Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

THE MICROMODELS SHOOTING PROCESS

Ricardo F. Garzia

JBG Software, Inc.
Columbus, Ohio 43213

1. ABSTRACT

This paper presents three new concepts for the discrete-event simulation environment. They are: *self-generating models, micromodels adjustments*, and *automatic simulations*. The impact of these new concepts can be summarized as follows: the development of the model will be accomplished in a few minutes, rather than days, weeks or even months; and what is most important, they place modeling and simulation at the manager's fingertips, facilitating the daily decisions needed in Automated Manufacturing Plants. These new concepts are implemented in two software systems: AMMG (Automated Manufacturing Models Generation), and SHOOTING (The Micromodels Shooting Process).

2. INTRODUCTION

Since their introduction into the engineering environment, discrete-event simulation languages have always been used in the same manner. Once the model is developed, it is validated and finally simulated. This process takes a long time for its completion, thus it is not suitable for day-to-day management decisions, which may often require changes in the model's parameters and model's logic. This situation becomes more evident for the management of Automated Manufacturing Plants.

This paper describes a novel approach which allows modeling and simulation to become an effective management tool in the administration of manufacturing plants.

Although discrete-event simulation languages have evolved into sophisticated modeling support tools, the application of these simulation languages in the manufacturing environment, as well as in today's automated manufacturing environment, has not undergone any drastic changes. A model representing the operational behavior of the manufacturing assembly for example, in which the facility's management can base their decisions, requires the ability of fast updating to adjust the products in manufacturing to the current operations of the facility. Management decisions will be made upon the results of the simulation.

Within the above framework a model must be developed describing the factory operation. For this task, the professional services of a modeler with expertise in these techniques will be required, first he/she has to learn the process to be modeled, develop an accurate model and then validate it. In using this approach several questions arise, some of these are:

1. What are the effects of daily factory changes?
2. How can they be evaluated?
3. How can management use the modeling and simulation approach?

Thus in today's automated manufacturing environment management must make decisions concerning a completely dynamic environment. In order to find the best approach, since there is still the need for a model of the factory, it is necessary to introduce new concepts in the simulation environment to satisfy our increasing demands. These new concepts are:

Self-Generated Models- Model generation must be easy to accomplish by an English dialogue, and must not involve any modeling methodology concepts. The model is generated through a series of questions related to the manufacturing processes.

Micromodels Adjustments- The micromodels adjustments are used on-line (during the simulation) to modify the model's parameters, and also, if necessary, to change the model's logic (as in the case of evaluating disruption analysis). Each micromodel adjustment is created through a series of English questions concerning the management decisions to be made.

Automatic Simulations- This feature allows the execution of large simulations, subject to the model alterations, issuing partial simulation reports as requested.

These three new concepts have been developed in two software systems, AMMG and SHOOTING, which allow the self-generation of the models and the file to run the simulation session.

The application of these new concepts allows the automatic execution of a simulation, for long simulation times (a day, week, month, or even years), subject to model parameters changes and/or model logic changes at specified simulated times during the simulation session. They also provide simulation reports as requested. The simulation reports can be obtained during its execution without interrupting the run.

The combination of a discrete-event simulation language plus these concepts, provides a dynamic environment in which the model developed will run subject to the alterations prescribed by the user.

3. REQUIRED LANGUAGE CHARACTERISTICS

The discrete-event simulation language must have special characteristics to allow the implementation of these new concepts in a very simple manner. These include:

No Compilation- When the model is completed, it is ready to be executed. There is no need to compile.

Numbered Statements- Each statement in the model must be associated with a number.

Numbering System- For statement numbers the model must accept integer numbers as well as real numbers. Therefore, there is no practical limitation to the number of statements which can be placed between any two consecutive statements.

Overlapping Condition- Any new statement, with the same number as a previous statement in the model, will replace the former.

Statement Deletion- The delete command in the discrete-event simulation languages must be such that indicating only the number of the first statement and the number of the last statement, all the statements in between can be deleted.

