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

INTRODUCTION TO SIMAN

David T. Sturrock
C. Dennis Pegden
Systems Modeling Corporation
The Park Building
504 Beaver Street
Sewickley, PA 15143

ABSTRACT

This paper discusses the concepts and methods for simulating manufacturing systems using the SIMAN simulation language. SIMAN is a general purpose simulation language which incorporates special purpose features to greatly simplify and enhance the modeling of material handling components in a manufacturing system.

1. INTRODUCTION

This paper discusses the use of SIMAN simulation language for modeling manufacturing systems (Pegden 1982). SIMAN is a general-purpose SIMulation ANalysis program for modeling combined discrete-continuous systems. The modeling framework of SIMAN allows component models based on three distinct modeling orientations to be combined in a single system model. For discrete change systems either a process or event orientation can be used to describe the model. Continuous change systems are modeled with algebraic, difference, or differential equations. A combination of these orientations can be used to model combined discrete-continuous models.
SIMAN incorporates a number of important features which have contributed to its rapid growth in popularity. Some of the significant features include the following:

1. SIMAN is designed around a logical modeling framework in which the simulation program is decomposed into a model frame and an experiment frame.

2. SIMAN incorporates a number of unique and powerful general-purpose modeling constructs which represent a natural evolution and refinement of existing language designs.

3. SIMAN imbeds within this general-purpose framework a set of special-purpose constructs which are specifically designed to simplify and enhance the modeling of manufacturing systems. Many general-purpose languages lack the special-purpose manufacturing features provided by SIMAN. On the other hand, existing special-purpose manufacturing languages XCELL+ (Conway et al. 1987), and SIMFACTORY (CACI Products Company, 1986) are intended for a restricted class of manufacturing systems and are not applicable to systems in general.

4. SIMAN runs on mainframe, mini, and microcomputers. All versions, including the microcomputer versions, are completely compatible. Models can be moved between computer systems without modification.

5. SIMAN includes an interactive graphics capability for both building models and experiments, and displaying the outputs from the model.

6. SIMAN incorporates an interactive debugger which allows you to interactively monitor and control the execution of the simulation. Errors can be isolated and corrected during the execution of the simulation without the need to recompile, relink, and rerun the simulation.

7. SIMAN models can be used within the CINEMA system (Systems Modeling 1985) to generate real-time, high-resolution color graphics animation of the system dynamics. This provides an extremely powerful new tool for both understanding and explaining the dynamics of a system.

8. SIMAN's modular structure and powerful modeling capabilities encourage integration with other analysis technologies and interfaces such as SimStarter (Suri and Tomsicek 1988), providing sets of compatible tools for rapid modeling and analysis.

SimStarter provides the user with the ability to easily convert a Manuplan II analytical model to a SIMAN simulation model. Thus, a user may use Manuplan II to quickly create an analytical model of a system, then use the SIMAN model created by SimStarter to perform detailed simulation analysis, evaluating the effects of such system characteristics as buffer size limits and specific scheduling policies which cannot be analyzed using analytical tools. This seamless integration of tools allows the use of the "right tool at the right time" without duplication of effort as the modeling process proceeds from analytical model to detailed simulation.

Since its introduction more than seven years ago, SIMAN has been continually enhanced with new releases. In 1989, however, a totally new version of SIMAN has been completed. This new version is called SIMAN IV Release 1.0 and represents a complete re-write of the SIMAN language. SIMAN IV shares no code with its previous version. This massive development effort was undertaken to significantly improve the modeling features of the language and to enhance the internal design and maintainability of the SIMAN code.

SIMAN IV incorporates many new constructs which greatly enhance the modeling power of the language. In addition, many of the restrictions on operands and syntax have been removed. Despite the many additions and enhancements, SIMAN IV remains compatable with earlier versions of the language.

In this paper, we will describe both the general-purpose and special-purpose manufacturing features of SIMAN IV. The emphasis, however, will be on the manufacturing features.

