HOW TO AVOID "HARD CODING" VARIABLES IN YOUR SIMULATION MODEL

Haldun Çelik

Production Modeling Corporation
Three Parklane Boulevard
Suite 1006 West
Dearborn, Michigan 48126, U.S.A.

John Shore

Production Modeling Corporation
Three Parklane Boulevard
Suite 1050 West
Dearborn, Michigan 48126, U.S.A.

ABSTRACT

The purpose of this paper is to demonstrate the ability of AutoMOD as a flexible modeling tool. Simulation models of material handling systems like those found in automotive paint shops & assembly systems often require many variables, conveyor speeds, and complex logic. These systems have many different variables associated with them. The most important ones are: conveyor speeds, carrier counts, routing information, floating carrier numbers between two control points, active and inactive station information, station operation times, and shift schedules. In order for a simulation analyst to control or experiment with a material handling system, he or she will usually have to vary one or more of these variables. In the past, this has meant that these "hard-coded" variables would have to be changed one by one through menus for each material handling segment or system. Other methods including the use of AutoStat or AWK could be used to change certain parameters without having to visit the edit menus in AutoMOD. This paper will address how users can use external data files to control these traditionally "hard-coded" variables, thereby allowing simulation analysts and their customers to drive their material handling systems as they have traditionally used external files to provide process information for their models.

PMC's approach was to create a flexible and robust model that imports all of these system parameters from external data files. This allows for an increase in model efficiency if model experimentation is large and involves many different physical changes. Using this method, a potential customer can modify these parameters for experimentation with a run-time version of AutoMOD simulation software.

1. INTRODUCTION

Discrete event simulation can be one of the most powerful tools available to industrial engineers. Simulation programs allow us to predict overall system performance like the throughput of a system. It is also beneficial in identifying bottlenecks, evaluating proposed alternatives for eliminating bottlenecks, identifying under utilized resources, and determining buffer sizes just to name a few. Some simulation packages are more flexible than others. Simulation tools like AutoMOD offer users the ability to investigate many diverse process systems from automotive manufacturing plants to hospital operating wings.

Traditionally, simulations are developed based on a pre-determined base layout with appropriate assumptions and data. However, in the past few years, companies are starting to use simulation in the very early stages of process design. In this phase, many of the parameters that simulation modelers have come to expect to be available for use as an input to model are unavailable. During this phase, system variables such as location of conveyors or stations, operation times, pallet or carries counts, shift patterns, etc. are changing continuously. If all these variables and parameters are hard-coded into the model, too much time, resources, and overhead will be spent to make changes in the model. To prevent this, a flexible and reusable simulation model should be developed by simulation engineers.

Even if the model is created later in the design phase, creating flexible models becomes an important issue. Here more information is known about the process layout; however, behavior of that system under many different parameters may not be known. During the experimentation phase of an analysis, for example, most simulation engineers have to conduct different what-if scenarios and alternatives to determine the effect on system performance measures like the throughput of the system. Based on the findings and result of experiments, the customers may want to run more scenarios by changing the speed of a conveyor, the location of a station, or the cycle time of the operator. With most software packages, the simulation engineer has to update many dialog boxes or menus to vary even the simplest parameters of a model. The probability of making errors can be very high leading to faulty results. In addition, if there is a gap between the time the model was originally created and the next time it is used, there is a greater possibility that the simulation engineer will not

remember how to modify the simulation model. This will also increase the likelihood of an error.

2. WHY DEVELOP THE FLEXIBLE SIMULATION MODEL?

One of the approaches to create a flexible model is to import the system variables and process variables into the model from external data files. The ability to configure models like this enables the model creator and users alike to change the attributes of the system without "hard coding" each element for each new change. This is advantageous to both parties because it minimizes the amount of actual simulation modeling knowledge needed by the end user. It also puts the configurable data in a form more recognizable to the person using the model.

