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

Modeling Element	Definition
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, whick 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 A-2: Commonly Used Statements

Statement	Definition
begin	Start of a process or of a block of statements
end	End of a process or of a block of statements
set	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
send to	Send an entity to the start of another process
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
free	Free one or more units of a resource placing them in the idle state. Same as: make <resource> idle
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 windows 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_Arrive.
    f. Modify P_Depart so that data is not collected on the parts initially in the system, where **type** is a built-in load attribute: **if type = L_Part tabulate (ac – A_ArriveTime) in T_LeadTime**
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_Arrive to create a batch. Whenever the total number of arrivals to P_Arrive (**P_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_Arrive total % V_Batchsize** will be zero when a P_Arrive total is a multiple of the batch size. Recall that % is the remainder operator.
   a. If it is NOT met: **wait to be ordered on OL_BatchList // hold load on batch list**
   b. If it is met: **send to P_WSA**
2. Save and exit. Define the order list OL_BatchList by giving its name and description.
3. Modify P_WS to process a batch. Between get R_WS and free R_WS, add the following
   a. Wait for the setup time: **wait for 45 min**
   b. Use a while <condition> do loop to model processing each item in the batch individually
      i. **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
        set C_Inventory current = V_TargetInventory

// Set the capacity of a resource (number of units) to the number of machines at a station
        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
        open "StateTrace.csv" for writing save result as V_TraceFile

// write the header to the trace file
        print “Clock, New State” to V_TraceFile

        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
        set A_Machine    = R_MachineA          // Machine A is available
end
else
begin
        set A_Machine    = R_MachineB          // Only Machine B is available
end

