Proceedings of the 1989 Winter Simulation Conference
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

A TUTORIAL ON SEE WHY AND WITNESS

Andrew R. Gilman
Claire Billingham
ISTEL Incorporated
3659 Green Road
Beachwood, Ohio 44122, U.S.A.

ABSTRACT

SEE WHY and WITNESS are companion products for visual interactive simulation. SEE WHY is a general purpose discrete event simulation system that offers special integrated features for graphical display and run time interaction. It also provides user programmable extensions to both the display and interaction capabilities. These facilities are highlighted in the discussion of SEE WHY.

WITNESS has been developed from SEE WHY. It is a simulation program for manufacturing operations that fully exploits the visual and interactive capabilities of SEE WHY in bringing simulation modeling to the non-specialist. The phases of WITNESS modeling are covered in some detail as well as the characteristics of the user interface. The paper concludes with an overview of the ability to extend the WITNESS program with an interface to SEE WHY.

1. OVERVIEW OF SEE WHY AND WITNESS

SEE WHY was a pioneering development in simulation software when it was commercially introduced in 1980. It was designed from the beginning to help overcome the problem of communication between modelers and decision makers. It was therefore designed to be a visual and interactive simulation system.

Experience has shown that the original design objectives have been satisfied. Visual interactive SEE WHY models invite participation from decision makers in all phases of a simulation study. As a result, the recommendations of the study are used more confidently and more correctly in the decisions taken concerning the system under study.

As a general purpose simulation system, SEE WHY has been used in the improvement of many kinds of systems: manufacturing, retail, distribution, packaging, air and ground traffic, etc. Its flexibility makes it possible to model the control logic of virtually any type of discrete or continuous system.

It has been recognized within ISTEL, as well as the simulation community as a whole, that manufacturing systems provide one of the richest areas for simulation application. It is also recognized that many manufacturing models contain similar types of model constructs and model logic. Hence, there is duplication of effort among simulation specialists who work in the area of manufacturing. It is almost certain that there are more manufacturing problems worthy of simulation study than there are specialists to model them. All of these factors led to the development of WITNESS.

WITNESS is a menu driven simulation program created for the study of manufacturing operations. WITNESS has been built with SEE WHY, and therefore carries forward the powerful features for real-time animated color graphics and run-time interactions found in SEE WHY. The modeling elements of WITNESS are higher level constructs such as Parts, Machines, Buffers, Conveyors, Labor, Tracks, and Vehicles. While the terminology is manufacturing based, the simulation community will recognize that these same elements can be used to model in a non-manufacturing setting.

In order to expand WITNESS's ability to deal with the intricacies of a particular problem, a SEE WHY submodel can be merged into the overall simulation model. This capability yields an unprecedented combination of ease of use and modeling flexibility.

In the tutorial that follows, we will look first at the general purpose facilities for visual interactive simulation found in SEE WHY. We will then cover in more detail the modeling capabilities of WITNESS and the characteristics of the user interface. Finally, we will discuss the facilities for extending WITNESS with SEE WHY.

2. MODELING IN SEE WHY

SEE WHY models are written using the event scheduling orientation. The modeler codes a series of (usually) Fortran subprograms that call upon the SEE WHY simulation library for virtually all simulation related tasks. This same library provides the means for color display and animation as well as interactive facilities.

The SEE WHY library supports all of the standard requirements for discrete event simulation. SEE WHY routines handle the (real) simulation clock, time updating, event handling, events list (known in SEE WHY as the TIMESET) maintenance, event scheduling and event initiation. Handled simultaneously are the tasks required for time dependent statistics gathering and simulation run control. Within this structure there is the additional provision for interruption of the simulation run for interactions by the user.

SEE WHY supplies pseudo-random number generation and a series of standard theoretical probability distributions with corresponding sampling routines. This capability can be extended with the user defined distributions element listed below.

