

Introduction

This tutorial places you in the position of a process manager for a specialty electronics manufacturing firm that makes small lots of prototype boards for medical device manufacturers.

Your organization has just received an order for 40 units/week of a new component to be used in magnetic resonance imaging (MRI) equipment. As the process engineer, it is your responsibility to ensure that the process will be able to deliver the required weekly throughput.

Goal: 40 units/week of new component.

Step 1: Creating a Process Simulator Model

The first step in using Process Simulator is to create a simulation model of your process in the Visio layout. This is accomplished by adding activities, entities, and resources to the Visio layout and then connecting them with arrival and routing connections.

You may either build a model from scratch or you may open an existing Visio flowchart of your process, and convert it to a Process Simulator model.

For the purposes of this tutorial, we will begin with a small Visio flowchart that has already been converted to a Process Simulator model.

We will then add objects to complete the model, and then define properties for the objects in the model.

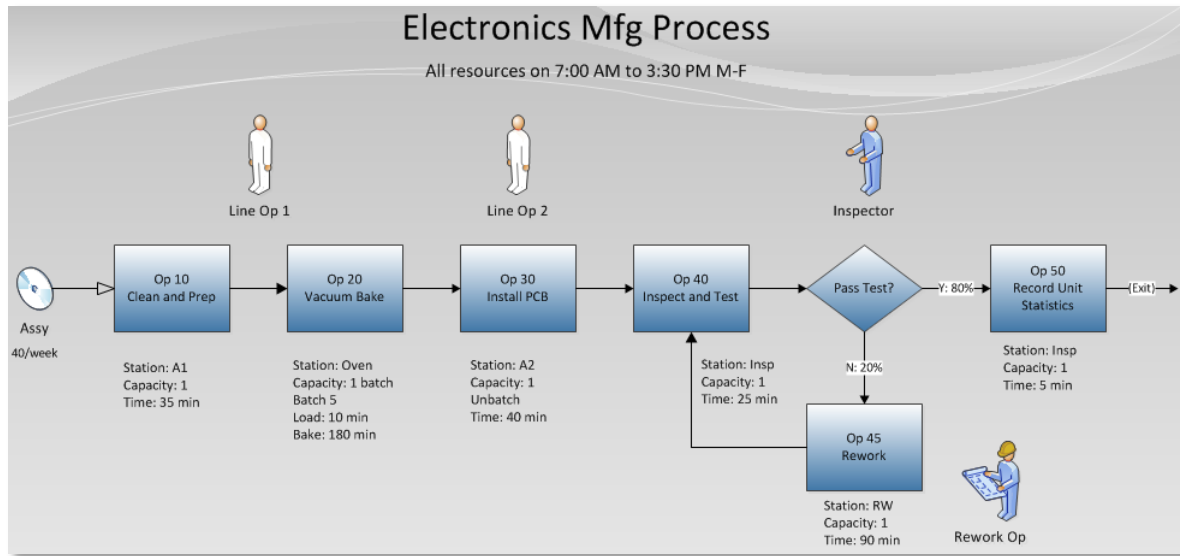
Open Tutorial File in Visio

Launch Process Simulator by double clicking on the Process Simulator icon on the Windows desktop or by clicking on the Start button and selecting All Programs -> ProModel Solutions -> Process Simulator 2014. From the File menu open the file, Tutorial.vsd, which is located in the Tutorial directory within the Process Simulator working directory.

(Typically: My Documents\Process Simulator\Tutorial\Tutorial.vsd).

If you think you might run through this tutorial again in the future, you will want to save the Tutorial.vsd file as a different name, preserving the original.

When you open the tutorial.vsd file, you will see the following process:



Summary of Electronics Mfg Process:

The electronics manufacturing process consists of six operations:

- Op 10 is a Clean and Prep activity where a purchased assembly is cleaned and placed into a special fixture by Line Op 1.
- Op 20 is a 3-hour baking operation. Parts are carried to the oven by Line Op 1 and loaded into the oven. The bake cycle will not begin until a batch of five parts has accumulated. The loading process takes 10 minutes and requires Line Op 1.
- Op 30 is an assembly operation, where a printed circuit board is installed. Each batch of 5 assemblies arriving from the oven is unbatched by Line Op 2 and then processed individually. The operation takes 40 minutes.
- Op 40 is an Inspect and Test station. Parts are taken to the inspect/test area by Line Op 2 and placed in an input buffer (a special cooling apparatus) that can hold only 8 parts at a time. The Inspector visually inspects each unit and runs a series of tests using a multi-function Test Unit. The inspection and test operation takes 25 minutes. 70% of the units pass the test on the first pass, while 30% of them fail and are sent to Rework.
- Op 45 is the Rework operation. All units can be reworked manually, which takes 90 minutes on average. A Rework Op completes the task and takes the units back to the Inspect/Test Queue.
- Op 50 is the final operation, where the Inspector records the results of the test.

Add Objects to the Model

The Electronics Mfg Process already contains most of the objects that are needed to simulate this process: seven activities, four resources, one entity, and eight connections. However, we still need to add another resource, Test Unit. We will also add a variable with a counter to track the work in progress (WIP) for the entire process.

Adding a Resource

The Inspector resource works in conjunction with a Test Unit resource to inspect and test parts as they pass OP 40. However, our model does not yet contain a Test Unit resource, so we will add one to the layout now.

1. From the Process Simulator Stencil, click on the Review (Activity) shape and drag it onto the layout above the Op 40 Inspect and Test activity.
2. With the Review shape still selected, type Test Unit for its new name (you can also double click on the shape to change its name).
3. Right-click on the Test Unit and select Resource from the right-click menu. This changes the shape type from an Activity to a Resource.

Adding a Variable

By adding a variable, called Assy_WIP, and incrementing its value when entities arrive in the process and decrementing the value when the entities leave, we can see how many entities are in the process at any given time during the simulation. Additionally, when the simulation is complete, we can view the average number of entities in the system.

1. Open the Model Elements dialog from the Process Simulator ribbon by clicking Variables.
2. Create a new variable named Assy_WIP.
3. Define its Type as an Integer.
4. Initial Value = 0.
5. Stats = Time-Weighted.
6. Check the Graphic check box, to display a counter for the variable on the Visio layout.
7. Close the Model Elements dialog, and then click and drag the counter graphic to position it below the Assy entity.

We will add the logic to increment and decrement the variable as the Assy entities arrive and leave the process later in the tutorial when we cover [adding logic](#) to the model.

Define Properties

With all of our objects on the layout, we will want to define the simulation properties for each of our objects. These properties include processing time at activities, batching of entities, routing decisions for quality control, cost values, shifts, etc. We will add logic statements and calendar files to our objects in the next topic: [Add Logic to the Model](#).

Defining Activity Properties

Immediately below or above each activity is text describing the properties for that particular activity. We will add those properties to our activities:

1. Right click on the Vacuum Bake activity and select Properties from the right-click menu.
2. In the Properties dialog for Vacuum Bake enter the information provided in the text associated with the activity:
 - Time Duration: 180 (leave time unit as “minutes”)
 - Click on the Batching button and, in the dialog that appears, select “Batch” for Batching Before Activity. Leave the Batch Size as 5. Then click OK.