2. GENERAL-PURPOSE MODELING FEATURES OF SIMAN

The SIMAN modeling framework is based on the systems theoretic concepts developed by Zeigler (1976). Within this framework, a fundamental distinction is stressed between the system model and the experimental frame. The system model defines the static and dynamic characteristics of the system. The experimental frame defines the experimental conditions under which the model is run to generate specific output data. For a given model, there can be many experimental frames resulting in many sets of output data. By separating the model structure and the experimental frame into two distinct elements, different simulation experiments can be performed by changing only the experimental frame. The system model remains the same.

Given the system model and the experimental frames, the SIMAN simulation program generates output files which record the model state transitions as they occur in simulated time. The data in the output files can then be subjected to various data analyses such as data truncation and compression, and the formatting and display of histograms, plots, tables, etc. Within the SIMAN framework, the data analysis and display function follow the development and running of the simulation program and are completely distinct from it. One output file can be subjected to many different data treatments without re-executing the simulation program. Data treatments can also be applied to sets when performing an analysis based on multiple runs of a model or when comparing the response of two or more systems.

Although SIMAN permits component models to be developed using a process, event, or continuous orientation, we will focus our attention in this paper on the process orientation in which models are constructed as block diagrams. This orientation is the one best suited for modeling most manufacturing systems. These diagrams are linear top-down flowgraphs which depict the flow of entities through the system. The block diagram symbol shapes indicate their function. The sequencing of blocks is depicted by arrows which control the flow of entities from block to block through the entire diagram.

These entities are used to represent "things" such as workpieces, information, people, etc., which flow through the real system. Each entity may be individualized by assigning attributes to describe or characterize it. For example, an entity representing a workpiece might have attributes named DueDate and ProcessingTime corresponding to due date and processing time for the workpiece. As the entities flow from block to block, they may be delayed, disposed, combined with other entities, etc., as determined by the function of each block.

There are ten different basic block types in SIMAN. These include the OPERATION, HOLD, TRANSFER, QUEUE, STATION, BRANCH,

PICKQ, QPICK, SELECT, and MATCH blocks. The OPERATION, HOLD, and TRANSFER blocks are further subdivided into several different block functions depending upon their operation type, hold type or transfer type. These types are specified as the first operand of the block, and consist of a verb which is descriptive of the specific function which the block is to perform. For example, the operation type ASSIGN specifies that the block is to assign a value to an attribute or variable; and the operation type DELAY specifies that the block is to delay entities.

Each block function of SIMAN is referenced by a block function name. In the use of the QUEUE, STATION, BRANCH, PICKQ, QPICK, SELECT and MATCH blocks, each block type performs only one function and the block function name is the same as the basic block name. However, in the case of the OPERATION, HOLD and TRANSFER blocks, each basic block type performs several different functions. The block function name for each of these blocks is the operation type, hold type or transfer type specified as the first operand of the block. We will frequently refer to a block by its function name--for example, we will refer to the OPERATION block with the DELAY operation type as the DELAY block.

All of the basic block types, including the OPERATION, HOLD, and TRANSFER types, have operands which control the function of the block. For example, the CREATE block has operands which prescribe the time between batch arrivals, the number of entities per batch arrivals, the number of entities per batch, and the maximum number of batches to create.

Blocks may optionally be assigned a block label, one or more block modifiers, and a comment line. A block label is appended to the left side of a block and can be of any length. A block label is used for branching or referencing from other blocks. Block modifiers are special symbols appended to the right or bottom of a block and either modify or extend the standard function to be performed by the block. The comment line, if specified, is entered to the right of the block and serves to document the model.

A block diagram model can be defined in either of two equivalent forms referred to as the diagram model and the statement model. The diagram model is a graphic representation of the system using the ten basic block symbols. The statement model is a transcription of the diagram model into statement form for input to the model processor. There is a one-to-one correspondence between blocks in the diagram model and input statements in the statement model.

The modeler normally proceeds by first constructing the Block Diagram which is then transcribed into the equivalent statement form for input to the SIMAN language. An interactive preprocessor named BLOCKS allows the modeler to graphically build the block diagram model directly on a graphics terminal. The BLOCKS program then automatically generates the corresponding statement version of the model for input to the SIMAN language. A preprocessor named ELEMENTS provides a similar method for defining the experiment frame.

