Proceedings of the 1983
Winter Simulation Conference
S. Roberts, J. Banks, B. Schmeiser (eds.)

# SIMULATION MODELING WITH INSIGHT

Stephen D. Roberts, Ph.D.
Regenstrief Institute for Health Care
1001 West Tenth Street
Indianapolis, IN 46202

Using INSIGHT, simulation models can be built and evaluated with less time and effort than previously possible. INSIGHT defines a vocabulary for systems analysis and provides a small set of modeling symbols to represent the basic processes found in real systems. By combining the symbols graphically and filling in needed information, immediately useful simulation models can be constructed quickly. Simple models can be greatly embellished using the rich variety of INSIGHT specifications without resorting to any computer programming. The visual model is simply translated into INSIGHT statements for automatic execution. Procedures incorporated in INSIGHT enable sound statistical analysis to be used routinely. Such features reduce the cost and difficulty in obtaining representative results. The net result is that attention is focused on modeling and analysis.

## INTRODUCTION

Available simulation languages range from simulation programming languages to parameterized simulation models. Simulation programming languages are rooted in general purpose programming and require the modeler to code simulation procedures. Although highly flexible, these languages require considerable programming and simulation knowledge. More recent attempts to incorporate higher level built-in procedures with network or process facilities provide easier access to the language, but they fundamentally rely on general purpose programming procedures written by the modeler for extensive models. Parameterized simulation models such as those for assembly line balancing or flexible manufacturing or warehousing offer greater ease of use and require no programming. However, only highly specific systems can be studied with little possibility of generalization.

INSIGHT (INS) has been designed to be a high level, general purpose, discrete event simulation language. It does not depend on any competence with general purpose programming and modelers need not code any procedures. Its few fundamental concepts are easy-to-learn and easy-to-use immediately, making simulation modeling accessible to a greater number of persons. There is little sacrifice of generality because of the rich variety of INSIGHT specifications which extends the modeling concepts. 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. Using a modeling language, simulation models can be built with much less time than previously possible and the time consuming activities of debugging and remodeling are minimized. Additionally INSIGHT is portable, fully supported, extensively tested, and well documented.

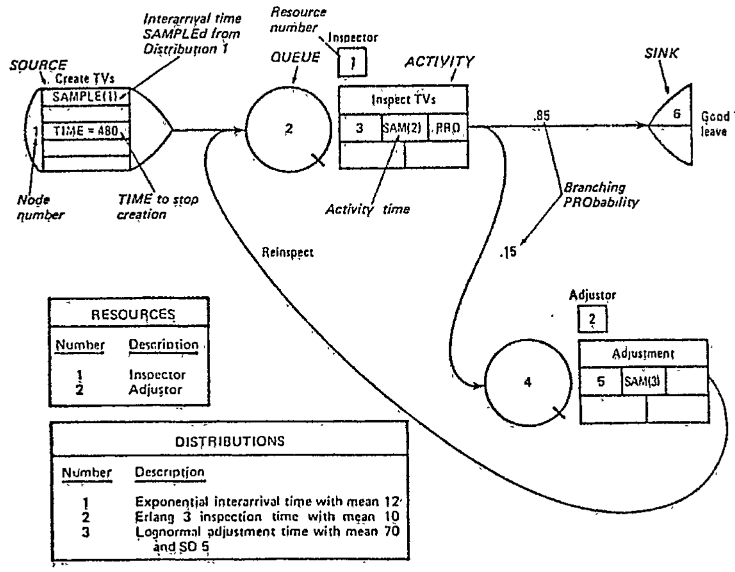
Modeling with INSIGHT (INS), the simulation modeler graphically conceives of the system to be simulated as a network of elemental processes. INS provides a set of modeling symbols for creating a representation of the system and a vocabulary for describing the system. Building the simulation model involves connecting modeling symbols, such as queues and activities, into a network that corresponds to the system being studied.

The network is translated into INS statements and INS automatically executes the simulation and provides output. The modeler 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 a next event simulation structure.

The modeler deals with a direct interpretation of the problem focusing attention on modeling issues. INS is concise, easy to document, and visually appealing. INS models make 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 probability of eventual implementation. To date, INS 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.

In this paper, we will illustrate the use of INS in simulation modeling and describe a few of its characteristics. The textbook (Roberts, 1983) fully describes the language and includes numerous examples exhibiting its many modeling features. A user's manual (INSIGHT, 1983) provides more specific information on implementation and details several specific issues including error recovery, time and space use, and statistical considerations.