Parameters and variables that can be converted to this flexible form include:

System Variables

1) Conveyor speeds
2) Load variation for various carrier types
3) The number of carriers, or pallets, or loads
4) Routing Information
5) Floating carrier numbers between two control points
6) Ratio information for merging and splitting lines

Station Variables

1) Active/Inactive toggle effect
2) Station operation times
3) Hourly loading rates
4) Part delivery amounts into system

Shift Variables

1) Shift operating patterns
2) Individual shift events (shift lengths, breaks, time between shifts)
3) Model run length (for example; 15 day run length)

The model also can be designed to create simulation output reports based on statistics generated from model runs, reports to contain critical variables and customized parameters. The output reports include:

Output Reporting

1) Throughput of the system
2) Load, resource, or carrier utilization,
3) Number of loads, or carriers, in the system,
4) Number of times each carrier was used throughout specified time period
5) Station operation information such as cycle time, utilization, and availability.

After determining which variables impact the analysis, the simulation engineer can set up variables that should be controlled externally instead of hard-coding their values into a model. In order to do this, it is very important to choose the right software that will be able to read and write data from an external data file. AutoMOD has the capability of reading different types of variables from external data files, setting the speed of conveyors from external files, and controlling the capacity of counters while the model is running.

3. CREATING THE FLEXIBLE MODEL IN AUTOMOD.

3.1 Speed Information Files

As an example, let's imagine that an analyst is modeling a manufacturing plant, which includes different type of material handling system, buffers, or stations with different operation times.

If this analyst needed to readily change the speeds of different conveyor systems " on the fly" or had a customer who wanted to configure those conveyor speeds from a text file. The analyst can create a text file that contains the conveyor, name of the section, and speed information (see Figure 3.1).

Conv1	*conv:chain299*	*0.06*
Conv2	*conv:chain432*	*0.75*
Pow_free1	*pow_free:chain3*	*0.75*
Pow_free2	*pow_free:chain541*	*1.00*
Pow_free3	*pow_free :chain46*	*0.75*
Pow_free4	*pow_free:chain476*	*3.00*

Figure 3.1 An Example of a Conveyor Speed External Data File.

The source code to set the speed of one particular section in AutoMOD would be written as follows:

```
set vse_ConvSection ( 1 ) velocity = vr_Conv_Speed ( 1 )
set vse_ConvSection ( 2 ) velocity = vr_Conv_Speed ( 2 )
```

The variable that holds the conveyor section name is: vse_ConvSection (X). The speed information associated with the conveyor section is: vr_Conv_Speed.

3.2 Operation Time Information Files

In this same model, the analyst may want to control the cycle time of a particular station from an external file. Figure 3.2 shows a sample operation time contained in an external data file. In this case, there is a triangular distribution associated with this particular station. The AutoMOD code reads the file from left to right. The first string of characters is the name of the station ("st_Loading1"). The minimum operation time value is 8 minutes, the most likely value is 10 minutes , and the maximum operation time value is 11 minutes.

st_Loading1	*8*	*10*	*11*
st_Loading2	*7*	*10.5*	*11.5*
st_Operation1	*17*	*20*	*23*
st_Operation2	*12*	*14*	*18*
st_Operation3	*8*	*10*	*11*
st_Unloading1	*9*	*13.1*	*16*
st_Unloading2	*8*	*12*	*18*

Figure 3.2 Sample Operation Time Data in an External File.

The AutoMOD "Read" action is used to capture the operation time information for a particular station. The following command line demonstrates how the information would be used after it is read into the model.

wait for triangular vr_OptTime (1,2), vr_OptTime(1,3), vr_OptTime(1,4) min

The variable vr_OptTime(X,X) stores the data representing the minimum, the most likely, and the maximum operation time imported from the external data file.

3.3 Carrier or Pallet Information Files

Another variable easily controlled via an external data file is the number of pallets introduced into a material handling system. Figure 3.3 shows a sample data file, which contains Carrier Information.

