Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

MODELING AND SIMULATION WITH INSIGHT

Stephen D. Roberts
Regenstrief Institute
1001 West Tenth Street
Indianapolis, IN 46202, U.S.A.
and
SysTech, Inc.
P.O. Box 509203
Indianapolis, IN 46250, U.S.A.

ABSTRACT

The INSIGHT simulation language describes systems in a quick, simple, and compact fashion using a network representation. This description can be entered and simulated using novel interactive facilities that relieve the user of needing to know specific syntax, while promoting a greater understanding of model behavior. Statistics summarizing the simulation are produced automatically, but can be greatly enhanced by various input models and output analysis mechanisms. Use of the language does not require programming and complex models use the descriptive features of simple ones, incorporating more elaborate specifications and more sophisticated concepts. INSIGHT is available for most computers and is portable across machines. The language has been extensively applied and its scope of applications has ranged from manufacturing to service environments. Using INSIGHT the process of simulation modeling and the results from the simulations combine to provide "insight" into problem solving.

1. INTRODUCTION

The INSIGHT (INS) simulation language is a high level, general purpose, discrete event simulation language that allows simulation models to be described quickly and compactly. Its fundamental concepts are easy-to-learn and easy-to-use. The language does not depend on any special competence with computer programming and yet the models can be easily extended though a rich variety of specifications. The emphasis on nonprocedural facilities and high level concepts makes INSIGHT a *simulation modeling language* rather than either a simulation programming language or a parameterized simulation model. Because INSIGHT is a simulation modeling language, simulation models can be built rapidly and the time consuming activities of debugging and remodeling are minimized, allowing users to focus on problem-solving rather than simulation mechanics.

As important as the simulation language is to describing systems, its use in problem-solving can be greatly enhanced by the *simulation environment*. On microcomputers running PC-DOS or MS-DOS, like the IBM PC and its compatibles, or on minicomputers running UNIX, such as the DEC VAX, the HP 9000, etc., INSIGHT is implemented in a fully **interactive** environment. Models are constructed and analyzed by *interacting* directly with the computer. Help is immediately available and support exists for selecting input distributions, writing specifications, and diagnosing errors. Interactive simulation means that modelers obtain immediate feedback on the acceptability of their model and they can participate directly in the simulation to study both the dynamic and static behavior of the system, while it is executing.

1.1 Specific Features

INSIGHT differs from other simulation languages in several specific and important ways:

1. Incorporating many general and unique modeling concepts which can be imitated in other languages only by resorting to programming. These include reneging, free queues, algorithmic resource decision making, multiple and simultaneous resource requirements, activity abortion, multilevel preemption, early/late arrivals, process synchronization, arbitrary gather grouping and queue departure processing, queue capture, set identification and attribute inheritance, etc.

2. Using a specification language that permits run-time evaluation so that specifications can be state-dependent and arbitrary functions of the entities, the system status, and the statistics. All information describing the simulation is directly available without calls to subroutines or procedures. Expressions are not limited to arithmetic, conditional, and logical operations, but can incorporate assignments, decisions, and iterations to generalize and extend the modeling concepts and features.

3. Providing nonprocedural methods of statistics collection and display of simulation information. In addition to automatically produced statistics, which can be modified, a broad range of other statistics and displays are available. For these, the modeler simply specifies what is needed and INSIGHT determines how to collect and display the results. Advanced statistical procedures for constructing confidence intervals and employing variance reduction are directly available without special routines or user-specified procedures.

4. Possessing a wide variety of statistical input mechanisms for reflecting a broad range of input models. INSIGHT has a full set of standard built-in distributions, and facilities for arbitrary discrete and continuous distributions, such as those obtained from data. Additionally, a time-varying Poisson process generator is available to model time-dependent processes and a multivariate Johnson system generator can be used for arbitrarily related multivariate input. The INSIGHT "help" facility aids in identifying and specifying appropriate input models.

5. Being portable between mainframe, mini, and micro computers and running similarly in different environments and on different machines. INSIGHT uses common random number and variate generators for all implementations and all versions are completely compatible, maintaining thirty-two bit accuracy. The micro and mini computer environment is fully interactive and can be used to construct models which can be uploaded when execution demands greater execution speed.

6. Is fully supported and extensively tested, by being used in practice and in classroom. There is a textbook, *Simulation Modeling and Analysis with INSIGHT* by Stephen D. Roberts (1983), and a user's manual exits for the mainframe version (SysTech, Inc. 1985) and the interactive INSIGHT version (Roberts 1986). These documents provide specific information on implementation details, error recovery, time and space use, and statistical features. The language and its environment is distributed and supported by SysTech, Inc.

1.2 Background

The INSIGHT language and its simulation environment have evolved from extensive actual experience over the past twelve years. For the past eight years it has been used at Purdue University in the senior course in Industrial Engineering in Systems Analysis and Design where students make extensive use of it in their projects (Roberts 1982). Much of the evolution of the language and its concepts and features have been motivated by an interest in an easy to learn and use simulation language which has general applicability and does not demand special programming or computer expertise. The primary emphasis has been the use of simulation in problem-solving.

Because the INSIGHT user deals with direct interpretation of the system, attention is focused on modeling issues. INSIGHT models are visually appealing and easy to document. These models provide an excellent communication medium between the modeler and the client. Participation by the client in the modeling activity greatly enhances the credibility of the work and increases the chances that the findings will be implemented. INSIGHT has had routine application to a variety of problems involving production planning, scheduling and dispatching, staffing, bottleneck analysis, material handling, robotics, inventory control, facilities planning, resource balancing, cost analysis, and productivity improvement in a variety of industrial and service environments.

2.0 BASIC INSIGHT CONCEPTS AND FACILITIES

When modeling with INSIGHT (INS), the modeler (user) graphically conceives of the system to be simulated as a network of elemental processes. INSIGHT provides a set of modeling symbols for creating a representation of the system and a vocabulary for describing the system. Building the simulation model involves connecting modeling symbols summarized in silhouette as Figure 1, into a network that corresponds to the system being studied.