Figure 1
TV INSPECTION AND ADJUSTMENT NETWORK



| RESOURCES | |
|---|---|
| Number | Description |
| 1 | Inspector |
| 2 | Adjustor |

| DISTRIBUTIONS | |
|---|---|
| Number | Description |
| 1 | Exponential interarrival time with mean 12 |
| 2 | Erlang 3 inspection time with mean 10 |
| 3 | Lognormal adjustment time with mean 70 and SD 5 |

## BASIC INS CONCEPTS

The INS 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 hair cuts, ships entering a harbor, or customers coming into a gas station. The nodes within the network 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 INS 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 requiremnts 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 service customers.

## EXAMPLE 1: 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 INS network needed is shown in Figure 1.

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 adjustor who is needed at the adjustment activity. All nodes have an identifying node number located on the left hand 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 a statistical distribution numbered 1 and TVs will be created until simulation TIME is 480.

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 one time because we have only one inspector and one adjustor. 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 Activty 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. TVs which successfully pass inspection are no longer needed and leave the network at the sink node. Appended to the network are glossaries identifying the resources available to the network and defining the set of statistical distribution references.

The network is intuitively appealing since it corresponds visually 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

Figure 2
INS STATEMENTS FOR TV INSPECTION AND ADJUSTMENT

```
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 ADJUSTMENT TIME WITH MEAN = 70 MINUTES, 5 MIN SD
$
SOURCE = 1  CREATE TVS,, SAMPLE(1),, TIME TO STOP CREATING TVS = 480
   BRANCH TO, 2
$
QUEUE = 2  INSPECT WAIT
ACTIVITY = 3  INSPECT TVS, PROBABILISTIC BRANCHING AND ACT TIME = SAMPLE(2)
   SELECT,,, 1 THE INSPECTOR
   BRANCH TO, 6 WITH PROBABILITY = .85
   BRANCH TO, 4 WITH PROBABILITY = .15
$
QUEUE = 4  ADJUST WAIT
ACTIVITY = 5 ADJUST TVS,, SAMPLE(3)
   SELECT,,, 2 THE ADJUSTOR
   BRANCH TO, 2 FOR REINSPECTION
$
SINK = 6  GOOD TVS LEAVE
FINISH
```

created from such simple, basic processes by the careful assembling of 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 clients or decision makers in an effort to encourage their involvement in the modeling process.

The network must be translated into a form suitable for the computer. Nodes and branches along with the glossaries are expressed by INS statements called the statement model, given in Figure 2. INS statements contain fields separated by commas or equal signs. A dollar sign ($) precedes comments on a line. Comments are permitted within a field after the field's information has been specified. The use of comments throughout not only documents and clarifies the statements but also makes the input readable.

The SIMULATION statement introduces the simulation problem and supplies identifying information. The glossaries define three statistical DISTRIBUTIONs and two RESOURCEs. INS has fourteen built-in statistical distributions including those constructed from user-supplied data and time-dependent processes. Resources and nodes may be given names in the second field. The node and branch statements are written as they appear in the network with the BRANCH statement following its node. The third field of the SOURCE and ACTIVITY statements is used to control the method of branching. TVs branch PRObabilistically from Activity 3; at other nodes, TVs go directly to the indicated node. The SELECT statement tells INS which resource services transactions at the activity. Multiple commas in a statement are used to skip unused fields for which INS automatically provides defaults.

## EXAMPLE 2: FLEXIBLE PRODUCTION DEPARTMENT

A department has three machine groups --drills, mills, and heat treatment -- which are used in a job shop fashion. 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.