get A_Machine  // get selected machine
wait for 15 min  // perform operation
free A_Machine // free selected machine
```

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 interest 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

Askin, R. G. & Standridge, C. R. (1993). *Modeling and analysis of manufacturing systems*. John Wiley and Sons, New York.

Askin R.G. & Estrada, S., (1999). Investigation of cellular manufacturing practices. *Handbook of cellular manufacturing systems*, S. Irani, ed., John Wiley & Sons, Inc., New York.

Askin, R. G. & Goldberg, J. B. (2002). *Design and analysis of lean production systems,* John Wiley & Sons, New York.

Balci, O. (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of Operations Research*, 53:121-173.

Balci, O. (1996). Principles of simulation model validation, verification, and testing. *International Journal in Computer Simulation.*

Banks, J., Carson II, J. S., Nelson, B. L. & Nicol, D.M. (2009). Discrete-event system simulation, 5th ed**.** Prentice Hall. Englewood Cliffs, NJ.

Bozer, Y. A., & White, J. A. (1984). Travel-time models for AS/RS. *IIE Transactions*, 16(4), 329-338.

Bozer, Y. A., & White, J. A. (1996). A generalized design and performance analysis model for end-of-aisle order-picking systems. IIE Transactions, 28(4), 271-280.

Buzacott, J. & Hanifin, L. (1978). Models of automatic transfer lines with inventory banks: a review and comparison. *AIIE Transactions*, 10, 197-207.

Carson II, J. S. (2002). Model verification and validation. *Proceedings of the 2002 Winter Simulation Conference***.** Paper presented at the 2002 Winter Simulation Conference: San Diego, California. Retrieved June 12, 2010 from http://www.informs-sim.org/wsc02papers/008.pdf

Conway, R., Maxwell, W., McClain, J. O., & Thomas, L. J. (1988). The role of work-in-process inventory in serial production lines. *Operations Research*, 35(2), 291-305.

Devore, J. L. (2008). *Probability and statistics for engineering and the sciences, 7th ed.,* Duxbury Press, Belmont, CA.

Duinkerken, M. B., Ottjes, J. A., & Lodewijks, G. (2006). Comparison of routing strategies for AGV systems using simulation. Proceedings of the 2006 Winter Simulation Conference. Paper presented at the 2006 Winter Simulation Conference: Monterey, California. Retrieved September 12, 2011 from http://www.informs-sim.org/wsc06papers/193.pdf

Elsayed, E. A. & Unal, O. I. (1989). Order batching algorithms and travel-time estimation for automated storage / retrieval systems. *International Journal of Production Research*, 27(7), 1097-1114.

Ekren, B. Y. & Heragu, S. S. (2009). Simulation based regression analysis for rack configuration of autonomous vehicle storage and retrieval system. Proceedings of the 2009 Winter Simulation Conference, Paper presented at the 2009 Winter Simulation Conference: Austin, TX. Retrieved October 16, 2011 from http://www.informs-sim.org/wsc09papers/232.pdf.

Flanigan-Wagner, M. A. & Wilson, J. R. (1995). Graphical interaction simulation input modeling with bivarate bezier distributions. *ACM Transactions on Modeling & Computer Simulation*, 5(3), 163-189.

Flanigan-Wagner, M. A. & Wilson, J. R. (1996). Using univariate bezier distributions to model simulation input processes. *IIE Transactions*, 28(9), 699-712.

Ferrin, D. M., Miller M. J., & Muthler D. (2005). Lean sigma & simulation, so what's the correlation? *Proceedings of the 2005 Winter Simulation Conference*. Paper presented at the 2005 Winter Simulation Conference: Orlando, Florida. Retrieved June 5, 2010 from http://www.informs-sim.org/wsc05papers/249.pdf.

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).

Grimard, C., Marvel, J. H. & Standridge, C. R. (2005). Validation of the re-design of a manufacturing work cell using simulation. *Proceedings of the 2005 winter simulation conference.* Paper presented at the 2005 Winter Simulation Conference: Orlando, Flordia. Retrieved September 9, 2010 from http://www.informs-sim.org/wsc05papers/170.pdf.

Han, M.-H., McGinnis, L. F., Shieh, J. S. & White, J. A. (1987). On sequencing retrievals in an automated storage / retrieval system, *IIE Transactions*, 19(1), 56-66.

Hopp, W. J. & Spearman, M. L. (2007). Factory physics: foundations of manufacturing management, 3rd edition, McGraw-Hill/Irwin, New York.

Hyden, P., Roeder, T., & Schruben, L. (2001). Resource graphs for modeling large-scale, highly congested systems. *Proceedings of the 2001 winter simulation conference***.** Paper presented at the 2001 Winter Simulation Conference: Arlington, Virginia. Retrieved September 9, 2010 from http://www.informs-sim.org/wsc01papers/068.PDF.

Irani, S. A., Subramanian, S., & Allam, Y. S. (1999). Introduction to cellular manufacturing systems. *Handbook of cellular manufacturing systems*, S. Irani, ed., John Wiley & Sons, Inc., New York.

Jing, G. G., Kelton, W. D., Arantes, J. C., & Houshmand, A. A. (1998). Modeling a controlled conveyor network with merging configuration. Proceedings of the 1998 Winter Simulation Conference, Paper presented at the 1998 Winter Simulation Conference: Washingon, DC. Retrieved September 27, 2011 from http://www.informs-sim.org/wsc98papers/142.pdf.

Jones – Lang- Lasalle. (2008). Lean practices in the supply chain. Retrieved May 8, 2011 from http://www.joneslanglasalle.com/Documents/JLL-LeanPracticesInSupplyChain.pdf

Keller, G. (2001). *Applied statistics with Microsoft Excel*, Duxbury Press, Belmont, CA.

Koo, L. Y., Chen, Y., Adhitya, A., Srinivasan, R., & Karimi, I. A. (2006). Evaluating refinery supply chain policies and investment decisions through simulation-optimization. Proceedings of the 2006 Winter Simulation Conference. Paper presented at the 2006 Winter Simulation Conference: Monterey, California. Retrieved September 12, 2011 from http://www.informs-sim.org/wsc06papers/181.pdf

Law, A. M. (2007). Simulation modeling & analysis, 4th edition, McGraw-Hill, New York.

Law, A. M. & McComas, M. G. (2001). How the EXPERFIT distribution fitting software can make your simulation models more valid. *Proceedings of the 2001 Winter Simulation Conference.* Paper presented at the 2001 Winter Simulation Conference: Arlington, VA.

Law, A. M. & McComas, M. G. (1996). EXPERFIT: total support for simulation input modeling. *Proceedings of the 1996 Winter Simulation Conference.* Paper presented at the 1996 Winter Simulation Conference: Coronado, California.