The majority of the SEE WHY library is devoted to providing routines for the definition and manipulation of the SEE WHY modeling elements. The routines available for manipulating entities, attributes, and sets, as well as inquiring on their current states, are similar to those found in other implementations of discrete event simulation. It is not the purpose of this tutorial to detail theses functions. Rather, we are concentrating on the special characteristics of the SEE WHY modeling elements and the SEE WHY system as a whole.

SEE WHY maintains all of the simulation data that represents the state of the simulation (plus some information concerning the display) within its own flexible internal data storage. This flexible storage scheme allows any combination of modeling elements within the available space. As an example, any entity can have any number of attributes.

Specifically, SEE WHY provides the following elements for modeling statistics collection, and user defined probability distributions:

ENTITIES
SETS
VESSELS
TIMESERIES
HISTOGRAMS
DISTRIBUTIONS

A key characteristic of these modeling elements is that they are given implicit display-related parameters by the modeler when they are initially defined within the model (with the exception of distributions which are not displayed). In other words, the modeler gives the element display characteristics such as screen location, color, height, and spacing when it is defined. From that point on, the modeler concerns himself only with the modeling logic. The changes in system state, for example an entity moving from one set to another, is automatically reflected on the screen by the system. This includes the updating of timeseries and histogram displays as each new simulation value is observed.

Another important characteristic of these elements is that the system provides standard interactions that allow the user to inspect and change almost any parameter of the defined modeling elements by stopping the simulation at any time during the run. Examples of this capability are:

- inspecting or changing any attribute of any entity
- finding and moving entities among sets
- changing screen location of any element
- changing input rates, capacity, display size of any vessel
- erasing, displaying, rescaling any timeseries or histogram
- respecifying user defined distributions

When the user has interrupted the model, he may also, through standard system interactions, change the run mode and perform a variety of utility tasks concerning the simulation run. For example, the user can:

- run the model in batch mode or step mode
- save the complete state of the simulation
- restore a previously saved state
- restart the simulation
- reset the simulation time
- reschedule or deschedule events
- restart random number streams
- redirect printed output to the file or printer

Last, but not least, among the system interactions, is the facility for coding interactions specific to the model. These are referred to as "OWN" interactions. When the model has been interrupted by the user the keyword "OWN" causes a subprogram written by the modeler, to be invoked. In this subprogram, the modeler can employ prompting routines from the SEE WHY library to query the user for the type of changes to be made, or establish menus of options. The modeler can then alter model parameters, model state, or even model logic using any of the routines of the SEE WHY library.

To augment the implicit display characteristics of the modeling elements, SEE WHY also provides routines for generation of static color displays. Lines, text, rectangles, and areas of color can provide enhancements to element displays in order to make the screen display recognizable as the corresponding physical problem area.

For both static displays and element displays, SEE WHY utilizes a powerful technique to control which part of the model is currently displayed on the screen. Any display information written to the screen, either by the system or directly by the modeler, is written using one of the one hundred "pens" available in SEE WHY. It is the pen that has a particular combination of foreground and background colors. The pens are also either active or inactive. When inactive pens are used to update the display, no change is seen. Pens can be grouped together into "pictures". By activating one picture and deactivating another, different portions of the model display can be viewed selectively. The pens and pictures yield a wide variety of methods for displaying a model that requires more than one screen of information.

The flexibility for modeling, display and interaction found in SEE WHY has made possible the construction of extremely user-friendly simulation models. SEE WHY models can truly be run, experimented with, and understood by decision makers. These same features have allowed the development of WITNESS, which fully exploits the features of SEE WHY to bring manufacturing modeling to the non-specialist.

3. MODELING IN WITNESS

WITNESS models are described in a manner very different from the general purpose orientation of SEE WHY. WITNESS is also a visual interactive simulation program, which views the world as separate manufacturing components or elements, that interact in specified ways. The basic elements of WITNESS version 5 are:

PARTS	ATTRIBUTES	MACHINES
FILES	CONVEYORS	FUNCTIONS
BUFFERS	LABOR	DISTRIBUTIONS
TRACKS	PART FILES	TIMESERIES
VEHICLES	VARIABLES	HISTOGRAMS

There are three main phases involved in building a WITNESS model: Define, Display and Detail. There is also the phase of executing (running) the model, which has several "options" allowing the user to control execution. Each of the phases will be described below.

