18.4: Summary

This case study shows how system operational algorithms can be included in models. The use of modeler defined resource states is included. Inventories and other resources are shared between processes in the model. The simulation experiment compares alternative system configurations.

Problems

1. Based on the process steps in the simulation model, tell why the bin states: occupied with a subassembly of type one, two, three, or four that is committed to the final assembly process are necessary.

2. Validate the function SearchOne by searching from rack location row 2 column 2 as shown in Figure 18-2 in the right and up direction. List the first ten values of RowIndex and CollIndex computed in SearchOne, including the infeasible bin location values that cause the loops in SearchOne to end.

3. In the Arrival process model, why is it not necessary to check if the function SearchRack located a bin in the IDLE state?

4. Tell why the quantity: Number of final assembly process requests waiting for a subassembly is not an effective performance measure for the simulation experiment in this chapter.

5. What impact would running the simulation experiment until all subassemblies had been moved to the final assembly process have on the validity of the performance measure estimates?

6. Explain why the average waiting time for the SR machine increases when the larger rack size is used especially considering that there is no waiting for an empty bin.

7. Would you expect the utilization of the SR machine to increase or decrease when the larger rack size is used? Justify your answer.

8. Use Little's Law to estimate the average number of subassemblies waiting at the pick point. How much buffer space would you use at the pick point?

9. Compute the expected time to store a carrier in a bin after the SR machine is obtained.

10. Visit a manufacturing facility and observe the automated material handling equipment that is in use.
11. Make a list of the automated material handling equipment you have observed in the service systems you encounter regularly.

12. How much improvement is there in the AS/RS system if the speed of the SR machine increases by 100%.

13. How much improvement is there in the AS/RS system if the time between requests from the second manufacturing process is uniformly distributed between 10 and 30 seconds?

14. Perform additional simulation experiments to find the smallest difference between the starting time of the storage process (currently 6:00 A.M.) and the retrieval process (currently 8:00 A.M.) for which the system can effectively operate.

15. The current rack configurations are about one story high. Suppose a two story high configuration was preferred, specifically 18 bins high and 10 bins wide. Compare system performance using this configuration to the 10 bins high and 18 bins wide configuration.

16. Embellish the model in this chapter with acceleration and deacceleration of the SR machine. Assume the acceleration (deacceleration) distance is one bin in either direction and the average time to traverse this bin is twice that of other bins.

Case Problem

The benefits of AS/RS technology have been effectively realized in libraries. The amount of floor space required for books and periodicals has been reduced by ten-fold or more. The number of librarians required was reduced as well. Reshelving errors were eliminated. The location of each item while in the library is known with certainty. Despite these benefits, it is estimated that a few (less than 12) mini-load AS/RS systems have been installed in libraries.

This case problem involves determining the saturation point for a mini-load AS/RS system installed in a particular library. This is done by creating a graph of the cycle time for retrieving a book or periodical versus the arrival rate for such requests. The arrival rate resulting in the longest acceptable retrieval time is the saturation point. The smallest arrival rate of interest is 10 requests per hour. Assume that the arrival rate for retrievals is the same as the arrival rate for returns.

The mini-load AS/RS system installed in one particular library has a capacity of 250,000 books and periodicals. There is a single aisle with identical racks on each side. The system is installed inside a secured vault for safety and security reasons.

Books and periodicals are stored in carriers that are 4 feet deep and 2 feet wide. Each carrier row is one of three heights: 10, 12, or 15 inches. Each item is stored in the shallowest carrier in which it can stand. Thus, vertical space is used most efficiently. Assume that the number of books and periodicals of each height is the same.

There are 36 carrier rows on each side of the single aisle. The height of the first row is 10 inches, the second 12 inches, the third 15 inches, the fourth 10 inches and so forth. There are 60 carriers in each row.

The S/R machine travels at a high rate of speed: 12.6 feet/second horizontally and 4.3 feet/second vertically. Assume that the S/R machine must travel either horizontally or vertically but not diagonally.

The process of retrieving a book or periodical is the following. A patron makes a request using the electronic library catalog system. The AS/RS fills one request at a time. The location of the item is completely random. The S/R machine moves from its idle location to the required carrier, extracts the carrier in 3 seconds, and places the carrier in the pick and delivery station. A librarian must remove the desired item from the carrier and record its status in the information...
system. This takes 7 seconds. The S/R machine remains idle at the pick and delivery station.

Next the librarian determines whether any item that needs to be returned to storage is of the same size as the carrier. If so, the item's new carrier location is recorded in the information system and the item placed in the carrier. Both steps combined take 7 seconds.