3. Click on another activity in the layout to switch the Properties dialog to that activity, and enter its corresponding information. Repeat for all other activities according to the following specifications:

Activity Name	Properties to Modify
Op 10 Clean and Prep	Time: 35 Resource: Line Op 1
Op 20 Vacuum Bake	Time: 180 Batching: Before Activity: Batch: Batch Size 5
Op 30 Install PCB	Time: 40 Resource: Line Op 2 Batching: Before Activity: Unbatch
Op 40 Inspect and Test	Time: 0 (The actual time for this activity will be added in the Logic Field, which is described in the next topic.) In Buffer: 8 Hourly Cost: 17.5 (we will track only the cost associated with Inspection)
Op 45 Rework	Time: 90 Resource: Rework Op
Op 50 Record Unit Statistics	Time: 5 Resource: Inspector

4. Please note that the Pass Inspection activity does not need any additional properties.

Defining Routing/Arrival Properties

If you still have the Properties dialog open, you may click on connections in the layout to begin defining properties for them. If you have closed the properties dialog, you may reopen it by right-clicking on a connection and choosing Properties from the right-click menu.

Define your connections using the following specifications:

Routing/Arrival Properties	
Routing or Arrival Name	Properties to be defined
Arrival: Assy at Op 10 Clean and Prep	Type: Periodic Occurrences: Infinite Interval: 1 weeks Quantity Per Arrival: 40 Start Time: 0 minutes

Routing from Op 10 Clean and Prep to Op 20 Vacuum Bake	Move Time: 1 minute Resource: Line Op 1 Priority: 1
Routing from Op 20 Vacuum Bake to Op 30 Install PCB	Move Time: 0 minutes
Routing from Op 30 Install PCB to Op 40 Inspect and Test	Move Time: 1 minute Resource: Line Op 2 Priority: 1
Routing from Op 40 Inspect and Test to Pass Test	Move Time: 0 minutes
Routing from Pass Test? To Op 45 Rework	Move Time: 0 minutes Percent: 30
Routing from Pass Test? To Op 50 Record Unit Statistics	Move Time: 0 minutes Percent: 70
Routing from Op 45 Rework to Op 40 Inspect and Test	Move Time: 1 minute Resource: Rework Op Priority: 1

Defining Resource Properties

We will add cost properties to our resources according to the following specifications:

Resource Properties	
Resource Name	Properties to Define
Line Op 1	Hourly Cost: 12
Line Op 2	Hourly Cost: 15
Test Unit	Hourly Cost: 0
Inspector	Hourly Cost: 25
Rework Op	Hourly Cost: 20

Add Logic and Shifts to the Model

By defining your shape's properties, you have already added basic logic to your model. We will now use the Logic Builder to add advanced logic to some of our activities for resource usage and incrementing and decrementing the WIP variable. We will also assign shifts to our resources and some of our activities in this topic.

Adding Advanced Logic to Activities for Resource Usage

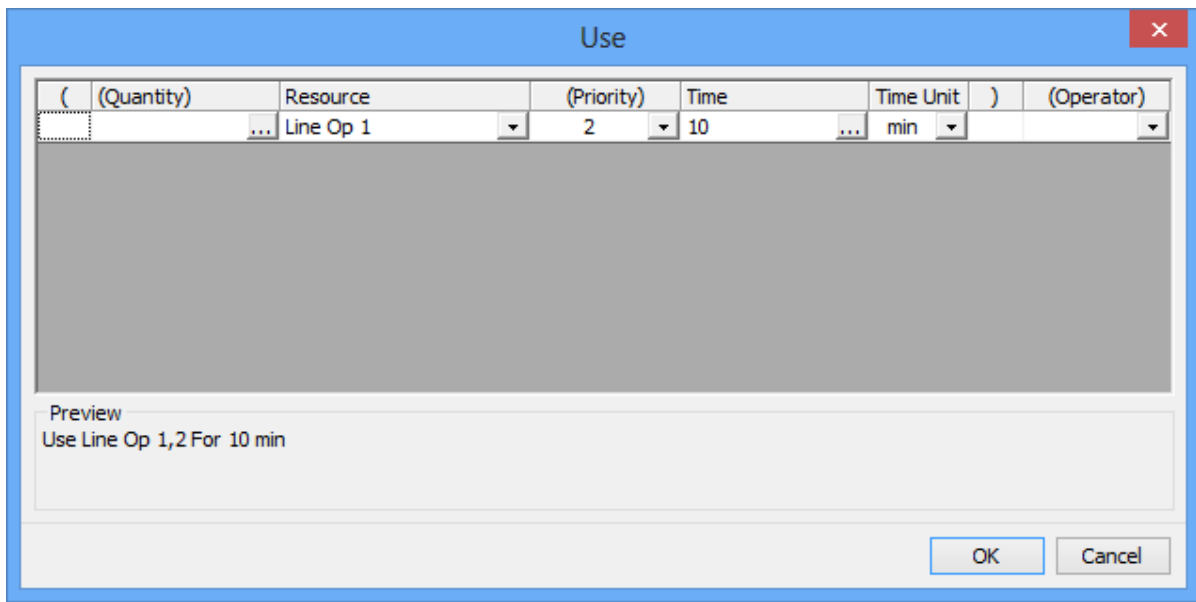
For simple resource usage, we have assigned resources for work on entities at certain activities using the Resource field in the Activity dialogs. For example, this was the case for Op 30 Install PCB.

However, Op 20 and Op 40 require more advanced resource logic. In this case, we leave the Resource field blank, and then use the Logic Builder to add more advanced resource usage logic.

To begin, Line Op 1 does not need to be used at the Op 20 activity for its entire processing time; therefore, we do not want to simply put Line Op 1 in the Op 20 activity Resource field. Instead we will use the Logic Builder to specify that the resource is to be used for only 10 minutes to load the oven.

1. Right click on Op 20 Vacuum Bake, and select Properties to open the Activity dialog.
2. Click on the Logic button to open the Logic Builder.
3. Double click on the USE statement on the left side of the Logic Builder. This statement allows you to assign resources to work on entities at activities.

4. A dialog will appear showing the available options for the statement. For information on this dialog, press F1 while using the dialog.



5. Fill in the fields of the dialog as shown in the image above.
6. Click OK to place the statement (Use Line Op1, 2 For 10 min) in the logic builder.
7. Close the Logic Builder to save the logic.

Next we will enter a different method for capturing, using, and then freeing multiple resources at activity Op 40 inspect and Test.

1. Right click on Op 40 inspect and Test, and select Properties to open the Activity dialog.
2. Click on the Logic button to open the Logic Builder. If you make an error, you can edit statements by double clicking on any line.
3. Double click on the GET statement on the left side of the Logic Builder. This statement allows you to capture resources to work on entities at activities.

((Quantity)	Resource	(Priority))	(Operator)
	...	Inspector			And
	...	Test Unit			

☒ Get resources jointly

Preview
Get Jointly Inspector And Test Unit

OK Cancel

4. Fill in the fields as shown above (be sure to check "Get resources jointly") and click OK.
5. Next, double click on the WAIT statement, enter a Value of 25 with time units of minutes, and click OK. This will cause the resource to work on the entity for 25 minutes.
6. Finally, double click on the FREE statement, choose ALL for the Resource value, and click OK. This will free the resources after the 25 minutes, so they will be available for work on other entities.
7. Double check that the Logic Builder contains the following lines for Op 40:

```

1  Get Jointly Inspector And Test Unit
2  Wait 25 min
3  Free ALL

```

8. Close the Logic Builder to save the logic.

Adding Advanced Logic to Activities for WIP Variable

We will now add the logic to increment and decrement the variable Assy_WIP as the Assy entities arrive and leave the process.