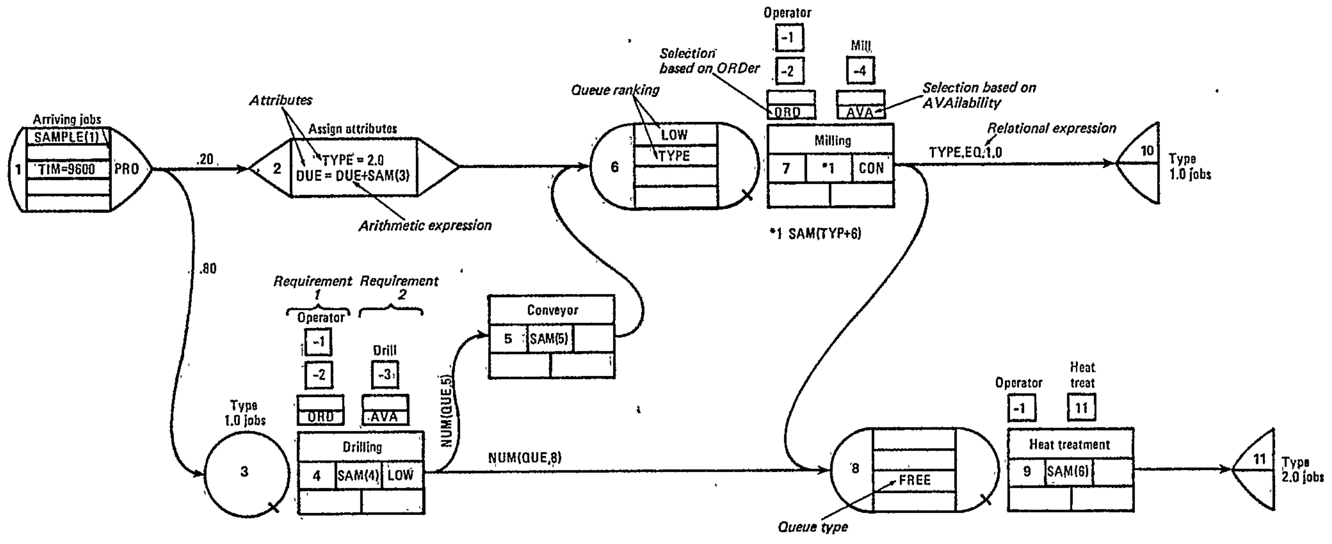
A model of the department would use transactions to represent the jobs and resources to represent the operators and the machines. The INS network model is given in Figure 3. Activities 4, 7, and 9 represent the drilling, milling, and heat treatment activities respectively. Jobs are created at Source 1 until the simulation TIMe is 9600 minutes (20 eight-hour days). Each transaction is assigned a TYPe and a DUE date by associating these values with 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 a DUE date set to the CURrent simulation TIMe plus a SAMple from Distribution 2. The specification of CUR(TIM) is just one of several System-Defined Functions (SDFs) available in INS which may supply information about the network known by INS. Other uses of SDFs include the use of SAMples from distributions.

Although 80 percent of the transactions move in to 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 any 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 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

Figure 3
FLEXIBLE PRODUCTION DEPARTMENT NETWORK



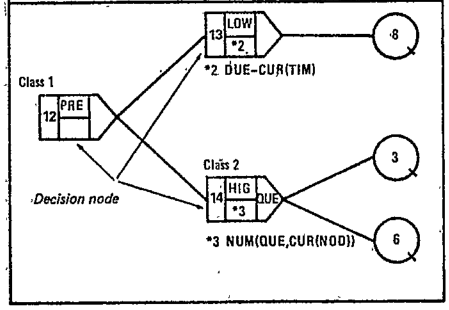
RESOURCES

| 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) |

DECISION TREES



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 |

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.

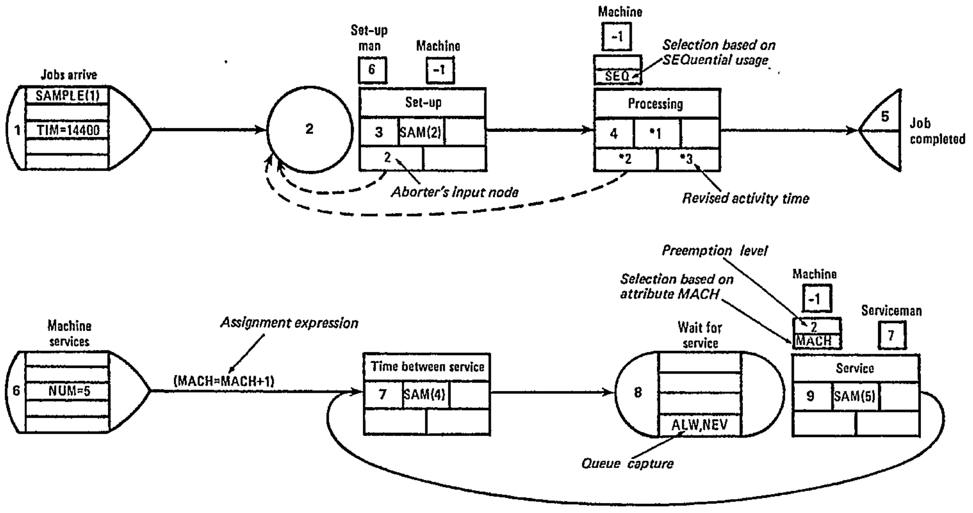
However, when an operator finishes an activity, its next service must be determined. This decision is accomplished by reference to the primary service nodes of the resources. Each of the machine groups service single queues but the operators serve one or more. Their choices are represented by one (or 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 a 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, INS easily accommodates some of the most complex aspects of modeling job shops. 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. The specifications clearly document the specific actions as well as depict the general system under investigation. Furthermore, the modeler did no programming.