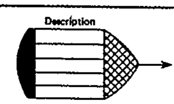
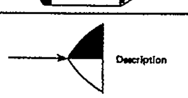
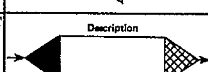
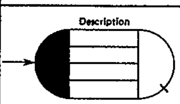
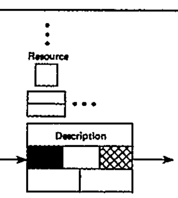
Node Type	Symbol	Processing Function
SOURCE		Creates transactions and schedules their arrival to the network
SINK		Removes transactions from the network
ASSIGNMENT		Assigns values to attributes
QUEUE		Delays transactions before an activity due to resource unavailability or a requirement to gate transactions
ACTIVITY		Performs an activity on transactions which may utilize resources
DECISION		Represents the decision process used by a resource
BRANCH		Connects two nodes in a network according to their precedence and specifies how transactions branch

Figure 1: INSIGHT Modeling Symbols in Silhouette

The INSIGHT network is constructed about the flow of units of traffic called *transactions*. A transaction is a general term that is interpreted by the modeler in the problem context. For example, transactions may represent TVs coming into an inspection station, people arriving for haircuts, ships entering a harbor or customers coming into a gas station. The nodes within the network are used to create transactions, assign attributes, cause queuing, perform activities, synchronize flow, and eventually remove transactions from the network. The branches route transactions from one node to another.

Transactions may require *resources* to process them at activities. The resource in INSIGHT is also a general term applied to an entity that services transactions at one or more activities. Several resources may be required simultaneously at some activities. Resources may exercise independent decision making in fulfilling their service requirements throughout a network. They may be preempted by other more important service requirements or they may be unavailable for service by leaving the network from time to time. Examples of resources are: inspectors who inspect TVs, barbers who cut hair, tugs which assist ships in a harbor, and gas station attendants who serve customers.

2.1 A Simple Example: TV Inspection and Adjustment

As a portion of their production process, TV sets are sent to a final inspection station. Some TVs fail inspection and are sent to an adjusting station. After adjustment, the TVs are returned for reinspection. The simple INSIGHT network needed is shown in Figure 2.

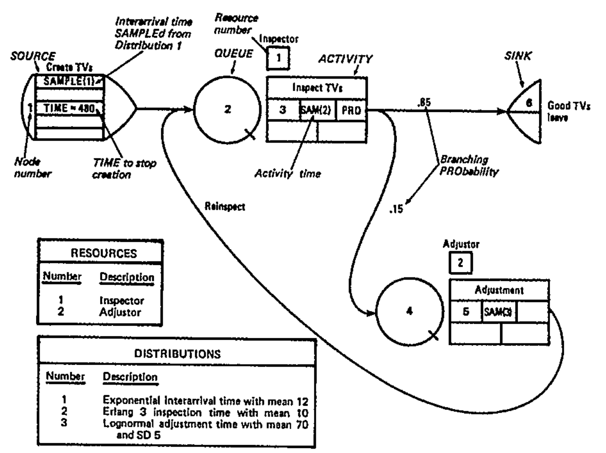


Figure 2: Network Model of TV Inspection and Adjustment

The INSIGHT Network Model. Transactions will represent TVs since they are the units of traffic. Our resources will be an inspector who is needed at the inspection activity and an adjuster who is needed at the adjustment activity. All nodes have an identifying node number located on the left side of the node. The arrow pointing out of the node is called a *branch* and indicates where the TVs go next. The TVs enter the network at a *source node*. The source node controls when and how many TVs will arrive. At Source node 1, interarrival times are determined by SAMples from statistical distribution number 1 and TVs will be created until simulation TIME is 480. The SAMple specification is just one of the many *System-Defined Functions (SDFs)* available to help in writing specifications.

Inspection and adjustment of TVs are represented by *activity nodes*. Activities are places in the network where transactions usually receive service and are delayed in their journey through the network. Only one TV can be inspected or adjusted at a time because we have only one inspector and one adjuster. TVs that are forced to wait for service do so in *queue nodes*. Queue nodes are always adjacent to activity nodes so no branching from them is required. TVs wait at Queue 2 (shaped like a Q) until the inspector (identified as Resource number 2) can inspect them. Inspection time at Activity 3 is obtained by a SAMPLE from Distribution 2. Eighty-five percent of TVs departing Activity 3 are good and leave the network while 15% are routed to adjustment. Branching from inspection is denoted by the PRObabilistic branching method associated with the node. TVs which successfully pass inspection are no longer needed and leave the network at the *sink node*. Appended to the network are the glossaries identifying the resources available to the network and defining the set of statistical distribution references.

Thus the INSIGHT network visually corresponds to our understanding of the real system. The symbols within the network not only convey individual processes but also contain relevant numerical data that control the processes. Very large or complex models can be created from such simple, basic processes by carefully assembling nodes and branches. The focus of modeling is confined to the construction of the network. Because the network has an intuitive appeal, it can be explained to decision makers in an effort to encourage their involvement in the modeling process.

Using the INSIGHT Modeler. The INSIGHT Modeler is a part of the Interactive INSIGHT system and facilitates model construction by conversing with the user during the model construction process. The user simply tells the Modeler what general elements are in the model, like an activity node, and the Modeler then queries the user about its characteristics, such as the activity time and resource requirements. The following illustrates only a portion of an introductory session. User input occurs only after a ":" or a "?" prompt and is shown in full caps, although lower case is acceptable. The <CR> is used to denote where there is only a carriage return.

```
C> MODEL

Filename for Model? TV
[creating file tv.mod]

type in knowledge level (range 1-7) = 2

:SOURCE

network section
          node number = 1? <CR>
          node name = SOURCE1? CREATE TVS
          interarrival time = 0.0? SAMPLE(1)
          PARSER: nonexistent distribution 1 in SAM SDF
          PARSER: expression value = <run time expression>
          time to begin creation = 0.0 <CR>
          termination criterion = NUM? ;HELP

          Alphanumeric beginning with:
            NUM:   create the NUMber of blocks of transactions
                   specified by termination value
            TIM:   create blocks of transactions until the
                   TIMe specified by termination value
            or Expression whose value must be:
             OPTion(NUMber) or OPTion(TIMe)

          termination criterion = NUM? TIME
[checking interarrival time value ...]
          PARSER: nonexistent distribution 1 in SAM SDF
          PARSER: expression value = <run time expression>
          termination value = 0.0? 480
          PARSER expression value = 480.00000
          block size = 1? ;SKIP
          number of branches = <unresolved>? 1
            branch sub-entity #1
                   branch number = 1? <CR>
                   destination node = <unresolved>? 2
[warning -- nonexistent node reference]
                   branch condition = [true]? <CR>
                   *where next = ;CO? <CR>
```

The Modeler operates by responding to commands or accepting responses to questions. At the *command* level, denoted by the ":" prompt, the user issues instructions to the Modeler. Most of these instructions designate elements of the model, such as a resource, queue, or activity. Some instructions are housekeeping or status commands which enable the user to manipulate the model and produce various files. At the *response* level, denoted by the "?" prompt, the user answers questions posed by the Modeler with respect to model elements. For instance, the Modeler will question the user about the node number or node name or some characteristic of the node. Generally, the questions provide a "default" answer, so you can enter a specific answer or accept the default answer by a carriage return. There are a set of special responses which allow the user to issue commands at the response level, the most common of which is to obtain ";help."