Emulation Process- When implementing an emulation process using the discrete-event simulation language it is necessary to assign the initial values of any matrices defined. Being able to do this task, during a simulation session, is of paramount importance for the creation of the dynamic environment.

Invisibility Concept- Some of the entities related to model animation can be changed from visible status to invisible status. The user will find that this invisibility concept is an important feature when we are dealing with model entity changes.

GPSS/ *PC*^{TM} has special characteristics allowing the implementation of these new concepts in a very simple manner, for this reason it is used in our applications.

4. MICROMODELS ADJUSTMENTS

With these special features, using GPSS/PC ^{TM}, we can design the micromodels adjustments which can be loaded after the generic model has been loaded and obtain the necessary parameter changes or model logic changes in the generic model. Let's introduce an example to understand how the changes are implemented.

Let's assume that our model has declared three STORAGE control statements, as follows:

```
120 BATCH  STORAGE   5
130 RECOR  STORAGE   6
140 LOCAL  STORAGE   4
```

The three previous STORAGE control statements, name BATCH, RECOR, and LOCAL have a capacity of 5, 6, and 4, respectively. The previous capacities will be changed to 7, 5 and 6. In order to implement these changes we write a file the same as above but with the new capacity assignments. After the generic model has been loaded, we load the new file. Due to the overlapping characteristics, the new declaration statements replace the old statements written in the model. It is important that the same line numbers be used to accomplish our objective of changing the storage size.

In doing these changes, we are not limited to changing only the storage capacity, we can also change the name. For example, the storage name BATCH above could have been changed to, say, FIRST.

If in addition, our changes affect the number of storages to be declared, we need to take care of the number of statements involved. The following rules need to be followed very closely

Rule # 1- Never renumber the original model before making a model alteration, since this action will change the statement numbers.

Rule # 2- All the statements to be changed need to be removed first. The deletion of these statements will be accomplished by means of the command

DELETE n1 n2

where n1 is the number of the first statement affected, and n2, is the number of the last statement affected in the set of statements to be removed. The DELETE command will delete all the statements included between n1 and n2.

Rule # 3- Always include the updating statements in the delete range in such a manner that the first updating statement is numbered n1, and the last statement n2. It could happen that a set of statements will be replaced by only one statement. In this case, you should include a dummy comment statement numered n2.

5. AUTOMATIC SIMULATIONS

After we understand how to update models, we are in a position to implement the concept of automatic simulations. To this end, and assuming that in our generic model, the storage declarations are:

```
150 OPER0102  STORAGE   2
160 OPER0304  STORAGE   2
```

The manager of an automated manufacturing facility would like to run a simulation session subject to the following changes: 1) For the first two hours of operation there are only 2 operators available. 2) For the following period, and until the end of first shift, the number will be increased to 4.

We need to prepare two micromodels adjustments, which will produce the model parameters needed. When this is done we can prepare a process file that executes the simulation session.

The first micromodel designated *micro1.adj* is

```
150 OPER0102  STORAGE   1
160 OPER0304  STORAGE   1
```

The second micromodel designated *micro2.adj* is

```
150 OPER0102  STORAGE   2
160 OPER0304  STORAGE   2
```

The process file to execute the simulation is

```
REPORT GPSS1.GPS
@MICRO1.ADJ
START 1
REPORT GPSS2.GPS
@MICRO2.ADJ
START 6
```

The analysis of the previous example allows us to see that the implementation of the micromodels which provide the alteration of the model, requires some additional commands, and these commands have an almost fixed structure: Request of the simulation report at the end of the model alteration; micromodel adjustment, and the simulation duration.

