularConverter*, *Turnplate*) into the model:

   - Left-click on the object in the Toolbox. Left-click on the designated location in the frame to insert the starting point. The cursor will change into drawing mode and show the angle and length while you move the mouse. Left-click again on the frame to insert the next arc point. Press the Esc-key to the drawing mode.

   - Drag and drop the object from the Toolbox into the frame. It will be inserted with its default properties. You can edit the properties afterward using the dialog property window.

3. Double-click on the *Line1* to open its dialog property window. In the Attributes tab, change the Length to "*5*" and uncheck the box of the attribute "*Accumulating*". This means that all succeeding MUs will stop moving when other MUs in front of them cannot exit the *Line*. Repeat this step for *Line2*.
4. For length-oriented objects that consist of multiple segments, you need to define the properties for each segment, i.e. its length and position. To edit the properties, navigate to the Curve tab and click on the **Segments** button. A new window will show the *Tangential angle [°]* and the *Length [m]* for each segment.
5. To add a new segment, press the Enter-key on the last row to create a new row. Enter the tangential angle and the length of the new segment and click **Apply/OK**. The figure below shows the properties of the AngularConverter.

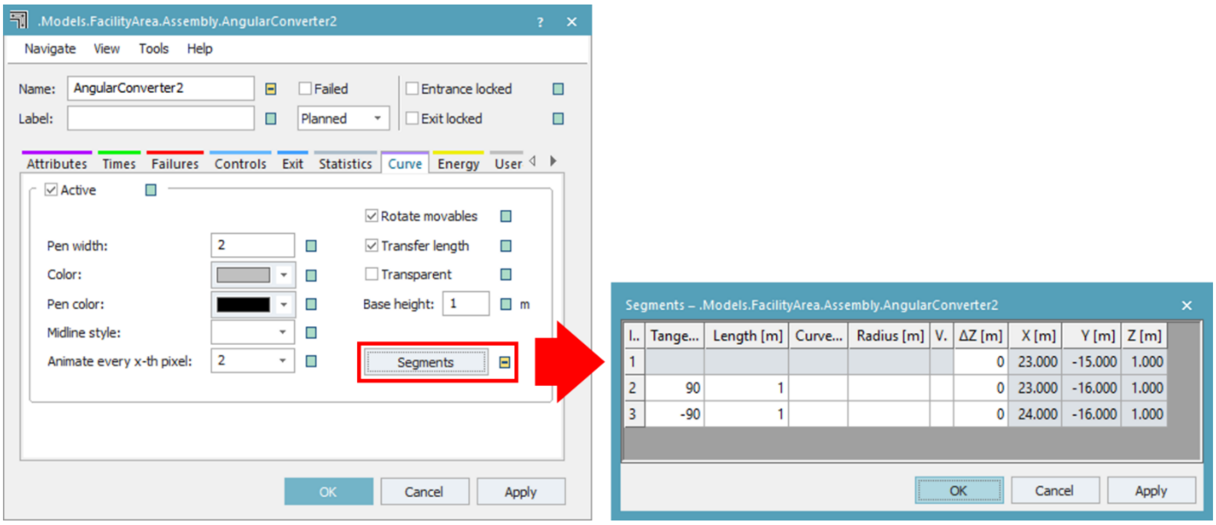

Figure 19: Edit segment in length-oriented objects

6. Connect the objects using **Connector**:
   - For each **Assembly** station, the order of the connectors from its predecessors should be (1) predecessor *Line*, (2) *Buffer_Housing*, (3) *Buffer_Pipe*, (4) *Buffer_Screws*
   - For each **FunctionTest** station, the order of the connectors to its successors should be (1) predecessor *Line*, (2) *Rejects* drain.
   - For the **Packaging** station, the order of the connectors from its predecessors should be (1) *Pallet_Source*, (2) predecessor *Line*.
   - For the **Transport** station, the order of the connectors to its successors should be (1) *Exit* interface, (2) *Pallet_Drain*.
7. Double-click on the **Assembly** station. Choose "*Predecessors"* from the dropdown menu of the *Assembly table*. Click on the **Open** button. Enter the predecessor connector number and the number of parts that the Assembly station should take

from its predecessors. The Assembly station should take each one Housing, one Pipe, and four Screws from the corresponding Buffer.

8. Select *"New MU"* from the dropdown menu of the *Exiting MU* option. Click on the button, choose **Select Object**, and select the *Radiator* entity. The MUs exiting the Assembly station will now be the final product.

9. Navigate to the Times tab. Set the processing time to "*3:30*".

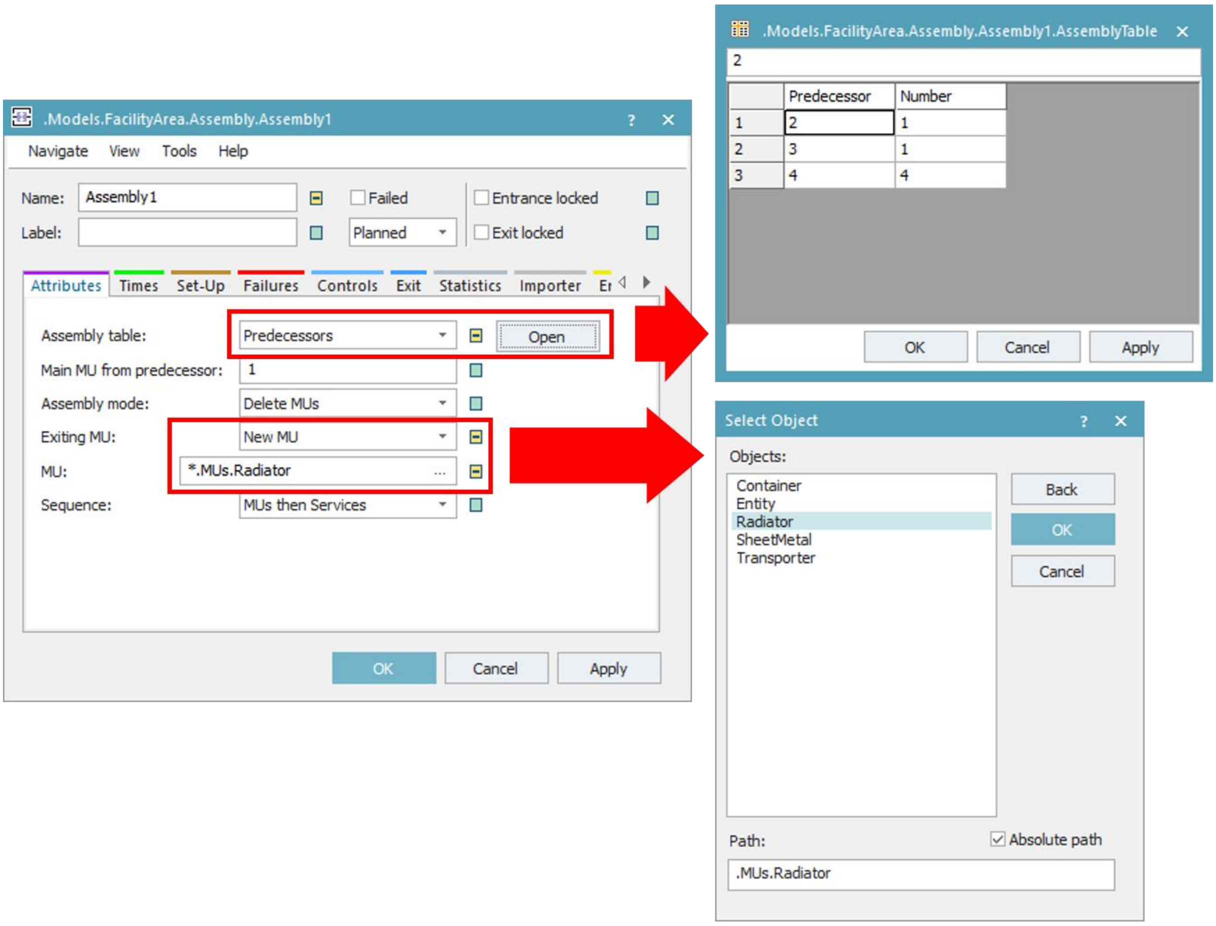

Figure 20: Dialog properties of the Assembly station

10. Double-click on the **FunctionTest** station. Set the processing time to "*3:30*". Navigate to the Exit tab and change the strategy to "*Percentage*". Modify the list so that 98% of the parts go to the succeeding line and 2% of the parts are rejected.

11. Double-click on the **Packaging** station. We want to load the finished products onto pallets. Therefore, the main MU should be the *Container* from the *Pallet_Source* and the *Assembly mode* should be set to *"Attach MUs"*. Each pallet can load two of the radiators. Change the *Assembly table* to predecessors and enter the required amount of the radiators. Finally, navigate to the Times tab and set the processing time to *"2:30"*.

12. Double-click on the **Transport** station. This is a dismantle station, where the radiators are sent to shipment and separated from the pallets. Set the *Dismantle mode* to

"Detach MUs" and enter the correct number of the successor. The radiators should be passed on to the *Exit* interface, while the containers are sent to the *Pallet_Drain*. Finally, navigate to the Times tab and set the processing time to *"1:00"*.

Download: The final state of the simulation model up to this exercise can be downloaded here Tutorial_Model_03.spp.

Running the simulation model

The **EventController** runs the simulation model and manages the events during simulation run. By default, the root frame that is created automatically when you start a new model will include an EventController by default. You can start a simulation by double-clicking on the EventController object in the frame or in the *Home* ribbon tab and pressing the **Start/Stop** button.

During a simulation run, Plant Simulation will show the animation of the material flow. Material flow objects also show their current state using LED icons in different colors on top of their default icon:

- **Red** Object is failed
- **Blue** Object is paused
- **Green** Object is working
- **Yellow** Object is blocked
- **Brown** Object is setup
- **Cyan** Object is in recovery
- **Orange** Object is waiting for resources

These animations can lead to a slower execution speed in a large model. Click on the icon **MUs** on the *Home* ribbon tab to switch on/off the animation of MUs during simulation run. Click on the icon **Icons** on the *Home* ribbon tab to switch on/off the animation of LED icons.

