

**SMART MODELING – BASIC METHODOLOGY AND ADVANCED TOOLS**

Arvind Mehta

Lanner Group, Inc.  
 11000 Richmond Avenue, Suite 680  
 Houston, TX 77042, U.S.A.

**ABSTRACT**

The paper discusses how a complex simulation project can be executed efficiently and effectively following simple basic methodology, and using advanced modeling features provided by the simulation tool. The paper explains the methodology that should be followed for the successful outcome of a simulation project. The paper also discusses and illustrates some of the advanced modeling capabilities provided by a simulation tool “Witness”, that enable the user to build complex models very quickly and at the same time, incorporate desirable characteristics like high flexibility, sharability, and re-usability.

**1 INTRODUCTION**

Usually simulation projects come into existence with very clear objectives. The decision to either subcontract the project or to execute it in-house is based on various factors such as availability of time, skills, and tools. In spite of all the investment of time, effort and money in the simulation project, many of the projects can’t make up to the expectations of the customer. Most of the times the short-fall is attributable to the inadequate planning of the project.

The planning becomes easier and effective if there are proper guidelines for the simulation modelers to break up the project in different stages. In other words the modeler should have the knowledge of simulation methodology, and should follow it religiously.

In addition to following the simulation methodology, the simulation software should provide tools that facilitate flexible modeling, easy sharing of the simulation efforts, and effective utilization of the work already done in the past thereby avoiding the need of duplication of efforts. This speeds up the model building process, and saves more time for model validation and what-if analysis.

**2 BASIC SIMULATION METHODOLOGY**

Simulation project can be broken down into four major phases.

1. Project Definition
2. Model Building and Testing
3. Experimentation
4. Project Completion

These four major phases can be further broken down into some more stages as shown in the Figure 1.

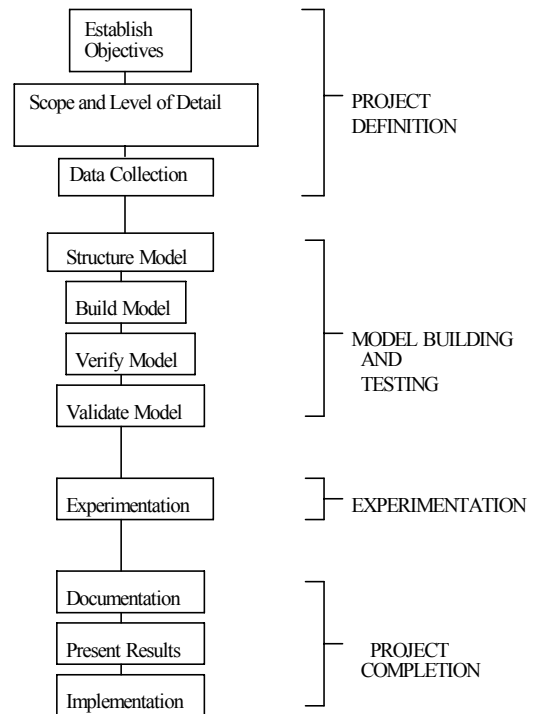


Figure 1: Flowchart showing the Basic Simulation Project Methodology

## 2.1 Project Definition

### 2.1.1 Establish Objectives

The first step in any simulation project is to set the objectives. The objectives should be a statement of what should be achieved. Typical objectives may look like those given below:

1. Increase plant throughput by 20%.
2. Reduce WIP by 10%.
3. How many AGVs are required?
4. What schedule achieves best throughput?

There may be some general project objectives related to time scale, run speed or visual presentation.

### 2.1.2 Scope and Level of Detail

The scope and level of detail determine what should be included in the model and how much detail should be modeled. The model should have a minimum amount of detail required to achieve the project's objectives. More detail doesn't necessarily mean more accuracy.

### 2.1.3 Data Collection

In a simulation project, data is usually required to build the simulation model, to set the initial level of various factors in the model, and also to validate the model once built. Data usually falls into one of three categories:

1. Available
2. Not Available, but Collectable
3. Not Available and Not Collectable

In order to deal with the third category, the data may need to be estimated. The data can be estimated by using the experience of people who are familiar with the operation and by referring to manufacturers' specifications. Data can also be estimated by making the model create the data (e.g. based on the knowledge of the throughput of the system that is being modeled, a model could be built to produce this throughput by varying the unknown parameters).

Once the project's objectives, the scope and level of detail, and the data required have been sorted, it is useful to map out a project specification that summarizes these areas and lists any assumptions made. This can be useful in gaining agreement to a project and in validating the approach.

## 2.2 Model Building and Testing

### 2.2.1 Structuring the Model

Before building the model, it is extremely useful to map out or structure the model on paper. The structure or map

of the model can help to get an overview of the whole model and map out any of the complex areas where further investigation is needed.

### 2.2.2 Model Building

Model building should be approached by taking small steps, one at a time: coding, documenting, and verifying. Before continuing onto the next step, the modeler should be confident that the step completed is working as intended.

### 2.2.3 Model Verification

The verification should consist of checking the code, inspecting output reports and checking that the modeled elements work the way the real world elements do.