1. Right click on the Op10 Clean and Prep activity, and choose properties from the right-click menu.
2. From the properties dialog, click the Logic button, which will display the Logic Builder.
3. In the list of statements, double click on the **Inc** option.
4. From the **Increment** dialog, select the Assy_WIP variable, and then click OK.

This will cause the Assy_WIP variable to increment by one each time an entity enters the process. We will now enter a Dec, so that the variable counter will decrease by one each time an entity exits the process.

1. Right click on the exit connection leaving Op 50 Record Unit Statistics, and choose properties from the right-click menu.
2. From the Routing dialog, click the Logic button, which will display the Logic Builder.
3. In the list of statements, double click on the **Dec** option.
4. From the **Dec** dialog, select the Assy_WIP variable, and then click OK.

Adding Shift assignments to Resources and Activities

All resources, except the Inspector and the Test Unit are available 8 hours per day, from 7:00 AM to 3:30 PM, with a ½ hour unpaid break from 11:30 to 12:00. These resources are not allowed to work overtime, so they leave immediately when their shift is over.

Adding shift assignments is done by:

- Open the properties dialog for a resource or activity.
- Click on the Calendar button.
- Click Browse to enter the path to the shift file.
- Select the appropriate .pmcal file, which is in the same directory on your computer as the Tutorial.vsd file.
- Select the Interrupt checkbox if needed, indicating they will interrupt their work to go off shift.
- Repeat for all resources and activities where shifts will be applied.

Assign shifts to resources and activities according to the following specification:

Shift Assignments

Resource or Activity Name	Shift File to Assign
Op 10 Clean and Prep	7a to 330p.pmcals, with Interrupt option checked
Op 20 Vacuum Bake	7a to 330p.pmcals
Op 30 Install PCB	7a to 330p.pmcals, with Interrupt option checked

Op 40 Inspect and Test	7a to 330p Test Insp.pmcals
Op 45 Rework	7a to 330p.pmcals, with Interrupt option checked
Op 50 Record Unit Statistics	7a to 330p Test Insp.pmcals
Line Op 1	7a to 330p.pmcals, with Interrupt option checked
Line Op 2	7a to 330p.pmcals, with Interrupt option checked
Test Unit	7a to 330p Test Insp.pmcals
Inspector	7a to 330p Test Insp.pmcals
Rework Op	7a to 330p.pmcals, with Interrupt option checked

Simulate the Model

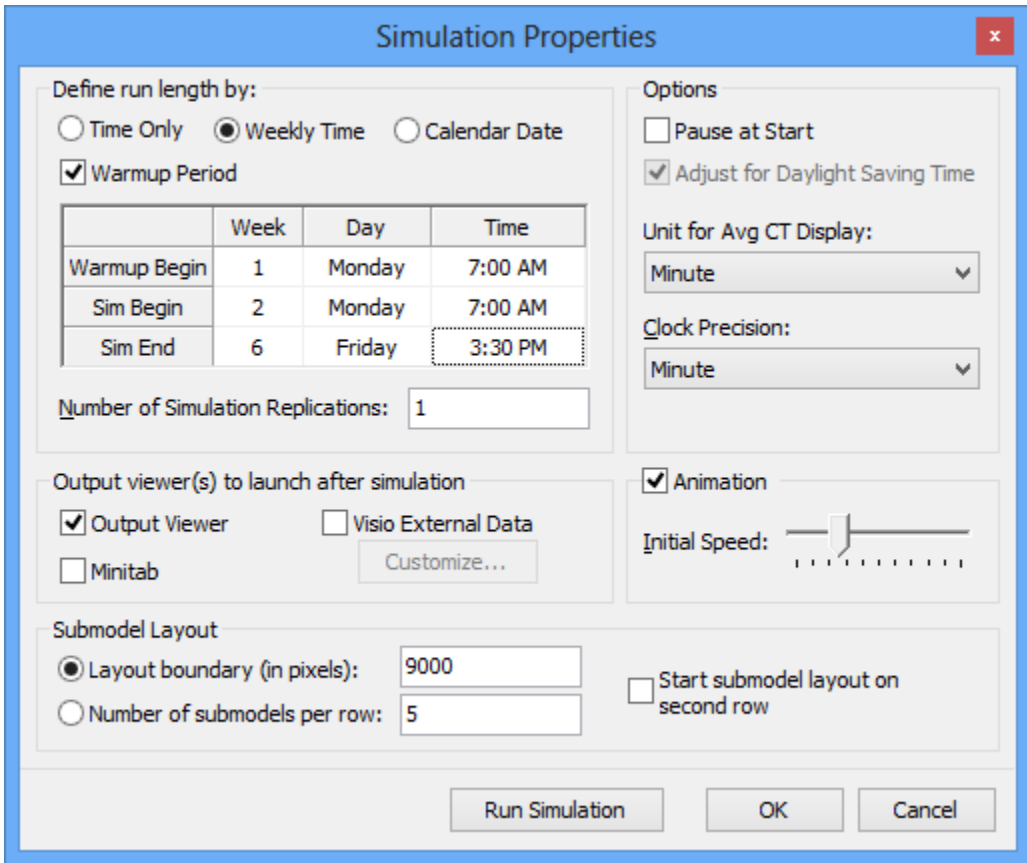
With all your objects defined and logic entered you are ready to simulate your model to see if we can achieve the goal of 40 units/week.

Simulating the Model with One Replication

Before we begin the simulation, we need to define the simulation run length:

1. From the Process Simulator ribbon, select the Simulation Properties button. The Simulation Properties dialog will appear.

2. Modify the properties as shown below:



The image shows the 'Simulation Properties' dialog box with the following settings:

- Define run length by:**
 - ☐ Time Only
 - ☒ Weekly Time
 - ☐ Calendar Date
- ☒ Warmup Period
- Table:**

	Week	Day	Time
Warmup Begin	1	Monday	7:00 AM
Sim Begin	2	Monday	7:00 AM
Sim End	6	Friday	3:30 PM
- Number of Simulation Replications:** 1
- Options:**
 - ☐ Pause at Start
 - ☒ Adjust for Daylight Saving Time
 - Unit for Avg CT Display:** Minute
 - Clock Precision:** Minute
 - ☒ Animation
 - Initial Speed:** (slider bar)
 - ☐ Start submodel layout on second row
- Output viewer(s) to launch after simulation:**
 - ☒ Output Viewer
 - ☐ Visio External Data
 - ☐ Minitab
 - Customize...
- Submodel Layout:**
 - ☒ Layout boundary (in pixels): 9000
 - ☐ Number of submodels per row: 5

Buttons at the bottom: Run Simulation, OK, Cancel.

3. Click **OK** to save the changes. This will cause the simulation to run for six weeks and collect statistics for the five weeks where the simulation is in a steady state.

Starting your simulation is as easy as clicking the **Simulate** button from the Process Simulator ribbon.

When the simulation begins the view will change from the Visio layout to the Process Simulator run-time layout. As the simulation runs, you will see the Assy entities move through your process. You will also notice the color of the resources change to reflect their state:

- No color change = available to do work.
- Green = in operation.
- Red = off shift.

There are several options available in the Run-Time layout, which are described in the Process Simulator Help System.