Exercise: Simulation run & model presentation

In this exercise, you will set the Start and End time of the simulation and run the simulation that you created. Finally, you will modify the presentation of the frames so that you can see the complete facility on one frame.

Step-by-Step:

1. Open the *.Models.Plant* frame. Double-click the **EventController** object in the frame or click on the EventController icon on the Home ribbon tab.
2. Navigate to the Settings tab. Enter value on the field *End* so that the simulation runs exactly for one whole day. The time format is *DDD:HH:MM:SS.XX*. Check the box *"Show summary report"* to automatically create a summary report after the simulation run. Click on **Apply**.
3. Navigate to the Control tab and press the **Start/Stop** button to start the simulation. Watch the simulation run and view the summary report afterward.

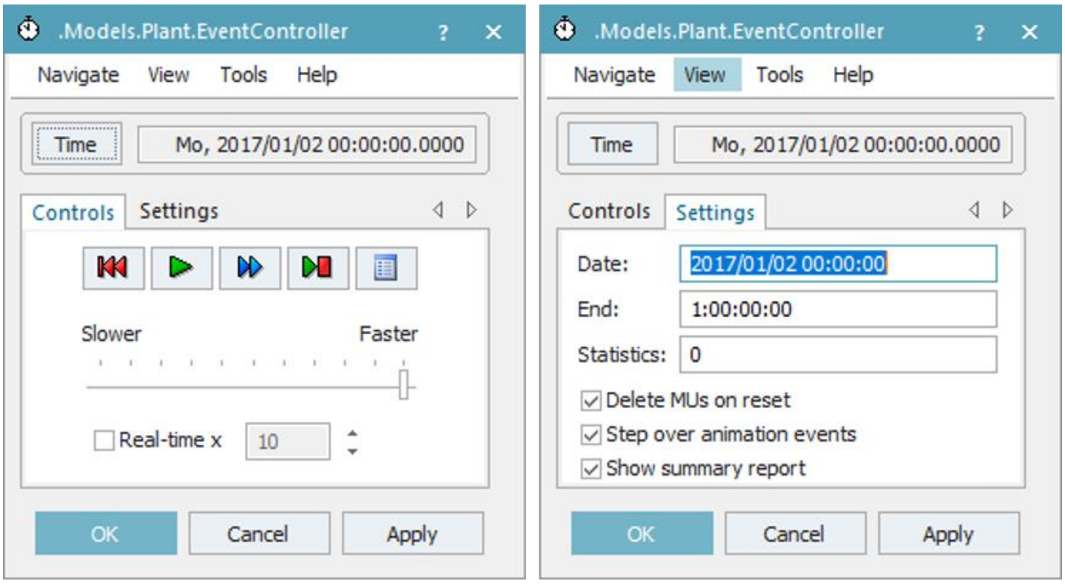

Figure 21: EventController

4. If you double-click on the Manufacturing frame and the Assembly frame inside of the Plant frame, you can view the animation of the MUs. However, you can only see the animation for each frame separately this way and not the animation of the whole process.
5. Open the **Manufacturing** frame. Click on the icon **Representation**  from the *General* ribbon tab. Change the *Representation mode* to *"Contents"* and enter the value as shown in the figure below. Repeat the steps in the **Assembly** frame.

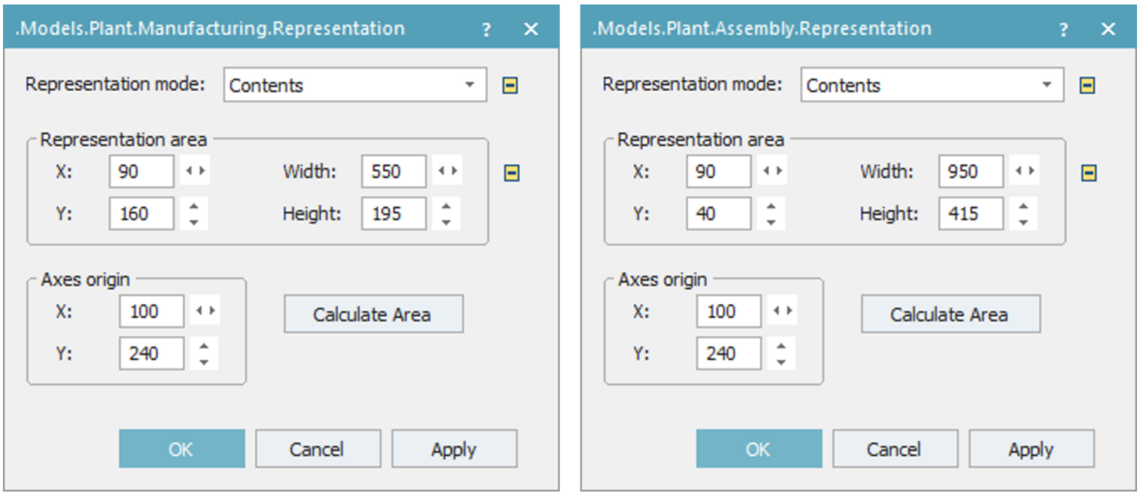

Figure 22: Dialog properties for representation of the Manufacturing frame (left) and Assembly frame (right)

6. Navigate back to the *Plant* frame. You will see that both frames are now showing their content in this frame. You might need to adjust the position of the Source and Drain so that the connection between them is shown correctly. **Reset** the simulation and **Start** the simulation again.

Download: The final state of the simulation model up to this exercise can be downloaded here Tutorial_Model_04.spp. Your simulation model should look similar to the figure below.

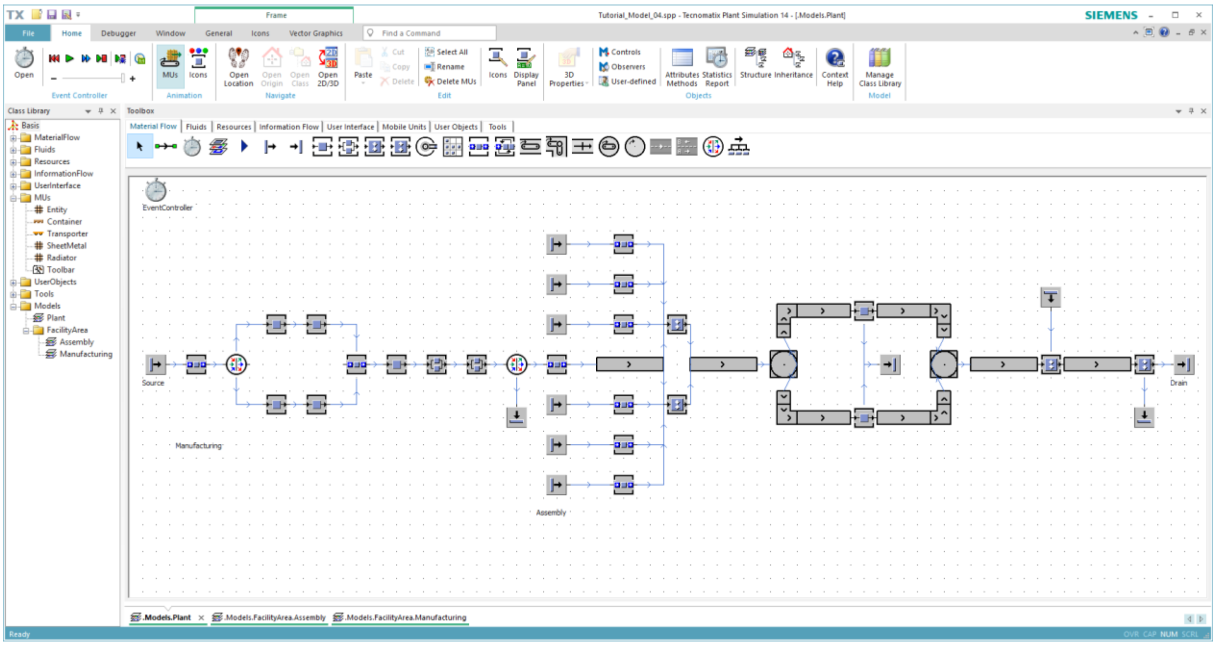

Figure 23: Final model of this chapter

previous chapter | next chapter

Chapter 4: Modeling in Plant Simulation – Resources

This chapter will introduce you to the modeling of resource objects in Plant Simulation. After completing this chapter, you will be able to:

- Understand resource objects in Plant Simulation
- Extend your model with workers and resource objects

Modeling workers and resources

In the previous chapter, you have created the processing stations of your production facility. Plant Simulation provides additional resource objects to integrate human factors and shift calendars to the processing stations.

Worker represents a person who works at a processing station, such as *SingleProc*, *ParallelProc*, *Assembly*, or *DismantleStation*. Workers can provide different types of services and have different parameters, such as efficiency and speed, to determine their behavior.

Worker can only work at a processing station if there is a **Workplace** attached to this station, which is not already occupied by another worker.

WorkerPool introduces *Worker* to the simulation. It creates the number of *Worker* instances with given parameters according to the creation table. Every *Worker* starts from the WorkerPool. They return to the WorkerPool during shift breaks and at the end of the shift, and remain in there when they do not have any work assigned. The WorkerPool also determines how workers travel between the WorkerPool and the assigned station. Workers can either move freely within the area, walk along footpaths, or teleport directly to the workplace.

FootPath connects the *WorkerPool* to workplaces. It is a length-oriented object, which means that the length of the footpath and the speed of the worker will be used to determine the required travel time during the simulation.

Broker manages the service requests of all processing stations and the provided services of the workers. If the provided service of a worker matches the requested service of a processing station, the Broker assigns the worker to this station and triggers the event for the worker to leave his WorkerPool and go to the requested station.