## EXAMPLE 3: MACHINE ADJUSTMENT

A department has five semi-automatic machines which are operated almost continuously. A set-up man loads the machine, sets the tooling, and starts the machine. It runs automatically, discharging its product into tote boxes which are taken away. The department has a serviceman who must periodically service the machine. When service is needed, the machine may be stopped, if it is running, and taken out of production for service. If the machine is in the process of being

Figure 4
MACHINE ADJUSTMENT NETWORK



RESOURCES

| Number | Description | Type | Primary Service Node |
|---|---|---|---|
| 1.TO.5 | Machines | -1 | 2 |
| 6 | Set-up man | | 2 |
| 7 | Serviceman | | 9 |

ATTRIBUTES

| Label | Description | Scope | Initial Value |
|---|---|---|---|
| MACH | Machine number | IND | 0 |
| ACT | Prior unsatisfied activity time | IND | 0 |

DISTRIBUTIONS

| Number | Description |
|---|---|
| 1 | Interarrival time of jobs |
| 2 | Set-up time |
| 3 | Processing time |
| 4 | Time between services |
| 5 | Service time |

EXPRESSIONS

| Number | Description |
|---|---|
| *1 | .IF.ACT.GT.0 |
| | .THEN.ACT |
| | .ELSE.SAM(3) |
| *2 | .IF.NUM(IDL,-1).GT.0 |
| | .THEN.2+(ACT=TRA(REM))*0 |
| | .ELSE.0 |
| *3 | TRA(REM)+5.0 |

*.IF.–.THEN.–.ELSE. expression*

"set-up," then the job to be processed is made available to be set up on another machine. If the machine is in production, then the job will wait until the machine is serviced if no other machines are available to take over the job's production. If another machine is available, it will assume the remainder of the job's production.

The network in Figure 4 describes our modeling approach. There are two disjoint network segments: one which depicts the flow of jobs and their processing and the second which depicts the machine "services." The connection between the network is that they have resources in common, the five machines. A set-up man and serviceman are also resources.

The arrival of jobs at the department is generated by Source 1. A job enters Queue 2 and competes for the set-up man and a machine. When both resources are available, the job is set-up. The new specification in Activity 3 is the aborter's input node, which designates what happens to the job if the machine is preempted for service. In this case, the job is simply returned to Queue 2 to compete again for resources. Activity 4 represents the job processing. It uses the machine SEQuentially from the set-up activity. No queue is needed for Activity 4 since its resource will always be available. The activity time is given by expression *1. In this case, we footnoted all expressions and placed them in a network glossary. Expression *1 states in an .IF-.THEN.-.ELSE. expression that if this job had any prior unsatisfied ACTivity time, then that time should be the activity time. Otherwise, the activity time should be SAMpled from Distribution 3. Prior ACTivity time is an INDividual transaction attribute that is given a non-zero value if the job was being processed but interrupted because the machine was serviced. Expression *2, the aborter's input node, determines if and when the job aborts. In this case, the job aborts Activity 4 only if another machine is idle (i.e., NUM(IDL, -1).GT.0). It will abort to Queue 2 and assign its ACT attribute the value of the TRAnsaction's REMaining time at Activity 4 (the assignment is multiplied by zero to delete the effect). If no other machine is available, the input node has a value of zero, meaning the job does not abort but simply waits at Activity 4 until the machine is serviced. When the machine is serviced, it will return to resume its processing of the interrupted job and use the revised activity time, expression *3, as the activity time. Expression *3 specifies the revised activity to be the TRAnsaction's REMaining time plus 5.0 minutes to account for some restart requirements. When the job completes, it exits at Sink 5. Thus the "dashed" lines in the job processing network represent the possible flow of aborting transactions.