All of the commands for constructing and running WITNESS models are provided by menus and prompts within the single environment of the WITNESS program. There is no code to be compiled and linked.

There are virtually no restrictions on the order in which the phases are employed to create and run WITNESS models. At any time, the user is free to move between Define, Display, Detail and the execution phases of WITNESS modeling. Elements can be added, parameters or display characteristics changed, at any time, even while the model is running. The user need not leave WITNESS to change models or restore saved model states. No run termination time need be specified, the model may be halted, saved, restored, or restarted at will.

Unlike the old technology of creating a model and then running it, the recommended approach of WITNESS is to build up the model cumulatively. The Define, Display and Detail modes may be used for each component or group of components. The WITNESS user does not have to wait for the model to compile and link before running - it runs immediately. WITNESS is truly interactive modeling.

The primary WITNESS menu provides the following options:

DEFINE — The elements of a WITNESS model may be defined.

DISPLAY — WITNESS elements may be displayed on the screen.

DETAIL — The timings of the model and the flow of Parts may be established.

INITIALIZE — Input certain starting conditions for the model.

GO — Activates starting conditions for the model.

KEEP Save the parameters of the WITNESS model.

LIST List on the screen the elements of the WITNESS model.

DELETE Remove unnecessary elements.

END End the WITNESS session.

After the model's starting conditions have been activated (i.e. GO has been selected), the user will see these same options, along with others, appearing in a blue box in the top right hand corner of the screen. This is called the interact box and will be familiar to SEE WHY users as an area of the screen reserved for interaction. All interaction after the primary WITNESS menu is controlled through the interact box.

3.1 WITNESS ELEMENTS

A WITNESS simulation is comprised of physical elements, control elements, and reporting elements. The Define phase is used to create and name each new physical, control or reporting element. Machines, Parts, Conveyors, Buffers, Labor, Tracks, and Vehicles, are WITNESS physical elements, and are used to represent the manufacturing plant. The control elements, which are comprised of Attributes, Variables, Distributions, Functions, Files, and Part Files, are used to control and monitor the plant. The reporting elements, Timeseries and Histograms, are used to accumulate and graphically display performance measures of the plant.

3.1.1 WITNESS PHYSICAL ELEMENTS

PARTS are the individual items which are processed by the rest of the system being modeled. They move into, out of, and between other elements. They may be combined together in an operation (as in an assembly machine) or split apart (as output from a production machine). Parts are referred to by name. Effectively, an unlimited number of each Part name may be introduced into the system. Routings through the other modeling elements may be based on Part name. Attributes may be associated with Parts to provide further differentiation.

In addition to the Part's name and its associated attributes, a part may optionally be supplied with a Routing. The Routing specifies the other physical elements (e.g. Machines) through which the Part must be Routed as it is processed in the plant.

MACHINES represent the physical facilities that work on a Part in some way. Machine types available are Single, Batch, Assembly, Production or General, which are distinguished by the number of parts entering and leaving the machine during a single cycle.

Machines go through a series of states in an attempt to process Parts. The duration of these states reflects the times specified by the user in the Detail phase. Their duration also reflects times imposed because of dependencies on the state of other elements in the model. For a Machine, the user may specify times and requirements for Part processing, setup, and breakdowns. As the model runs, the Machine status will change appropriately among the states of idle, busy, blocked, setting up, down, and waiting for Labor. This is represented graphically by a change of color for the machine display.