To illustrate the general purpose modeling approach of SIMAN, consider the simple manufacturing system in which workpieces arrive, are processed in order on a single machine, and then, depart the system.

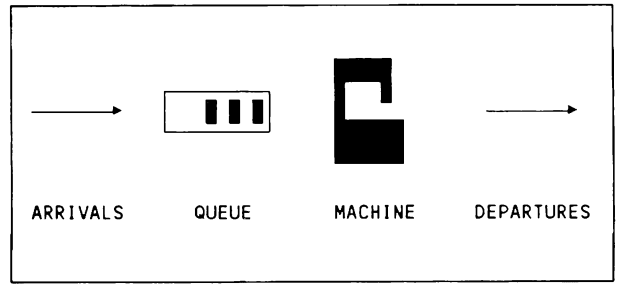

Figure 1: Schematic of a Simple System

We will assume that the parts arrive in batches of size ten, with the time between arrivals exponentially distributed with a mean of twenty. The processing time on the machine is assumed to be uniformly distributed between one and two.

The block diagram model for this example is shown in Figure 2. The workpieces enter the system at the CREATE block. The operands for this block specify that the workpieces enter in batches of 10 and that the interarrival time between batches is exponentially distributed with a mean of 20. The workpieces continue to the QUEUE block named Buffer where they wait in turn to seize a unit of the

resource Machine. Once a workpiece seizes the Machine, it enters the DELAY block where it is delayed by the processing time which is specified as a sample from a uniform distribution between 1 and 2. Following this delay, the workpiece releases the resource Machine which allows it to be reallocated to workpieces waiting in the QUEUE block. The symbol attached to the bottom of the RELEASE block is called the DISPOSE modifier and models the departure of the workpiece from the system.

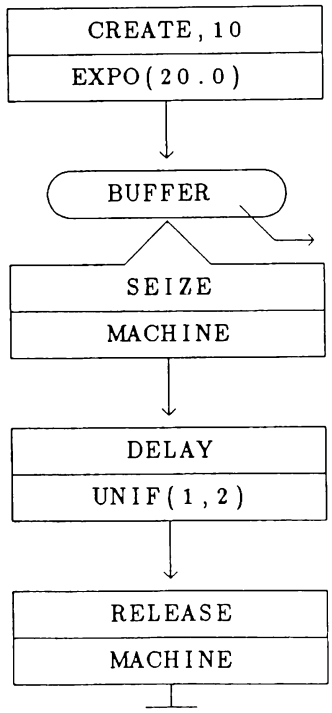

Figure 2: Block Diagram for Simple System

An example experimental frame for this model is shown in Figure 3. The experimental frame specifies the experimental conditions associated with the model and includes the definition and capacity of the resources employed, a specification of the number and length of each simulation replication, etc. Note that in the Resources element, the capacity of the resource named Machine is set to 1. We can add a second machine to our system by simply increasing this resource capacity to 2.

```
Begin;
Project,Simple System,Pegden;
Queues: Buffer;
Resources: Machine,1;
Replicate, 1, 0, 480;
End;
```

Figure 3: Experiment for Simple System

3. CHARACTERISTICS OF MANUFACTURING SYSTEMS

Manufacturing systems exhibit a number of unique characteristics which make them awkward to model within the framework of a strictly general purpose simulation language. These characteristics are present in both the classical jobshop system as well as the modern automated version of the jobshop referred to as an FMS. The most significant of these characteristics include the following:

a) Large manufacturing systems typically consist of a number of different workcenters or cells. A natural way to model such systems is to decompose the large system into its workcenters, modeling each workcenter separately, and then combine the workcenter models into an overall system model. The general purpose simulation languages typically do not provide a logical format for modeling separate workcenters.

b) Often several workcenters within a manufacturing system are functionally equivalent. As a result, it is possible to develop a single functional description which can be used to model all the similar workcenters within the system. General purpose simulation languages generally do not provide any features to exploit this property.