The machine services are created at Source 6. One is created for each machine (the source node limits the NUMber created to 5). When they exit Source 6, they have their MACHine number attribute incremented by one in the assignment expression. This assignment identifies the machine each service request represents. Activity 7 is the delay between services, obtained by SAMples from Distribution 4. A service request leaving Activity 7 enters Queue 8 to obtain service from the machine and serviceman. The specific machine is selected based on the MACHine attribute. The preemption level of 2 for this resource requirement means that it can preempt any other transaction having the needed resource where the preemption level is lower. Since, by default, all other requirements have a level of 1, this machine will be obtained immediately. Because the serviceman may not be immediately available since he may be busy servicing another machine, the preempted machine is ALWays captured in Queue 8 along with the service request. As soon as the repairman is free, service at Activity 9 can be started.

Thus important realistic complications like resource preemption along with its attendant impact on other resources and transactions can be specified directly in the network. More powerful specification expressions generalize several network concepts by making actions conditional on what is happening in the network at the time. Transactions can select resources in many ways and capture the resources at a queue or activity. Although such embellishments greatly extend the modeling potential, they are readily incorporated into the network without sacrificing its visual clarity.

## EXAMPLE 4: A BUS STOP

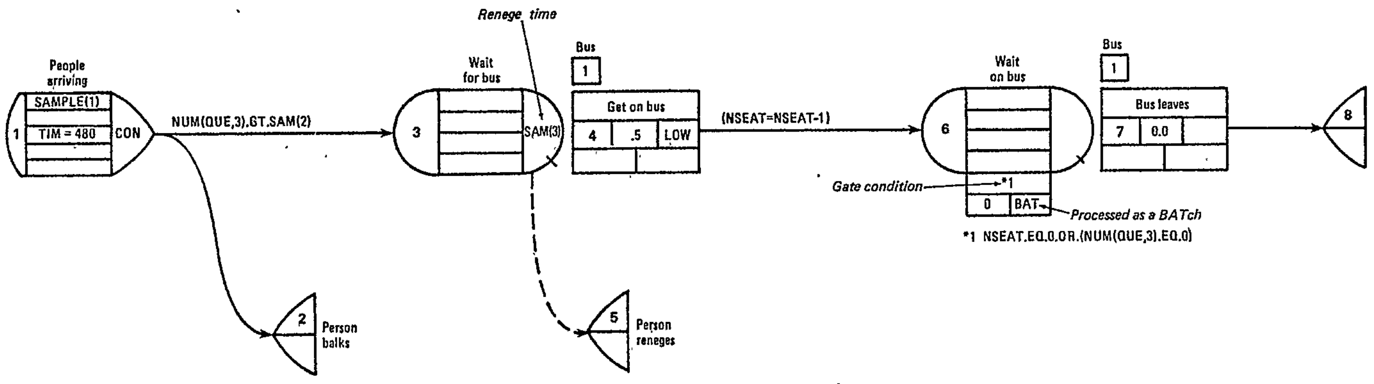
People arrive to a bus stop to board a bus which arrives periodically. Some of the arrivals will see the number waiting and choose not to wait (i.e., they balk) while others may wait for a while and leave (i.e., they renege). The number of available seats for each arriving bus varies.

The INS network in Figure 5 might be used to model the system. The arriving people are represented by transactions and the bus is represented by a resource. The number of seats available when the bus arrives is given by the global RUN attribute NSEAT. People arrive via Source 1. If the NUMber in QUEue 3 is greater than a SAMple from a "number to balk" distribution, then the person balks (is branched) to Sink 2. Otherwise, the person waits in Queue 3 for the bus. Time for one person to get on the bus is .5 minutes. Each person getting on the bus decrements the Number of SEats available by 1 as he exits Activity 4. This change uses an assignment expression. Queue 6 represents the time those entering the bus wait before the bus leaves. A gate condition at Queue 6 insures that the bus cannot leave until all available seats are filled (NSEAT.EQ.0) or there is no one else to get on the bus (NUM(QUE,3).EQ.0). The bus will take all transactions in Queue 6 and process them as a BATch.

However, we need to control the actions of the bus to insure that the right number of people are able to get on the bus. The decision tree is used for the bus.

