

## A TUTORIAL FOR MODELING WITH THE WITNESS VISUAL INTERACTIVE SIMULATOR

W. Bruce Thompson

AT&T Istel, Inc.  
 Burlington, MA 01803, U.S.A.

### ABSTRACT

The paper will describe AT&T Istel's WITNESS Visual Interactive simulator. Benefits of using WITNESS are described and illustrated by explaining the steps required to build any WITNESS model and then, at a more advanced level, by creating a sample model.

### 1 INTRODUCTION

Witness is a comprehensive discrete event and continuous process simulator. It is designed to allow a person knowledgeable about the process under study to rapidly, incrementally, and accurately build, debug, validate, verify and exercise complex models. Model building time has been reduced by displaying color coded status icons for each element, allowing portions or all of previously built models to be used (Sub-Models), user definable element library, (Designer Elements), the intelligent use of a graphical interface, form based modeling elements, pre-defined Input/Output rules, and built-in error checking. Modeling accuracy has been attained through the creation of an Action Language, a host of system and user-defined functions, and user interactivity. An example will be used to illustrate where each of these items comes into play during the model building process

### 2 BUILDING A WITNESS MODEL

#### 2.1 A Mini-mart Process

Thompson's QuikStop, a multi-million dollar chain of mini-marts with gas pumps, would like to know if their operation in Bird-in-Hand Pennsylvania should be expanded by another island of gas pumps and a store expansion (Figure 1). Based on surveys of the other QuikStops in similar environments across the country, the Business Analysis Department of the Main Office has estimated customer arrivals to the location to be

every 1.1 minutes and negative exponentially distributed. For the sake of this example, we'll assume a non-terminating system and that the mean time between arrivals is constant. Arriving customers will enter the queues in front of the gas pumps or mini-mart. From either location, they could also use the other service. Table 1 presents estimates of the percentage of arrivals requiring the various services.

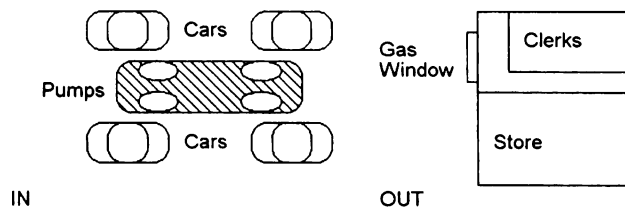


Figure 1: Mini-mart Example Site Layout

Table 1: Mini-mart Service Parameters

<i>Service</i>	<i>Number of Servers</i>	<i>Percent of Arrivals</i>	<i>Service Time</i>
Gas only	8	40	7
Store only	3	20	3
Gas and store		10	12
Store and gas		30	8

The department has used WITNESS to model the gas pumps and store for their other locations. In addition, they have created a library of operations. In this case, they have a model of an island with 4 pumps and a mini-mart. They would like to use this library to create a model that will determine the average waiting time in each of the queues, the number of customers served, and total dollars earned.

The methods that we will use to construct this model greatly reduce the model building time. However, they

do require a basic knowledge of how WITNESS works. WITNESS models are built in 3 steps:

- 1) Define the elements to be included in the model
- 2) Display the necessary elements
- 3) Detail the elements.

## 2.2 Defining Model Elements

In order to construct a simulation model, the user must first tell WITNESS what types of elements the model will contain. WITNESS contains 22 pre-defined elements that fall into 3 categories: 1) Physical, 2) Logical, and 3) Reporting. The user would enter the Define mode through a menu bar, select the element type to be defined, and provide a name and quantity required for the element.

### 2.2.1 Physical Elements

Table 2 lists the Physical elements that are available in both manufacturing and generic terms.

Table 2: WITNESS Physical Elements

<i>Manufacturing</i>	<i>Generic</i>
Parts	Entities
Machines	Activities
Module	Module
Labor	Resources
Tracks	Tracks
Vehicles	Vehicles
Buffers	Queues
Conveyors	Conveyors
Shifts	Shifts
Fluids	Items
Tanks	Store
Processors	Process
Pipes	Stream
Module	Module

### 2.2.2 Logical Elements

Table 3 lists the Logical elements. These elements provide the modeler greater control over execution of the model. Logical elements include:

Table 3: WITNESS Logical Elements

Variables	Attributes
Functions	Distributions
Files	Part Files

### 2.2.3 Reporting Elements

Table 4 lists the Reporting elements that allow user selected information to be graphically displayed while the model is running. Reporting elements include:

Table 4: WITNESS Reporting Elements

Histograms	Timeseries
Pie Charts	

In addition to these graphical reporting elements, the variables previously mentioned in the logical elements section can also be displayed.

## 2.3 Displaying Model Elements

Elements are displayed by clicking on the Display option from the menu bar and selecting the element to be displayed or, if the element is already displayed, the modeler may double-click on the required element, with the right mouse button, to bring up the Display form for that element. Once an element is selected, a sub-menu specific to the element type is presented. As an example, the sub-menu for a queue would contain selections for the Queue Name, Entity-in-the-Queue representation, and Icon graphic. The user would choose an option and place the chosen item accordingly. Should the initial placement of elements be unsatisfactory, a net is available to move an individual or group of elements to a new position.