Shift calendar is used to model work shifts in your plant. You can define as many shifts as you need and set individual break times for each shift. You can also define exceptions for days on which no or reduced work is done in your plant, e.g. on a national holiday.

Exercise: Resource objects

In this exercise, we will extend the previous simulation with resource objects. We want to model two types of *Worker*. Each will provide a different type of service and work either in the manufacturing or in the assembly area. We also want the workers to work in two shifts according to a *ShiftCalendar*. Similar to previous exercises, it is highly recommended that you only make changes in the frames in the folder *FacilityArea* in the Class Library.

Step-by-Step:

1. Expand the *Resource* folder in the Class Library. Right-click on the *Worker* class and select **Derive** from the context menu.
2. Double-click on the derived Worker class and rename it to *"WorkerManufacturing"*. Deactivate the inheritance of the provided services. Click on the **Services** button and add a new service type *"Manufacturing"* as shown in the figure below.
3. Repeat the previous steps to derive a new class *"WorkerAssembly"* from the *Worker* class. Change the attributes so that the *WorkerAssembly* class will provide a new service type *"Assembly"*.

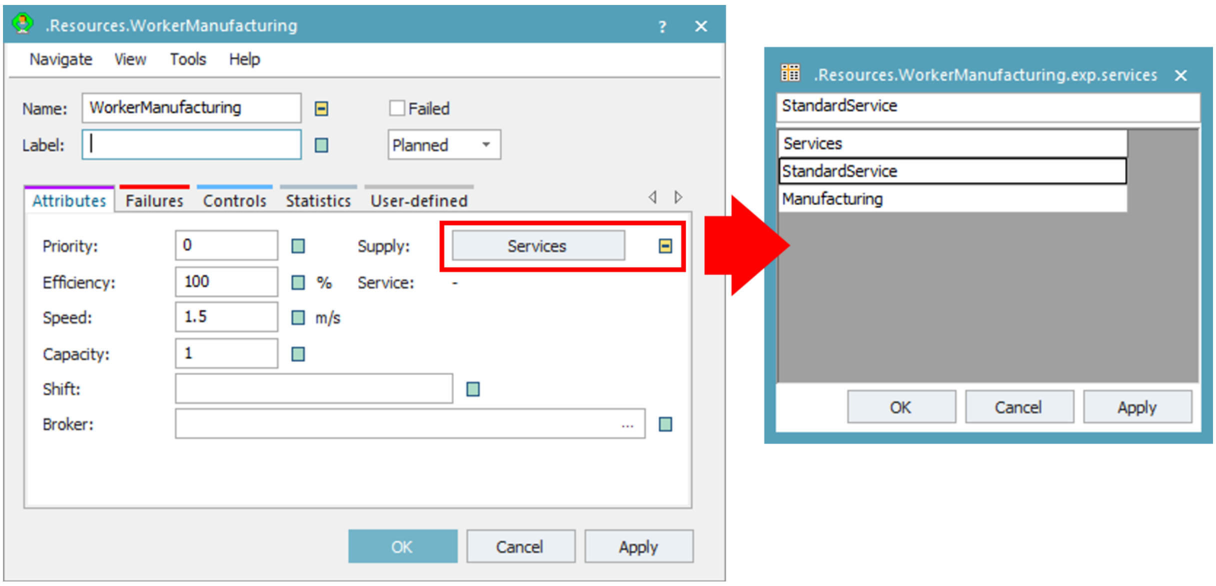

Figure 24: Services dialog properties of the WorkerManufacturing class

4. Open the *Manufacturing* frame under the folder *FacilityArea*. Navigate to the Resource tab in the Toolbox. Drag & drop the **Workplace** object from the Toolbox

onto the frame. We want each single processing station to have one workplace and each parallel processing station to have two workplaces.

Tips: If you place a *Workplace* near a processing station when you use drag & drop in the frame, it will be linked automatically to the processing station. Alternatively, you can also drag & drop the *Workplace* onto the processing station. Check the Attributes tab of the *Workplace* to make sure that the workplace is linked to the correct station.

5. Insert a **WorkerPool**, **ShiftCalendar**, and **Broker** from the Toolbox into the frame.

6. Draw **FootPath** to connect the *WorkerPool* to each *Workplace*. The beginning and the end of each footpath is connected by *Connectors*. This way, you can connect a footpath to several workplaces as well as to several other footpaths. The finished model in your Manufacturing frame should look similar to the figure below.

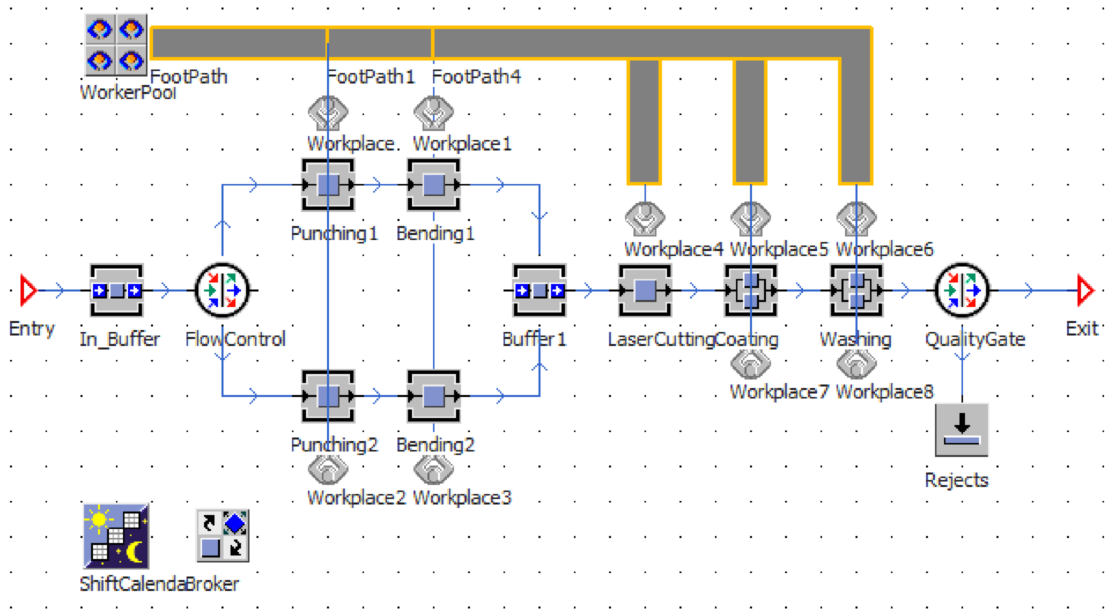

Figure 25: Extended model of the Manufacturing frame

7. Drag & drop the **ShiftCalendar** and the **Broker** onto the **WorkerPool** instance in the frame to link them.

8. Double-click on the *WorkerPool*. In the Attributes tab, the *WorkerPool* should automatically refer to the assigned *Broker* and *ShiftCalendar*.

9. Deactivate the *Workers* inheritance and click on the button **Creation Table**. Drag & drop the **WorkerManufacturing** class from the Class Library onto the row below the *Worker* column. Set the values in the creation table as shown in the figure below. The *WorkerPool* will have 9 manufacturing workers for each shift with reduced efficiency in the second shift.

Tips: The *ShiftCalendar* creates two shifts by default. Make sure that the shift name in the *WorkerPool* matches exactly with the shift name defined in the *ShiftCalendar*. Double-click on the *ShiftCalendar* to see the defined shift times.

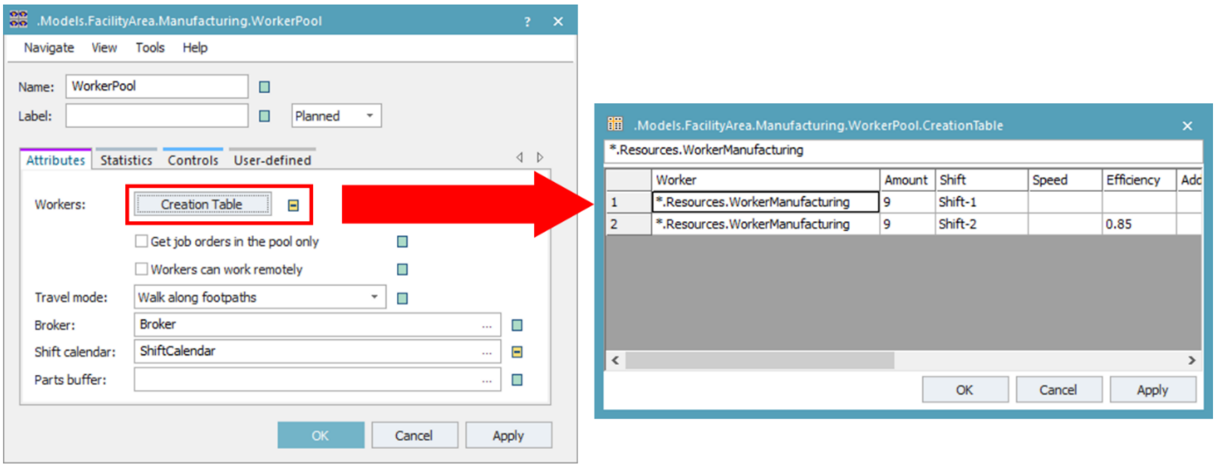

Figure 26: WorkerPool dialog properties

10. Now you will need to link the processing stations to the *Broker* so that they are able to send service requests. Drag & drop the **Broker** instance onto each processing station in the frame.
11. Double-click on a processing station, e.g. **Punching1** station. Navigate to the <u>Importer</u> tab. You should see here that the correct *Broker* is assigned to this station.
12. Set the Importer to be *Active*. Deactivate the service inheritance and click on the **Services** button. Set the required service to *"Manufacturing"* and enter the required amount, as shown in the figure below. Repeat this step for each processing station in the Manufacturing frame. Note that the required amount must match the number of workplaces linked to this station.