c) The workpieces which move through a manufacturing system typically each have a unique process plan. This means that each entity in the model must have its own routing sequence through the workcenters as well as its own setup, processing time, tool requirement, etc., within the workcenter. Since the general purpose simulation languages do not provide any features for this, a considerable

amount of effort can be consumed in incorporating logic within the model for maintaining process plans on each workpiece and controlling the flow from workcenter to workcenter.

d) In most manufacturing systems, there is an uneven distribution of workload between work-centers. One way that this varying workload is accommodated is through the use of different operating schedules for the different workcenters based upon their workloads. Hence, it is desirable to be able to assign each workcenter an operating schedule which it will follow during the simulation. Again, this is often awkward to do with general purpose simulation languages.

e) An essential element of most manufacturing systems is the material handling component. This includes devices such as robots, AGV's, conveyors, power and free monorail systems, etc. These types of devices can be extremely difficult to model with many general-purpose simulation languages.

It should be noted that although in theory a general-purpose simulation language can, given enough effort, accurately model these characteristics of manufacturing systems, the modeling effort involved can be enormous. This is particularly true when the system of interest contains a major material handling component.

4. MANUFACTURING MODELING FEATURES OF SIMAN

In this section, we will describe the special features included in the SIMAN simulation language for modeling the special characteristics of manufacturing systems.

4.1 Modeling Workstations

In manufacturing systems, it is frequently desirable to model distinct workcenters within the system. This can be done within SIMAN by employing the STATION block which defines the beginning of a station submodel. An entity is entered into the station submodel by sending the entity to the STATION block using a TRANSFER block. The TRANSFER block is used to represent entity movements between station submodels.

Each station submodel is referenced by a station name (e.g. Inspection) and a station number which corresponds to a physical location within the system. The station name and number can be used interchangable to refer to the station. The station name or number is an operand of both the STATION block and the TRANSFER block.

When an entity enters a STATION block, the entity's station attribute, M, is set by SIMAN to the station of the STATION block. The entity carries this special attribute with it as it proceeds through the sequence of blocks which comprises the station submodel. The entity remains within the station submodel until it is disposed, or it is sent to a new station submodel via a TRANSFER block. The block sequence within a station submodel defines the processes through which the entities flow. The processes normally involve the queueing of entities due to limited resources such as operators, tools, etc.

To illustrate the concept of a workcenter submodel, consider the block diagram submodel shown in Figure 4. This block diagram contains the frequently occurring sequence QUEUE-SEIZE-DELAY-RELEASE which can be used to model both a single-server and a multi-server queueing system, depending on the capacity for the resource which is specified in the experimental frame.

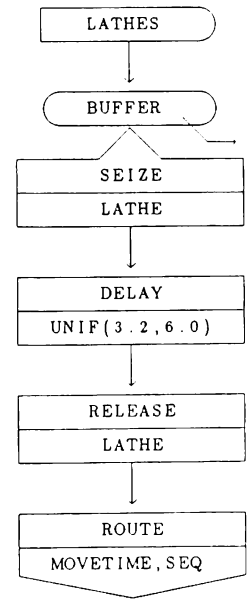

Figure 4: Block Diagram of a Workcenter

The workpieces arriving to this submodel enter the STATION block named Lathes, proceed through the QUEUE-SEIZE-DELAY-RELEASE blocks and then enter the ROUTE block. The ROUTE block is a TRANSFER block which routes the workpieces to their next workcenter.

The block sequence in this example is particularly simple and employs only a small subset of the features of SIMAN. Once the analyst become familiar with the wide range of block functions included in SIMAN, complex workcenters can be modeled with similar ease.

4.2 Macro Submodels

One particularly useful feature for modeling workcenters in SIMAN is the macro submodel. This powerful feature permits the development of a single macro submodel to represent a set of two or more similar yet distinct workcenters. For example, a typical jobshop consists of several different workcenters (lathes, planers, etc.) which are functionally equivalent, and differ only in their number and type of machines, buffer sizes, etc. We can model such a jobshop by constructing a single macro sub-model which represents the process encountered by a job at a general jobshop workcenter. This single macro submodel can then be used to model a jobshop of arbitrary size.