CONVEYORS are transportation devices where all parts being carried move in unison. Typically, Conveyors are used to represent real world conveyors. WITNESS provides two types, Fixed and Queueing. Fixed Conveyors stop when a part reaches the end but cannot move (or be moved) off. Queueing Conveyors allow parts to accumulate in a line from the front of the Conveyor until the Conveyor is full. Conveyors generally connect two or more elements (Machines, Buffers, or other Conveyors) when transportation over a physical distance imposes a significant delay on the availability of parts between the two connected elements.

BUFFERS represent storage areas for Parts between processes, in particular, Machines, Conveyors, and Vehicles. Buffers are completely passive. Parts enter or leave the Buffer depending on the actions of other simulation elements that proceed or follow the Buffer. Buffers may have a specific internal organization that specifies the position at which Parts may enter, or from which they may leave. Buffers may also specify a Delay time for each Part before it may leave the Buffer. WITNESS also provides for dedicated Input and Output Buffers that are tied to a particular Machine. These are specified when Defining or Detailing Machines and are separate from the Buffer element.

LABOR represents resources which provide a service to Machines or Conveyors in order that a particular task (or tasks) may be carried out. Machines may require Labor for setup, production cycle, or repair. Conveyors may require Labor for repair. Each type of Labor in the model can be given a name and quantity of operatives. In describing the detail of a Machine, specific Labor types can be chosen individually for set up, production and repair.

Labor rules may also give a number of different options for the Labor required for each task, depending on availability of each Labor type, or on other conditions in the model. In a more general sense, Labor can be thought of as a moveable but limited resource. It might therefore be used to represent an expensive jig or fixture that is shared by several machines.

VEHICLES represent material handling devices that can load a Part or Parts and move via a Track or series of Tracks to a destination where the Parts are to be unloaded. The Vehicle and Track elements together provide a powerful framework for modeling Automatic Guided Vehicles. The user can model simple systems where vehicles travel a loop looking for work, or more complex systems where central control is employed to dispatch specific vehicles to meet specific demands for pickup. In general, Vehicles can also be used to model other devices such as cranes, lift trucks, or over-the-road trucks.

TRACKS represent the paths over which Vehicles travel. They provide routing information to the Vehicles at intersections. They also provide the mechanics for collision avoidance. Most importantly, the end of a Track provides the destination for Vehicles to load or unload.

3.1.2 WITNESS CONTROL ELEMENTS

ATTRIBUTES are associated with Parts. They can be used to distinguish Parts from one another. An Attribute is referred to by name, and can be assigned a real or an integer value. Part numbers, model numbers and process times are good examples of Attributes. Parts may be associated with common groups of Attributes where the group is defined by a reference number assigned to Attributes. Attributes can be used as part of the flow logic.

VARIABLES are storage units for numeric values. They may be used anywhere in a WITNESS model for the purpose of holding process times, interarrival times, mean time between Machine breakdowns, production quantities, etc. A Variable may be used as part of the logic, recording or display.

DISTRIBUTIONS: WITNESS offers you the use of 13 statistical distributions such as the uniform, exponential and poisson distributions. In addition to these standard distributions the user may define "customized" Distributions. These Distributions are created from actual data entered into the system as values and weights. WITNESS determines the probabilities, creates a probability density diagram and takes random samples from this Distribution as specified by the modeler.

FUNCTIONS: WITNESS allows you to make changes to the values of variables and attributes in Action statements related to each physical element. Certain special commands such as calling a Vehicle or changing a Part into a different Part may also be performed in Actions. Functions consist of Action statements which are grouped together to avoid duplication, or to enhance the "readability" of the logic. Functions may also be used instead of Variables in expressions such as cycle times, where several possibilities exist. Functions may be thought of as analogous to Subroutines in a programming language.

FILES may be used to incorporate data and parameters in the model which are held in an ASCII file on the computer. The values are read from the file and allocated to Variables as specified by the user. Values may also be written to files to allow the user to monitor certain parameters in the model when required. Files allow the user to easily use data contained in a spreadsheet or MRP program, for example, within the simulation model. Files also facilitate multiple experiments using different parameters.