Assume the library is open 16 hours per day, 7 days per week.

Embellishment: The AS/RS system tests the carrier for weight restrictions. One in 100 tests fail. In this case, the librarian must remove the item as well as the newly entered location from the information system in 7 seconds. In either case, the S/R machine replaces the carrier and returns empty to its idle location.

Embellishment: Find the saturation point when the following procedure is used. The S/R machine does not replace a carrier that is at a pick and delivery station until the next retrieval request is made. At that time, a carrier is first stored and then the next carrier retrieved.

Embellishment: Limit the number of carriers stored at the pickup/dropoff station to a total of three. When the fourth carrier arrives, it is immediate returned to the same storage location by the AS/RS machine.

Case Problem Issues:

1. How should carriers be modeled?
2. How should the location of the carrier containing the book or periodical requested be determined?
3. How should S/R machine travel time be computed?
4. Specify the process for book and periodical returns.
5. What are good initial conditions for this simulation experiment?
6. What performance measures, other than cycle time, would be of interest?
7. What is the expected utilization of the SR machine?
8. How should verification and validation evidence be obtained?

AutoMod Summary and Tutorial for the Chapter 6 Case Study

A.1. Introduction

AutoMod modeling constructs and experimental specifications generally needed for modeling arrivals, operations, and detractors such as rework, downtime, and setup / batching are presented. Example models illustrating routing and inventory dynamics are given as part of the application studies. A tutorial gives step-by-step instructions for building and simulating the model associated with the single workstation case study in Chapter 6.

A.2. AutoMod Modeling Elements

The application studies use primarily AutoMod modeling elements defined in Table A-1.

Table A-1: AutoMod Modeling Elements

<table>
<thead>
<tr>
<th>Modeling Element</th>
<th>Definition</th>
</tr>
</thead>
</table>

https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Book%3A_Beyond_Lean_-_Simulation_in_Practi…

Updated: Wed, 26 Aug 2020 00:49:44 GMT
Powered by
**Process**  
The steps used to model entity processing at a workstation as well as upon arrival or departure

**Loads**  
Entities

**Attributes**  
Entity attributes

**Resources**  
Resources

**Resource Cycles**  
The pattern of state changes of a resource due to the breakdown and repair cycle

**Counters**  
Resource-like variables used to model inventories

**Queues**  
Buffers or waiting areas

**Order Lists**  
A list of loads. Loads remain on the list until ordered to leave.

**Variables**  
State variables used throughout a model such as parameters of a processing time or characteristics of a resource

**Tables**  
The collection mechanism for performance measure observations not automatically maintained by AutoMod

**Random Streams**  
Pseudo-random number streams

In AutoMod, loads (entities in the text) flow through one or more processes. A process is described by a set of statements. AutoMod has many statements. Table A-2 describes some of the commonly used statements. A complete definition of each statement is provided in the AutoMod help system along with examples.

The user needs to be aware of one quirk in AutoMod, which expects models to have a visual component. Thus, entities must always be where they can be displayed graphically. For right now, this place is in a queue. Thus, while an entity is being processed by a resource, it must be in a queue. Thus, a single queue preceding a resource will contain the loads in the buffer as well as the loads in processing that is all the loads at the workstation. Alternatively, the user can employ one queue to represent the buffer where entities wait for a resource and a second queue to represent where an entity is graphically while it is being processed by the resource. The former approach will be used in this tutorial.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>begin</td>
<td>Start of a process or of a block of statements</td>
</tr>
<tr>
<td>end</td>
<td>End of a process or of a block of statements</td>
</tr>
<tr>
<td>set</td>
<td>Assign a value to a variable or attribute as well as changing the state or number of units of a resource or the value of a counter</td>
</tr>
<tr>
<td>send to</td>
<td>Send an entity to the start of another process</td>
</tr>
</tbody>
</table>

Table A-2: Commonly Used Statements
tabulate  Record the value of a performance measure (observed type)
clone  Create copies of an entity and send the copies to a process
move into  Enter a queue
wait for  Time delay for a process step
wait until <condition>  Delay until the condition (logical expression) becomes true
get  Acquire one or more units of a resource that are in the idle state. Same as: wait until <resource> is idle; make <resource> busy
increment  Add to the value of a variable or attribute as well as increasing the number of units of a resource or the value of a counter
decrement  Subtract from the value of a variable or attribute as well as decreasing the number of units of a resource or the value of a counter
wait to be ordered  Enter an order list
order  Send one or more loads on an order list to a process
while <condition> do begin end  While loop.

