Proceedings of the 1984
Winter Simulation Conference
S. Sheppard, U. Pooch, D. Pegden (eds.)

INTRODUCTION TO SIMAN

C. Dennis Pegden

Associate Professor
Industrial and Management Systems Engineering
207 Hammond Building
The Pennsylvania State University
University Park, PA 16802

President
Systems Modeling Corp.
226 Highland Ave.
State College, PA 16801

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

1. INTRODUCTION

This paper discusses the use of the SIMAN simulation language for modeling manufacturing systems [5]. SIMAN is a new 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.

With the growing number of simulation languages available to the practitioner interested in manufacturing systems, the reader may be questioning the need to add one to their number. SIMAN differs from existing simulation languages in five important ways.

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 represents a natural evaluation 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. Existing general purpose languages such as GPSS [6] and SLAM [4] lack the special purpose manufacturing features provided by SIMAN. On the other hand, existing special purpose manufacturing languages such as GALS [1] and SPEED [3] are intended for a restricted class of manufacturing systems and are not applicable to systems in general.

4) SIMAN runs on mainframe, mini, and 16 bit 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 animating the outputs from the model.

In this paper, we will describe both the general purpose and special purpose manufacturing features of SIMAN. The emphasis, however, will be on the special purpose manufacturing features as they relate to the modeling of a manufacturing system.

2. GENERAL PURPOSE MODELING FEATURES OF SIMAN

SIMAN is a new 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.

The SIMAN modeling framework is based on the system theoretic concepts developed by Zeigler [7]. 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. These same data files can also be used to drive a graphical animation of the system. 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 is constructed as a sequence of blocks whose 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 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.

The general purpose attributes of an entity are denoted by the real array A(*). In the example in the previous paragraph, A(1) could be used to record the due date and A(2) the job processing time. Each entity has its own unique attribute array which is "carried along" with the entity as it moves from block to block within a model.

There are ten different basic types in SIMAN. The symbol and functions for each of the ten block types are summarized in Table 1.

The OPERATION, HOLD and TRANSFER blocks are further subdivided into several different block functions depending upon their <u>operation</u> type, <u>hold</u> type or <u>transfer</u> 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 CREATE 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 <u>block function name</u>. 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 <u>label</u>, one or more block <u>modifiers</u>, and a <u>comment line</u>. A block label is appended to the lower left side of a block and can consist of up to eight alphanumeric characters. 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.

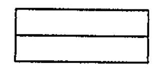
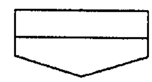
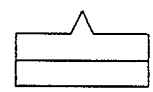
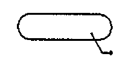
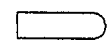
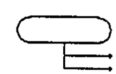
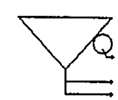
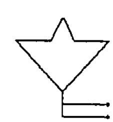
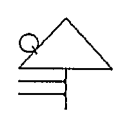
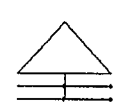
Name	Symbol	Function
OPERATION		The OPERATION block is used to model a wide range of processes such as time delays, attribute assignments, etc. Also see Table 1-2.
TRANSFER		The TRANSFER block is used to model transfers between stations via material handling systems. Also see Table 1-4.
HOLD		The HOLD block is used to model situations in which the movement of an entity is delayed based on system status The HOLD block must be preceded by a queueing facility to provide a waiting space for delayed entities. Also see Table 1-3.
QUEUE		The QUEUE block provides a waiting space for entities which are delayed at following HOLD or MATCH blocks.
STATION		The STATION block defines the interface points between model segments and the material handling systems.
BRANCH		The BRANCH block models the conditional, probabilistic and deterministic branching of entities.
PICKQ		The PICKQ block is used to select from a set of following QUEUE blocks.
SELECT		The SELECT block is used to select between resources associated with a set of following OPERATION blocks.
QPICK		The QPICK block is used to select from a set of preceding QUEUE blocks.
MATCH		The MATCH block delays entities in a set of preceding QUEUE blocks until entities with the same value of a specified attribute resides in each QUEUE.

Table 1

A block diagram model can be defined in either of two equivalent forms referred to as the <u>diagram model</u> and the <u>statement model</u>. The diagram model is a graphic representation of the system using the ten basic block symbols in Table 1. The statement model is a transcription of the diagram model into statement from 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. This latter step is straightforward and in fact has recently been automated by Professor Randall Sadowski at Purdue University as part of an interactive graphics preprocessor named BLOCKS which he developed for SIMAN. BLOCKS allows the modeler to graphically build the diagram model directly on the graphics display of an IBM Personal Computer. The BLOCKS program then automatically generates the corresponding statement version of the model for input to the SIMAN language.

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. A schematic of this system is shown in Figure 1.

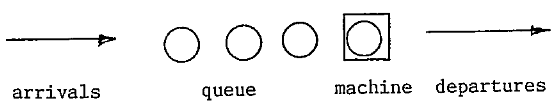


Figure 1.