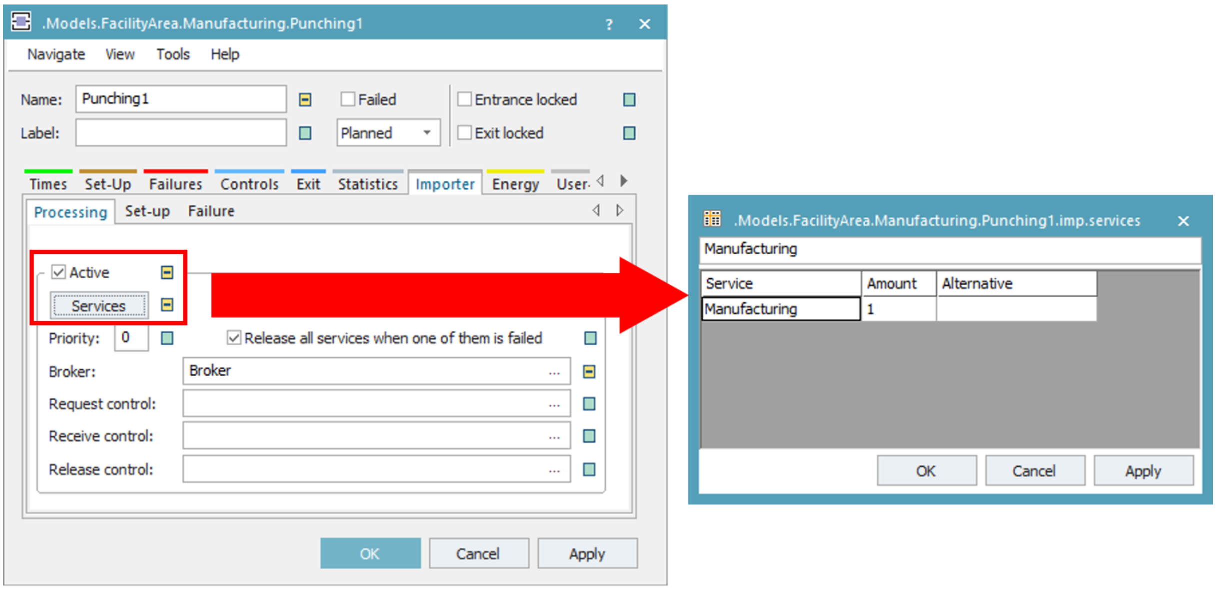

Figure 27: Importer for defining required services of a station

13. Repeat all previous steps to create the **WorkerPool**, **Broker**, **ShiftCalendar,** and **Workplaces** in the *Assembly* frame. The following parameters differ from the *Manufacturing* frame:

    - The **Packaging** station should have two workplaces. The **Transport** station does not need any workplace. All other stations should have one workplace each.
    - Use the **WorkerAssembly** class for the creation table in the WorkerPool. Set the number of workers to *"6"* for each shift.
    - Set the travel mode to *"Move freely within area"*. You will not need to draw any footpath in the assembly area.
    - Use the **Assembly** service for the Importer for each processing station.

14. Open the **Plant** frame and run the simulation. Did you see your workers moving? If you open the summary report after the simulation, did you notice any difference to the previous simulation run?

Download: The final state of the simulation model up to this exercise can be downloaded here Tutorial_Model_05.spp. Your simulation model should look similar to the figure below.

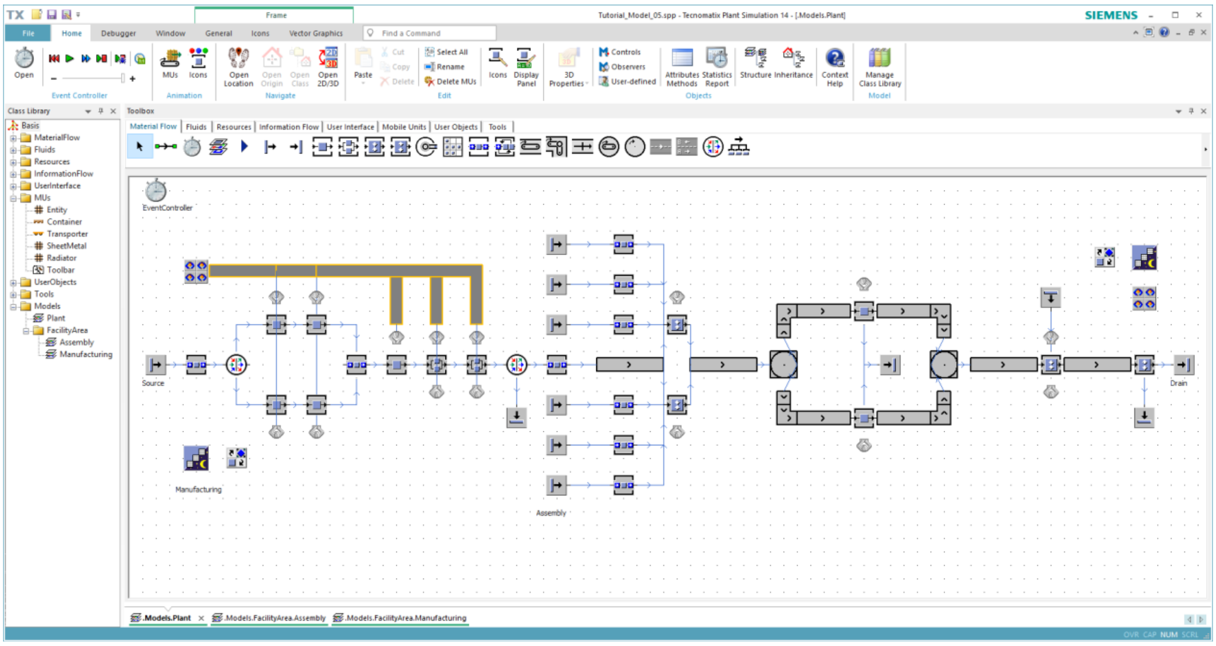

Figure 28: Final model of this chapter

previous chapter | next chapter

Chapter 5: Modeling in Plant Simulation – Information Flow

This chapter will introduce you to the information flow objects and user-defined controls in Plant Simulation. After completing this chapter, you will be able to:

- Store information in table files and use them in your simulation
- Understand the Plant Simulation programming language SimTalk
- Extend your model with user-defined control functions

Table files

Plant Simulation provides multiple list and table objects to store information. They can be used to provide the simulation with input data which will be used during the simulation runs, as well as to record the simulation results.

TableFile is a list with multiple columns. Each individual cell in the table is indexed by its row and column number and can be accessed individually. Additionally, a table in a TableFile can contain another table.

CardFile, **QueueFile**, and **StackFile** are similar to the *TableFile*. However, they only consist of one column.

Exercise: Set MU attribute from TableFile

In this exercise, we will extend the previous simulation with *TableFile*. We will define three different types of MUs in a TableFile and link the Source to the TableFile so that it will produce the MUs according to the attributes defined in the table. Similar to previous exercises, it is highly recommended that you only make changes in the frames in the folder *FacilityArea* in the Class Library.

Step-by-Step:

1. Open the *Plant* frame. Navigate to the Information Flow tab in the Toolbox. Insert a *TableFile* object from the Toolbox into the frame next to the *Source* and rename it to *“SourceTable”*.

2. If you double-click on the *SourceTable*, you will see that each column currently has the same data type. You can show/hide the data type of the columns and rows using the icon **Data Type** from the *List* ribbon tab.
3. In the *Plant* frame, drag & drop the **SourceTable** onto the **Source** to link them.
4. Double-click on the **Source** to open it. In the Attributes tab, you will see that instead of referring to the MUs directly, the Source now creates the MUs by referring to the *SourceTable*. Change the *MU selection* to *"Percentage"*.
5. Double-click on the **SourceTable** to open it. You will notice that the structure of the table has changed. When a *TableFile* is linked to another object, the structure of the table and its data types will be automatically formatted according to the linked object.
6. Drag & drop the *SheetMetal* entity from the folder *MUs* in the Class Library onto the rows below the first column. Edit the values of each row in the columns *Portion*, *Number*, and *Name* as shown in the figure below. The *Source* should produce in 50% of the time one *RawPartA*, 30% one *RawPartB*, and 20% one *RawPartC*.
7. For each row of the *Attributes* column, set the value to *"x"*. This will automatically create a sub-table for each row where you can further specify the MU attributes. Set the *Name of Attribute* to *"Type"* and enter the corresponding attribute of the string type in the fourth column.