Decision 9 DESignates service to Queue 3 as long as there are seats available (NSEAT.GT.0) and there are people waiting to board the bus (NUM(QUE,3).GT.0). Transactions leaving Activity 4 will remain in Queue 6 until the gate condition at Queue 6 is met. This means the bus will use its decision tree each time Activity 4 is complete. However, when the last available seat is filled

Figure 5
A BUS STOP NETWORK



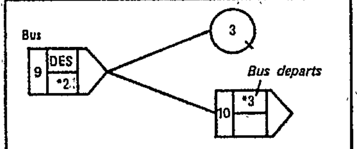
RESOURCES

| Number | Description | Primary Service Node |
|---|---|---|
| 1 | Bus | 9 |

ATTRIBUTES

| Label | Description | Scope | Initial Value |
|---|---|---|---|
| NSEAT | Number of seats | RUN | SAM(4) |

DECISION TREE



*2 .IF.NSEAT.GT.0.AND.
NUM(QUE,3).GT.0
.THEN.3
.ELSE.10

*3 SAM(5)+0•(NSEAT=SAM(4))

DISTRIBUTIONS

| Number | Description |
|---|---|
| 1 | Interarrival time |
| 2 | Number to balk |
| 3 | Renege time |
| 4 | Available seats |
| 5 | Time between bus arrivals |

(NSEAT.EQ.0) or if Queue 3 empties during loading (NUM(QUE,3).EQ.0), then the gate is satisfied and because the gather specification will be satisfied, the bus is captured to start Activity 7. The bus processes the group in the queue as a BATch, after which it will be able to re-examine its decision tree. Now Decision 10 is DESignated and the bus departs the network to represent the arrival of another bus in an amount of time SAMpled from Distribution 5. During the evaluation of the time the bus departs, NSEAT is given a new value SAMpled from Distribution 4.

Thus more complicated behavior like reneging and balking are added without complex changes to the network. Furthermore a broader definition of waiting at a queue now includes waiting for a gate condition or some transaction synchronization in the network. Added to the concept of servicing a transaction at an activity is the ability to process groups of transactions.

## INS OUTPUT

The most elegant simulation model would be useless if it did not generate any information about the behavior of the sustem. INS automatically provides a comprehensive set of outputs of three basic types. Compilation output consists of a listing of the INS statements as in Figure 2 and an echo of the model which includes the values for all specified and defaulted fields. This is useful in debugging the model and model verification. Execution output is controlled by the modeler. It may include an English-like trace (Figure 6) which is turned on and off by the SDFs START(TRACE) and STOP(TRACE), respectively. The modeler may also use SDFs to request any of the forms of summary output during execution.

Summary output contains the statistical information gathered by INS during execution. The Summary Report is automatically printed after all runs have been made and consists of three reports. Figure 7 contains an abbreviated report for the TV inspection problem. The Network Status Report describes the current state of the network. The Node Statistics describe: the number of transactions and the 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. All this information is generated without any specific instruction from the modeler.

The modeler, however, can request additional statistics with a TABLE statement. A unique feature is that it employs a nonprocedural format. The desired statistic is described on the TABLE statement by a phrase such as TIME IN ACTIVITY = 3 or UTILIZATION OF RESOURCES = 1,2,3. In addition, the statistic can be broken down to gain more detailed information. For example, Queue 6 in Example 2 which contains both types of jobs could have the following table:

```
TABLE = 1  TIME WAITING, (7)INCLUDING ZERO WAITS IN QUEUE, 6
   BREAKDOWN TABLE, 2,,TYPE, 2, 1=TYPE 1,
                                2=TYPE 2
```

This will produce waiting time statistics (including zero waits) for Queue 6 broken down by job TYPe. Tables can also specify histograms. For example, the TV inspection queue generates the histogram in Figure 8 by using this statement:

```
TABLE = 1  NO. IN INSPECTION QUEUE,
      20 CELLS STARTING AT = 1 TV WITH CELL WIDTH = 1 TV,, NUMBER IN QUEUE=2
```

Tables can be specified on any statistic collected by INS as well as on attributes. Furthermore, statistical SDFs can obtain the MEAN OF (TABLE,1),