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 using parameter set 1 and random stream 1 (the actual parameters used for the distributions are specified in the experimental frame in the PARAMETERS element). The workpieces continue to the QUEUE block 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 using parameter set 2 and random stream 1. 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.

An example experimental frame for this model is shown in Figure 3.

```
BEGIN;
PROJECT, Single Machine Ex.,
         Pegden, 2/15/84;
DISCRETE, 20, 0, 1;
RESOURCES: 1, MACHINE, 1;
PARAMETERS: 1,20:2,1,2;
REPLICATE,1,0,480;
END;
```

Figure 3. Experiment for Single Machine Example

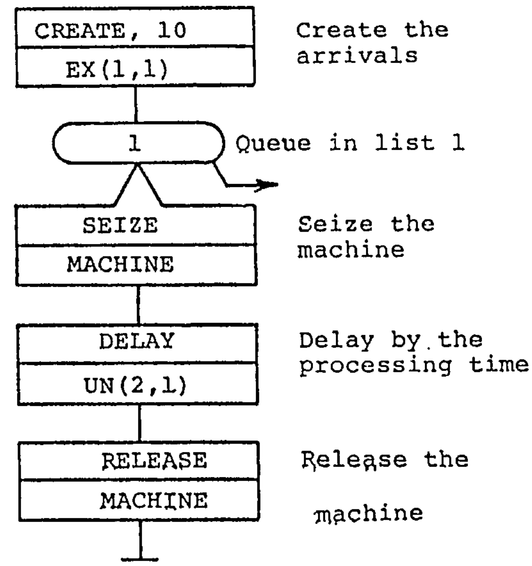


Figure 2. Block Diagram for Single Machine Example

The experimental frame specifies the experimental conditions associated with the model and includes a specification of the parameter values references by the distributions in the model, 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 resource number 1 named MACHINE is set to 1. We can add a second machine to our system by simply increasing this resource capacity to 2.

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 job shop 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 are typically comprised 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 to reduce the modeling effort.

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 requirements, 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 workcenters. 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 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 positive integer which is called the station number. This number corresponds to a physical location within the system. The station 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 number 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.

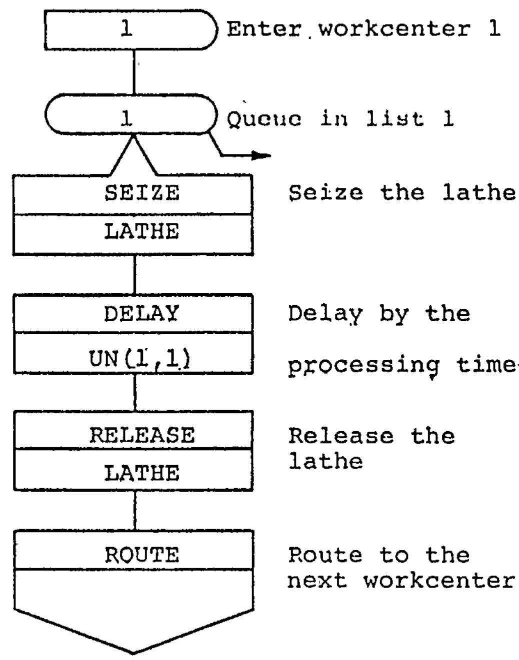


Figure 4. Example of a Workcenter Submodel

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 multi-server queueing system, depending on the capacity for the resource which is specified in the experimental frame.

The workpieces arriving to this submodel enter the STATION block, 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 becomes 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 submodel 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 station numbers 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 number 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 number to which the entity was sent.

When a macro submodel is employed, the station attribute is typically incorporated in the operands or 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 number of the entity. For example, the list number at a QUEUE block can be specified as a function of M.

The station attribute can also be used to specify a resource to be seized or released through the use of indexed resources. An indexed resource allows a single name to be assigned to a set of two or more different resource types, with each resource in the set having its own capacity. The resource types within the set are distinguished by an dex appended to the resource name. FOr example, MACHINE (1) and MACHINE (2) represent two distinct resources with different capacities. These resources are completely independent other than sharing the common name MACHINE.

To illustrate the use of indexed resources within a macro submodel, consider the block diagram shown in Figure 5. When an entity arrives to the STATION block, the station attribute M of the entity is set to its current station number. Since this macro models stations 1 through 6, M in this case will be between 1 and 6. The entities proceed to the QUEUE-SEIZE combination where they attempt to seize a unit of MACHINE (M). If all units of the resource are busy, the entity is placed into list number M where it waits for an idle unit of MACHINE (M). Once a unit is seized, the entity continues to the DELAY block where it is delayed by its processing time given by attribute 1. The entity then arrives to the RELEASE block where one unit of MACHINE (M) is released. The entity is then sent via the TRANSFER block to its next 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. For example, the general purpose attribute A(1) was used in our macro submodel shown in Figure 5 to specify the processing time on the machine. Thus an additional function of the TRANSFER block is the updating of attribute A(1) to correspond to the processing time at the next 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 three different visitation sequences.

```
SEQUENCES:1,5,EX(1,1)/3,UN(2,1):
        2,4,UN(3,1)/2,10.3/6,EX(4,1):
          3,2,EX(5,1)/6,EX(6,2);
```