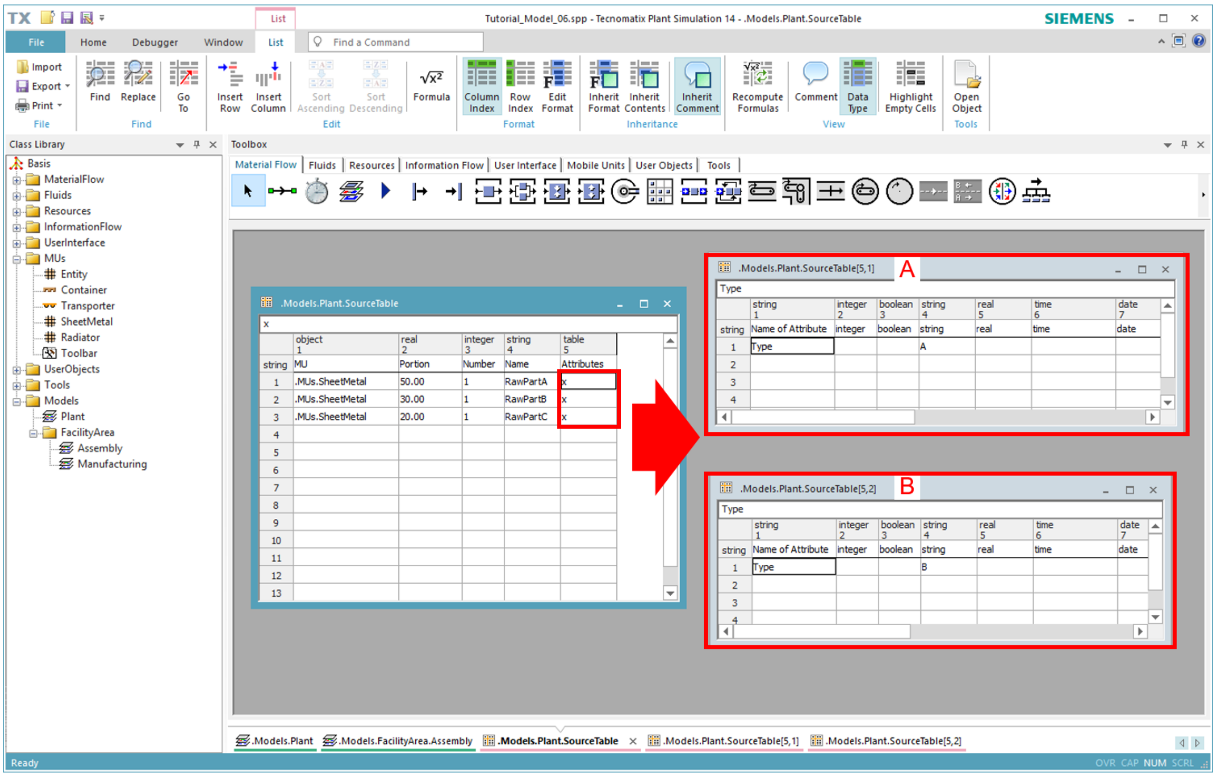

Figure 29: TableFile for defining MU attributes

8. Open the *Assembly* frame in the folder *FacilityArea* in the Class Library. Double-click on the first *Line* which transports the parts from the inbound buffer to the assembly stations.
9. Navigate to the Exit tab. Change the strategy to *"MU Attribute"* and click **Apply**. You will be able to further specify the attribute type using a list. Set the *Attribute type* to *"String"* and click **Apply**.
10. Click on the button **Open List**. We want to distribute parts with the attribute type "A" to the first succeeding station (*Assembly1*) and parts with the attribute type *"B"* or *"C"* to the second succeeding station (*Assembly2*). Enter the attribute *"Type"* in the table and set the values according to the figure below. Click on **Apply**/**OK**.
11. Go back to the *Plant* frame and run the simulation. Did you notice any difference?

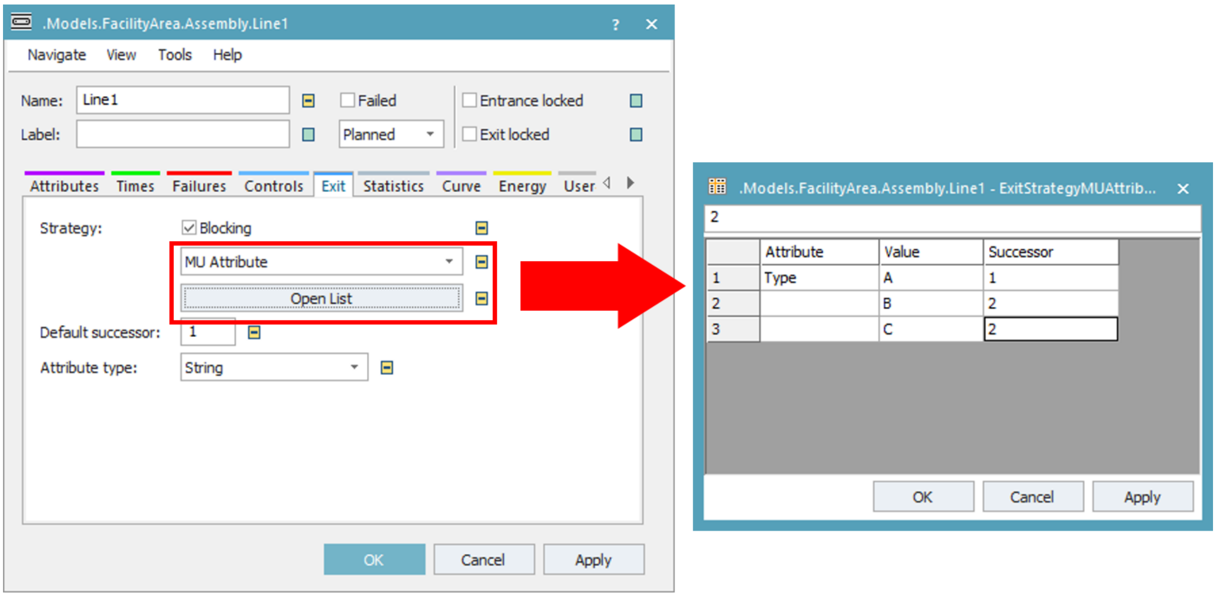

Figure 30: Exit strategy based on MU Attribute

Download: The final state of the simulation model up to this exercise can be downloaded here Tutorial_Model_06.spp.

Methods

Method represents a control object which is executed during the simulation run. You can program a specific action or define an individual behavior using *Method*. *Method* can also be built-in in material flow objects as an entrance or exit control. In this case, the action will be executed each time an MU enters or leaves the material flow object.

There are also special types of methods which are called by the *EventController*. They are indicated with a special icon:

- **Init Method** is executed when you start the simulation model after resetting it.
- **EndSim Method** is executed at the end of a simulation run.
- **Reset Method** is executed when you click on the reset simulation button in the *EventController*.

Method contains source code written in the **SimTalk** programming language. The structure of a Method generally consists of parameters declaration, followed by the data type of the return value, local variables, and source code of the action itself.

You can write the code syntax in the **SimTalk 1.0** syntax or in the newer **SimTalk 2.0** syntax. One model can contain several methods with either SimTalk 1.0 or SimTalk 2.0, as they can run parallel. It is also possible to convert methods written in the old syntax to the new syntax. In SimTalk 2.0, the is-do-end constructor is obsolete. This results in a leaner and simpler code. The figure below shows an example of the notation in SimTalk 1.0 and SimTalk 2.0.

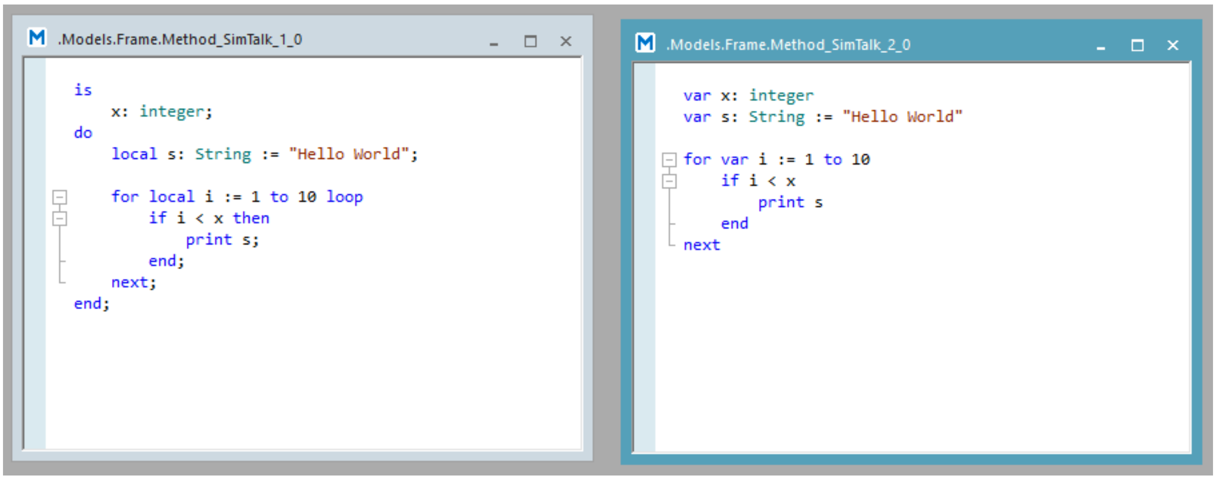

Figure 31: Notation of SimTalk 1.0 (left) vs. SimTalk 2.0 (right)

SimTalk consists of built-in methods and standard constructors, such as loops and conditional branching. To refer to an object in the simulation or query the path to an object, SimTalk provides the following anonymous identifiers:

- **"@"** refers to the MU that triggered the control. This may be the MU that entered the object or the MU that is ready to exit the object.
- ***"root"*** refers to the topmost (root) frame in the model frame hierarchy.
- ***"current"*** refers to the current frame in which the called Method object is located.
- ***"?"*** refers to the material flow object or the Method that triggered the control.
- ***"self"*** refers to the Method that is currently executed.

Exercise: Set MU attribute using Method

In this exercise, we will extend the previous simulation and model flow control based on MU attributes created using a *Method*. In the previous exercise, we modeled the *QualityGate* in the manufacturing area to sort the bad parts from the good parts and to only pass on the good parts. Instead of using the fixed percentage in the *QualityGate*, now we want to randomly assign the attribute whether an MU fulfills the quality requirement and make the decision based on the current attribute of the MU.

Similar to previous exercises, it is highly recommended that you only make changes in the frames in the folder *FacilityArea* in the Class Library.

Step-by-Step:

1. Open the *Manufacturing* frame under the folder *FacilityArea* in the Class Library. Navigate to the Information Flow tab in the Toolbox. Insert a *Method* object from the Toolbox into the frame and rename it to *"CheckPartOK"*.
2. Drag & drop the **CheckPartOK** method onto the **Washing** station. The icon of this *Method* should change to the icon of an exit control *Method*. Double-click on the *Washing* station and navigate to the Controls tab. Make sure that the *CheckPartOK* method is entered correctly in the *Exit* field.
3. Double-click on the **CheckPartOK** method and enter the following SimTalk source code, as shown in the figure below. The *Method* will be executed each time an MU leaves the station.