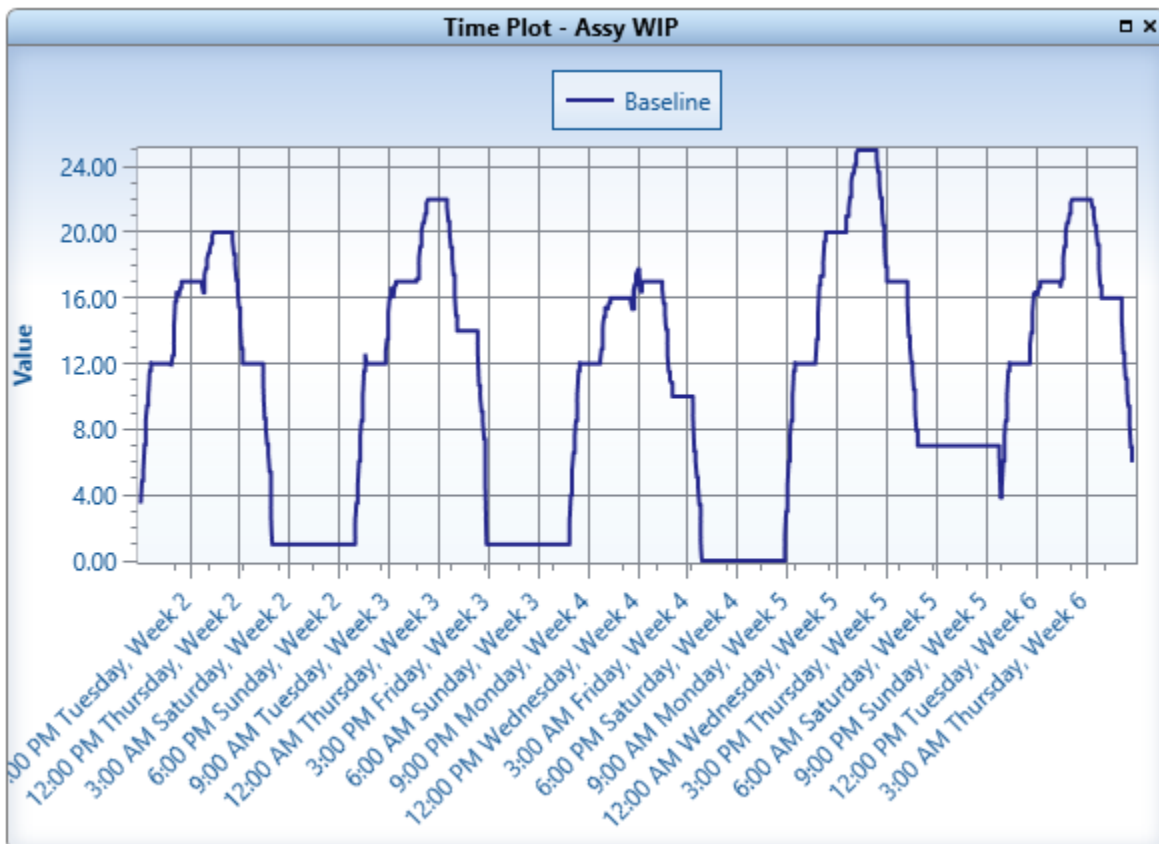
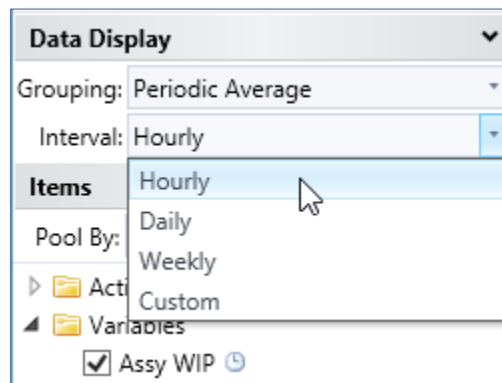
Viewing Output Results for One Replication

Upon completion of the simulation, you will be prompted to view the output statistics. Choose yes, which will launch the Output Viewer.

From the statistics displayed in the output viewer, you will notice:

- In the Report, Total exits for Assy is 197, which does not quite meet our goal of 40 units/week. This is derived by dividing the 197 by the number of weeks for which we are collecting statistics, which is 5 weeks. The first week of simulation was warm-up and is not counted in the statistics.
- In the Single Capacity Activity State Chart we see there appears to be no resource constraints other than a small amount of waiting at Op 40.
- If you create a Time Plot chart of the Assy_WIP variable (Output Viewer Ribbon -> Time Plot -> Time-weighted values and choosing to display Assy_WIP) we see that the inventory is clearing out each week as expected.

Please Note: By default, the Interval displayed is for "Weekly" time. In order to achieve the same results as the graphic below, please change the interval to "Hourly."



The only variability in our model so far is the 30/70% routing decision for our Test and Inspect activity. Next, we will average out this variability by running the simulation for 20 replications, and see if our goal of 40 units/week can be met.

Simulating the Model with 20 Replications

Return to Process Simulator, and in the Simulation Properties dialog, change the number of replications from 1 to 20.

Run the simulation again, and view the new statistics.

Notice that Total Exits now equals 201.30, which meets our goal of 40 units/week.

Please Note: If you are not seeing the same results as described above, you may need to double check your model against the reference tutorial model for this point in the tutorial, Tutorial2.vsd, which is in the same directory as the Tutorial.vsd file.

Summary

We probably could have calculated throughput of 40 units/week with some simple arithmetic, and most likely did not need to use simulation to come to this conclusion. If our goal was 40 units/week, we could then assume that we have achieved our goals for this process.

However, does this number, which was calculated using constant numbers, really reflect the reality of our process? In the real world, activities do not always process entities at a constant rate. Variability is a critical factor to consider, before we declare that this process will produce 40 Assy units/week.

Therefore, we will add variability, in the form of statistical distributions, to the process times at activities in order to more closely reflect reality. We can then determine if our process will continue to meet the throughput goal of 40 units/week, before we implement the actual process.

View Step 1 Solution Video (optional)

<http://www.promodel.com/services/training/PCS/StandardTutorial/9.1/Tutorial1.html>

Step 2: Adding Variability to the Model

In order to better reflect the reality of the process, we will now add variability to our model.

We will add variability to the model for the following areas:

- Process times at activities.
- Arrival quantities.
- Resource availability.

Add Distributions to Activity Times

Many process activities in the real world do not complete their processing in a fixed amount of time, but rather vary somewhat from one instance to the next. To simulate this variation, we will use the static process times as a basis for the new distributions we will use for activity times.

In the activities dialog (right-click on activity and choose Properties) change the activity times from the constant (or deterministic) value to the distributions shown below:

<u>Activity Name</u>	<u>Properties to Modify</u>
Op 10 Clean and Prep	Time Duration: N(35,5)
Op 20 Vacuum Bake	No Change
Op 30 Install PCB	Time Duration: N(40,5)
Op 40 Inspect and Test	Logic: Open Logic Builder and double click on the WAIT 25 line of logic. In the WAIT dialog, change the value to WAIT N(25,3), by clicking on the "..." button under Value the column, and selecting Distribution. In the distribution builder, select the Normal distribution and enter the mean and standard deviation as 25 and 3 respectively. For more information on using the distribution builder, press F1 while using the distribution Builder.
Op 45 Rework	Time Duration: T(70,90,180)
Op 50 Record Unit Statistics	Time Duration: U(5,1)

You may type the distributions directly into the Time Duration field, or use the distribution builder to ensure your distributions are syntactically correct.