The beginning of a macro submodel is defined by a STATION block. The range of stations represented by the macro submodel is specified as the operand of the block. An entity is entered into the macro submodel by sending it to the STATION block using a TRANSFER block. All entities sent to a station in the specified range of the STATION block are processed as arrivals to the block. Upon entering the STATION block, the macro station attribute M of the entity is set by SIMAN to the station to which the entity was sent.

When a macro submodel is employed, the station attribute is typically incorporated in the operands of one or more of the blocks which follow the STATION block. In this way, the operation of the macro submodel can depend upon the station of the entity. For example, the station attribute could be used to specify a queue number or the entity could be branched within the sub-model based on its current station.

4.3 Visitation Sequences

As illustrated in the previous example, a workpiece is sent to its next workcenter using a TRANSFER block. However, the TRANSFER block must have some way to determine which workcenter is next in sequence for a particular workpiece. In addition, it may be necessary to update one or more attributes of the workpiece to correspond to the processing parameters at that workcenter.

The workcenter visitation sequence and corresponding attribute update values are specified in SIMAN using the SEQUENCES element which is included as part of the experimental frame. The following SEQUENCES element defines two different visitation sequences.

```
SEQUENCES:
1, Drills, ProcessTime=EXPO(10.5) &
   Lathes, ProcessTime=UNIF(1,2) &
   Exit:
2, Planers, ProcessTime=UNIF(5,18) &
   Drills, ProcessTime=10.3 &
   Exit:
```

Sequence number 1 consists of three workcenter visits. The first visit is to the station named Drills and assigns to the first attribute of the entity a sample from an exponential distribution with a mean of 10.5. The second visit is to the station named Lathes and assigns to the first attribute of the entity a sample from a uniform distribution between 1 and 2. The last visit in this sequence is to the station named Exit. Sequence number 2 consists of visits to the stations named Planers, Drills, and Exit with assignments made to the first attribute of the entity as shown.

Each workpiece in the system has two special attributes which are used in conjunction with the SEQUENCES element to determine its next station and attribute update values at the TRANSFER block. The first special attribute is NS which specifies the number of the visitation sequence which the workpiece is to follow. The value is typically assigned to the entity when it first enters the model.

The second special attribute is IS which keeps track of the current index within the sequence. An index value of k means that the workpiece is at the kth workcenter within its visitation sequence. The index attribute is automatically updated by SIMAN whenever the entity arrives to a TRANSFER block.

Although the example discussed here is relatively simple and involves only a single attribute update, the constructs provided by the SEQUENCES element in combination with the special attributes NS and IS are very flexible. Additional attributes could be employed to specify setup times, special tool requirements, etc., which might be part of the process plan. In addition, by resetting the value of IS for a given entity within a workcenter submodel, a portion of a given sequence could be repeated or skipped. Likewise by setting the value of NS, the sequence which that workpiece follows could be changed.

4.4 Resource Schedules

As noted earlier, the workcenters within a manufacturing system often operate according to different work schedules as a result of their differing loads. Within SIMAN, this characteristic can be easily modeled through the use of the SCHEDULES element which is included in the experimental frame. The SCHEDULES element is used to define a work schedule by specifying a resource capacity over time. A resource capacity within the model can then be directed to follow a given work schedule. For example, the resources in workcenter 1 might be directed to follow schedule number 1 and the resources in workcenter number 2 might be directed to follow schedule number 2.

The following SCHEDULES element defines two different work schedules:

```
SCHEDULES:
1,  1*8, 0*16:
2,  1*EXPO(5.5), 0*UNIF(.5,1);
```

In schedule number 1, the capacity is 1 for 8 time units, then 0 for 16 time units, and then this cycle repeats. In schedule number 2, the capacity is 1 for a duration which is sampled from an exponential distribution, and then 0 for a duration which is sampled from a uniform distribution, and then this cycle repeats. Note that schedule number 2 could be used to represent an unscheduled breakdown and repair activity for a resource such as a machine.