Carrier1 Part1	*20*
Carrier2 Part2	*30*
Carrier3 Part3	*40*

Figure 3.3 Sample Pallet Count Information Found in an External Data File.

This data file contains:

a) The carrier type (Carrier1)
b) The load type which that particular carrier carries (Part1)
c) The maximum carrier number (20).

3.4 Routing Information Files

Material handling carrier routing is a more powerful and robust example of external data control. In this example, the modeler will control the route an entity will take in a model simply by changing a number or sequence of numbers in an external file. Based on the data read from routing information data file, the load, pallet, or carriers can be send to the particular stations in a particular order (see Figure 3.4).

Part1	2, 3, 5, 4, 6, 7, 9, 4, 2, 0
Part2	3, 2, 4, 5, 6, 13, 14, 12, 1 , 0

Figure 3.4 Sample Routing Data Found in an External Data File.

The first line of the data file in Figure 3.4 is:

Part1 2, 3, 5, 4, 6, 7, 9, 4, 2, 0

This can be interpreted as: part1 should proceed to station2, then to station3, then to station5, etc.

3.5 Buffer Capacity Information Files

Data files can also be used to control buffer sizes between stations. The data file in Figure 3.5 can be used to set buffer capacity information, in this case the maximum. It would control the maximum entities (loads, pallets, carriers, etc.) between two consecutive control points.

Buffer_Loading1_Station1	2
Buffer_Station1_Station2	3
Buffer_Station2_Station3	2

Figure 3.5 Sample Buffer Data File

In AutoMOD, counters can be used to control the maximum number of loads between two control points. An example of how this would be done is as follows:

set cLoadBufferMax(1) capacity = vi_LoadBufferMax(1,2)

3.6 Schedule Information Files

Many times shift patterns of a facility can significantly impact the output of that operation. At the very least, the shift patterns will disrupt the flow of materials, especially in facilities that have many manual operations. In order to observe the impact that a shift pattern will have on a facility, it needs to be configured into the model in such a way that it can be easily changed from one experimental run to another.

Figure 3.6 contains the daily scheduling data information such as working hours, break times, and shift start and stop time for a particular facility. The first data line of the data file identifies the shift and the operating time span for that shift. The next data lines include the start and stop time during the shift schedule. This would represent any break times or maintenance times associated with a particular shift. This information is repeated for each shift.

Shift Numbers Required:	*1*	*2*	*0*
1.Shift(Begin-End):	*6.00*	*14.00*	
1.Break Time(Begin-End):	*9.00*	*9.17*	
2.Break Time(Begin-End):	*12.00*	*12.50*	
3.Break Time(Begin-End):	*13.00*	*13.10*	
4.Break Time(Begin-End):	*0.00*	*0.00*	
5.Break Time(Begin-End):	*0.00*	*0.00*	
6.Break Time(Begin-End):	*0.00*	*0.00*	
2.Shift(Begin-End):	*14.20*	*22.20*	
1.Break Time(Begin-End):	*15.00*	*15.17*	
2.Break Time(Begin-End):	*18.00*	*18.50*	
3.Break Time(Begin-End):	*20.00*	*20.17*	
4.Break Time(Begin-End):	*0.00*	*0.00*	
5.Break Time(Begin-End):	*0.00*	*0.00*	
6.Break Time(Begin-End):	*0.00*	*0.00*	
3.Shift(Begin-End):	*14.20*	*22.30*	
1.Break Time(Begin-End):	*15.00*	*15.17*	
2.Break Time(Begin-End):	*18.00*	*18.50*	
3.Break Time(Begin-End):	*20.00*	*20.27*	
4.Break Time(Begin-End):	*0.00*	*0.00*	
5.Break Time(Begin-End):	*0.00*	*0.00*	
6.Break Time(Begin-End):	*0.00*	*0.00*	

Figure 3.6 Sample Shift Schedule File

For example, the first line in the file:

1.Shift(Begin-End) : 6.00 14.00

can be interpreted as the first shift. This shift begins at 6:00 AM and ends at 2:00 PM. The next line:

1.Break Time(Begin-End): 9.00 9.17

can be interpreted as the first break time. This break time begins at 9:00 AM and ends at 9:17 AM.