Please Note: Microsoft Windows operating systems other than English may use different separator styles for distributions. For example the English OS uses a comma separator as in N(35,5), while a German OS uses a semi-colon as in N(35;5). Process Simulator will use the correct separator for your

OS when you use the Distribution Builder to enter distributions; however, if you enter distributions manually, you must use the correct separator based on your OS settings

Add Variability to Arrivals

In our example, we assume our Electronics Mfg Process is dependent on a supplier to deliver the Assy entity. Therefore, we cannot always assume the supplier will be able deliver exactly 40 units a week to our process at Op 10 Clean and Prep.

Therefore, we will define a distribution for the arrival of Assy to Op 10 to reflect this variability.

1. Open the Arrival Property dialog for the arrival of Assy to Op 10 (right click on the connection between Assy and Op 10, and choose Properties.)
2. In the Quantity Per Arrival field, change the fixed value of 40 to the distribution U(40,5)

The screenshot shows a dialog box titled "ARRIVAL: Assy AT Op 10_Clean and Prep". It has two main tabs: "General" and "Details". In the "General" tab, the "Type" is set to "Periodic". Below this are buttons for "Logic..." and "Calendar...", both of which are currently set to "(None)". There is also an unchecked checkbox labeled "Input to Submodel". The "Details" tab is active, showing the following settings: "Occurrences" is "Infinite", "Interval" is "1" with a unit dropdown set to "weeks", "Quantity Per Arrival" is "U(40, 5)", and "Start Time" is "0" with a unit dropdown set to "minutes". At the bottom of the dialog are two tabs: "Arrival" (which is selected) and "Notes".

Add Resource Constraints

When variability is added to the availability of resources, a constraint may be caused in the process. Process Simulator makes it easy to see where resource constraints are happening, as will be illustrated after making the following changes, and then simulating the model, which we will do in the next topic.

Change Resource Shifts and Availability

The times when breaks and downtimes for resources occur are not always known. Therefore, instead of using fixed break times in the shift file to specify the 80% availability for the Inspector and Test Unit resources, we will reassign the standard shift to these resources, and make the resources randomly available for 80% of the time while they are on-shift.

1. Open the Resource Properties dialog for the Inspector resource.
2. Click the Calendar button, and reassign the calendar file from "7A to 330P Test Insp.pmc" to "7A to 330P.pmc." Click **OK** to save the change.

3. Change availability to 80%, as shown below:

The screenshot shows the 'RESOURCE: Inspector' dialog box with the following details:

- Title Bar:** RESOURCE: Inspector
- General Tab:**
 - Units: 1
 - ☒ Show multiple units graphically
 - ☒ Collect Stats
 - Availability: 80
 - Calendar... 7a to 330p.pmc
- Cost Tab:**
 - Hourly Cost: 25
 - Cost Per Use: 0
- Bottom Tabs:** Resource (selected), Downtimes, Notes

4. Repeat steps 1-3 for the Test Unit Resource.

Change Activity Shifts

Since we changed the shift for the Inspector and Test Unit resources, we may also want change the shift for the two activities where they work:

1. Open the Activity Properties dialog for Op 40 Inspect and Test.
2. Click the Calendar button, and reassign the calendar file from "7A to 330P Test Insp.pmc" to "7A to 330P.pmc."
3. Click OK to save the change.
4. Repeat steps 1-3 for Op 50 Record Unit Statistics.

Simulate Model with Variability

Now that we have added variability to our process times, arrival, and resource availability, in order to make our process reflect reality, we will simulate the model again and see if we still meet our goal of 40 units/week.

Single Replication

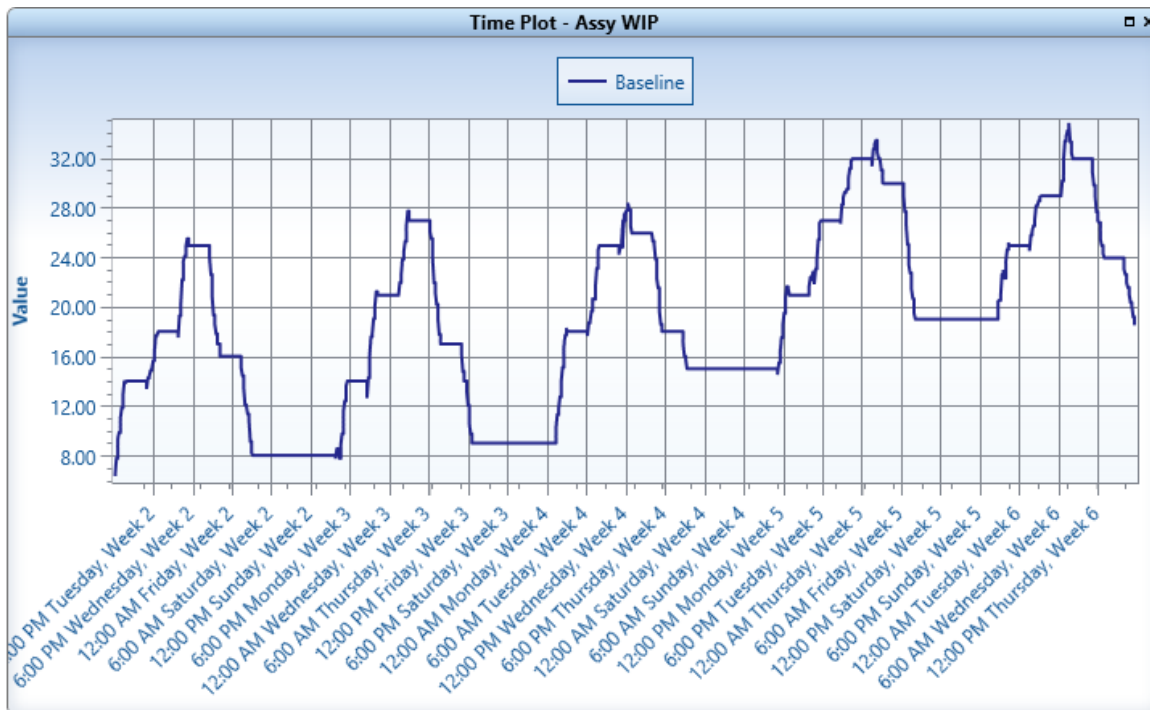
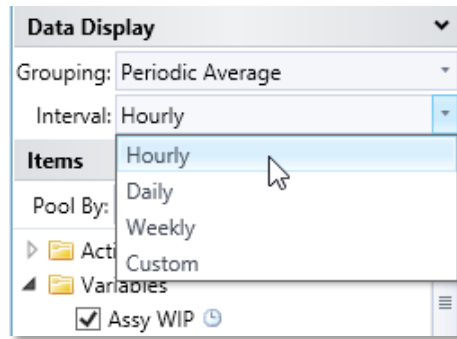
As we did when we ran the previous simulation, we will begin by running a single replication of the simulation.

1. In the **Simulation Properties** dialog, set the number of replications to 1.
2. Click the **Simulate** button to run the simulation.
3. When the simulation is complete, choose to view the output results.

In the output results, we immediately see that the throughput (Total Exits) has dropped to 179 for the five-week period.

Additionally, if you create a Time Plot of the Assy_WIP variable (Output Viewer Ribbon -> Time Plot -> Time-weighted values and choosing to display Assy_WIP) we see that the WIP is much more variable.

Please Note: By default, the Interval displayed is for "Weekly" time. In order to achieve the same results as the graphic below, please change the interval to "Hourly."