### 2.2.4 Model Validation

A validation of the model should be performed on completion of the model. This 'validation' should test the overall accuracy of the model and the ability to meet the project objectives. This phase helps to establish credibility by giving confidence in the results.

The model can be validated in various ways. An idea of the validity of the model can be obtained by demonstrating it to the people who know the system. The results produced by the model can be compared to historical data. Results from the real system and results from the model could be put in front of an expert to see if they could tell the difference. The model produced could also be compared to other models; e.g., mathematical models, deterministic models, and other simulation models.

## 2.3 Experimentation

Prior to starting the actual experimentation stage, number of issues given below need to be addressed.

1. Warm-Up Period
2. Run Length
3. Number of Replications
4. Experiments to be Performed

### 2.3.1 Warm-Up Period and Start-Up Conditions

The length of the warm-up period needs to be evaluated if the state of the model at time zero does not represent the normal working conditions of the actual system that the model is representing.

Warm-up period is the amount of time that a model needs to run before statistical data collection begins. This warm-up period could be estimated through experimentation with moving averages (flow times, throughput) and time series (WIP, buffer sizes). The point at which the

model seems real for the first time could be estimated as the warm-up time.

If the level of WIP is plotted on a graph, the plot rises steeply at the start of the model run and gradually reaches a relatively stable level representing the ‘typical’ condition. The time taken by the plot to get stable can be recorded as being the system’s warm-up period. It is always better to have a warm-up period that is too long rather than one that is too short. A warm-up time should be evaluated for each experiment. This period could be the same for every experiment because of time constraints as well as to provide a basis when comparing different what-if scenarios.

Starting conditions could be used as an alternative to warm-up period. This could be achieved by loading the model from the beginning of the simulation run. The disadvantage of this alternative is very obvious when two different systems are compared. A combination of starting conditions and warm-up periods could be used to alleviate this disadvantage. Terminating systems are usually dependent upon the initial values and quantities used when starting the model; therefore, the start-up conditions should reflect real world situations.

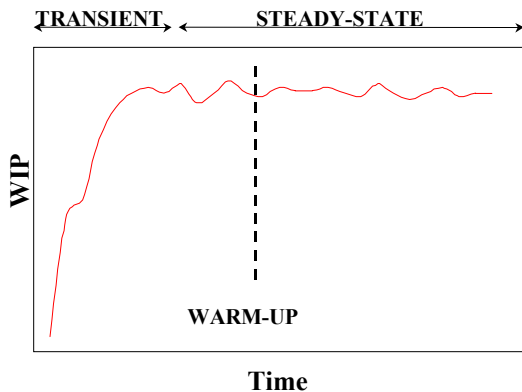


Figure 2: Figure showing Estimation of Warm-Up Period

### 2.3.2 Run Length

Once the model has warmed up, run length of the model has to be decided. One method for deciding this is inspecting the random numbers sampled. As a rule of thumb, a minimum of 15 to 20 random numbers from each random number stream should be used in the model. Typically, to ensure that this takes place for all of the random number streams, the least frequent event in the simulation model (usually breakdowns) should be selected and the model should be run until this event has happened 15 to 20 times (e.g. 15 to 20 breakdowns). The values of the random numbers can then be monitored (and plotted on a graph) to ensure that the numbers sampled follow the distribution required.

### 2.3.3 Number of Replications

Due to the very nature of random numbers, it is imprudent to draw conclusions from a model based on the results generated by a single model run. Replication is defined as executing the same model a number of times, but with different random numbers in each run. As a rule of thumb, a modeler should always perform at least three to five replications for each experiment. It is possible to do this automatically by using the WITNESS EXPERIMENT tool. Using WITNESS XA, standard statistical analysis can be performed, and confidence intervals can be put around the results from the simulation runs.

### 2.4 Project Completion

On completion of the project, a number of tasks may need to be done, e.g.,

1. Complete Documentation
2. Communicate Results
3. Produce a Report
4. Implement the Recommendations

It is also useful, at the end of a simulation project, to review the overall project process and perhaps refine the process before starting the next project.

## 3 USE OF SOME ADVANCED TOOLS

WITNESS simulation software provides many advanced tools that facilitate flexible modeling, easy sharing of the simulation efforts, and effective utilization of the work already done in the past. Use of these tools speeds up the model building process, and saves more time for model validation and what-if analysis. Some of the tools provided by Witness have been discussed below.

### 3.1 Designer Elements

These template elements, customizable by organization, allow the modeler to point to a recognizable piece of equipment and introduce it into a model. Combinations of elements, for example a manufacturing cell or an operating theater, may be captured into a designer element module to be re-used in the simulation model as necessary. Entities may be routed to the cell or the theater or to any element within the cell or theater.

The designer elements can be saved as \*.des file and shared between the users electronically.