.Models.FacilityArea.Manufacturing.CheckPartOK

```
-- Create a local random variable with the value between 0 and 1 using uniform distribution
var randomValue : real := z_uniform(1, 0, 1)

-- Add a new MU attribute "PartOK" of boolean
@.createAttr("PartOK", "boolean")

-- Set the value for the attribute "PartOK" based on the randomValue
if randomValue < 0.05
    @.PartOK := false
else
    @.PartOK := true
end

-- Finally, move the finished MU to the succeding station
@.move

-- Print the assigned randomValue in the console window (optional)
print "Current part: ", @.Name, ", Value: ", randomValue, ", Part is Ok: ", @.PartOK
```

Figure 32: Source code of the exit control method

4. Double-click on the **QualityGate** flow control object. Navigate to the Exit Strategy tab. Change the strategy to *"MU Attribute"* and click **Apply**. Set the *Attribute type* to *"Boolean"* and click **Apply**.

5. Click on the button **Open List**. We want to distribute parts of which the attribute *"PartOK"* is *true* to the succeeding assembly area and parts of which the attribute *"PartOK"* is *false* to the *Rejects* drain. Enter the values in the table as shown in the figure below. Click on **Apply/OK**.

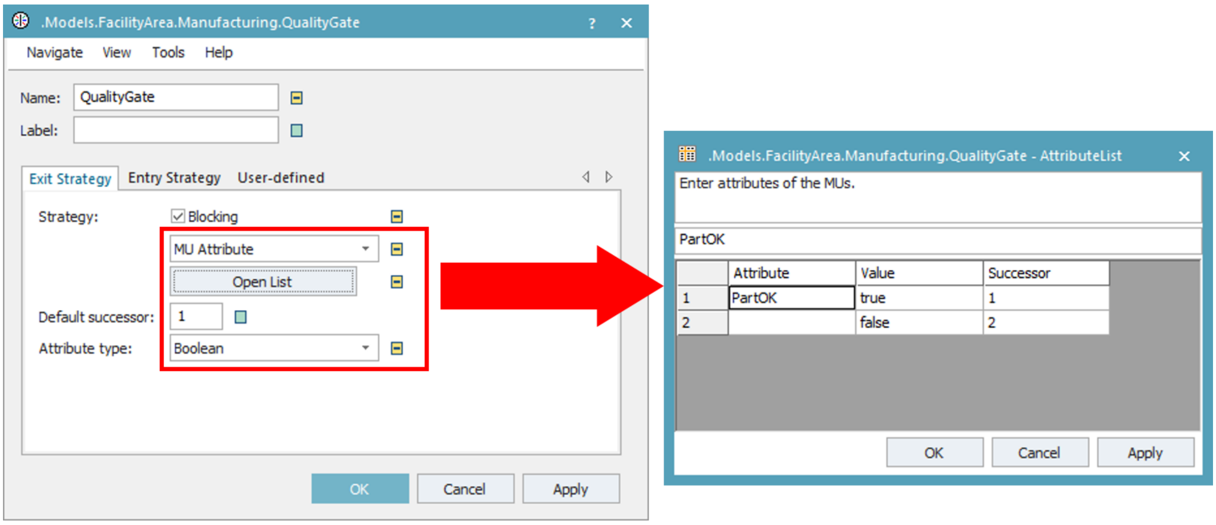

Figure 33: Exit strategy of the FlowControl

6. Double-click on the **QualityGate** flow control object. Navigate to the Exit Strategy tab. Change the strategy to *"MU Attribute"* and click **Apply**. Set the *Attribute type* to *"Boolean"* and click **Apply**.

7. Open the *Plant* frame and run the simulation. Did you notice any difference to previous simulation runs?

8. Try to experiment with different threshold values for the if-else-condition and see how it affects the simulation results. You can open the *Console* window to see which random values were generated by the simulation.

Download: The final state of the simulation model up to this exercise can be downloaded here Tutorial_Model_07.spp. Your simulation model should look similar to the figure below.

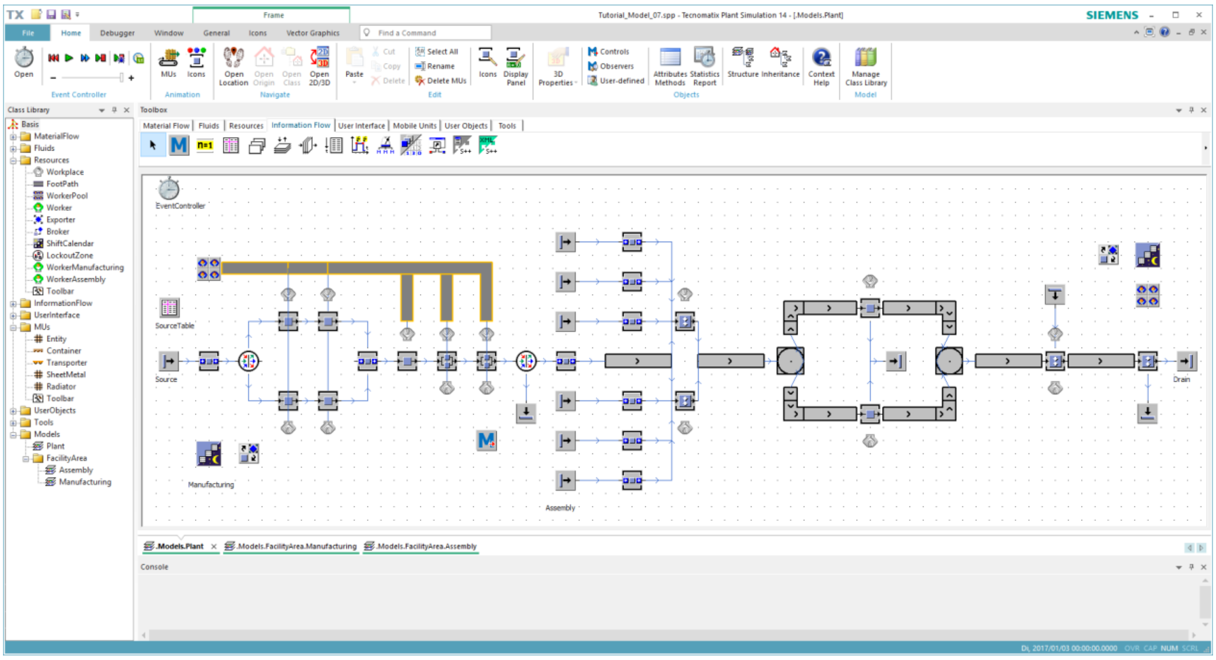

Figure 34: Final model of this chapter

previous chapter | next chapter

Chapter 6: Visualizing Simulation Results

This chapter will introduce you to user interface objects in Plant Simulation. After completing this chapter, you will be able to:

- Visualize resource statistics and buffer occupancy
- Visualize worker statistics

Charts

Plant Simulation provides different chart objects to visualize and analyze simulation results. Charts are updated during runtime. Therefore, if you view a chart during the simulation run, you will be able to see how the recorded values change over the course of the simulation.

Chart shows the statistics of data sets that were recorded during a simulation run. It collects the data either from a table file or directly from other objects defined in the input channels. There are three statistics types that can be displayed by the chart: **Resource Statistics**, **Energy Statistics**, and **Occupancy**.

The *Chart* object is also equipped with a *Statistics Wizard*, with which you can specify the classes of objects that you want to show their statistics values. Alternatively, if you drag & drop a material flow object onto the *Chart*, it will be stored in the input channel and used to record the statistics values.

WorkerChart shows utilization statistics of all workers located in a *WorkerPool*.

WorkerSankeyDiagramm visualizes the frequency of the paths taken by *Worker* that moves freely within the area. The flow of the *WorkerSankeyDiagram* is shown only in 3D.

Exercise: Resource utilization & buffer occupancy

In this exercise, we will visualize the resource utilization of all processing stations in our facility. We will also visualize the occupancy of the buffer between the *Bending* and *LaserCutting* station in the manufacturing area as well as the occupancy of the inbound buffer in the assembly area. Similar to previous exercises, it is highly recommended that you only make changes in the frames in the folder *FacilityArea* in the Class Library

Step-by-Step:

- Open the *Manufacturing* frame under the folder *FacilityArea* in the Class Library. Navigate to the User Interface tab in the Toolbox. Insert a *Chart* object from the Toolbox into the frame and rename it to *"ResourceChart"*.
- Drag & drop a processing station, e.g. the *Punching1* station onto the *ResourceChart*. A dialog window will appear where you can select the statistics type for configuring the chart. Select **Resource Statistics** from the dropdown menu and click **OK**.
- Right-click on the *ResourceChart* and select **Show** from the context menu. You should see the station in the x-axis. Drag & drop all other processing stations onto the *ResourceChart* to add them to the chart.
- Insert another Chart object from the Toolbox into the frame and rename it to *"BufferChart"*.
- Drag & drop the *Buffer1* onto the *BufferChart*. Select **Occupancy** from the dropdown menu in the *Statistics Type* dialog window and click **OK**. The icon of the *BufferChart* will automatically change.
- Repeat the same steps to create a *ResourceChart* and a *BufferChart* for the *Assembly* frame in the folder *FacilityArea*.
- Open the *Plant* frame and run the simulation. Decrease the speed and double-click either on the *Manufacturing* frame or the *Assembly* frame in the *Plant* frame to open. Right-click on the *Chart* objects and select **Show**. See how the statistics values change during the simulation. Compare the *BufferChart* of the *Manufacturing* frame with the *BufferChart* of the *Assembly* frame.