A bottleneck was formed due to variability and resource interdependencies.

Twenty Replications

Next, we will analyze the averaged effect of variability on the process.

1. In the Simulation Properties dialog, set the number of replications to 20.
2. Click the Simulate button to run the simulation.
3. When the simulation is complete, select "Yes" to view the output results.

We see in the output the following:

- Average throughput (Total Exits) is now 190.40 units for the five-week period, which is an average of 38.08 units a week.
- Average WIP has increased from 11.6 units for the constant simulation to 16.41 units in the current simulation with variability.
- Average cycle time has also increased from 75 hours to 98 hours.
- Average cost has increased by \$5.20.

The chart below shows a comparison of the simulation without variability to the simulation with variability.

<u>Variable Name</u>	<u>Without Variability</u>	<u>With Variability</u>	<u>Difference</u>
Throughput	201.30	190.40	-10.9
Avg WIP	11.60	16.41	4.81
Avg Cycle Time	76.8	95.8	19
Avg Cost	208.44	213.88	5.44
<u>Resource Utilization</u>			
Inspector	83.04	74.9	-8.14
Line Op 1	63.33	61.67	-1.66
Line Op 2	68.33	66.43	-1.90
Rework Op	64.75	74.65	9.90
Test Unit	72.79	67.52	-5.27

Please Note: If you are not seeing the same results as described above, you may need to double check your model against the reference tutorial model for this point in the tutorial, Tutorial3.vsd, which is in the same directory as the Tutorial.vsd file.

Summary

We will not meet our goal of 40 units/week. If this process had been implemented in the real world, based on the fixed values initially provided, the process never would have met its goal. Changes would have needed to be made post-implementation, which would have been costly in lost time and money.

However, with these results, we may explore changes to the process through simulation, in order to achieve the 40 units/week goal, before the process is actually implemented in the real world.

View Step 2 Solution Video (optional)

<http://www.promodel.com/services/training/PCS/StandardTutorial/9.1/Tutorial2.html>

Step 3: Make Changes to Improve Process

Based on our output results, we know that a bottleneck is occurring at Op 40 due to resource constraints. Additionally, we find that unreliable arrivals of Assy entities to the process is causing problems.

If we could relieve the Op 40 bottleneck and stabilize our arrivals, we may achieve our 40 units/week goal.

Make Changes to Model

We will need to make two changes to our model:

- Fix the Op 40 bottleneck created by resource constraints.
- Fix the inadequate arrivals of the Assy entity.

Fix Op 40 Bottleneck

In order to fix the problem of resource constraints causing a bottleneck at Op 40, we will make the Rework Op an alternate resource when the Inspector is not available. We also discover that an old test unit, which is not currently in use, could be used as a backup to the primary test unit.

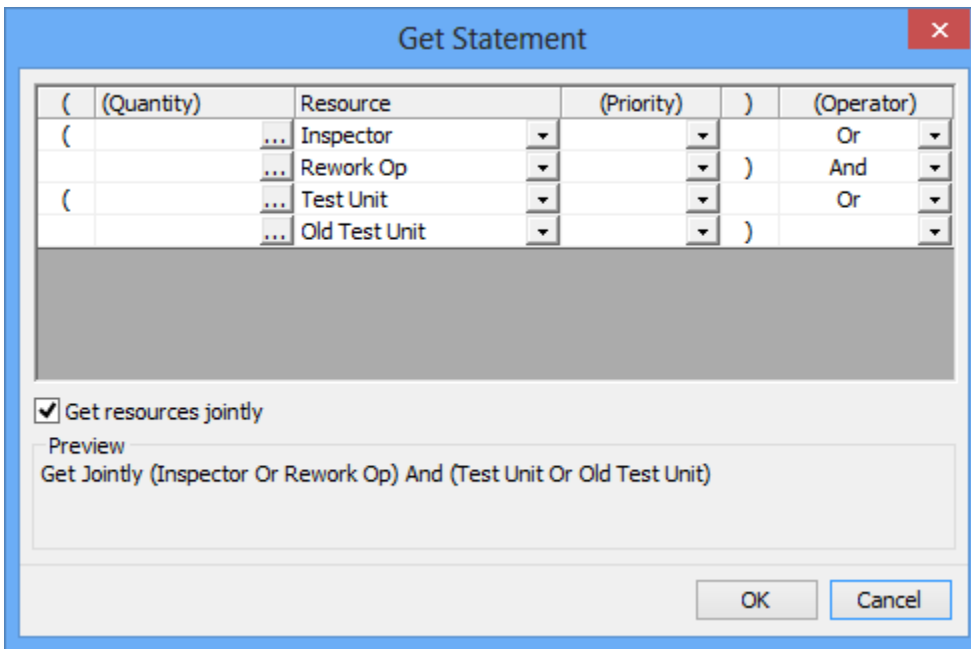
Complete these steps to implement the fix:

Add Old Test Unit Resource

1. Click and drag a Review (Activity) shape from the Process Simulator stencil to the layout, and place it under the Op 40 Inspect and Test activity.
2. Double click on the shape in the layout and rename it to "Old Test Unit."
3. Right-click on the Old Test Unit, and change its type to Resource.
4. Open the Resource Properties dialog for the resource, and assign the 7A to 330P.pmc shift.

Add Logic to Use Old Test Unit as Alternate Resource

1. Open the Activity Properties dialog for the Op 40 Inspect and Test activity.
2. Click on the Logic button to open the Logic Builder.
3. Double click on the GET JOINTLY statement we previously defined, in order to edit it.
4. Change the statement as shown below, and click OK to save the changes.



The 'Get Statement' dialog box is shown with a table for defining resource requirements. The table has columns for opening parenthesis '(', Quantity, Resource, Priority, closing parenthesis ')', and Operator. The table contains four rows of data. Below the table is a checkbox for 'Get resources jointly' which is checked, and a 'Preview' section showing the resulting statement: 'Get Jointly (Inspector Or Rework Op) And (Test Unit Or Old Test Unit)'. At the bottom are 'OK' and 'Cancel' buttons.

((Quantity)	Resource	(Priority))	(Operator)
(...	Inspector			Or
	...	Rework Op)	And
(...	Test Unit			Or
	...	Old Test Unit)	

☒ Get resources jointly

Preview
Get Jointly (Inspector Or Rework Op) And (Test Unit Or Old Test Unit)

OK Cancel

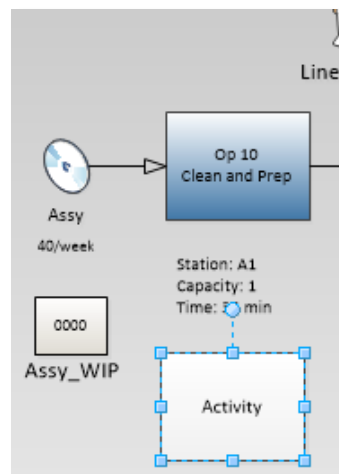
Fix Arrivals of Assy Entity

In order to fix the inadequate arrivals of Assy Entities, we could simply have our supplier send more to cover any deficits, but this may lead to an unnecessary increase in inventory.