As you answer questions about the model, the Modeler checks the acceptability of your response and will inform you if the response is inconsistent with the rest of the model or if the response is erroneous. You can change your response and alter the model immediately, without any more complex manipulations. The Modeler keeps track of all specifications and its advice is based not only on its general knowledge about INSIGHT, but also its growing knowledge about the model being built, typical of an *expert system*. The model will even accept complex, multi-line expressions and edit them conveniently while checking them for inconsistencies or misuse.

A special advantage in using the INSIGHT Modeler is that users need only be knowledgeable about the INSIGHT modeling concepts. Details about specific features will be answered through the questioning by the Modeler. This approach is particularly valuable to modelers who do not want to be involved in the great detail, so often required of simulation users. Also, as a special aide to learning and use, the Modeler has a "knowledge level" that stipulates the level of INSIGHT knowledge required, so that more advanced concepts can be hidden from routine use, and the Modeler won't request more advanced information from a user whose knowledge is less.

You can cause your entire model to be checked for general as well as specific errors, possibly to correct them now or at some later time. The model can be saved during any stage of its development and restored later to be completed. However when the user believes the model is complete, the INSIGHT *statement model* needs to be created. The Modeler will perform this function directly. A statement model for the TV problem is given in Figure 3.

```
 SIMULATION OF = TV INSPECTION AND ADJUSTMENT,,, RUNS = 10
$
    RESOURCE = 1  INSPECTOR,, 2
    RESOURCE = 2  ADJUSTOR,, 4
$
    DISTRIBUTION = 1, EXPONENTIAL INTERARRIVAL TIME WITH MEAN = 12 MINUTES
    DISTRIBUTION = 2, ERLANG INSPECTION TIME WITH MEAN = 10 MINUTES, 3RD ORDER
    DISTRIBUTION = 3, LOGNORMAL ADJUST TIME WITH MEAN = 70 MINUTES, 5 MIN SD
$
$*** THE NETWORK
$
    SOURCE = 1  CREATE TVS,, SAMPLE(1),, TIME TO STOP CREATING TVS = 460
       BRANCH TO, 2
$
    QUEUE = 2  INSPECT WAIT
    ACTIVITY = 3  INSPECT TVS, PROBABILISTIC BRANCHING WITH ACT TIME = SAM(2)
       SELECT,,, 1 THE INSPECTOR
       BRANCH TO, 6 WITH PRODADILITY = .85
       BRANCH TO, 4 WITH PROBABILITY = .15
$
    QUEUE = 4  ADJUST WAIT
    ACTIVITY = 5 ADJUST TVS WITH ACT TIME, = SAMPLE(3)
       SELECT,,, 2 THE ADJUSTOR
       BRANCH TO, 2 FOR REINSPECTION
$
    SINK = 6  GOOD TVS LEAVE
 FINISH
```

Figure 3: Statement Model of TV Inspection and Adjustment

The INSIGHT Statement Model. The statement model is the intermediate, computer readable form of the simulation model. From the statement model, INSIGHT automatically compiles and executes the simulation and provides output. The user is free of troublesome details such as event handling, statistics collection, and report writing. Furthermore, it is not necessary for the problem to be interpreted into some restricted simulation programming structure.

The Modeler will automatically generate the statement model from its interaction with the user, however where the interactive INSIGHT system is unavailable, the user can create and edit the statement model using a text editor. Each INSIGHT statement begins with a key word, followed by specifications in fields, separated by commas. Comments can be used throughout the statement model to both document the function and to make the input more readable.

The form of the statement model shows its close correspondence to the network model. Each network element has a statement equivalent. The *SIMULATION* statement introduces the simulation problem and supplies identifying information. The glossaries define the three statistical *DISTRIBUTIONs* and the two *RESOURCEs*. Notice the use of the built-in statistical distributions. Resources and nodes may be given names. The node and branch statements are written as they appear in the network with the *BRANCH* statement following its node. Each node from which transactions may depart also includes the method of branching. TVs branch Probabilistically from Activity 3; at other nodes, TVs go directly to the indicated node. The *SELECT* statement tells INSIGHT which resources serve transactions at the activity. A *FINISH* statement always terminates the model description.

The statement model is submitted to the INSIGHT Compiler to construct the form of the executing simulation. INSIGHT uses a two-pass compiler to construct an efficient executing simulation. The compiler acts quickly. In Interactive INSIGHT, you can observe the interpretation of the statement model and optionally obtain an *echo* of the model. The echo of the model includes the values of all specifications, both those given by the user and those assumed by INSIGHT, and can be very useful in debugging and model verification.

Executing the Simulation. With Interactive INSIGHT, the execution of the simulation provides for extensive interaction through a menu interface. The following is an example of the main menu at the beginning of the simulation.

```
C> simulate

FILENAME OF MODEL TO BE EXECUTED? tv

tv.lod
NOW LOADING ...

** EXECUTION OF THE SIMULATION CAN BE ATTEMPTED **

THE CURRENT RUN =   1    THE CURRENT TIME =     .00000

YOU MAY NOW:
   1. CONTINUE
   2. UNLOAD
   3. ENTER ONE EVENT MODE (which turns on the trace)
   4. LEAVE ONE EVENT MODE (which may turn off the trace)
   5. SCHEDULE THE NEXT INTERRUPT
   6. CHANGE THE FREQUENCY OF CURRENT TIME UPDATES
   7. ENTER THE ANALYZER
   8. STOP

CHOOSE ONE?
```

The display of the current run and current time will appear as the simulation is executed and can be controlled by changing the frequency of current time updates. The user may interrogate the simulation at any time during its execution, either by scheduling an interrupt from this menu or by striking the carriage return. In either case the main menu appears.