PART FILES are also ASCII files held on the computer. They are used to schedule the arrival of Parts into the model at varying times and with varying batch sizes and attribute values. Part Files allow the modeler to try out different production schedules. Part Files may also be written out to show the sequence and timing of parts which are "shipped" from the model. This file might then be used as input to a second model which contains the operation downstream of the original model.

3.1.3 WITNESS REPORTING ELEMENTS

TIMESERIES are graphs which show the pattern of certain recorded values over time, for example, output of each Part type per shift. Up to seven values may be recorded on one graph. The recording interval and values to be graphed are specified by the user. WITNESS controls the updating of the graph and shows each new value on the display as it is added to the graph, while the model is running.

HISTOGRAMS are graphs which show the pattern of a recorded value observed during the model run. Histograms are updated using a command which may be entered in any Actions section. Thus the

Histogram may be updated at regular time intervals. For example, each time a time a part is shipped, or each time a particular machine breaks down, WITNESS shows the new values on the display of the graph as the model is running.

3.2 DISPLAY PHASE

Ten of the fifteen WITNESS elements can be displayed in a WITNESS model. Machines, Conveyors, Buffers, Labor, Variables, Parts, Tracks, Vehicles, Timeseries, and Histograms are positioned using a mouse or the keyboard cursor control keys.

WITNESS users can position the elements on any one of four windows. The windows are analogous to the picture facility of SEE WHY. Only one window can be viewed at any one time. This is a particularly useful feature in modeling a very large system. The modeler can display various areas of the plant separately on the four windows.

WITNESS allows the simulator to have complete control over the colors, size, position, and shape of the model display. WITNESS has a library of 20 shapes (or icons) with which you can represent the physical shape of a Machine. This library can be extended through SEE WHY. There is also the option for these icons to change color depending on the status of the Machine.

3.3 DETAIL PHASE

The Detail phase is the phase of WITNESS modeling that allows the modeler to identify times in the model, flow of Parts, Labor requirements, breakdowns, and setups. This information is specified in a Detail Form.

Each element, except for Attributes and Variables, has a specific Detail Form. The modeler simply fills in the blanks to establish the model's parameters. These details such as Conveyor length, Machine type, process times, and lot sizes, may be changed at any time while creating and running the model.

Machine cycle time, Conveyor speed, Vehicle speed, Part interarrival time, setup time, mean time between Machine failure, repair time, etc. constitute the timings of the model. Essentially, any numeric response may be detailed as a constant, Variable, Attribute, sample from a Distribution, algebraic expression, dimensioned Variable, or dimensioned Attribute. When Variables and Attributes are used for times, the manipulation of these values may be performed in the Actions feature. Actions are described below.

3.3.1 INPUT/OUTPUT RULES

The flow of Parts through the system is controlled through input and output rules. Machines, Conveyors, and Buffers all have input and output rules. The modeler specifies the input and output for the elements through the Detail Forms.

The basic function of the input and output rules is to push and pull Parts through the Machines, Buffers, and Conveyors in the system. An output rule of push will cause a Part to be pushed to a Machine, Buffer, or Conveyor. Should the element to which the Part is being pushed not be prepared to accept the Part, then the element pushing will assume a blocked state. Pull will cause a Machine or a Conveyor to only receive input when it need input, and not when the previous operation is ready to provide it. In a WITNESS model, either push or pull systems can be modeled or a combination of push and pull can be modeled. There are eight input rules and nine output rules. This extensive variety gives the WITNESS user the ability to model complex routing. A description of each of the rules follows:

The eight input rules are called PULL, MOST, LEAST, %, SEQUENCE, SELECT, IF and WAIT.

PULL: This rule allows a Machine or Conveyor to pull a Part from specified Conveyors or Buffers. Several alternatives can be specified.

MOST: This rule is used when you wish to pull a Part from one of several elements. The Buffer or Conveyor from which the Part will be pulled has the most Parts in it, or has the most free capacity.