### 3.2 Modules with Password Protection

An extension of the idea of a designer element module is to allow the module to be password protected. This allows a

wide variety of different users with different experience levels to use exactly the same elements to create a model. For example, a breakfast cereal packaging line can be built, tested, encapsulated as a model, and password protected. Every engineer within the cereal company gets to use the same identical packaging line or ideally chooses from several pre-built configurations. In either case they do not need to build the line from first principles. The modeler can input certain characteristics, for example the speed of the line, breakdown characteristics, or cereal box sizes, even if the module structure is password protected.

The modules can be saved as \*.mdl file or as designer elements in \*.des file, and shared between the users electronically. All of the users sharing a module can link the module, from their model, to the parent module (\*.mdl file). This ensures that all the changes made to the parent module get propagated to the models that use it. Thus the author of the module can update the password protected \*.mdl file on the network and be assured of automatic updating of all the designer elements that are linked to it, and used by various users in the company in various simulation models.

### 3.3 User Input Dialog/System Functions

These functions allow an easy interaction between the model and the person running the model. These functions also enable the modeler to introduce immense amount of flexibility in the model. The user can modify the model or perform what-if analysis without having to delve into the details of model code. The user-input functions accept the data fed in the Input Boxes and modify the model accordingly. The Input Boxes are easy to use and let the user feed the information to the model by way of typing in the boxes, choosing listed options or pushing available option push buttons. Some examples of user input dialog functions are provided below.

1. COMBODLG – The function displays a combination box as shown in the Figure 3.
2. INPUTDLG – The function displays an edit box as shown in the Figure 4.
3. MSGDLG – It returns the number of the button that the user presses as shown in the Figure 5.

System functions allow the user input to configure the model. Simple commands such as

SET QUANTITY OF DRILL TO X

Or

SET CAPACITY OF STORE TO Y

are supplemented by other commands that can change any aspect of model behavior. This would include, for exam-

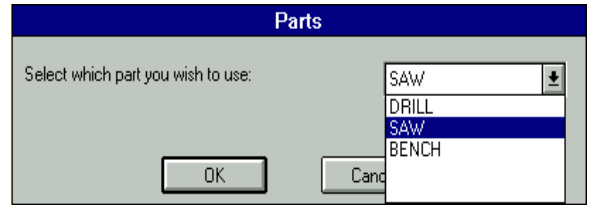


Figure 3: Figure showing the Input Box for COMBODLG Function

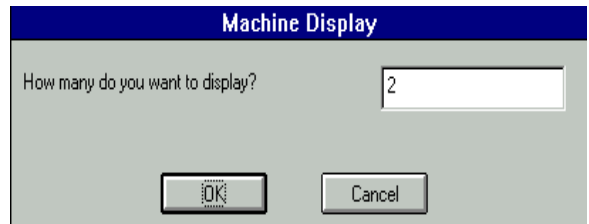


Figure 4: Figure showing the Input Box for INPUTDLG Function

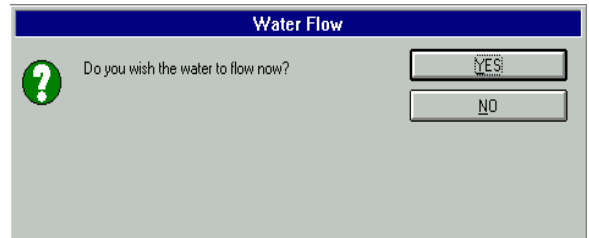


Figure 5: Figure showing the Input Box for MSGDLG Function

ple, the breakdown characteristics of a machine, the shift pattern of a worker, or whether or not a conveyor should accumulate parts.

### 3.4 Direct Input from and to External files

The model should be designed to handle the fast changing business scenarios without having the need to make changes to the model or even access the detailed simulation code. The model should therefore have the ability to read the input data to the model directly from external software like MS Excel, MS Access etc., and give the output accordingly. The model should also be able to output the data to the external software in the format desired by the modeler. This output can be used to drive another model or application. The newly introduced direct Excel and Access links in WITNESS enable the user to take the maximum advantages of these popular spreadsheet and database software. Other software can be used externally using OLE links to WITNESS.

## 4 CONCLUSION

The simulation methodology explained in this paper if followed and used in conjunction with some available advanced tools, can bring about significant improvement in the productivity of the modeler. The objective of any simulation project is not to build the simulation model, but to analyze a system, existing or proposed, and decide how the management objectives can be met. Its therefore imperative that the modeler should build an accurate simulation model in minimum possible project time and thus save a good amount of time for performing what-if analysis and giving viable recommendations to the management.

## REFERENCES

- Lanner Group, Inc. 1998. Learning Witness.  
Mehta, A. and I. Rawles. 1999. Business solutions using Witness. In *Proceedings of the 1999 Winter Simulation Conference*, ed., Farrington, P.A., Nembhard, H.B., Sturrock, D.T., and Evans, G.W., 230-233. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

## AUTHOR BIOGRAPHY

**ARVIND MEHTA** is a Technical Consultant & Training Coordinator with Lanner Group, Inc. He received his M.S degree in Industrial and Systems Engineering from Auburn University. He has BS degree in Mechanical Engineering. He has six years of work experience out of which three years is in the field of simulation. He is a member of IIE. His email address is <amehta@lanner.com>.