4.5 Modeling Material Handling Systems

Within a manufacturing system, the movement of entities between workcenters is accomplished by the material handling system. This is an extremely critical function in most manufacturing systems. Apple (1977) notes that the material handling function can easily account for 50 to 70 percent of the production activity.

In simple terms, the function of material handling is the movement of material from one point to another. There is a large variety of material handling devices which have been developed to support this function. From a system modeling perspective, the devices all perform one of two basic movement functions. The first function we will call the transport function and corresponds to the intermittent movement of items, one load at a time, along a fixed or varied path. The term load unit as applied here could denote a box, a roll of material, or a pallet containing a number of items grouped together. The second basic movement function we will call the convey function and corresponds to the continuous movement of items along a fixed path.

The categorization of material handling equipment into the two basic movement functions of transport and convey provides the basis for modeling these devices in SIMAN. Special blocks and experimental elements are included in SIMAN which allow both of these basic movement functions to be modeled in a straightforward manner.

The block functions which are used to model material handling systems employ the concept of a station submodel as discussed earlier. All movement functions are made relative to station names assigned to each station submodel. A material handling system is represented in SIMAN as a component of the system model which models the utilization of material handling devices to move from one station submodel to another. The travel time for entities between stations is based on the speed of the material handling device and the spatial relationship of the origin and destination stations relative to the

device. Both of these are specified by the modeler in the experimental frame.

The generic term transporter is used in SIMAN to denote a general class of movable devices which may be allocated to entities. Each transporter device has a specific station location in the system and required time to travel from one station to another. Examples of devices which might be modeled as transporters are carts, cranes, and mechanical manipulators.

There are two different types of transporters in SIMAN IV. The first, called standard transporters, were in earlier versions of SIMAN and are unconstrained in their movement between stations. These are useful in modeling situations where transporter traffic congestion is minimal. The second, called guided transporters, are new in SIMAN IV. These are used to model situations where the transporters are constrained to moving over a defined network consisting of links and intersections. Guided transporters are particularly useful for modeling Automatic Guided Vehicles.

The characteristics of each transporter type are specified in the experimental frame and include the transporter name, capacity, system map, and the initial station position and operational status of each of the transporter units for that type. The transporter name is an arbitrary alphanumeric name assigned by the modeler to each transporter type. The transporter capacity is the number of independent movable units of that transporter type. The system map is a cross-reference to a definition of the feasible travel paths and associated travel distances between pairs of stations which each transporter unit of that type may visit. In the case of guided transporters, the system map includes a description of the network of links and intersections. This system map is specified in the experimental frame.

Transporter units are allocated to entities at HOLD blocks using the REQUEST hold type. Once an entity has been allocated a transporter unit at a REQUEST block, the entity can be transported from one station to the next using the TRANSFER block with the TRANSPORT hold type. The duration of the transport is automatically computed by SIMAN based on the distance to the station and the speed of the transporter unit. For guided transporters, the travel time may be further influenced by other traffic in the system. At the end of the transport duration, the entity enters the STATION block of the destination station submodel.

A transporter unit may be released using the OPERATION block with the FREE operation type. The FREE block changes the status of the specified transporter from busy to idle.

The generic term conveyor is used in SIMAN to denote a general class of devices which consist of positioned cells which are linked together and move in unison. Each cell represents a location on the device and can be either empty or occupied. Entities which access the conveyor must wait at the entering station until the specified number of consecutive empty cells are located at the station. The entity then enters the conveyor and the status of the cells are changed from empty to occupied. The entity remains in the cells until the conveyor is exited at the entity's destination station.

Each conveyor in SIMAN IV can be defined to be either accumulating or non-accumulating. In an accumulating conveyor, an entity which is stopped on the conveyor forms a blockage point for other entities which continue to move on the conveyor. Entities arriving to the blockage point accumulate behind the blockage. In a non-accumulating conveyor, an entity which stops on the conveyor forces the entire conveyor to stop. As a result, there is no accumulation of entities. In earlier versions of SIMAN, only non-accumulating conveyors were supported.