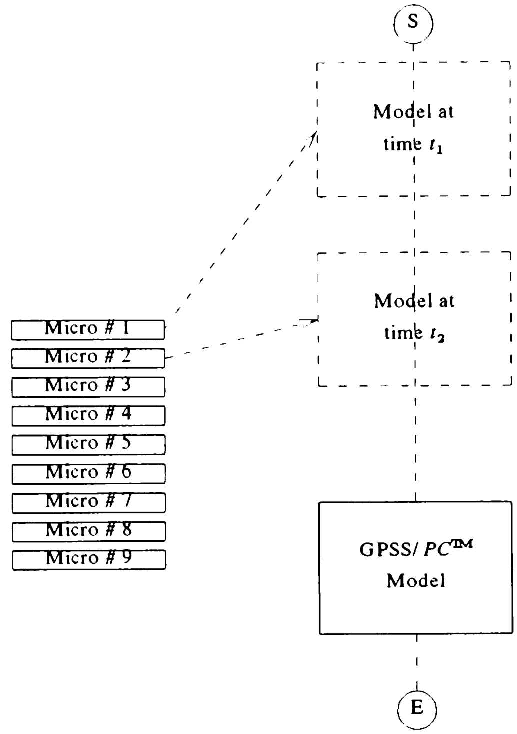

Figure 1- The Micromodels Shooting Process

6. MICROMODELS SHOOTING PROCESS

The Micromodels Shooting Process can be described by the pictorial shown in Figure 1. Let's idealize this process as a model that travels during the simulated time, starting the travel at point S and ending at point E. During this travel, the model is subject to changes which are provided by the micromodel adjustments. In other words, at simulated time t_1 the micromodel adjustment designated as *Micro #1*, impact our model producing the corresponding changes; at time t_2 the micromodel adjustment *Micro #2* will impact the model making the corresponding changes; and so on.

In order to produce the model changes, before shooting the micromodel adjustment the simulation is placed on hold, the model changes are made, and then the simulation continues. All these changes occur without user intervention.

We observe in the figure that this process consists of several micromodel adjustments which are scheduled to impact the model, and therefore to provoke the intended model changes at simulated times specified by the user or by the system. The first category of changes are related to operators change, input to the model changes, etc. In the second category we have disruption occurrences, such as materials out of stock.

The Micromodels Shooting Process which provides the self-generated models and process file to run the simulation session consists of two programs:

- **AMMG (Automated Manufacturing Models Generation) Program**- This program generates the needed model for a specific application such as:
  1. *Automated Storeroom Facility (Dual Conveyors)*
  2. *Automated Storeroom Facility (Single Conveyor)*
  3. *FMS- Flexible Manufacturing Systems*
  4. *Material Handling Systems*
  5. *Circuit Pack Assembly Line*

  In addition to the GPSS/*PC* TM model generation, this program produces a control file. This control file provides the necessary information for the SHOOTING program.

- **SHOOTING Program**- This program produces the process file which executes the simulation session according to the model changes requested by the user.

With the utilization of these two programs the manager is not required to know modeling and simulation to develop a model of his/her automated facility and run the simulation subject to the changes required. His/her daily decisions are based on the simulation report.

Before discussing some of the automated manufacturing facilities, it is necessary to define the following terms:

Unit of Machine Allocation- By definition the unit of machine allocation is the set of similar machines which are dedicated to perform the same manufacturing operation. We will allocate a given number of operators to each one of the units of machine allocation. This is a practical approach for the operator allocation.

Operator Allocation- By definition this is the number of operators allocated to each unit of machine allocation.

Maximum Number of Machines Available- By definition this is the number of available machines to perform a given manufacturing operation. This number plays an important role in the model to be developed. If the number of machines available is less than the number of machines required, the batch products should be scheduled to use them following the order in which they have been entered in our model.

7. AN INDUSTRIAL APPLICATION

We will present the application of these concepts to a circuit pack assembly line in which the applicability of these concepts makes possible an effective administration of these facilities. The facility administrator can modify the circuit pack assembly line to adjust their representation for a weekly simulation. Besides evaluating the operation performance of the facility, he/ she can recognize the *long runners* in this assembly line, schedule the weekend production, and know before hand if the production level is met.