A-3. Tutorial – Model Building

This section shows how to build the single workstation model as specified in the chapter 6 case problem in AutoMod step-by-step.

1. Start AutoMod as you would any window program.
2. Choose FILE from the menu bar and then NEW. Specify the location you want for the model files in the directory structure.
3. Design the model.
   a. Decide what processes are necessary. In this case, use three processes: one for entity arrival, one for entity departure, and one for the workstation.
   b. Decide what attributes are necessary. In this case, arrival time is sufficient.
4. Define the arrival process. By convention, process names begin with P_. Choose PROCESS from the process system menu and then NEW. Give the name of the process (P_Arrive is good) and enter a title as documentation.
5. Select EDIT arriving procedure and the text editor appears. The statements for P_Arrive can be entered.
   a. Enter begin on the first line and end on the second line to delimit the procedure. Insert a comment line after
the first line to describe the procedure. Comments start with //. Comments may be placed on the same line
as statements.

b. The procedure P_Arrive must accomplish two things. The first thing is assigning the value of the time
between arrivals load attribute to the arrival time: set A_ArriveTime = ac, where ac is the current simulation
time (absolute clock).

c. The second thing is to send the arriving entity to the process for the workstation: send to P_WSA.

d. Terminate the edit using FILE then SAVE and FILE then EXIT. Notice that AutoMod will object that the load
attribute (A_ArriveTime) as well as the workstation process (P_WSA) have not as yet been defined. The
strategy that we are using is to define them at this point. In the error box for A_ArriveTime, choose define
and load attribute. In the attribute definition box, enter the name and a title for documentation as well as the
type as real. In the error box for P_WSA, choose define and process and then simply hit return to take all of
the defaults.

e. In the Edit a Process window, select OK.

6. Next choose PROCESS from the process system menu and edit P_WSA in the same way that P_Arrive was
created. The procedure must accomplish the following.

a. Enter the buffer of the workstation: move into Q_WS
b. Acquire the workstation resource: get R_WS
c. Perform processing: wait for RS_WS uniform 7.5, 1.5 min
d. Free the workstation resource: free R_WS
e. Send the load to the process for departing entities: send to P_Depart

7. Next choose FILE then SAVE and FILE then EXIT. Note that one queue, one resource, one random number
stream, and a process must be defined. Define a queue by specifying its name, a title, and capacity. The capacity
of Q_WSA is INFINITE.

a. Define a resource by specifying its name, a title, and default capacity (number of units), in this case one.
b. Define a random number stream by specifying its name: RS_WS.

8. P_Depart must accomplish the following.

a. Observe entity time in the system: tabulate (ac – A_ArriveTime) in T_LeadTime
b. Destroy the entity: send to die.

9. Choose File then SAVE and FILE then EXIT.

a. A table is defined by specifying its name and a title.

10. Define the load type for parts. From the process system menu, select Loads and then select New for a new load
type. Name the load L_Part.

a. Next select New Creation to specify the arrival process for loads.
b. Specify the time between arrivals as exponentially distributed with a mean of 10 minutes.
c. Specify the first arrival at time 0: Constant 0 in the First One at field.
d. Specify the first process as P_Arrive.

11. Define the load type for initial parts at the workstation at the start of the simulation. From the process system
menu, select Loads and then select New for a new load type. Name the load L_InitPart.

a. Next select New Creation to specify the arrival process for loads.
b. Specify the number of creations to be 3.
c. Specify the time between arrivals as a constant 0 so all the parts arrive at time 0
d. Specify the first arrival at time 0: Constant 0 in the First One at field.
e. Specify the first process as \( P_{\text{Arrive}} \).

f. Modify \( P_{\text{Depart}} \) so that data is not collected on the parts initially in the system, where \( \text{type} \) is a built-in load attribute:

\[
\text{if type} = \text{L Part tabulate (ac} - \text{A Arrive Time) in T Lead Time}
\]

12. Specify the length of the run as 168 hours. Select Run Control and new. Specify the snap (replicate) length as 168 hours.

13. Save the model.

14. Export the model: File/Export

15. Use the zip utility to create a zip file containing the exported (archive) version of the model: Programs/AutoMod/Utilities/Model Zip and select the model archive.

Note: The exported version of the model is a condensed version of the model suitable for sending by email. This is the version of the model that should be submitted.

**A-4. Tutorial – Model Execution**

The model can be run as follows.