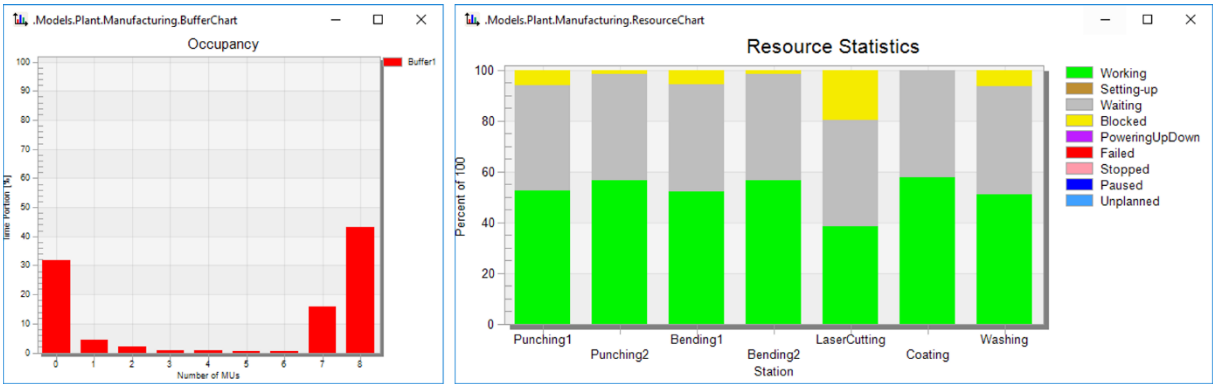

Figure 35: Buffer occupancy chart (left) and resource statistics chart (right)

Exercise: Worker statistics

In this exercise, we will visualize the worker statistics in the assembly area of our production facility. Similar to previous exercises, it is highly recommended that you only make changes in the frames in the folder *FacilityArea* in the Class Library.

Step-by-Step:

1. If the *WorkerChart* object is not present, click on the icon **Manage Class Library** from the *Home* ribbon tab. Navigate to the Libraries tab and check the box *Worker Chart* under the *Tools* group. Click **OK**.
2. Navigate to the Tools tab in the Toolbox. Now you will see the *WorkerChart* object here.
3. Open the *Assembly* frame under the folder *FacilityArea* in your Class Library. Navigate to the Tools tab in the Toolbox. Insert the *WorkerChart* from the Toolbox into the frame.
4. Drag & drop the **WorkerPool** onto the **WorkerChart** in the frame. A dialog window will appear. Select *"Group Workers by Individuals"* to show each individual Worker in the statistics and *"Occupancy with respect to Working time"*. Click on **Apply** & **OK**.
5. Open the *Plant* frame and run the simulation. Double-click on the *Assembly* frame in the *Plant* frame to open it. Right-click on the *WorkerChart* and select **View Chart**.

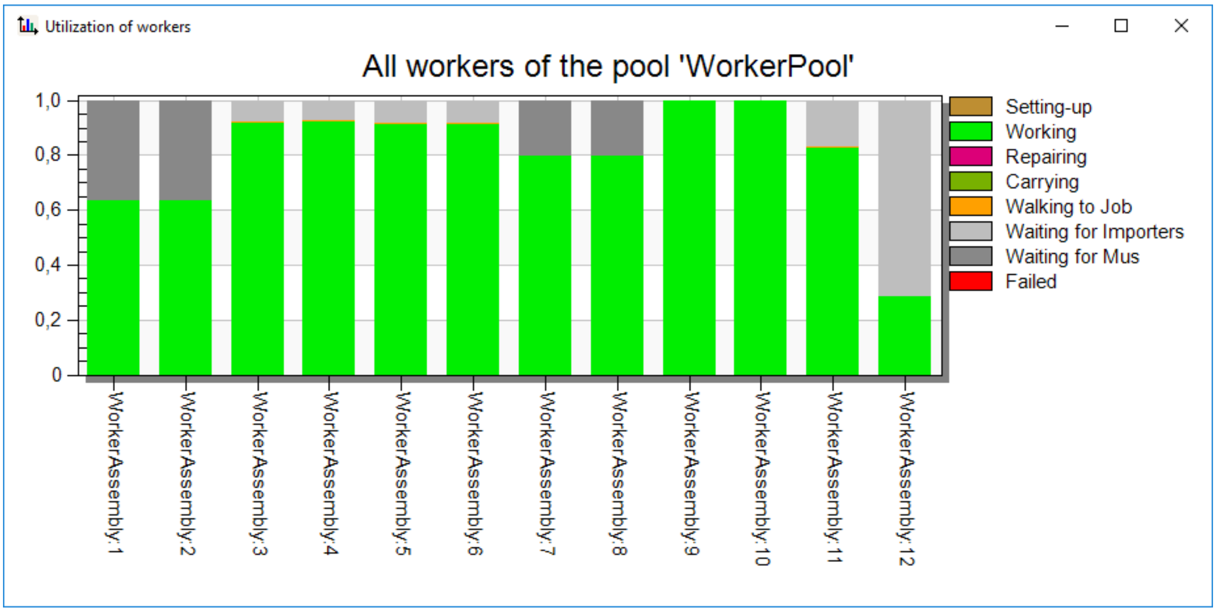

Figure 36: Visualization of worker utilization

Download: The final state of the simulation model up to this exercise can be downloaded here Tutorial_Model_08.spp. Your simulation model should look similar to the figure below.

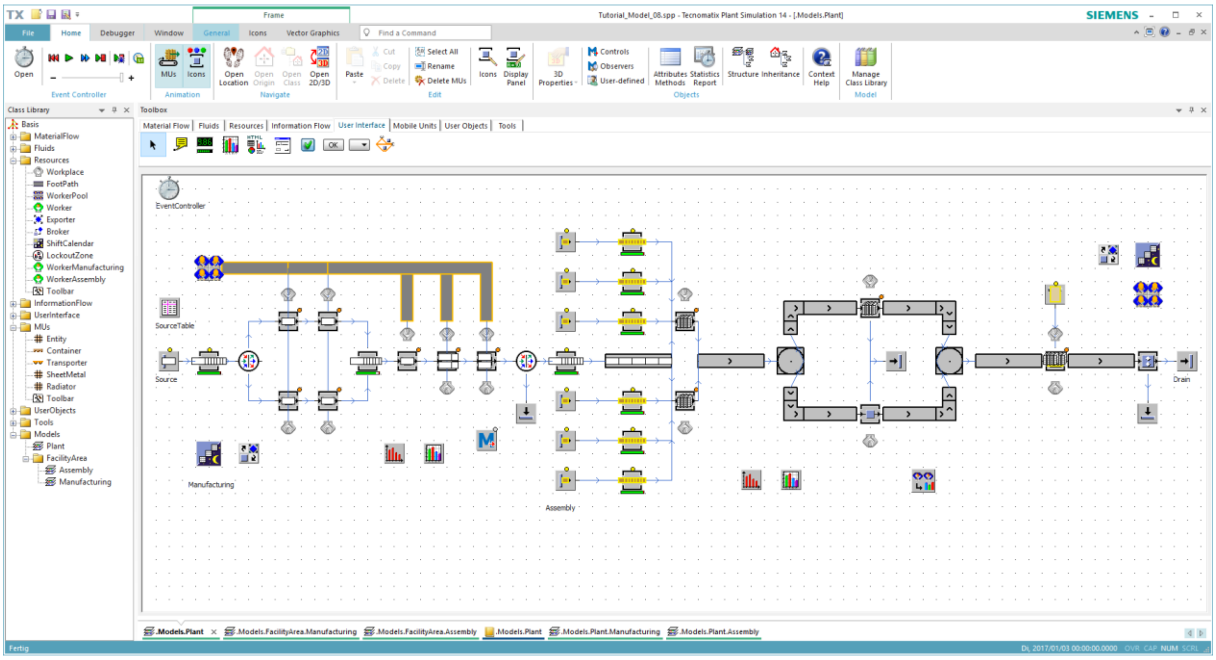

Figure 37: Final model of this chapter

previous chapter | next chapter

Chapter 7: Basics of 3D Modeling in Plant Simulation

This chapter will give you a short introduction to the basics of 3D modeling in Plant Simulation. After completing this chapter, you will be able to:

- View and run simulation in 3D
- Understand 3D object properties
- Create 3D models using standard library elements

3D modeling in Plant Simulation

Plant Simulation provides a **3D Viewer** to create and visualize simulation models in a three-dimensional space. When you create a new model, you can create your model directly in the 3D Viewer. Alternatively, you can also create your 2D simulation model first and activate the 3D visualization later on.

Exercise: 3D Model

In this exercise, we will run the simulation in the 3D Viewer and exchange a material flow object in 3D. Similar to previous exercises, it is highly recommended that you only make changes in the frames in the folder *FacilityArea* in the Class Library.

Step-by-Step:

1. Open the *Assembly* frame in the folder *FacilityArea*. Click on the **Open 2D/3D** icon from the Home ribbon tab. If this is your first time creating the 3D model, Plant Simulation will ask if you want to create the 3D model with default graphics. Click on **Yes**. Plant Simulation will open the 3D viewer of this frame.

2. If you click on the icon **Planning View** in the *View* ribbon tab of the *3D* ribbon group, Plant Simulation will change the view to the view from the top. This is especially helpful when you model length-oriented objects as you can edit them with more precise using the grids.

3. If you set the *Worker* to move freely within the area, Plant Simulation will use the material flow objects as obstacles for the workers. You can click on the icon

Obstacles in the *View* ribbon tab to display these worker obstacles around the material flow objects, as shown in the figure below.