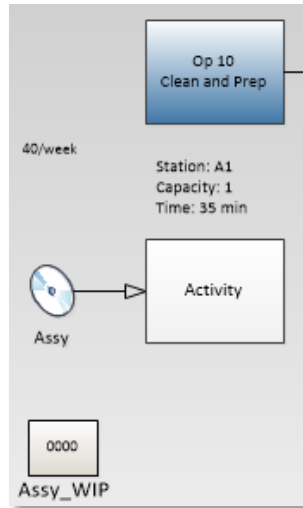
Instead, we will define an order, which will automatically order more Assy entities when the Assy entity inventory has dropped below 20.

To implement this automatic order, complete the following steps:

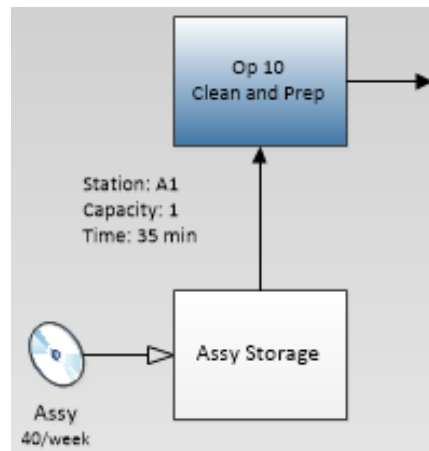
1. Add a storage activity by doing the following:
 - a. Underneath Op 10, add a new activity, as shown below:



2. Right click on the new activity, and change its type to a Storage type.
3. Double click on the new activity, and change its name to Assy Storage.
4. Move the Assy entity and its arrival connection, so that arrivals will occur at the new Assy Storage, instead of Op 10, as shown below: (Feel free to move the Assy_WIP counter to make space for the Assy entity)



- a. Add a routing connection from Assy Storage to Op 10 as shown below:



2. Make changes to the properties of the affected objects by doing the following:
 - a. Open the properties of the arrival connection from Assy to Assy Storage.
 - b. Change the Type to Ordered, and define the lead time to be $U(3,1)$, which gives a lead time of 2 to 3 days. The properties should be as shown below:

ARRIVAL: Assy AT Assy Storage

General

Type: **Ordered**

Logic... (None)

Calendar... (None)

☐ Input to Submodel

Details

Lead Time: **U(3,1)** days

Arrival Notes

- c. Open the properties for Assy Storage, and change its properties to the following:

STORAGE: Assy Storage

General

Capacity: **999**

Entity: **Assy**

Initial Level: **15**

Logic... (None)

Trigger

Type: **By Level** Level: **20**

Order

Entity: **Assy** Quantity: **10**

Source: **Assy @ Assy Storage**

Storage Notes

- d. The storage activity will now order 10 more Assy entities whenever its inventory drops below 20.

- e. Since the new Assy Storage acts as a buffer for Op 10, we can remove Op 10's buffer. Open Op 10's properties dialog and change the In Buffer to 0, as shown below:

Simulate and Analyze Results

With our fixes in place, we will once again run the simulation to see if the changes now allow us to meet our goal of 40 units/week.

1. In the Simulation Properties dialog, set the number of replications to 100.
2. Click the Start button to run the simulation.
3. When the simulation is complete, choose to view the output results.

We see in the output the following compared to the simulation with variability:

- Average throughput has increased from 193.05 units/5 weeks (38.61/week) to 206.16 units/5 weeks (41.23 units/week).
- Average cost has decreased by \$2.02 per unit.

The chart below shows a full comparison of the simulation with variability to the simulation with fixes.

	<u>With Variability</u>	<u>With Fixes</u>	<u>Difference</u>
Throughput	190.40	207.66	17.26
Avg WIP	16.41	20.12	3.71
Avg Cycle Time	95.8	109.41	13.61
Avg Cost	213.88	211.83	-2.05
<u>Resource Utilization</u>			
Inspector	74.9	70.89	-4.01

Line Op 1	61.67	67.75	6.08
Line Op 2	66.43	73.44	7.01
Old Test Unit	N/A	11.45	N/A
Rework Op	74.65	90.63	15.98
Test Unit	67.52	69.68	2.16

With these changes, we now know that our process will achieve its goal of 40 units/week.

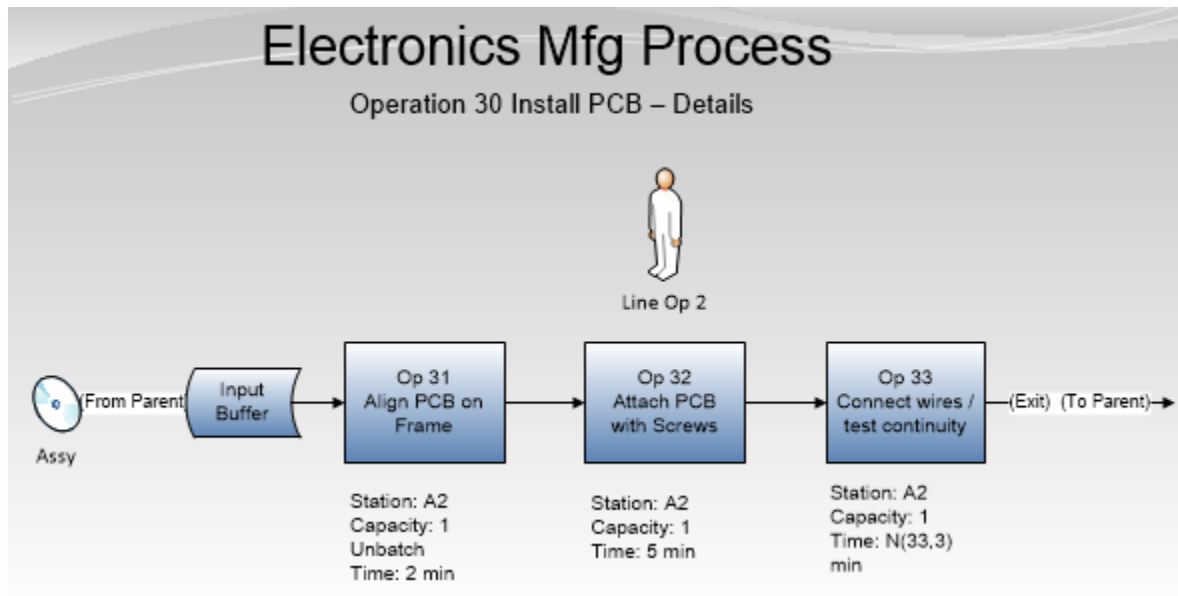
Please Note: If you are not seeing the same results as described above, you may need to double check your model against the reference tutorial model for this point in the tutorial, Tutorial4.vsd, which is in the same directory as the Tutorial.vsd file.

View Step 3 Solution Video (*optional*)

<http://www.promodel.com/services/training/PCS/StandardTutorial/9.1/Tutorial3.html>

Step 4: Implement Hierarchical Modeling

This step shows how we can add a submodel to show more detail of the processes taking place at an activity. For the tutorial, we will be connecting the submodel shown below to the Op 30 Install PCB activity.



This submodel details the three processes that occur in the main model's Op 30 Install PCB activity. Next, we will learn how to link the submodel to the main model, and then make necessary changes to the submodel to handle the input and output of entities from the main model to submodel.

Link to Submodel

The submodel for the Op 30 Install PCB activity is in the TutorialSubmodel.vsd file, which is in your Process Simulator working directory in the Tutorial folder. Please locate this file and open it in Process Simulator. You may note that this submodel is itself a complete model that can be simulated.