Each conveyor device moves along a fixed path defined by one or more segments. A segment is a section of a conveyor path which connects two station submodels. Segments can be connected to form either open or closed loop conveyor paths. A closed loop path is one in which an item on the conveyor can return to a station by continuing on the device. An open loop path is one that is not closed. The segments defining a conveyor path are specified in the experimental frame.

Conveyor cells are allocated to entities at HOLD blocks with the ACCESS hold type. Once a conveyor has been accessed, the entity can be conveyed to its destination workcenter using the TRANSFER block with the CONVEY transfer type. A conveyor may

be stopped and started using the STOP and START blocks, respectively.

5. SUMMARY OF NEW SIMAN IV FEATURES

As noted earlier, SIMAN IV is a completely new implementation of the SIMAN language. This effort was undertaken in order to incorporate improvements in internal design and to make extensive additions to the modeling constructs supported by the language. In this section we summarize some of the more important new features in the SIMAN IV language.

1. The transporter constructs have been extended to make it much easier to model fixed-path devices such as Automatic Guided Vehicles. These constructs involve extensions to existing blocks and elements as well as several new blocks and elements. The language automatically handles the logic associated with traffic congestion along the path.

2. The conveyor constructs have been extended to make it straightforward to model a much wider range of conveyor devices. These constructs have been incorporated as extensions to the current conveyor blocks and elements. Using these new features, both accumulating and non-accumulating conveyors can be easily modeled.

3. Users can now define their own variables and attributes which are referenced by user-defined symbolic names. One and two - dimensional user-defined arrays for both variables and attributes are also supported. The index of an array can be any expression and may include other arrays.

4. Objects such as queues, stations, counters, etc. which were previously referenced by numbers can now be referenced by user-assigned names. This, combined with the use of symbolic names for attributes and variables, makes SIMAN IV models much easier to read and understand.

5. Constraints on the form that specific numeric operands can be entered have been eliminated. Any numeric operand in the model can now be entered as any valid SIMAN expression.

6. READ and WRITE blocks have been added to allow for input-output from the model without the need to link user-code. These blocks support both formatted and unformatted, and sequential and direct-access files. In addition, Lotus style WKS spreadsheet files can be read and written directly by these blocks.

7. The interactive debugger has been enhanced to make it easier to find and correct modeling errors.

8. A new event calendar design has been incorporated which greatly improves execution speed for models with a large number of scheduled events. This new design is a hybrid between a simple link list and a modified heap.

9. PICKUP and DROPOFF blocks have been added to make it easier to handle multi-load transfers between stations.

10. The numbering of multiple entries within an experiment element is now automatically done by the experiment processor. This makes it much easier to add or delete entries within an element, since manual renumbering is no longer required.

11. The attributes of entities within queues and groups can now be directly accessed without having to remove or copy the entity.

12. Output files for multiple replications now go into a single file for each output variable. This greatly simplifies file management for experiments involving a large number of replications.

13. User-assigned labels and names can be of any length and are case-insensitive.

14. All column restrictions have been eliminated. The block label and/or statement can be entered anywhere on a line.

15. Numerous new status functions have been added to provide additional information on the state of the system.

16. Model and experiment compile and link speeds have been greatly improved.

17. The parameters for random distributions can now be specified either indirectly using parameter

sets or directly as part of the distribution specification. Random number streams associated with the distribution may be defaulted.

18. The REPLICATE element has been extended to make it easier to discard the warmup period for multiple replications with both terminating and non-terminating systems.

19. All text output such as error messages, trace information, summary report headings, etc., can be displayed in languages other than English by simply replacing a data file. Error messages, summary reports, and trace messages may be directed to a file. The general formatting of reports has been improved.

20. Additional commands have been incorporated into the Output process to make it easier to interpret the results from a simulation.

21. The more modern Pascal-like relational operators (>, <, >=, <=, ==, <>) can be optionally used in place of the older FORTRAN-like operators.