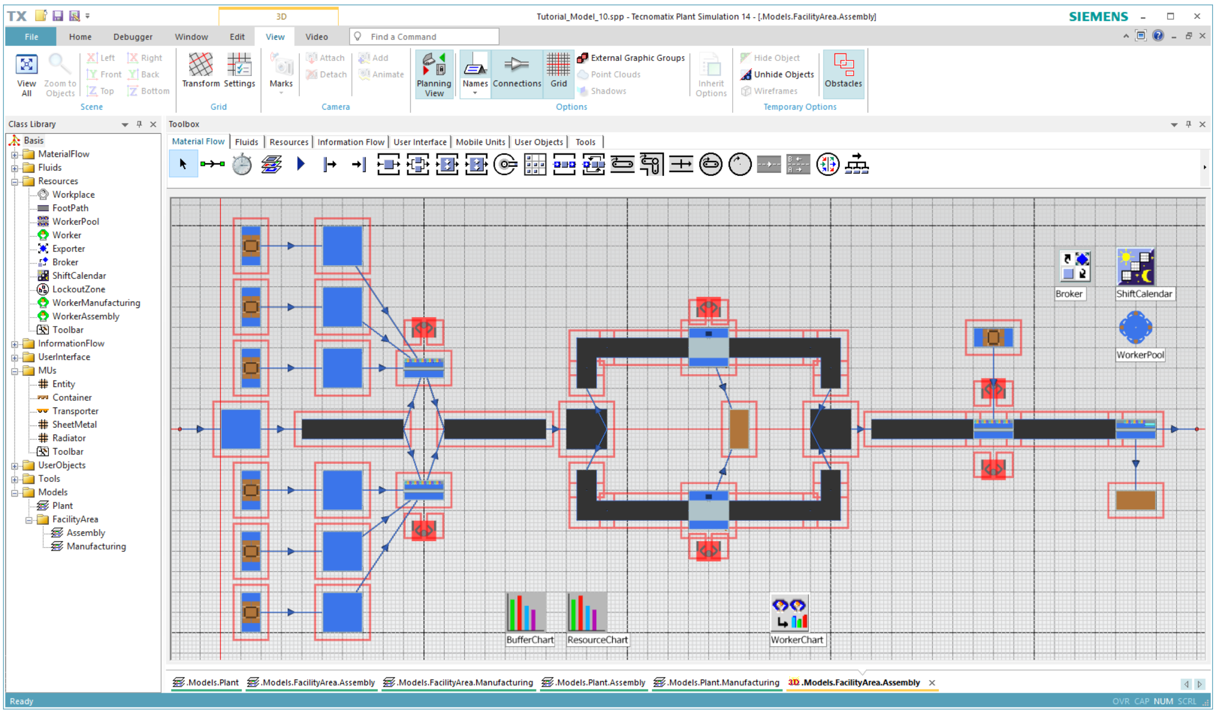

Figure 38: Planning view of the assembly frame with worker obstacles shown

4. Now we want to replace the first *Turnplate* with a *PickAndPlace* object. This can be done in the 2D viewer as well as in the 3D viewer. Hold the **Alt-key** and drag & drop the **PickAndPlace** from the Toolbox onto the **Turnplate** in the frame to replace it. Click on **Yes** to copy all attributes in the appearing dialog window.
5. Deactivate the *Planning View*. You will see in the 3D viewer that a *PickAndPlace* robot is placed in place of the *Turnplate*. You can see this change in the 2D viewer as well.
6. If you want to use another 3D model instead, right-click on the object and select **Exchange Graphics…** from the context menu. Choose the s3d graphics file that you want to use.
7. Return to the *Plant* frame and open it in the 3D viewer as well. Run the simulation and open the *Assembly* frame of the *Plant* frame. You will notice that the 3D model animation of the *Radiator* MUs hovers above the assembly station. This is because the size of the default 3D MU model differs from the MU size we defined in the previous step.
8. Right-click on the Radiator entity in the MUs folder in your Class Library and select **Edit 3D Properties…** from the context menu. Navigate to the Graphics tab and click on the button **Adjust Graphics to MU Size**, as shown in the figure below. Click on **Apply/OK**.

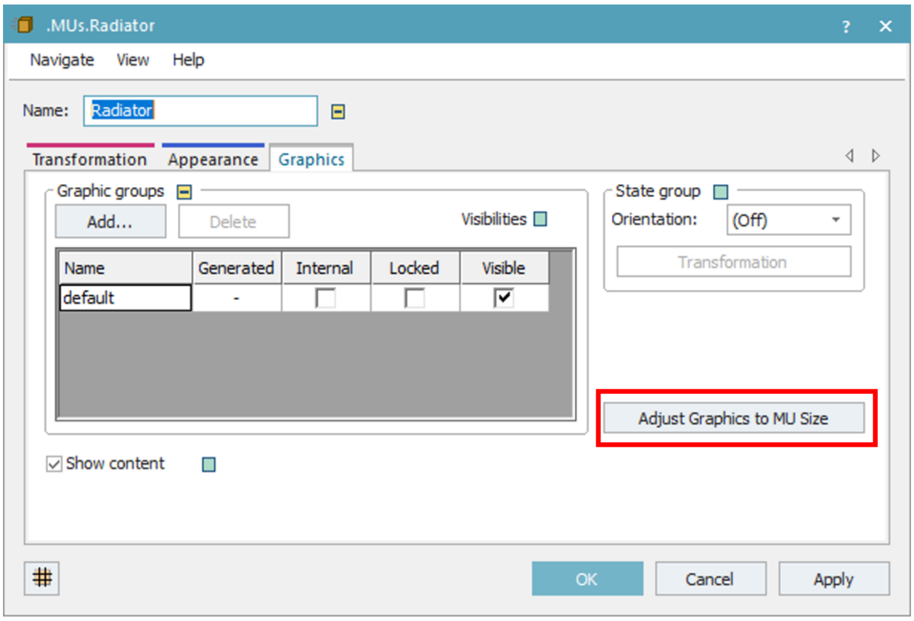

Figure 39: 3D properties of the MU

9. Reset the simulation and run it again. Do you see any difference? Feel free to explore the 3D model more on your own.

Download: The final state of the simulation model up to this exercise can be downloaded here Tutorial_Model_09.spp. Your simulation model should look similar to the figure below.

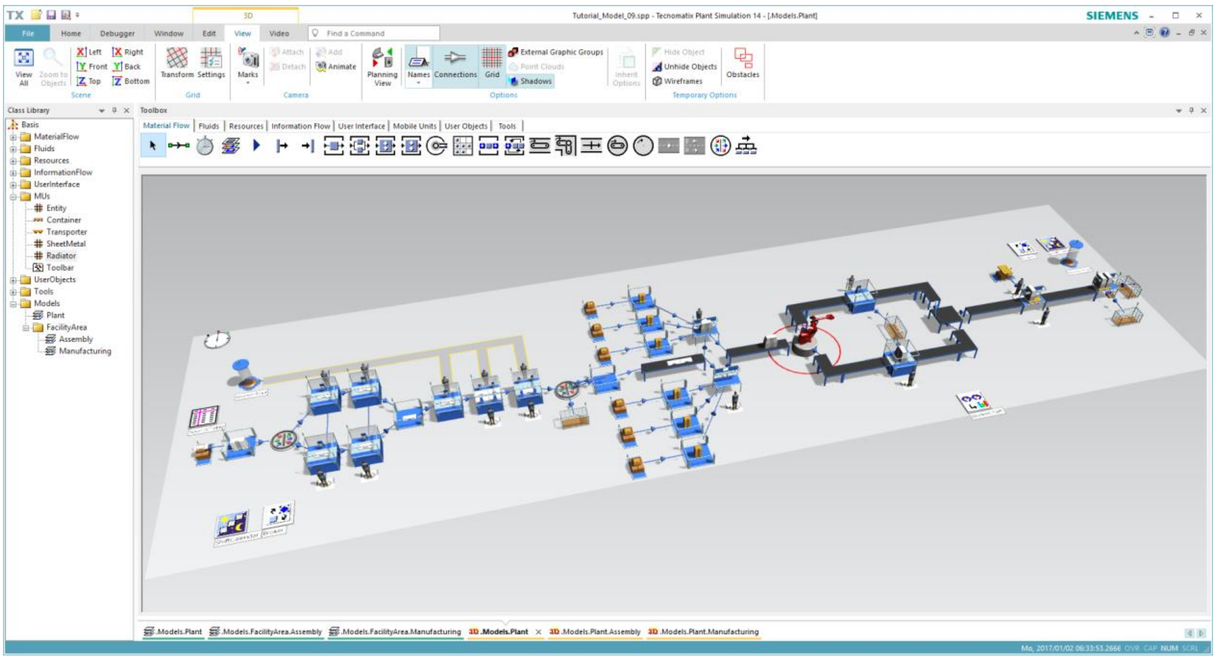

Figure 40: Final model of this chapter

previous chapter | next chapter

Summary & Conclusion

Congratulations! You have finished the Tecnomatix Plant Simulation 14 Online Tutorial. So far, you have covered the following topics:

- General understandings of simulation
- Understanding the working environment and GUI of Plant Simulation
- Understanding the object-oriented modeling approach of Plant Simulation
- Modeling material flow and resource objects in simulation model
- Enhancing the simulation user-defined control functions and behavior
- Analyzing and visualizing simulation results
- Viewing and running the simulation in the 3D viewer

You can download the complete tutorial as PDF (without example model files) here.

Advanced training

This tutorial only covers the basics of Plant Simulation. There are many more advanced functionalities of Plant Simulation that you can use. If you want to take your training to next level, there are several helpful additional resources:

- The integrated Tecnomatix Plant Simulation Help provides in-depth references for setting parameters of standard objects and built-in functions in SimTalk
- Plant Simulation also comes with several small example models. They are especially helpful to understand how a specific object works.
- Join Siemens Plant Simulation Forum to ask a specific modeling question or see if there are already answers to your questions

We also offer various training and individual coaching for Plant Simulation. Training is either in German or English. Training options are for example:

- Modeling in Plant Simulation for advanced users
- Advanced 3D modeling and visualization
- SQLite and database interface in Plant Simulation
- Statistics and distribution functions

Please do not hesitate to contact us for further inquiry.