## 2.4 Detailing Model Elements

Detailing a model involves the entering of cycle times, setup conditions, capacities, etc. Again, from the menu bar, a Detail option may be selected or, if the element is already displayed, the modeler may double-click on the required element, with the left mouse button, to bring up the Detail form for that element.

Each element type has a specific Detail form. The form for a Buffer/Queue would contain items such as the Name, Capacity, Delay Time (for black box analysis), Input Position, Output Position, as well as further conditions for Input and Output positions. The user would simply enter the relevant information into the appropriate field.

### 2.4.1 Input/Output Rules

WITNESS provides 8 pre-defined rules for Pushing or Pulling Parts/Entities through the model. At a basic level, elements such as Machines/Activities would either

Pull from within the Input Rule editor or Push from within the Output Rule editor. At a more advanced level, IF/ELSEIF logic or User Defined Functions may be employed to move the Parts/Entities.

### 2.4.2 Action Language

Few, if any, models are straight forward enough to use only Push/Pull rules. More likely, movement of Parts/Entities would be contingent on conditions in some other part of the model. In addition, Parts/Entities are not always vanilla flavored. These items might also have lot sizes, specific routings or failure percentages, to name a few. WITNESS' Action Language allows these types of data to be added to the model at the beginning or end of each type of event. For example, a pass/fail determination needs to be made at the completion of a test. FOR/NEXT and WHILE/ENDWHILE constructs allow the user to change or monitor entity specific attributes or global variables thereby allowing greater control over model creation.

## 2.5 Basics Summary

So, in a nutshell, the steps necessary to create a simple model of an activity with an entity going through it would be to:

```
Define the entity:      A
Define the activity:   M1
Display the activity
Detail the activity
  Input Rule: Pull A from World (Pre-defined
  WITNESS location)
  Output Rule: Push to Ship (Pre-defined WITNESS
  location)
  Cycle Time: 1
```

Click on Run on the menu bar and the model is running. Had we forgotten the Cycle Time, WITNESS would have displayed a screen prompting the user for the Cycle Time. Had we forgotten an Output Rule, WITNESS would have displayed M1 in a magenta color indicating the activity is blocked. This makes it very easy to get the model debugged and running. If we had different entities going through M1, each of which had a different Cycle Time, we could have defined an attribute for the entity called CYC\_T. When the entity was created, we could have assigned a value to CYC\_T and referenced the attribute directly in the Cycle Time field of M1's Detail screen, etc., etc. Oh, by the way, all this could have been performed without resetting the model back to zero. WITNESS' Interactiveness is built into the

software. You say that's fine for a small model but, what about a larger model. Back to the mini-mart example.

## 3 MINI-MART EXAMPLE

You remember I said that the Business Analysis group had performed simulations of different mini-marts in the past. Well, being the foresighted group that they are, they decided to do two things. They Designed a library of elements commonly found in a mini-mart environment and they saved Sub-Models, or pieces, of their earlier models.

### 3.1 Designer Elements

Creating custom element libraries requires a basic knowledge of WITNESS. Since you already have that, let's move on. The menu bar has a DESIGNER selection. Clicking on this selection brings up a Designer Window. Once in the window, you Define an element. WITNESS will use the first three characters to create a sequentially numbered name each time you use the element in the model. For instance, ACT might be the name I give an activity. When the activity is used in the model, WITNESS would append 001 to it making the activity name ACT001. Once defined, the element needs an icon representation and positions for Entities and Labor, if necessary.

As above, clicking on Display on the menu bar brings up the display sub-menu for the element. From there, the items would be displayed in the Designer window.

Finally, if there are any Details of the element that would be common when placed in the model, these Details would be filled in on the Detail form of the element. Once the Design of the element is complete, adding a new element of this type to the model requires only that you click once on the element in the Designer Window, at which point the cursor will turn into a crosshair, and click at the point in the model at which you want the element to appear.

For our mini-mart example, the group has Designed 2 Module elements, one for the islands and one for the store. The powerful Module element allows the user to assign a Sub-model to a Designer element icon. However, before we get into Modules, a little background on Sub-models is necessary.

### 3.2 Sub-Models

Sub-Models are pieces of other models that the modeler might want to use over. Creating a Sub-Model requires clicking on File, Save, Sub-Model, from the menu bar and then selecting which elements will be included in

the Sub-Model. Selection may be from a list of elements, by mouse or by placing a net around a group of elements. Once saved, any reference to elements not contained in the Sub-Model will be listed at the completion of the save. Using the Sub-Model requires that you click on File, Open, Sub-Model and select the appropriate Sub-Model file. You would then be prompted for what to do in case and element name in the Sub-Model matches an element name in the model into which you are placing it. Options are to Prompt, or Ignore. If Change is selected, further selections are Prompt, Replace the first letter, or Add a first letter. After the Sub-Model is read in, a list of unresolved references is displayed. These references are still in the Sub-Model, they have, however, been commented out.