1. Select RUN and then RUN MODEL.
2. The model will be compiled and a new window opened.
3. In the new window, select CONTROL and CONTINUE to run the simulation.
4. To make the model run faster, turn off animation: CNTL-G.
5. At the end of the run (or during the run), examine the reports for Processes, Queues, Resources, and Tables using VIEW and then REPORTS.
6. Use the information in the reports to obtain verification evidence.

**A-5. Tutorial – Modeling Extension**

Next close the execution window and return to the model. Save the model under a new name so that the modifications to follow are kept distinct from the original model.

The first modification is to model setup and batching at the workstation using the logic described in chapter 6. First determine the batch size using the computations in chapter 6. The enter setup and batching into the model as follows:

1. Modify \( P_{\text{Arrive}} \) to create a batch. Whenever the total number of arrivals to \( P_{\text{Arrive}} \) (\( P_{\text{Arrive total}} \)) is a multiple of the batch size, a batch is created. Thus, when a load arrives, test whether or not this condition if met. The expression:

\[
P_{\text{Arrive total}} \% V_{\text{Batchsize}}
\]

will be zero when a \( P_{\text{Arrive total}} \) is a multiple of the batch size. Recall that \( \% \) is the remainder operator.

a. If it is NOT met: \text{wait to be ordered on OL_BatchList} // hold load on batch list
b. If it is met: \text{send to P_WSA}

2. Save and exit. Define the order list OL_BatchList by giving its name and description.

3. Modify \( P_{\text{WS}} \) to process a batch. Between get R_WS and free R_WS, add the following

a. Wait for the setup time: \text{wait for 45 min}

b. Use a while <condition> do loop to model processing each item in the batch individually

i. \text{set V LoopIndex} = 0
ii. while V_LoopIndex < V_BatchSize do

iii. begin

iv. wait for RS_WS uniform 7.5, 1.5 min

v. increment V_LoopIndex by 1

vi. end

4. After free R_WS, send each individual load to P_Depart:
   a. order (V_BatchSize-1) loads from OL_BatchList to P_Depart

5. Save the model.

The second change is to add rework of a part to the model. This requires a little thought since loads in P_WS represent batches not parts. Here is one way this can be accomplished. Incrementing V_LoopIndex means that the part successfully completed. Thus, incrementing V_LoopIndex with the probability of completing a successful part would model part rework.

If RS_Rework uniform 0.5, 0.5 > 0.05 then increment V_LoopIndex by 1 // 0.05 is the probability that a part needs rework

The third change in the model involves a downtime repair cycle. Your tasks are as follows:

1. Create a new resource cycle and name it C_Bdown. Select Resources and then New for resource cycles. Select OK, edit to create the resource cycle.
2. Select MTTF/MTTR and fill in the required information.
3. Edit the resource WS to attach the resource cycle.
4. Save the model.

Follow the directions in IV above to make sure the model works by obtaining verification evidence.

A-6. Tutorial – Conducting Experiments with AutoStat

AutoStat is the component of the AutoMod simulation environment that is used to conduct simulation experiments. AutoStat is used after the model is built as well as verified and validated using the graphical execution component.

Start AutoStat from the build component menu: RUN, Run AutoStat. The AutoStat setup wizard will ask several questions. Answers can be modified later by selecting Properties from the menu bar. In answer the setup wizard questions, use the following information.

1. The model is random.
2. Answer no to the second question.
3. The model does not require warm-up.
4. The snap length is 168 hours.
5. It is fine to have the method of common random numbers as the default method.

Next conduct a simulation experiment as follows:

1. Define a new analysis of type single scenario.
2. In the pop-up box, give the analysis a name, specify 20 replications. Next select: OK do these runs.

3. Next from the main AutoStat window, select new responses to extract from the simulation runs the performance measure statistics of interest. In this case, select the mean lead time. This is done by choosing Table as the AutoMod entity and mean as the statistic of interest. A name should be specified as well. This step can be repeated for all performance measures of interest, such as utilization and maximum lead time.

4. View the performance measure values by selecting Analyses from the main AutoMod window and then the Run Results item under the name of the analysis of interest.

5. Copy the results to an Excel spreadsheet from the window where the run results are displayed. Select Edit/Copy Entire Table. In Excel, select Edit / Paste Special / Unicode Text.

One through five above should be done for each model, the original workstation model and the one with detractors

6. Analyze the simulation results using Excel. Create three columns: Replicate number (1-20), Lead Time for Original, Lead Time with detractors. Use the Excel function Transpose to place the simulation results in the proper column. Compute the difference in cycle time replicate by replicate in a fourth column. Compute summary statistics and t confidence intervals as appropriate. Use the Excel function TINV to return the appropriate critical values from the Student's t distribution with n-1 degrees of freedom.