Further execution of the simulation can employ *one-event* mode or direct execution. In one-event mode the trace or English-like description of each event is displayed. An example of the Interactive INSIGHT system trace is the following:

```
THE CURRENT RUN =   1   THE CURRENT TIME =      .000000
-----------------------------------------------------------------------
TRACE OF RUN 1

CUR(TIM) =       .0000
 INSPECTOR       1 HAS ARRIVED
 INSPECTOR       1 IS IDLE

NEXT EVENT? y

CUR(TIM)          .0000
 ADJUSTOR         2 HAS ARRIVED
 ADJUSTOR         2 IS IDLE

NEXT EVENT? y

CUR(TIM) =        .0000
 TRANS       1 CREATED      AT  CREATE TVS         1
 TRANS       1 ENTERING         INSPECT WAIT       2 NUM(QUE,*) =      0
 TRANS       1 DEPARTING        INSPECT WAIT       2 TIME IN QUEUE =        .000
 INSPECTOR         1 BUSY       AT INSPECT TVS       3  ON TRANSACTION     1
 TRANS       1 BEGINNING    AT  INSPECT TVS        3

NEXT EVENT? y

CUR(TIM) =      10.2421
 TRANS       1 COMPLETED  AT  INSPECT TVS          3 ACTIVITY TIME =     10.242
 INSPECTOR          1 IN BUFFER STORAGE
 TRANS       1 DESTROYED AT GOODS TVS LEA          6 CUR(RTC) =        0
 INSPECTOR          1 IS IDLE

NEXT EVENT?
```

Anytime during the simulation, the current state of the simulation may be *unloaded*. When a model is unloaded, its entire status, including statistics collected, is saved. This state may be reloaded at some later time and the simulation continued or the state of the system examined. This interaction can give a user new perspectives on the system being examined.

Analyzing the Simulation. By choosing the *analyzer* option from the main simulation menu, you can examine and possibly alter the simulation model. Within the analyzer, you can review or print reports, review the status of the model, cause actions, and edit the model without re-compilation. The following is the main Analyzer menu:

```
THE FOLLOWING OPTIONS ARE AVAILABLE:
      1. INSIGHT REPORTS
      2. STATE OF THE NETWORK AND NETWORK ATTRIBUTES
      3. NUMBER (QUANTITY) OF TRANSACTIONS AND RESOURCES
      4. STATUS OF THE CURRENT TRANSACTION
      5. STATUS OF RESOURCES
      6. VIEW SPECIFIC STATISTICS
      7. MOVE THE TRANSACTION POINTER
      8. CAUSE ACTIONS
      9. EDIT MODEL
     10. UNLOAD
     11. EXIT ANALYZER
```

The analyzer provides interactive access to all information provided by INSIGHT including the SDFs, attributes, and statistics. For instance, if during the simulation you wanted a confidence interval on the number of transactions in Queue 6, you could experience the following interaction:

```
THE FOLLOWING STATISTICS TYPES ARE AVAILABLE:
      1. NUMber of transactions in a specific queue node
      2. time in a specific queue INCluding zero waits
      3. time in a specific queue EXCluding zero waits
      4. NUMber of transactions in a specific activity node
      5. activity TIMe in a specific activity node
      6. total time in the NETwork when leaving at a specific sink node
      7. total time in QUEues when leaving at a specific sink node
      8. total time in ACTivities when leaving at a specific sink node
      9. statistical information on a specific table
     10. statistical information on a specific breakdown value
           or period of a specific table
     11. UTilization of a specific resource(s)

CHOOSE ONE? 1

WHICH NODE NUMBER? 6

DO YOU WANT A CONFIDENCE INTERVAL FOR YOUR MEAN? y

THE FOLLOWING CONFIDENCE LEVELS ARE AVAILABLE:
      1. 99 PERCENT
      2. 95 PERCENT
      3. 90 PERCENT

CHOOSE ONE? 2

MEAN OF ALL RUN OR BATCH MEANS OF THE
NUMBER OF TRANSACTIONS IN QUEUE     6 =       1.,00796
WITH A 95% CONFIDENCE INTERVAL HAVING       2 d.f. = [  .09222, 1.92370]
```

INSIGHT automatically collects a comprehensive set of statistics about the simulation. A variety of reports based on these or additional statistics can be generated and be printed to the screen or to a file. The *Summary Report* presents the entire set of statistics and Figure 4 displays the summary report for the TV inspection problem at the end of the simulation, although it could be examined at any time. Without any specific instruction from the user, the summary report consists of three reports. The Network Status Report describes the current state of the network. The Node Statistics describe: the number of transactions and time spent in each queue (including and excluding zero times); the number of transactions and the time spent in each activity; and the time in the network, in queues, and in activities for the transactions which exit the network at each sink node. Resource Statistics include utilization and availability information.

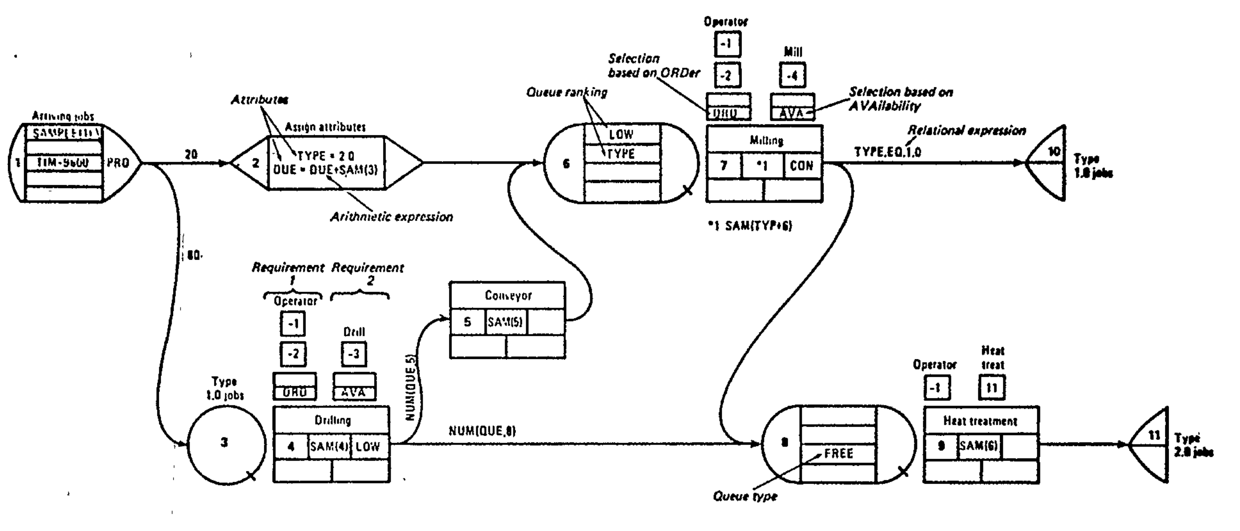
2.2 A Few More Modeling Concepts: Flexible Production Cell