Although we will present in our tutorial, an on-line application for this type of facility, we will review the English dialogue involved for the self generation of this model and the creation of the process file to execute the simulation. The English dialogue to generate this model using **AMMG** follows:

First: Select the specific application: Automated Storeroom Facility (Dual Conveyors); Automated Storeroom Facility (Single Conveyor); FMS- Flexible Manufacturing System; MHS- Material Handling System; Circuit Pack Assembly Line; etc.

Second: Define the number of operations in this particular assembly line and the name given to each one of them. Using **AMMG** the user has two options: Provide the name of the work station using up to five characters or let the software system assign the name.

Third: Depending on your selection, you need to enter the processing operation name or let the system define it for you. In addition you need to enter the unit of machines allocation; the maximum number of machines available in the assembly shop; and the number of operators allocated for the unit of machine allocation.

Fourth: Select the disruption analysis to be implemented in the model: Machine breakdown; Unusual large set-up time; Shortage of components for a particular product; Operator maintenance work; or no disruption analysis. For each implementation selected, there is some particular information requested by the system. For example, if we select the machine breakdown, the system will request the MTBF (Mean Time Between Failures) and the MDT

(Machine Down Time).

Fifth: Information on the batch size of each product and the number of processing operations for each one of them.

Sixth: The information requested is related to the processing of each batch for each processing needed such as: Operation Number; The distribution to be used for describe the service time and the parameters required to define the probability distribution selected.

Seventh: If we are interested in including the batch transportation time.

Eigth: If we would like to define the simulation duration for a given time or for a specific number of finished a specific number of batch.

Ninth: If there is interest in obtaining a histogram of the batch processed.

Tenth: If there is interest in obtaining a histogram of the residence time of each batch under process.

Eleventh: If there is interest in creating a microwindow indicating the remaining memory available in the PC.

Twelfth: If there is interest in creating a microwindow indicating the elapsed time of the simulation.

Thirteenth: If there is interest in creating a microwindow indicating the remaining time to end the simulation.

The English Dialogue to generate the process file using SHOOTING involves modifications such as:

- Changing the unit of machine allocation
- Changing the operators allocated to each unit of machine allocation
- Changing the interarrival times of the batch products
- Changing the service time provided by each machine per batch product
- Changing the batch sizes
- Changing the routine of each product
- Deleting any batch product
- Adding a new batch product

8. CONCLUSIONS

From the material contained in this paper we can make the following observations:

Creating the Model- To create a model using **AMMG**, the English dialogue involves only questions related to the manufacturing plant description and does not include any questions related to the modeling methodology used. In addition, since the self-generated model is developed using GPSS/*PC* ^{TM}, it could be possible for the user to include some specific details not covered by the questionnaire. Parallel to the model creation, the system provides a control file with the purpose to be used as an input to the SHOOTING program.

Creating a file for the Automatic Simulations- The SHOOTING program allows by means of an English dialogue the creation of the process file to run the simulation session. In addition, it also provides a description of the self-generated model.

Modeling and Simulation at the Manager's Fingertips- The use of the approach of *Micromodels Shooting Process* furnish management with the bridge to use modeling and simulation in the daily decision making process. For the case discussed in this paper, *circuit pack assembly line*, from management's point of view, the previous experience managing these facilities cannot be used as a base for future decisions. The approach presented is a very viable way of providing support for the decision making process.

9. REFERENCES

AMMG (*Automated Manufacturing Models Generation*)- Reference Manual, JBG Software, Inc., 220 Old Trail Drive, Columbus, Ohio 43213.

GPSS/PC ^{TM}- Reference Manual, Minuteman Software, P.O. Box 171, Stow, Massachusetts.

SHOOTING- Reference Manual, JBG Software, Inc., 220 Old Trail Drive, Columbus, Ohio 43213.

10. AUTHOR BIOGRAPHY

Ricardo F. Garzia is the Executive Director at JBG Software, Inc., and an Instructor in Computer Science at Capital University.