Figure 6
TRACING BEHAVIOR IN THE TV MODEL

```
  TIME    ENTITY       NUMBER    ACTION            NODE TYPE      NUMBER        RELATED INFORMATION

 0.000 INSPECTOR          1    HAS ARRIVED
                               IS IDLE

 0.000 ADJUSTOR           2    HAS ARRIVED
                               IS IDLE

 0.000 TRANSACTION        1    CREATED      AT    CREATE TVS        1
                               ENTERING           INSPECT WAIT      2  NUM(QUE,*) =       0
                               DEPARTING          INSPECT WAIT      2  TIME IN QUEUE =        0.000
       INSPECTOR          1    BUSY         AT    INSPECT TVS       3  ON TRANSACTION         1
       TRANSACTION        1    BEGINNING    AT    INSPECT TVS       3

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

20.008 TRANSACTION        2    CREATED      AT    CREATE TVS        1
                               ENTERING           INSPECT WAIT      2  NUM(QUE,*) =       0
                               DEPARTING          INSPECT WAIT      2  TIME IN QUEUE =        0.000
       INSPECTOR          1    BUSY         AT    INSPECT TVS       3  ON TRANSACTION         2
       TRANSACTION        2    BEGINNING    AT    INSPECT TVS       3

28.087 TRANSACTION        2    COMPLETED    AT    INSPECT TVS       3  ACTIVITY TIME =        8.079
       INSPECTOR          1    IN BUFFER STORAGE
       TRANSACTION        2    DESTROYED    AT    GOOD TVS LEA      6  CUR(RTC) =         0
       INSPECTOR          1    IS IDLE

36.199 TRANSACTION        3    CREATED      AT    CREATE TVS        1
                               ENTERING           INSPECT WAIT      2  NUM(QUE,*) =       0
                               DEPARTING          INSPECT WAIT      2  TIME IN QUEUE =        0.000
       INSPECTOR          1    BUSY         AT    INSPECT TVS       3  ON TRANSACTION         3
       TRANSACTION        3    BEGINNING    AT    INSPECT TVS       3

42.777 TRANSACTION        4    CREATED      AT    CREATE TVS        1
                               ENTERING           INSPECT WAIT      2  NUM(QUE,*) =       0
```

Figure 7
SUMMARY REPORT FOR THE TV MODEL

```
                                      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   NUMBER IN NODE   ZERO WAIT COUNT
                                                    COU(*)                          ZER(*)

        QUEUES        INSPECT WAIT       2          449             0               122
                      ADJUST WAIT        4           67             0                23

        ACTIVITIES    INSPECT TVS        3          449             0
                      ADJUST TVS         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

                 ----------- OBSERVATIONS OF THE MEANS ----------- -- ALL OBSERVATIONS --  CURRENT RUN OR BATCH ONLY
 NODE                       TIME OF     STANDARD     STANDARD                                             TIME OF
NUMBER    NAME      MEAN   OBSERVATION  DEVIATION     ERROR       SMALLEST     LARGEST      MEAN        OBSERVATION
                 MEA(NUM,*) OBS(NUM,*)  STD(NUM,*)  STE(NUM,*)   SMA(NUM,*) LAR(NUM,*)  CME(NUM,*)     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
                         *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   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    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)                     MEAN       NUMBER OF      STANDARD       STANDARD
 OR TYPE        NAME       UTILIZATION  OBSERVATIONS    DEVIATION        ERROR        SMALLEST       LARGEST
                           MEA(UTI,*)    OBS(UTI,*)     STD(UTI,*)    STE(UTI,*)     SMA(UTI,*)    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 8
ADDITIONAL STATISTICS PROVIDED BY TABLES AND HISTOGRAMS

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

                                          ** TABLE NUMBER    1 **

                                          NUMBER IN INSPECT QUEUE

                   ------------ OBSERVATIONS OF THE MEANS ------------ --- ALL OBSERVATIONS ---    CURRENT RUN OR BATCH ONLY
                                  TIME OF      STANDARD     STANDARD                                            TIME OF
                     MEAN       OBSERVATION    DEVIATION      ERROR       SMALLEST     LARGEST        MEAN       OBSERVATION
                   MEA(TAB,*)   OBS(TAB,*)    STD(TAB,*)   STE(TAB,*)   SMA(TAB,*)  LAR(TAB,*)    CME(TAB,*)    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       RELATIVE      CUMULATIVE       ---CELL LIMIT---
 OBSERVATION    FREQUENCY     FREQUENCY       LOWER           UPPER
                REL(*,*)      CUM(*,*)                                        0.0       0.2       0.4       0.6       0.8       1.0
                                                                              +    +    +    +    +    +    +    +    +    +    +
.000000E+00     0.000         0.000         -INFINITY       .000000E+00       +                                                 +
3860.12         0.565         0.565         0.000000E+00    1.00000           +***************************C                     +
1076.19         0.158         0.723         1.00000         2.00000           +*********                          C             +
678.177         0.099         0.822         2.00000         3.00000           +*****                                   C        +
453.342         0.066         0.889         3.00000         4.00000           +***                                        C     +
217.096         0.032         0.920         4.00000         5.00000           +**                                           C   +
100.134         0.015         0.935         5.00000         6.00000           +**                                            C  +
76.8508         0.011         0.946         6.00000         7.00000           +*                                             C  +
14.9505         0.002         0.948         7.00000         8.00000           +                                              C  +
20.8436         0.003         0.951         8.00000         9.00000           +                                               C +
53.0242         0.008         0.959         9.00000         10.0000           +                                               C +
65.4680         0.010         0.969         10.0000         11.0000           +                                               C +
58.7343         0.009         0.977         11.0000         12.0000           +                                                C+
28.5891         0.004         0.982         12.0000         13.0000           +                                                C+
67.7557         0.010         0.992         13.0000         14.0000           +                                                 C
31.2228         0.005         0.996         14.0000         15.0000           +                                                 C
23.8638         0.003         1.000         15.0000         16.0000           +                                                 C
2.78857         0.000         1.000         16.0000         17.0000           +                                                 C
-----------                                                                   +    +    +    +    +    +    +    +    +    +    +
6829.15                                               LOWER LIMIT INCLUSIVE   0.0       0.2       0.4       0.6       0.8       1.0
```