To illustrate a few more modeling concepts within INSIGHT, consider the following problem. A flexible production cell has three machine groups, drills, mills, and heat treatment, which are used with the flexibility of a job shop. The machines are operated by two categories of operators. One category can operate all three machine types but gives priority to heat treatment. The second category operates only the mills and drills. Two general job types are processed through the department and their dispatching at each machine group depends on job type at the mills and on job slack time at the heat treatment. The only important material handling occurs between the mills and the drills using a conveyor.

A model of the cell would naturally use transactions to represent the jobs and resources to represent the operators and the machines. The INSIGHT network model is given in Figure 5. Activities 4, 7, and 9 represent the drilling, milling, and heat treatment activities respectively. Jobs are created at Source 1 until the simulation TIMe is 9600 minutes (20 eight-hour days). Each transaction is assigned a TYPe and DUE date by associating these values with the transactions through INDividual transaction attributes. The attributes are declared in a glossary to the network. Each transaction initially created has a TYPe of 1.0 and DUE date set to the CURrent simulation TIMe plus a SAMple from Distribution 2. The specification of CUR(TIM) is another one of the SDFs available in INSIGHT which may supply information about the network and will be evaluated during the execution of the simulation.

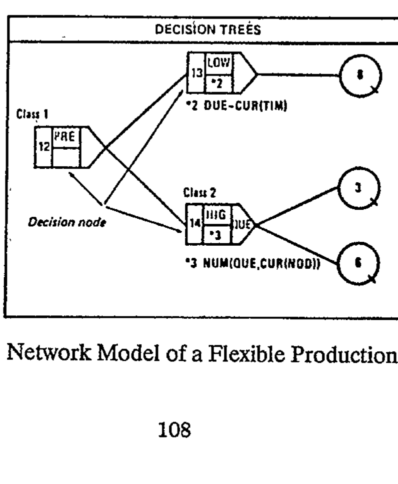
Although 80 percent of the transactions move into Queue 3, 20 percent enter Assignment node 2 where their TYPe and DUE date are changed to reflect a different (the second) job type. Assignment nodes are used to change the values of attributes in the network. Notice an arithmetic expression can be used to compute specific values. Attribute values may be used throughout the network. At Queue 6, transactions are ranked by putting those with the LOWest value of TYPe first (i.e., jobs from drilling are ranked ahead of jobs just arriving). The activity time for milling is SAMpled from the distribution computed as the value of TYPe plus 6 (SDF arguments may be expressions). Branching from the milling activity (Activity 7) is CONditioned on the value of TYPe. If TYPe is equal to 1.0 (a relational expression), then the job exits the department at Sink 10. Jobs of TYPe 2.0 branch to Queue 8 to wait for heat treatment.

Branching from the drills (Activity 4) is based on the



RESOURCES			
Number	Description	Type	Primary Service Node
1.TO. 2	Class 1 operators	-1	12
2 TO. 5	Class 2 operators	-2	14
5.TO. 7	Drills	-3	3
7.TO.10	Mills	-4	6
11	Heat treatment		8

ATTRIBUTES			
Label	Description	Scope	Initial Value
TYPe	Job type	IND	1.0
DUE	Due date	IND	CUR(TIM)+SAM(2)



DISTRIBUTIONS	
Number	Description
1	Job interarrival time
2	Due date for Type 1.0 jobs
3	Due date for Type 2.0 jobs
4	Drilling time
5	Conveyor time
6	Heat treat time
7,8	Milling times

Figure 5: Network Model of a Flexible Production Cell

****
NETWORK STATUS
****

TIME = CUR(TIM) =	680.247	NODE STATISTICS LAST STARTED AT TIME =	0.000
RUN = CUR(RUN) =	10	RESOURCE STATISTICS LAST STARTED AT TIME =	0.000
RTC = CUR(RTC) =	0	NODE STATISTICS COLLECTION TIME SINCE LAST CLEAR =	680.247
NODE = CUR(NOD) =	0	RESOURCE STATISTICS COLLECTION TIME SINCE LAST CLEAR =	680.247
COMPLETED ACTIVITY = CUR(ACT) =	0		
TRANSACTION = CUR(TRA) =	0	TRANSACTIONS IN NETWORK = NUM(NET) =	0
RESOURCE = CUR(RES) =	0	TRANSACTIONS CREATED = NUM(CRE) =	40
		TRANSACTIONS DERIVED = NUM(DER) =	0

MAXIMUM SPACE UTILIZED = 725/ 5000 OR 14.50 PERCENT

	NODE NAME	NODE NUMBER	NODE COUNT COU(*)	NUMBER IN NODE	ZERO WAIT COUNT ZER(*)
QUEUES	INSPECT WAIT	2	449	0	122
	ADJUST WAIT	4	67	0	23
ACTIVITIES	INSPECT TVS	3	449	0	
	ADJUST TVS W	5	67	0	
SINKS	GOOD TVS LEA	6	382		
SOURCES	CREATE TVS	1	382		

** CURRENT SEEDS **

SYSTEM	=	610234094
DISTRIBUTION	1 =	44620978
DISTRIBUTION	2 =	406037205
DISTRIBUTION	3 =	192317993

****
NODE STATISTICS
****

QUEUE NODES NUMBER OF TRANSACTIONS IN QUEUE

NODE NUMBER	NAME	------------ OBSERVATIONS OF THE MEANS MEAN MEA(NUM,*)	TIME OF OBSERVATION OBS(NUM,*)	STANDARD DEVIATION STD(NUM,*)	STANDARD ERROR STE(NUM,*)	-- ALL OBSERVATIONS -- SMALLEST SMA(NUM,*)	LARGEST LAR(NUM,*)	CURRENT RUN OR BATCH ONLY MEAN CME(NUM,*)	TIME OF OBSERVATION COB(NUM,*)
2	INSPECT WAIT	1.42561	6829.15	1.64405	0.519893	0	16	0.487982	680.247
4	ADJUST WAIT	0.674818	6829.15	0.564294	0.178445	0	4	1.49339	680.247