Sequence number 1 consists of two workcenter visits. The first visit is to station number 5 and assigns A(1) a sample from an exponential distribution. The second visit is to station number 3 and assigns A(1) a sample from a uniform distribution. Sequence number 2 consists of visits to stations 4 then 2 and then 6 with the assignments to attribute 1 as shown. Sequence number 3 consists of visits to station 2 and then 6, with the assignments to attribute 1 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. This 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.

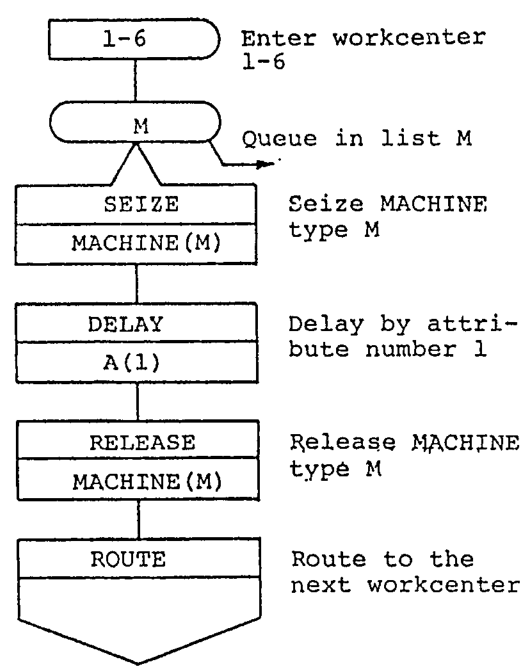


Figure 5. Example of Macro Submodel

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 resetting 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*EX(1,1),0*UN(2,1);
```

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 [2] notes that the material handling function can easily account for 50 to 70% 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. Within the material handling literature, these devices are frequently categorized into the following equipment classes.

a) Industrial Trucks - hand or powered vehicles used for intermittently moving items by maneuvering across a fixed surface. Common examples are forklift trucks, hand carts, and platform trucks.

b) Cranes, Hoists, and Manipulators - mechanical devices used for intermittently moving items through space. Common examples are overhead cranes, jib cranes, industrial robots, hoists, and monorail devices.

c) Conveyors - gravity or powered devices used in moving items continuously and simultaneously along a fixed path. Common examples are belt conveyors, bucket conveyors, hook conveyors, and trolley conveyors.

From a systems modeling perspective, the devices in the above three equipment categories 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 transport function is performed by devices in the first two equipment categories. Devices in the first equipment category perform the transport function along the ground, whereas the devices in the second equipment category perform the transport function above the ground. We will refer to the devices in either of these two equipment categories as transporters.

The second basic movement function we will call the convey function and corresponds to the continuous and simultaneous movement of items along a fixed path. The convey function is performed by the devices listed in the third equipment category. We will refer to devices in this category as conveyors.

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 numbers 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.

The characteristics of each transporter type are specified in the experimental frame and include the transporter name, capacity, distance set number, 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 distance set number is a cross-reference to a matrix containing the travel distances between all pairs of stations which each transporter unit of that type may visit. This matrix 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. 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 entities' destination station.

The representation of conveyor devices as linked cells which are either empty or occupied imposes the restriction that items can only enter the device at discrete points along the conveyor. This is representative of discrete spaced conveyor devices such as bucket, cable, and magnetic conveyors for which items can enter the device only at fixed positions along the conveyor. However, other devices, such as belts, permit continuous spacing of items along the device. These devices can be reasonably approximated by defining a small spacing between each cell. The number of cells accessed would be specified as the length of the item in cell widths.

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 OPERATION block with the STOP and START operation types, respectively.

5.0 SUMMARY

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

In this paper we have given only a brief overview of the modeling features of SIMAN with an attempt to highlight those features which are 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

1. Advanced Manufacturing Methods Group, "Introductory User's Manual for GALS," Illinois Institute of Technology Research Institute, Chicago, 1978.
2. Apple, J., Plant Layout and Material Handling, John Wiley, 1977.
3. Gross, J. and J. Ippolito, Simulation with Speed, ORSA/TIMS National Conference, San Diego, CA 1982.
4. Pegden, C. D. and A. A. B. Pritsker, Simulation Language for Alternatives Modeling, Simulation, Vol. 33, No. 5, Nov., 1979.
5. Pegden, C. D., Introduction to SIMAN, Systems Modeling Corporation, 1982.
6. Schriber, T., Simulation Using GPSS, John Wiley, New York, 1974.
7. Zeigler, B. P., Theory of Modeling and Simulation, John Wiley, 1976.