for example. Tables allow the modeler to collect, without procedural instruction, any desirable statistic and apply advanced techniques of output analysis. INS automatically handles the tasks of collecting, compiling, and displaying statistics.

## FORTRAN INTERFACE

To allow conversation with other software, a convenient interface with FORTRAN is incorporated which allows two-way communication of information. Using the interface, INS 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 to reduce execution time. SDFs and attributes may be used in the FORTRAN 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.

## STATISTICAL ANALYSIS

In addition to providing a high level approach to simulation modeling, INS 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 INS estimate variances by replications (separate simulation runs) or by batches (division of observations into subintervals). INS does not automatically provide a standard error from observations collected during a run because of their known lack of independence. This is why the output described in the previous section uses one observation per run to compute variances. Similar computations could have been obtained using batches when dealing with a steady-state simulation. Tables can be used for more complex statistics collection.

Start-up issues requiring special initial conditions can be specified directly within INS by a PRERUN statement 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 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 variables are available in INS. By default each distribution in the model has its own separate random number stream. There is no practical limit to the number of streams in INS. 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, INS employs inverse transform variate generators so that random numbers are inherently synchronized. By making these features a part of INS, statistical analysis can become an integral part of simulation modeling.

Special attention has been given to random number and random variate generators in INS. The random number generator (Marse, Roberts, 1983) is completely portable to any machine having a 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 INS 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, Roberts, 1983). More discussion of statistical issues in INS is found in (Roberts, Klein, 1982).

## CONCLUSIONS

INS provides an easy-to-use simulation capability that contains powerful modeling concepts that do not rely on general programming capabilities. Such an approach makes simulation modeling available to more people and extends the scope of simulation applications. Built-in procedures for statistics collection and automatic output generation mean that results are obtained easily and quickly.

INS is easy to learn because of its small symbol set and because INS networks closely correspond to the systems being simulated. Sophisticated INS models retain the simple modeling structure so that advanced concepts strengthen the fundamentals without expanding the network structure. Due to the ability to progress from simple to advanced concepts, INS has been taught at both the undergraduate and graduate level with equal success. The textbook (Roberts, 1983) details simulation modeling and analysis with INSIGHT. Many applications are described and it contains sections on simulation issues such as input modeling, random number generation, event processing, etc.

INS is written in standard (1966) FORTRAN and its portability has been demonstrated by running on many computers. INS is presently being used by a variety of organizations both nationally and internationally.

## REFERENCES

INSIGHT User's Manual, Regenstrief Institute for Health Care, Indianapolis, Indiana, 1983.

Klein RW, Roberts SD. A time-varying Poisson process generator. In review, 1983

Marse KJ, Roberts SD. Implementing a portable FORTRAN Uniform(0,1) generator. In press, Simulation.

Roberts SD. Simulation Modeling and Analaysis with INSIGHT. Regenstrief Institute for Health Care, Indianapolis, Indiana, 1983.

Roberts SD, Klein RW. Statistics collection, display, and analysis in INSIGHT. Joint National ORSA/TIMS Meeting, 1982.