QUEUE NODES TIME IN QUEUE INCLUDING ZERO WAITING TIMES

2	INSPECT WAIT	21.6831	10	22.1288	6.99773	0.0000E+00	197.3	6.91559	48
4	ADJUST WAIT	68.7826	10	46.7156	14.7728	0.0000E+00	272.8	126.984	8

QUEUE NODES TIME IN QUEUE EXCLUDING ZERO WAITING TIMES

2	INSPECT WAIT	29.7728	10	24.9016	7.87458	0.5038	197.3	9.76319	34
4	ADJUST WAIT	104.737	10	46.9386	14.8433	5.514	272.8	145.125	7

ACTIVITY NODES NUMBER OF TRANSACTIONS IN ACTIVITY

3	INSPECT TVS	0.663770	6829.15	0.117624	0.371959E-01	0	1	0.666321	680.247
5	ADJUST TVS W	0.683530	6829.15	0.163073	0.515683E-01	0	1	0.848396	680.247

ACTIVITY NODES ACTIVITY TIME

3	INSPECT TVS	10.0957	10	0.868443	0.274626	0.7268	33.72	9.44297	48
5	ADJUST TVS W	69.6705	10	1.97769	0.625402	59.25	82.07	72.1399	8

SINK NODES TIME IN SYSTEM

6	GOOD TVS LEA	61.6362	10	28.9446	9.15309	1.550	527.0	59.4551	40

****
RESOURCE STATISTICS
****

---------- OBSERVATIONS OVER RUNS OR BATCHES ----------
TIME EACH RESOURCE WAS OBSERVED = 6829.15

NUMBER(S) OR TYPE	NAME	MEAN UTILIZATION MEA(UTI,*)	NUMBER OF OBSERVATIONS OBS(UTI,*)	STANDARD DEVIATION STD(UTI,*)	STANDARD ERROR STE(UTI,*)	SMALLEST SMA(UTI,*)	LARGEST LAR(UTI,*)
1	INSPECTOR	0.66377	10	0.11762	0.03720	0.46288	0.83045
2	ADJUSTOR	0.68353	10	0.16307	0.05157	0.35170	0.84840

Figure 4: Summary Report for TV Inspection and Adjustment

LOWest value of the branching expressions following the activity. The SDF referring to NUMber in QUEue is used to supply the size of Queues 6 and 8. The shortest queue length consequently specifies where transactions branch. If movement is from Activity 4 to Queue 6, then transactions are delayed in Activity 5 to represent the movement on the conveyor. No resources are required for this activity.

Activities 4, 7, and 9 each require two resources, an operator and a machine, before the activity can begin. Both must be available simultaneously and the requirements are represented by the two columns of symbols above each activity. However, there are choices among alternatives for each requirement so the selection method is specified. Resources may be referred to by resource type using negative numbers. Thus resources of Type -2 (class 2 operators) are in preferred ORDer to resources of Type -1 (class 1 operators). This means that whenever a choice among resources is possible, Type -2 is considered before Type -1. Resources of the same type are chosen based on whichever has been AVAilable for selection (idle) the longest.

However, when an operator finishes an activity, its next service must be determined. This decision is accomplished by reference to the primary service nodes of the resources. Each machine group services a single queue but the operators may serve more. Their choices are represented by one (or more) decision trees, composed of decision nodes and queues. The decision trees are found in a network glossary. In this case, the class 2 operators serve a subset of the class 1 operators. Class 1 operators PREfer service at Queue 8. However, the service of a particular job from Queue 8 is based on LOWest slack time, given as the DUE date less the CURrent simulation TIMe. Notice that the relative dispatching of each job can change dynamically and therefore, Queue 8 is designated as being FREe to permit any job to be served regardless of its position in the queue. If Queue 8 is empty when considered or if the resource is Type -2 (class 2 operator) then Decision node 14 is employed. Queues 3 and 6 are examined and that QUEue having the HIGhest NUMber in the QUEue will be serviced.

Therefore, INSIGHT easily accommodates some of the most complex aspects of modeling manufacturing systems. Multiple and simultaneous resource requirements at activities are possible. Resource decision making can be incorporated. Branching involving conditions and expressions can be employed. Queue ranking and dynamic dispatching are easily handled, relying on the attributes of transactions. The importance of these extensions is that they did not clutter the network and the modeling framework retains its visual and communication benefits. In addition, even detailed specifications do not destroy the visual impact. Furthermore, all this was accomplished without any programming.

3. MORE MODELING AND SIMULATION FACILITIES

The previous example illustrates only extensions to the simplest features. Many other behaviors can be modeled similarly. Reneging and balking at queues can be directly incorporated. The definition of a queue can be broadened to include waiting for a gate condition and/or gather requirements to be satisfied. Transactions may be gathered and grouped at queues and then processed together in the associated activity.

Important realistic complications like resource preemption and its impact on other resources and transactions can be specified. Transactions can select resources in many ways and capture resources at a queue or activity. Many other features are incorporated, all within the context of the six INSIGHT modeling symbols. Although such embellishments greatly extend the modeling potential, they are readily incorporated into the network without sacrificing its visual clarity. Many are added to the node description and the Modeler will request such information at a higher "knowledge level."