### 3.2.1 Module Elements and Hierarchical Modeling

When bringing a Sub-model into the model proper, the user is asked if he/she wants it assigned to a Module element. Assuming the answer is yes, the elements in the Sub-model would be assigned to an icon. Once assigned, the Module can then be entered into the Designer Elements window and brought into the model by simply clicking. The beauty of this assignment is that the model being presented needs only show the Module icon. The detail associated with the specific module is in a different part of WITNESS' virtual screen. Double clicking on the Module icon will take the user to that part of the screen. What's more, the

The Business Analysts conveniently have Module elements of the gas islands with the 4 pumps, and the mini-mart with associated queues, clerks and customers. The mini-mart Module also contains the floor layout of the mini-mart that was imported from a .DXF CAD drawing. All that remains to complete the model would be a little logic to connect the Modules together once placed into the model.

### 3.3 Creating the Mini-mart Model

You now have the knowledge necessary to build the mini-mart model in a hurry. The required steps would be:

1. Open a New model screen.
2. Open the Designer elements sub-model that contains the Island and Mini-mart Module elements.
3. Place 3 Island Modules and 1 Mini-mart Module in the model.
4. Add 3 clerks to the Mini-mart (module only has 1).
5. Increase the arrivals per Table 1.
6. Run the model.

### 3.4 Experimentation

Now that the model is running, it's time to verify and perform some experimentation. WITNESS has a built-in feature that allows you to automatically change the random number streams and re-run the same model, or a number of different models, to statistically determine the model's performance. Since our model has no variability, let's change the Inter-arrival Time of the Customers to that of the Negative Exponential distribution mentioned above, and the operation cycle times to Lognormal distributions with a standard deviation of 5% of the mean. Creating an experiment requires defining a Situation/Scenario, determining the Warm-up period, Run Length, Continuous vs. Restart data collection, how the random numbers will be exercised, what the output file name will be, and running the experiment.

Defining the Situation/Scenario requires that you name the scenario and, if applicable, assign a model file to be run for that scenario. Since we'll be running the experiment inside of the mini-mart model, no model name is required. For the Warm-up period, assume I've put a Timeseries in the model to, every hour, look at the number of customers in the mini-mart. The number of customers appears to level off after 2 hours of simulation time so I'll use 2 hours as the Warm-up. The run length will be 1 week with data collection occurring at the end of each replication. Exercising the random numbers requires a little background.

WITNESS has 999 unique random number streams numbered 1 through 999. The option I choose is the Stream Offset. This option will use a random number stream that is a certain number higher than the one referenced in the model. For instance, assume I use the following distribution as a cycle time in my model:

**LOGNORML (2, .15\*2, 35)**

The mean is 2, the standard deviation is 15% of the mean and the random number stream is 35. If I use a Stream Offset of 2 for this iteration of the experiment, the actual number stream being used would be 37. Exercising the model 6 times by simply changing the Offset is very convenient. For our purposes, the streams will be exercised as shown in Table 5.

Table 5: Stream Offsets for Experimentation

<i>Iteration Number</i>	<i>Stream Offset</i>	<i>Regular or Antithetic</i>
1	0	Regular
2	1	Regular
3	2	Regular
4	3	Antithetic
5	4	Antithetic
6	5	Antithetic

One thing remains. Where are the system statistics (i.e. Time in system, Total Throughput, etc.) going to go? Although there are 6 options, output from these iterations would normally go to a Comma Separated Values (CSV) file. This file format can be directly imported by a number of software programs for statistical analysis. For our purposes, a CSV file, call it MINI-MART.CSV, will be used as it will be input to WITNESS' statistical analysis program XA. That's it. Now simply run the experiment and go home. The model will run 6 times and the output from each run will be placed in the output file MINI-MART.CSV

#### 4 XA

Packaged with WITNESS, although a separate program, is XA. XA will take a CSV output file from WITNESS, and calculate a Student's 't' statistic for each measurement over all runs of your experiment. It will also calculate 99%, 95%, and 90% confidence intervals using the 't' statistic. Measurements for which statistics are generated are user selectable. In addition, the format of the analyzed measurements is user definable.

#### 5 SUMMARY

WITNESS is a discrete event, Visual, Interactive simulator. It has been designed to quickly and accurately create a simulation. Real-time color coded element status indications, combined with an extremely flexible graphic environment greatly reduce the time required to build a model

Basic models are built using a Define, Display, Detail cycle with pre-defined modeling elements and Input/Output rules. More complex models would incorporate IF/THEN/ELSEIF Input/Output rules and the WITNESS Action Language. Modeling time can be significantly reduced by saving portions of previous simulations as Sub-Model for use in future simulations. In addition, elements common to the modeler's environment may be incorporated into a Designer library for quick Definition, Display and Detail.

An example of how to build a model of a mini-mart process was used to illustrate these time saving features. The example also described how to set up experiments from within WITNESS and lastly, how to analyze the data from the experiments using the XA statistical analysis package.

#### AUTHOR BIOGRAPHY

**W. BRUCE THOMPSON** is a Sr. Technical Consultant with AT&T Istel, Inc. He received his B.S. degree in Industrial Engineering from Northeastern University in 1981. His responsibilities for Istel include training, consulting, customer support and Editor of the company newsletter, The Interactive Modeler.