LEAST: This is similar to the MOST rule except the element chosen has the lese Parts or the least free capacity.

%: This rule allows you to pull Parts in a probabilistic manner, according to the percentage specified by the modeler. i.e.: 40% from element A, 30% from element B, 30% from element C.

SEQUENCE: This rule is used when you wish to sequence the input from a series of elements. e.g. if the first time you need a Part it comes from element A, the second time it comes from element B, and the third time it comes from element C, then you would use a sequence rule. When the sequence cannot be maintained, there are three alternatives: WAIT, NEXT and RESET. These allow you to control the sequence in different ways.

SELECT: This rule allows you to pull a Part from an element that is selected from a list of elements, based on the evaluation of an expression.

The selection mechanism works in a manner similar to a Fortran computed GOTO. The expression may involve Variables, Attributes, sampling functions, and status functions.

IF: This rule allows for evaluation of a logical condition (such as "IF SHIPPED >50") followed by the input element when the condition is true (THEN) and the input element when the condition is false (ELSE). If the condition is false, WAIT may be specified or further IF conditions may be entered.

WAIT: Do nothing.

There are nine output rules. They are: PUSH, MOST, LEAST, %, SEQUENCE, PART, SELECT, IF and WAIT. The output rules differ in several ways from the input rules, the main difference being that rather than an element pulling a Part for input, the element will push the Part somewhere for output. A second difference is the addition of the rule called "PART". This output rule allows the modeler to specify different output paths for different Parts.

Output rules may also take advantage of the Routing feature of a Part. If a Part is to be directed through the plant via its Routing, an output rule can specify a destination of "ROUTE." For simple part routing based logic, each machine would normally have an output rule of PUSH to ROUTE. For more complex logic the ROUTE destination can be combined with other destinations, for example, % Rule: 90% to ROUTE, 10% to REWORK.
Part Files provide two additional features that expand the flexibility of output rules. A Part File can be given an output rule causing Parts to enter the system according to a schedule contained in an external Part File. Conversely, any element may direct Parts to a Parts File by specifying the Part File's name in its output rule. This causes the Part File to record all parts leaving the system via this element.

The flow of Vehicles through a network of Tracks is controlled by Track output rules. All of the above output rules apply to Tracks, plus a rule for Vehicle type and Vehicle Destination.

3.3.2 ACTIONS

Much of WITNESS's power is derived from selectively using the input and output rules in conjunction with the Actions feature. Special actions can be programmed to take place at entry to or departure from some component of the system. For example, Actions on start of a machine cycle or Actions on Part creation can be specified.

The Actions are a series of steps. These steps are computations and assignments which can draw on Attributes, Variables, dimensioned Attributes, dimensioned Variables, and various system functions. Actions can also be used to read data from files for assignment to Attributes or Variables, as well as writing data to files. The Actions can be entered as a straightforward set of steps or utilize the IF action to provide logical branching in the procedure. Creation or modification of the Actions is carried out in the Detail mode using an editor.

3.4 FURTHER FEATURES OF THE DETAIL PHASE

3.4.1 SYSTEM FUNCTIONS

WITNESS offers several system functions that enable complex modeling logic. NPARTS will return the number of Parts in the Machine, Buffer, or Conveyor specified. NFREE returns the free capacity of a specified Machine, Buffer, or Conveyor. ISTATE will return an integer value associated with the particular status (idle, busy, blocked, setup, down) of a Machine, Buffer, or Conveyor.

These functions can be used wherever numeric expressions are required in the Detail Forms. Or, the values returned by these functions can be assigned to Variables and Attributes in Actions.

3.4.2 SYSTEM VARIABLES AND ATTRIBUTES

WITNESS also provides system Variables and system Attributes. The system Attributes will change Part descriptions and Part colors. TIME is a real Variable holding the current simulation clock time.