3.1 Getting "help"

A very important dimension of the INSIGHT facilities is the on-line help which supports INSIGHT documentation. The *help* can be invoked directly within other INSIGHT programs, such as asking about possible responses for network specifications, and the help program can be run in its own shell. The basic options if the help is requested directly, are:

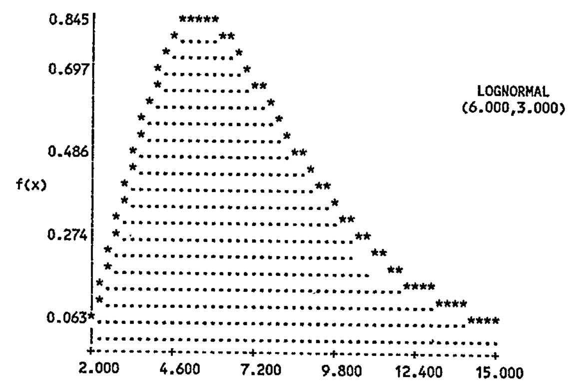
```
The following types of help are available:
          1. Distribution parameter conversions
          2. Distribution plots
          3. Error messages and recommendations
          4. Index for INSIGHT textbook
          5. Statement specifications(Appendix A)
          6. SDF specifications(Appendix B)

Enter q(uit) to terminate help session of 1-6 or help type:
```

Much of the help is oriented to supplying more detailed information about specifications and errors, while other help is more general such as supplying references and distribution plotting. Examining distributions is a part of the input modeling capabilities and one example plot is shown below for a Lognormal:

```
Enter the mean: 6

Enter the standard deviation: 3.0
```



3.2 INSIGHT Expressions

Much of the generalization in INSIGHT modeling is due to the powerful specification expressions. Expressions are combinations of primitive elements including constants, SDFs, attributes, and functions (which are themselves expressions) which are evaluated as the simulation executes. The SDFs represent system-defined elements, while the attributes and functions are user-defined. In addition to arithmetic, relational, and logical expressions, INSIGHT will accept assignment, decision, and iteration constructs. For example, the expression

```
(ACT = MAX(CUR(TIM),15-SAM(3)))
```

causes the attribute ACT to be assigned the MAXimum of the CURrent TIMe and 15 less a SAMple from Distribution 3. Furthermore, the value of the expression itself is the assigned value and can be used to specify, for instance, an activity time. A decision expression as

```
.IF.TYPE.EQ.1 .OR. NUM(QUE,4).GT.5
  .THEN. 5
  .ELSE. SIZE
```

has as its value either 5 or the value of SIZE depending on whether the condition, TYPE equal to 1 and NUMber in QUEue 4 greater than 5, is true or false. An iteration expression such as

```
.WHILE.(I=I+1).LT.NUMBER
   .DO.(TOT=TOT+TIM(BUSY,1))
```

adds to the present value of TOT the value of BUSy TIMe for resource I iterating from I to NUMBER.

3.3 Special Functions

To obtain information about any transaction in the network, INSIGHT provides an Internal Transaction Pointer (ITP) which the modeler can employ anytime an expression is evaluated. It is another SDF. For example,

```
ITP(3,5) + ABC
```

yields the value of ABC for the third transaction in node 5. The ITP plays an analogous role in modeling to pointer variables in modern programming languages.

Other special SDFs exist within INSIGHT to control many other actions. For example, statistics may be cleared or started/stopped, special resports can be generated during the simulation, activity times may be changed, the simulation run can be stopped, resources can be made to arrive differently, etc. These functions can themselves satisfy specifications or become part of expressions which specify the model.

Expressions may be made arbitrarily complex to reflect sophisticated specifications and behaviors without changing the visual structure of the model. Because they are evaluated at run time, their specification can be symbolic to take advantage of state-dependent information. Thus the network serves primarily as a structuring tool while the specifications actually prescribe the model behavior. Again, even with these considerable complications, there remains no need for programming outside the INSIGHT language.

3.4 Statistical Analysis

In addition to providing a high level approach to simulation modeling, INSIGHT also contains a number of built-in features to make accurate estimates of the variance of the sample mean and to perform variance reduction. The modeler can directly specify that INSIGHT estimate variances by *replications* (uniquely seeded runs of the same simulation model) or by *batches* (division of observations in to collections of observations). INSIGHT does not automatically provide an estimate of the standard error from observations known to lack independence. This is why the output in the previous sections uses one observation per run to compute variances. Similar computations could be obtained using batches when dealing with a steady-state simulation. Tables can be used for more complex statistics collection.

Tables are a unique feature of INSIGHT in that they employ a nonprocedural format. The desired statistics are described to the Modeler and INSIGHT automatically handles the tasks of collecting, compiling, and displaying statistics. For example if within the Modeler we requested a table as:

```
:TABLE
          table number = 1? <CR>
          table name = TABLE1? NUMBER IN INSPECT QUEUE
          statistic type = <unresolved>? NUM
          number of referenced entities = <unresloved>? 1
                queue/activity node number #1
                      node number = <unresolve>? 2
                      *where next = ;CO?
          number of histogram cells = 0? 10
          lower limit of first cell = 0.0? 1
          cell width = VAR? 1
          collection control = COM? <CR>
          table breakdown = NO? <CR>
```

which produces the following table and histogram in Figure 6.

Start-up issues requiring special initial conditions can be specified directly within INSIGHT by the PRERUN that initializes variables and attributes and inserts transactions in nodes. Truncating data during start-up is accomplished by SDFs that can clear specified statistics. Clearing can occur within the network or be activated at a specific time. Furthermore, run length or batch size can also be controlled

```
*********************
TABLES AND HISTOGRAMS
*********************
```

** TABLE NUMBER 1 **

NUMBER IN INSPECT QUEUE

	OBSERVATIONS OF THE MEANS				ALL OBSERVATIONS		CURRENT RUN OR BATCH ONLY	
	MEAN MEA(TAB,*)	TIME OF OBSERVATION OBS(TAB,*)	STANDARD DEVIATION STD(TAB,*)	STANDARD ERROR STE(TAB,*)	SMALLEST SMA(TAB,*)	LARGEST LAR(TAB,*)	MEAN CME(TAB,*)	TIME OF OBSERVATION COB(TAB,*)
AGGREGATE	1.42561	6829.15	1.64405	0.519893	0.000000E+00	16.0000	0.487982	680.247