A-7. Initialization of State Variables

Initialization of state variables, that is setting the value of a counter or a resource capacity (number of units of the resource) before the simulation begins, is important in some models. This is accomplished using the model initialization function, which AutoMod automatically executes before a model is simulated. There is at most one model initialization function per model.

A model initialization function is created as follows:

1. Select Source Files from the Process System panel.
2. Select New
3. For name, use logic.m
4. Select edit to open the editor.

The following example illustrates how to use the model initialization function. Assume the variables have been defined and given initial values in their definitions.

begin model initialization function
// Set the value of counter to target inventory value
// Note the current attribute of the counter must be referenced
\quad set C_Inventory current = V_TargetInventory

// Set the capacity of a resource (number of units) to the number of machines at a station
\quad set R_Station capacity = V_MachinesAtStation

return true //AutoMod requirement
end
A-8. Creating a Trace File in Comma Separated Value (.csv) Format

Consider the model of a single workstation with no detractors as described in section III above. Suppose a trace of all state changes: from idle to busy as well as from busy to idle is desired. This trace is to be written to a user defined comma separated value (.csv) file that can be opened in Excel. In the file, columns are delimited by commas. Every time Excel sees a comma, the following information is placed in the next column to the right. As well, such files can be opened in editors, like Notepad, in which the contents of the file including the commas can be seen.

The following example shows how to open .csv file in the model initialization function and write the column headers to the file.

begin model initialization function

// open the trace file; note that the variable V_TraceFile is of type file ptr (pointer)
// by Automod convention, the file will reside in the \arc directory for the model
\(\quad\)open "StateTrace.csv" for writing save result as V_TraceFile

// write the header to the trace file
\(\quad\)print "Clock, New State" to V_TraceFile

\(\quad\)return true //AutoMod requirement
end

Column values can be written in a similar way whenever desired. For example, the print statement to write the state change to busy to the trace file is as follows:

print ac, ", Busy" to V_TraceFile

A-9. Choose between Two Resources

Suppose an operation can be performed by either of two resources, R_MachineA or R_MachineB. The first resource with one unit in the idle state will be used. If both are available R_MachineA will be use. The following process fragment shows how to accomplish this. Note that A_Machine is load attribute of type resource ptr (resource name).

wait until R_MachineA remaining > 0 or R_MachineB remaining > 0 // wait for a machine
if R_MachineA remaining > 0 then
begin
\(\quad\)set A_Machine = R_MachineA // Machine A is available
end
else
begin
\(\quad\)set A_Machine = R_MachineB // Only Machine B is available
end
Distribution Function Fitting in JMP: Tutorial

B.1 Introduction

JMP is a general purpose data analysis software tool that includes fitting distribution functions to data. This tutorial leads the reader through a data fitting exercise for version 9 of JMP. Steps of the tutorial are shown in italics.

B.2. Procedures for Fitting Data to Distributions

Start up JMP in the usual way for a Windows program.

Select View / JMP Starter

Within JMP Starter, Select New Data Table.

Within New Data Table, Select File / Open to load the file with the data to be fit. The file is a .txt file. The data in the file will appear in a spreadsheet-like table.

Next select Basic from the category column.

Next select Distribution. Click in the box to the right of: Y, columns. Then double click on column 0. Then select OK.

A box appears containing statistical summaries of the data set. Examine these carefully.

Next see how well the data fits a normal distribution. Click the arrow next to the column label 0. Select Continuous Fit then normal distribution. Look at the normal distribution superimposed on the histogram.

Next test the fit. Click the arrow next to Fitted Normal. Select Goodness of Fit. Note that the fit to a distribution is not adequate.

Let go back and re-examine the data values. Assume that a zero value represents a no ship condition and that we are interested in the distribution of the volume shipped given that shipments were made. Let’s eliminate the zero values and refit the distribution. Select the first six rows in the data table by selecting the row numbers 1 through 6. Select the arrow next Rows and then Exclude / Unexclude.

Repeat the above process for fitting a distribution function to the data.

In addition, repeat all of the above for the gamma distribution. Which fits better in your opinion, the normal or the gamma?

Bibliography


Gorman, M. F., Hoff, J. & Kinion, R. (2009). Tales from the front: case studies indicate the potential pitfalls of misapplication of lean improvement programs. *Interfaces* 39 (6).