Create Hyperlink

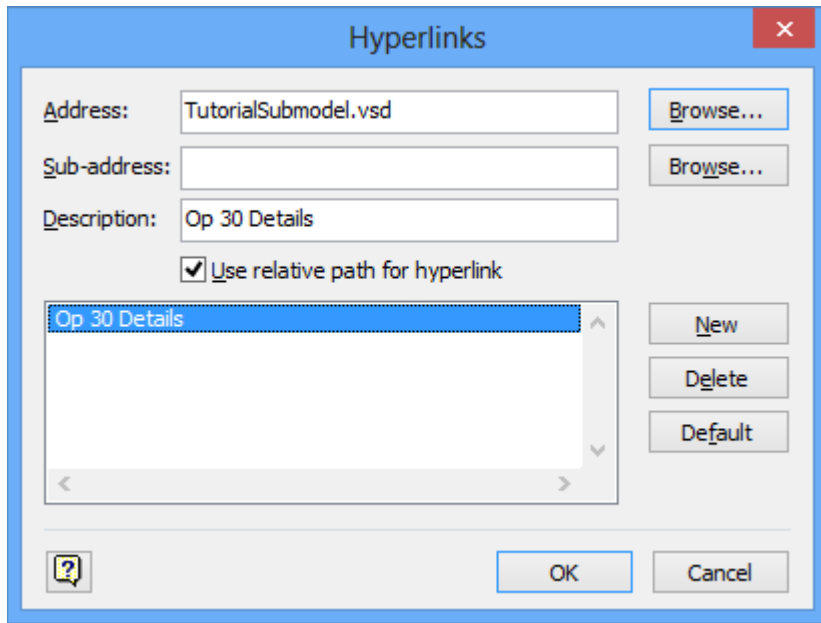
We will use a hyperlink to connect the Op 30 Install PCB activity on the parent sheet to its submodel.

1. Select the Op 30 Install PCB activity.



2. Open the Visio Hyperlinks dialog by using the keyboard shortcut Ctrl+K or clicking the Hyperlink button on the Visio **Insert** ribbon bar.

3. Click the Browse button for the Address, choose the local file option, and then select the TutorialSubmodel.vsd file.
4. Rename the Description to "Op 30 Details". (optional)



5. Click OK to save the hyperlink. If you move the mouse over the activity, you will notice that it shows a hyperlink icon next to the mouse cursor to signify the hyperlink.

Make Changes to Submodel

Now open the TutorialSubmodel.vsd file that we just linked to our parent model. In the TutorialSubmodel.vsd file you will see three activities that detail the subprocesses that occur in the main model's Op 30 Install PCB activity. The PCB is aligned on its frame, screws are attached, and the install is completed by connecting wires and testing the continuity.

Previously in the main model, the entire Op 30 Install PCB activity used a $N(40,5)$ distribution for its time. The submodel gives more detail to the time required to install the PCB, where the first two activities have fixed times of two minutes and five minutes respectively and the final activity has a distribution of $N(33,3)$ for its activity time.

Entity Arrival and Exits

An Assy entity is defined in the submodel. It is present so our submodel can initially be simulated independently of its main model; however, we would ultimately like to have our submodel receive its entities from the main model, process them, and then return them to the main model.

1. Right click on the connection from the Assy entity to the Input Buffer and choose Properties.
2. Check the Input to Submodel checkbox. Entities will now arrive in the submodel from the parent model. The connection will now say (From Parent).
3. Click on the (Exit) route.
4. In its properties dialog choose Output from Submodel. The connection now says (Exit) (To Parent). Entities exiting the submodel from this route will exit to the parent model, and re-enter the parent at the connection between Op 30 Install PCB activity and Op40 Inspect and Test.

Please Note: Be sure all calendar files are pointing to the correct path in the submodel using the Calendar button from the properties dialog or you may receive an error message that indicates, "Missing the calendar files" the next time you open your saved model.

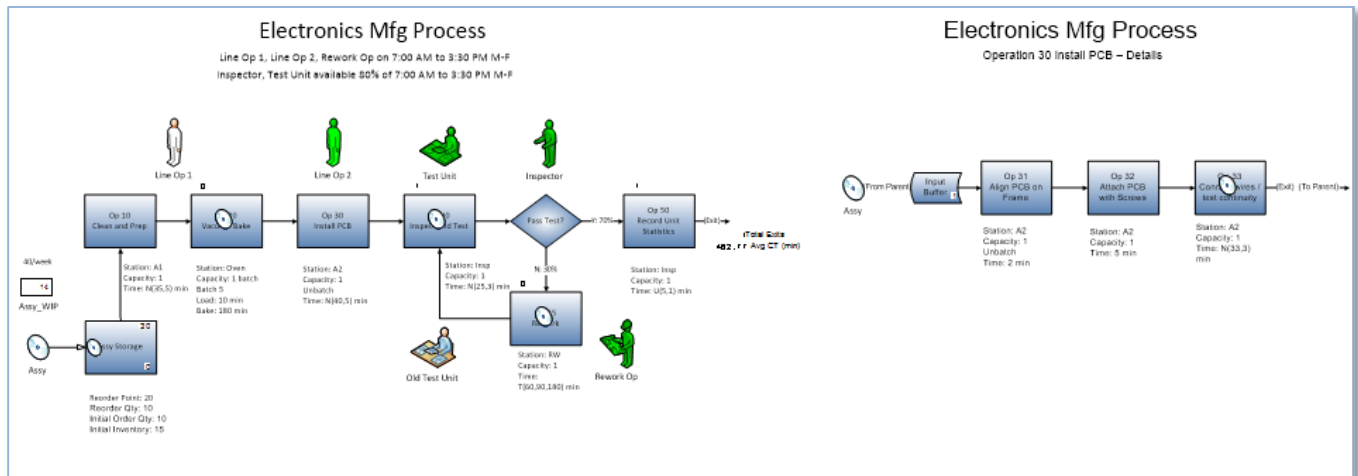
Submodel Resource Usage

The submodel also has a Line Op 2 resource. The resource is present so the submodel may be run independently, and also so the resource may be selected in the properties dialogs for activity and routing processing. When the submodel is run independently, the properties for its Line Op 2 resource are used; however, when the submodel is used by a simulated main model, the properties for the main model's Line Op 2 resource are used.

Simulate Main Model with Submodel

If we now return to our main model, we can choose 100 replications from the Simulation Properties dialog, and then click the Run button to simulate the main model with its associated submodel.

During simulation, the layout will now show the main model side by side with the submodel. As entities are routed to the Op 30 Install PCS activity, you will see them enter the submodel for processing.



Simulation Results

Since we have entered more detail about the subprocesses that occur at the Op 30 Install PCS activity we should now expect different results.

As you compare the output between the 100 replications that did not use the submodel with the 100 that did, you may find interesting differences.

View Step 4 Solution Video (optional)

<http://www.promodel.com/services/training/PCS/StandardTutorial/9.1/Tutorial4.html>

Conclusions

Although our initial simulation showed that we would achieve our goal of 40 units/week, its results were not reflective of reality, where variability impacts much of our process. Had we implemented this process in our organization, we would have expected no problems with productivity; however, in reality, we never would have met our goal due to variability.

After taking variability into account, we saw a significant drop in our throughput. The goal of 40 units/week could not be achieved without making some changes to the process. With the results of the simulation in hand, we could find which changes to make, and then test these changes in Process Simulator without making expensive changes to our actual, real-world process.

Having fixed the deficiencies in the process, we find that we will be able to meet our goal of 40 units/week and can implement the process in our organization with confidence.