** HISTOGRAM NUMBER 1 **

NUMBER IN INSPECT QUEUE

TIME OF OBSERVATION	RELATIVE FREQUENCY REL(*,*)	CUMULATIVE FREQUENCY CUM(*,*)	CELL LIMIT LOWER	CELL LIMIT UPPER
0.000000E+00	0.000	0.000	-INFINITY	0.000000E+00
3860.12	0.565	0.565	0.000000E+00	1.00000
1076.19	0.158	0.723	1.00000	2.00000
678.177	0.099	0.822	2.00000	3.00000
453.342	0.066	0.889	3.00000	4.00000
217.096	0.032	0.920	4.00000	5.00000
100.134	0.015	0.935	5.00000	6.00000
76.8508	0.011	0.946	6.00000	7.00000
14.9505	0.002	0.948	7.00000	8.00000
20.8436	0.003	0.951	8.00000	9.00000
53.0242	0.008	0.959	9.00000	10.0000
65.4680	0.010	0.969	10.0000	11.0000
58.7343	0.009	0.977	11.0000	12.0000
28.5891	0.004	0.982	12.0000	13.0000
67.7557	0.010	0.992	13.0000	14.0000
31.2228	0.005	0.996	14.0000	15.0000
23.8638	0.003	1.000	15.0000	16.0000
2.78857	0.000	1.000	16.0000	17.0000
6829.15			LOWER LIMIT INCLUSIVE	

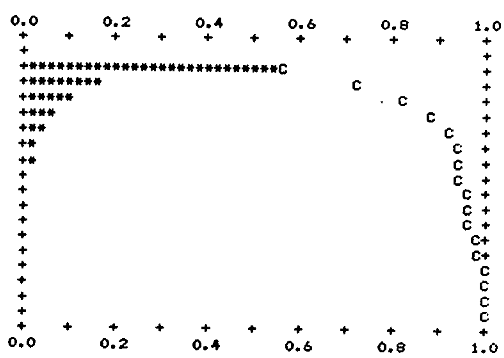


Figure 6: Additional Statistics Provided by Tables and Histograms

within the network model by employing expressions which can terminate a run or batch interval. Statistics collection can also be stopped and started.

Variance reduction techniques employing common, antithetic, and paired random variates are available in INSIGHT by direct specification. By default each distribution in the model has its own separate random number stream. There are over 2100 individual streams in INSIGHT. Thus different experiments with a model automatically use common sources of variation. Within an experiment, some runs may have common streams and other runs may be made antithetic automatically. To maximize the applicability of these variance reduction procedures, INSIGHT employs inverse transform variate generators so that random numbers are inherently synchronized. By including these features in INSIGHT, statistical analysis can become an integral part of simulation modeling.

Special attention has been given to random number and random variate generators in INSIGHT. The random number generator (Marse and Roberts 1983) is completely portable to any machine having 32 bit or greater word length while retaining excellent statistical properties. The implementation of the generator not only permits the automatic use of common variates but in combination with the variate generators causes INSIGHT simulation models to run identically on different computers. This makes simulation results portable. The variate generation process using inverse transforms is extensive, ranging from standard distributions to a time-varying Poisson process (Klein and Roberts 1984) and a new multi-variate Johnson system generator that permits up to 42 dependent variates. More discussion of statistical issues in INSIGHT is found in Roberts and Klein (1982).

3.4 Programming Interface

To allow conversation with other software, an interface is available which allows two-way communication of information. Using the interface, INSIGHT can be used in trace-driven simulations. The interface can be used to store model-generated data in a file which is then available for further analysis. Complex or frequently used procedures may be written in FORTRAN, C, or other linkable languages to reduce execution time. SDFs and attributes may be used in the routines to provide current status information, or cause simulation actions, or obtain statistical information which can be used to print reports tailored to the model. Facilities exist for user-determined events and statistics collection. Finally, a program can be written to execute a simulation many times while testing various model parameters to optimize an objective function.

4. CONCLUSIONS

INSIGHT provides an easy-to-use simulation capability that contains powerful modeling concepts that do not rely on general programming. Such an approach makes simulation modeling available to those with little prior experience and further extends the scope of possible simulation applications. Built-in procedures for statistics collection and automatic output generation mean that results are obtained easily and quickly.

INSIGHT is fully interactive and the modeling and simulation requirements are further reduced since the system can now provide so much help in modeling and simulation. Also since the simulation is interactive, the user can obtain both static and dynamic information about the simulation. INSIGHT runs under a variety of operating systems on mainframes, minis, and micros with results being fully portable. The language is supported by SysTech, Inc. at P.O. Box 509203, Indianapolis, IN 46250 which also is responsible for the distribution and maintenance of the simulation products.

REFERENCES

INSIGHT User's Manual (1985). SysTech, Inc., Indianapolis, Indiana.

Klein, R. W. and Roberts, S. D. (1984). A time-varying Poisson process generator. *Simulation* 43, 193-195.

Marse, K. J. and Roberts, S. D. (1983). Implementing a portable FORTRAN Uniform (0,1) generator. *Simulation* 41, 135-139.

Roberts, S. D. (1982). Teaching simulation to undergraduates. In: *Proceedings of the 1982 Winter Simulation Conference* (H. Highland, Y. Chao, O. Madrigal eds.). The Institute of Electrical and Electronics Engineers, San Diego, California, 706-707.

Roberts, S. D. (1983). *Simulation Modeling and Analysis with INSIGHT.* Regenstrief Institute, Indianapolis, Indiana. Distributed by SysTech, Inc.

Roberts, S. D. and Klein, R. W. (1984). Statistics collection display, and analysis in INSIGHT. Regenstrief Institute, Indianapolis, Indiana.

Roberts, S. D. (1986). *Interactive INSIGHT Simulation System.* SysTech, Inc., Indianapolis, Indiana.

AUTHOR'S BIOGRAPHY

STEPHEN D. ROBERTS is Professor of Industrial Engineering at Purdue University and Professor of Internal Medicine at the Indiana University School of Medicine. His academic and teaching responsibilities are in simulation modeling. He is Director of Health Systems Research at the Regenstrief Institute for Health Care Focusing on simulation of medical decisions. His methodological research is in simulation language design and includes INSIGHT (INS) a general purpose, discrete event language, and SLN for the Simulation of Logical Networks. He is also a principal in SysTech, Inc. which distributes the simulation languages and consults on their application.

He recieved his BSIE, MSIE, and PhD in Industrial Engineering from Purdue University and has held research and faculty positions at the University of Florida. He is active in several professional societies and in addition to presentations and chairing sessions at conferences, he was Proceedings Editor for WSC '83, Associate Program Chairman for WSC '85, and Program Chairman for WSC '86. Presently he is treasurer for SIGSIM and an Area Editor for Simulation.

Stephen D. Roberts
SysTech, Inc.
P.O. Box 509203
Indianapolis, IN 46250, U.S.A.
(317) 842-6586