22. The MATCH block can be used for assembly operations without the requirement to specify a match attribute.

23. The DUPLICATE block can send duplicate entities to any block in the model.

24. Multiple assignments can be made at a single ASSIGN block.

25. Multiple types of resources can be simultaneously seized and/or released. For example, both a Worker and Machine can be seized at a single SEIZE block.

26. The operation of the PREEMPT block has been enhanced.

27. Compilation and runtime error messages have been improved.

28. The ARRIVALS element has been enhanced to allow for arrivals at any block in the model.

29. DSTATS may now be recorded on arbitrary expressions containing both SIMAN and user-defined variables.

30. The INSERT block has been added to allow entities to be inserted into a queue at a specific rank.

31. The signal-wait function of SIMAN has been enhanced.

32. Many previously required operands now accept default values.

Along with the new SIMAN IV, Systems Modeling is creating completely new versions of Blocks, Elements, Cinema, and the Output Processor. These new product releases will contain enhanced user interfaces and portability across a wider range of systems.

6. SUMMARY

Since its introduction in 1982, SIMAN has been used in a wide range of applications by numerous companies throughout the world. Although most of the applications have been simulations of manufacturing systems, SIMAN has also been applied to general system simulation including health care, computer, and retail systems.

In this paper we have given only a brief overview of the modeling features of SIMAN IV with an attempt to highlight those features which are new and/or particularly relevant to manufacturing systems. Only a small subset of the block functions were discussed, and no attempt was made to describe the enhanced general purpose features included in the language. In addition, a detailed discussion of the event and continuous modeling orientations included in SIMAN was omitted from the paper.

REFERENCES

Apple, J. (1977). *Plant Layout and Material Handling*. John Wiley.

CACI Products Company (1986) SIMFACTORY User's Manual, La Jolla, California.

Conway, R., W. Maxwell, J. McClain, and S. Worona (1987) *User's Guide to the XCELL+ Factory Modeling System*, Second Edition, Scientific Press, Redwood City, California.

Pegden, C.D. and A.A.B. Pritsker (1979). Simulation Language for Alternatives Modeling. *Simulation*, Vol. 33, No. 5.

Pegden, C.D. (1982). *Introduction to SIMAN*. Systems Modeling Corporation.

Schriber, T. (1974). *Simulation Using GPSS*. John Wiley, New York.

Suri, R. and M. Tomsicek (1988). Rapid Modeling Tools for Manufacturing Simulation and Analysis. Proceedings of the 1988 Winter Simulation Conference (M. Abrams, P. Haigh, and J. Comfort, eds.) Institute of Electrical and Electronics Engineers, San Diego, California.

Systems Modeling Corporation (1985). CINEMA User's Manual, State College, PA.

Zeigler, B.P. (1976). *Theory of Modeling and Simulation*. John Wiley.

AUTHORS' BIOGRAPHIES

DAVID T. STURROCK is a Senior Systems Engineer in Software Development with Systems Modeling Corporation. Since joining Systems Modeling in 1988, he has worked as Project Director for SIMAN IV. He received a Bachelors Degree in Industrial Engineering from Pennsylvania State University, with concentrations in manufacturing and automation. Prior to joining Systems Modeling, he worked in several manufacturing facilities including more than 10 years at Inland Steel Company. He has applied simulation techniques to problem solving in the areas of transportation systems, scheduling, plant layout, capacity analysis, and process design. He is an active member of IIE, SME, CASA, and SCS.

DR. C. DENNIS PEGDEN is the President of Systems Modeling Corporation. Dr. Pegden has taught Industrial Engineering at The Pennsylvania State University and the University of Alabama in Huntsville. During his tenure at the University of Alabama, he studied simulation language design and led in the development of the SLAM simulation language. After joining the faculty of the Pennsylvania State University, he continued his work in simulation language design and led in the development of the SIMAN simulation language. Dr. Pegden received his Ph.D. in 1976 in Industrial Engineering from Purdue University where he studied optimization. His current research interests include both optimization and simulation.