Learnsigma. (2007). What is lean manufacturing? Retrieved February 13, 2009 from http://learnsigma.wordpress.com/2007/11/10/what-is-lean-manufacturing/.

Lehmer, D. H. (1951). Mathematical methods in large-scale computing units. *Annals of the Computing Laboratory of Harvard University*, 26,141-146.

Liu, R, Kumar, A., & Stenger, A. J. (2006). Simulation results for supply chain configurations based on information sharing. Proceedings of the 2006 Winter Simulation Conference. Paper presented at the 2006 Winter Simulation Conference: Monterey, California. Retrieved September 12, 2011 from http://www.informs-sim.org/wsc06papers/076.pdf

Marvel, J., & Standridge, C. (2009). A Simulation-enhanced lean design process. *Journal of Industrial Engineering & Management*, 2(1), pp 90-113. Retrieved June 5, 2010 from http://www.jiem.org/index.php/jiem/article/viewFile/61/18.

Marvel, J.H., Schaub, M.A., & Weckman, G.R. (2008). Assessing the availability & allocation of production capacity in a fabrication facility through simulation modeling: a case study. *International Journal of Industrial Engineering*, 15(2), 166-175. Abstract retrieved June 5, 2010 from http://ijietap.utep.edu/ojs/index.php/ijie/article/view/117.

Material Handling Industrial Association. (2011). Automated Guided Vehicle Systems Industry Group. Retrieved October 2, 2011 from http://www.mhia.org/industrygroups/agvs.

Miller, G., Pawloski, J. & Standridge, C. (2010). A Case Study of Lean, Sustainable Manufacturing. *Journal of Industrial Engineering & Management*, 3(1), 11-32. Retrieved September 27, 2011 from http://www.jiem.org/index.php/jiem/article/viewFile/156/50.

Mittal, S. & Wang, H.-P. (1992). Simulation of JIT production to determine number of kanbans. *International journal of advanced manufacturing technology*, 7, 292-308.

Nazzal, D. & McGinnis, L. F. (2006). An analytical model of vehicle-based automated material handling systems in semiconductor fabs. Proceedings of the 2006 Winter Simulation Conference. Paper presented at the 2006 Winter Simulation Conference: Monterey, California. Retrieved October 2, 2011 from http://www.informs-sim.org/wsc06papers/239.pdf

Pritsker, A. A. B. (1989). Why simulation works. Proceedings of the 1989 Winter Simulation Conference, Paper presented at the 1989 Winter Simulation Conference: Washington, D. C.

Pritsker, A. A. B. (1977). Modeling & analysis using Q-GERT networks. Halsted Press, New York.

Pritsker, A.A.B. (1967). Application of multichannel queueing results to the analysis of conveyor systems. *Industrial Engineering*, 17, 14-21.

Rosenblatt, M. J., Roll, Y. & Zyser, V. (1993). A combined optimization and simulation approach for designing automated storage / retrieval systems, *IIE Transactions*, 25(1), 40-50.

Rother, M. & Harris, R. (2001). *Creating continuous flow: an action guide for managers, engineers, and production associates*. The Lean Enterprise Institute, Brookline, MA.

Rubrich, L. & Watson, M. (1998). Manufacturing cells: One piece flow and the key to employee empowerment and ownership. In *Implementing World Class Manufacturing*. WCM Associates, Fort Wayne Indiana.

Sargent, R. G. (2009). Verification & validation of simulation models. Proceedings of the 2009 Winter Simulation Conference. Paper presented at the 2009 Winter Simulation Conference: Austin, Texas. Retrieved June 5, 2010 from http://www.informs-sim.org/wsc09papers/014.pdf.

Sargent, R. G. (2012). Verification and validation of simulation models. *Journal of Simulation* (2012), 1–13. doi:10.1057/jos.2012.20.

Schmeiser, B. W. (1980). Random deviate generation: a survey. Proceedings of the 1980 Winter Simulation Conference. Paper presented at the 1980 Winter Simulation Conference:Orlando, Florida.

Schonberger, R. J. (2011). Taking the measure lean: efficiency and effectiveness, parts I and II. *Interfaces* 41(2).

Schruben, L. W. (1983). Simulation modeling with event graphs. *Communications of the ACM,* 26(11), 957-965.

Schruben, L. W. (1995). Graphical simulation modeling and analysis: sigma for windows. Duxbury Press, Pacific Grove, CA.

Sekine, K. (1992). *One-piece flow: cell design for transforming the production process*. Productivity Press, Portland, Oregon.

Shannon, R E. (1975). System simulation: the art & science, Prentice-Hall, Englewood Cliffs, N.J.

Shapiro, J. F. (2007). *Modeling the supply chain 2nd edition,* Cengage Learning, Florence, KY.

Standridge, C. R., & Marvel, J. H. (2006). Why lean needs simulation. Proceedings of the 2006 Winter Simulation Conference. Paper presented at the 2006 Winter Simulation Conference: Monterey, California. Retrieved June 5, 2010 from http://www.informs-sim.org/wsc06papers/244.pdf

Standridge, C. R. & Heltne, D. R. (2000). An MSE-based simulation capability for strategic & tactical logistics. Proceedings of the 2000 Winter Simulation Conference. Paper presented at the 2000 Winter Simulation Conference: Orlando, Florida. Retrieved June 5, 2010 from http://www.informs-sim.org/wsc00papers/147.PDF.

Standridge, C. R. (1998). Manufacturing system simulation. In concurrent design of products, manufacturing processes, & systems, H-P Wang, ed. Gordon & Breach Science Publishers, London.

Taj S., Cochran, D. S., Duda, J. W. & Linck, J.(1998). Simulation and production planning for manufacturing cells. Proceedings of the 1998 Winter Simulation Conference, Paper presented at the 1998 Winter Simulation Conference: Washingon, DC. Retrieved September 9, 2010 from http://www.informs-sim.org/wsc98papers/131.PDF.

Vardeman, S. B. & Jobe, J. M. (2001). *Basic engineering data collection and analysis*. Duxbury / Thompson Learning.

Vasudevan, K., Lote, R., Williams, E, & Ulgen, O. (2009). High speed bottle manufacturing lines: case studies and simulation software selection techniques. Proceedings of the 2009 Winter

Simulation Conference, Paper presented at the 2009 Winter Simulation Conference: Austin, TX. Retrieved September 27, 2011 from http://www.informs-sim.org/wsc09papers/030.pdf.

Warber, M. & Standridge, C. R. (2002) Material handling expansion for a package routing hub. Proceedings of the 2002 International Mechanical Engineering Congress and Exposition. Paper presented at the 2002 International Mechanical Engineering Congress and Exposition: New Orleans, LA.

Wilson, J. R. & Pritsker, A. A. B. (1978). A procedure for evaluating startup policies in simulation experiments, *Simulation*, 31, 79-89.