3.4.3 REPORTS

A simple "Yes" entered next to "REPORTING" on the Detail Form will engage statistics collection for Machines, Parts, Buffers, Conveyors, Labor, Vehicles, and Tracks. The modeler can specify a particular warm-up period. During the time specified, the modeled system will be allowed to achieve steady state and then the statistics will be set back to zero. In addition, at any time, the statistics may be manually set to zero.

The reports provide important information including: average size of a Buffer, average time spent by a Part in a Buffer, percent of time a Conveyor was blocked, average work in proces over time, average time for a Part to move through the system, percent of time a Machine was blocked, percent of time a Machine spent waiting for an operator, average number of jobs waiting for an operator, number of jobs started, idle time spent by Labor, and Track and Vehicle utilization.

3.4.4 EXPLODE, SUMMARY AND USED

WITNESS provides interactions to help the user understand the structure and state of a model. The SUMMARY and USED interactions provide a quick means of looking at the information contained in the various Detail forms of the elements in the model. The EXPLODE interaction helps identify the locations and characteristics of the Parts and Vehicles currently in the system.

3.5 LIBRARY

A WITNESS model may be stored on the hard disk as a file which may be printed out or which may be re-used in another model. This file is known as a library file. It can be altered outside of WITNESS using a standard text editor. By editing library files, segments of models can be created, stored, and pieced together to create new models. These files can provide documentation for models, in hard copy form, for later reference.

3.6 LINKS TO SEE WHY

In order to expand WITNESS's ability to deal with a particular modeling intricacy, a submodel can be created in SEE WHY and linked into the WITNESS model. Internally, WITNESS is a generic SEE WHY model. WITNESS Parts, Vehicles, and Labor units are SEE WHY entities. WITNESS Machines, Buffers, Tracks, Conveyors, and operator job and idle queues are SEE WHY sets.

The WITNESS/SEE WHY interface allows Parts to be moved to and from a user written SEE WHY submodel. In the SEE WHY submodel, SEE WHY elements may be defined, events may be written, and system state information may be passed to and from WITNESS Variables with the complete facilities of Fortran I/O. Additions to the display can be made and special interactions may be coded just as in SEE WHY. The link to SEE WHY is also useful for performing calculations and decision rules that may be beyond the scope of WITNESS. The values can then be passed to a WITNESS model from a SEE WHY subroutine.

3.7 CONCLUSION

WITNESS, developed for modeling manufacturing systems, enables people with no simulation experience to develop models easily and quickly.

Little training is required for building simulation models with WITNESS. The system prompts the users for the model parameters through menus and questions. There is no compiling and linking required. The model is simply written in three phases and executed. Changes to the model may be made in any of the three phases, at any time.

Complex routing and logic is achieved with numerous input and output rules as well as special actions. In addition, SEE WHY may be linked into WITNESS in order to model operations of great complexity. This most important feature reflects ISTEL's commitment to combining flexibility and power, with ease of use, in manufacturing modeling.

REFERENCES

ISTEL Limited, SEE WHY Visual Interactive Simulation Reference Manual, January 1987.

ISTEL Limited, WITNESS User Manual, Version 5, July 1989.

AUTHORS' BIOGRAPHIES

ANDREW R. GILMAN is currently Manager, Management Consultant Program with ISTEL Incorporated. His previous assignments with ISTEL have included technical support, training, product management, and marketing management in support of the SEE WHY and WITNESS products. Before joining ISTEL, Mr. Gilman worked as an Industrial Engineer for the Xerox Corporation, where his duties included simulation projects in the North American Manufacturing Division. Mr. Gilman holds a B.S. degree in Operations Research from Cornell University.

CLAIRE BILLINGHAM is currently a Technical Consultant with ISTEL Incorporated, having transferred from a similar role at ISTEL Limited in Redditch, England. Before joining ISTEL, she worked as an Operations Research Analyst with Cadbury-Schweppes. Ms. Billingham holds a Master of Science in Operational Research from the University of Warwick (England), and a Bachelor of Science in Mathematics from the University of Leeds (England).