This data file is an example for a typical working schedule information. Once all the required data is read into the variables, they can be used to shut down or bring up the conveyor systems and resources in the simulation model.

3.7 Station Data Files

In a simulation model, the ability to turn stations on or off becomes important. For example, if the analyst is trying to determine the effect of having fewer stations on an assembly line, he or she could use an external data file to toggle various stations on or off. The analyst could also use this method to isolate certain stations from others to get their stand-alone throughput rates.

Loading Station #1		On
Station1		On
Station2	Off	
Station3		On

Figure 3.7 Sample Station Toggle Data File

Using the station data information, a pre-located station can be turned on and turned off based on the data that is read from external data files.

An example of AutoMOD source code that would control this action is:

```
if vs_StationToggle(1) = "On" then
   begin
      wait for vr_StationOpTime(1) min
      travel to vcp_Station(2)
   end
else if travel to vcp_Station(2)
```

In this example, if the string type variable *vs_StationToggle* is "ON" then load will wait at this station for a certain amount of time. If the same variable is set to "OFF," the load will pass right through to the next station.

4.0 USE OF FLEXIBLE MODEL

An appropriately designed flexible simulation can be used to do analysis quickly and easily. The model creator can also hand off their work to non-simulation experts. This attribute can be particularly attractive to customers who have only a runtime simulation. It is also beneficial when the same model must be reused for future investigations.

It is important to remember that the simulation package that will be used to develop the flexible model should have the ability to read data form external data files. It should have the ability to set certain variables on the fly such as resource information or material handling speed information. The software should also have the ability to update the animation of the model as the data values change. This will help the end user to understand the results of particular scenarios and minimize confusion associated with model data changes.

ACKNOWLEDGMENT

Edward J. Williams, Senior Technical Specialist, Ford Motor Company, contributed numerous suggestions to improve the clarity and organization of this paper.

APPENDIX: TRADEMARKS

AutoMOD is a registered trademark of AutoSimulations, Incorporated.

REFERENCES

Williams, Edward J. 1997. How simulation gains acceptance as a manufacturing productivity improvement tool. In *Proceedings of the 1997 European Simulation Multiconference*, eds. Ali Rıza Kaylan and Axel Lehmann, P-3 – P-7.

Williams, Thomas A. 1997. Changing the conveyor 'picture'. *Assembly* 40(8):34-40.

AUTHOR BIOGRAPHIES

HALDUN ÇELİK holds a bachelor's degree in Mechanical Engineering (İstanbul Technical University-İstanbul, Turkey - 1993). He worked at Seta Engineering, İstanbul, Turkey as the leader of the Automation Department from 1993 to 1995. From January 1996 to the present, he has been an Applications and Control Engineer at Production Modeling Corporation, Dearborn, Michigan. In this position, he has become highly expert in simulation and facilities-layout optimization systems, including AutoMOD, Taylor II, LayOPT, and ROBCAD. His work currently emphasizes real-time simulation using WONDERWARE, STEEPLECHASE, and PLC control programming tools. Additionally, he is studying toward a Master's degree in Engineering Management at the University of Michigan.

JOHN SHORE holds a Master of Science Degree in Manufacturing Systems Engineering from the University of Michigan – Dearborn. He also holds a Bachelor of Science Degree in Industrial Engineering from the University of Michigan – Ann Arbor. He currently works as the Operations Manager for Production Modeling Corporation in Dearborn, Michigan. He has worked in the past for General Motors Corporation as a Senior Industrial Engineer. There he worked as an internal throughput improvement consultant and as a simulation coordinator. He is a graduate of the "Jonah" program at the Avraham Goldratt Institute. He has used AutoMOD for over 7 years and is considered an expert user of the software.