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

# SLAM II, INCLUDING A MATERIAL HANDLING EXTENSION

Jean J. O'Reilly
Pritsker & Associates, Inc.
P.O. Box 2413
West Lafayette, IN 47906, U.S.A.

## ABSTRACT

SLAM II was the first simulation language which allowed a modeler to formulate a system description using process, event, or continuous world views or any combination of the three. Since its initial release in 1981, SLAM II has undergone continual development and application. This paper will provide an introduction to the modeling language and describe the most recent developments in SLAM II itself and support software.

## 1. INTRODUCTION

SLAM II, the Simulation Language for Alternative Modeling, was the first simulation language which allowed a modeler to formulate a system description using any of three approaches (world views) or any combination of the three. SLAM II integrates the process-oriented world view of Q-GERT®(Pritsker, 1977) and the discrete-event/continuous world views of GASP IV™ (Pritsker, 1974) in order to free the modeler to select the approach which best represents his system or with which he feels most comfortable. This integrated framework allows the SLAM II user to take advantage of the simplicity of the process-oriented (network) approach and to extend a model with discrete event constructs should the network approach become too restrictive. Continuous variables may be used in conjunction with a network or discrete event model whenever this is the most convenient way to represent system elements. The ability to construct combined network-event-continuous models with interactions between each orientation makes SLAM II an extremely flexible tool for simulation.

Since its introduction, SLAM II has been regularly enhanced to significantly increase its flexibility and efficiency. A microcomputer version has been available since 1984. Most recently, a Material Handling Extension (MHEX) has been released which addresses the problems of modeling material handling movements, a few of which are vehicle assignment priorities, speed and distance calculations, and contention. This paper will introduce a few of the modeling techniques available with SLAM II with some simple examples.

## 2. NETWORK MODELING

A simulation model normally begins with a network, or flow diagram, which graphically portrays the flow of entities (people, parts, or information, for example) through the system. A SLAM II network is made up of "nodes" at which processing is performed. Twenty node types in SLAM II, shown in Figure 1, provide for such functions as entering or exiting the system, seizing or freeing a resource, changing variable values, collecting statistics, and starting or stopping entity flow based on system conditions. Nodes are connected by branches, called "activities", which define the routing of the entities through the system. Routing may be deterministic, probabilistic, or based on system variables. Time delays on activities may represent processing times, travel times, or waiting times. Entities which proceed from node to node over activities may have unique characteristics, called "attributes", which control their processing. Entities may reside in "files", or ordered lists of entities which are waiting for some change in system status. The graphical framework for representing a network model simplifies model development and communication.

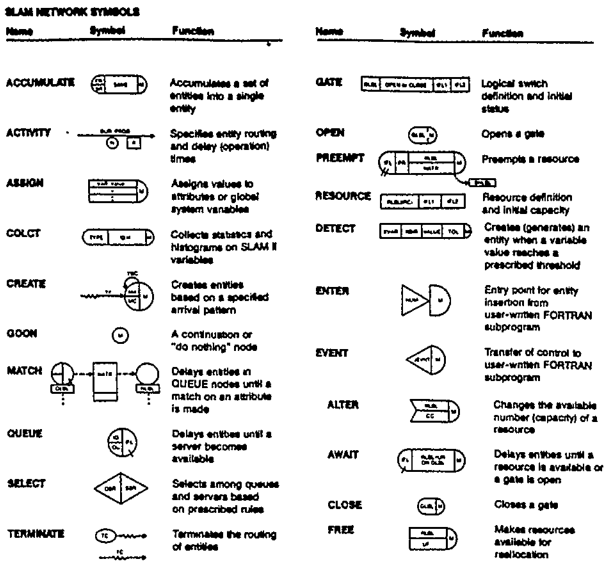

Figure 1: SLAM II Network Symbols

The process of building a SLAM II network model consists of choosing the symbols which can represent system processes and combining them in a diagram which represents the entity flow. A single-server queueing model (representing, for example, a bank teller) is shown in Figure 2. The network begins with a CREATE node which generates the first customer arrival at simulated time 0. and continues to generate arrivals at a rate drawn from an exponential distribution. A QUEUE node is used to delay arrivals until the server is available. The server, whose processing time is sampled from a normal distribution, is represented by the ACTIVITY, or branch, following the QUEUE. Upon completion of the activity, a COLCT node records the interval between departure time and the customer's arrival time, which was stored in attribute 1.

Unless the network was constructed using TESS (Pritsker, 1984), the diagram is then translated into a set of input statements as shown in Figure 3. Each symbol corresponds to an input statement, and each statement may be followed by a comment which describes the processing being performed. The output from this model would automatically report statistics on customer waiting time, queue length, server utilization, time in system and throughput.

## 3. USING DISCRETE EVENT CONCEPTS

In the discrete event orientation of SLAM II, the modeler identifies the discrete points in time at which the state of the system can change and develops the logic associated with each such "event". SLAM II provides support subroutines which perform such common simulation tasks as scheduling events, moving entities into and out of files, collecting statistics, and obtaining random samples. Most models built with SLAM II are not strictly network or discrete event but a combination of the two approaches.

Several interfaces are possible between a SLAM II network and user-written FORTRAN inserts. One is the EVENT node, which is a "do-it-yourself" node. The EVENT node invokes a user-written subroutine in which highly complex logic may be performed. Support subprograms provide information on system status and allow that status to be changed. Two additional interfaces are illustrated in the network segment shown in Figure 4.

The CREATE node which begins this segment creates only one entity, at time 0. The ACTIVITY following the CREATE node has a duration specified as USERF. This is a user-written function which in this example reads the entity description (attributes) from a disk file. USERF is set equal to the delay time required until the entity should enter the network. The GOON node at the end of the activity releases two entities, one to enter the network and one to activate USERF to read the next entity description. The branch back to the CREATE node is conditional, and the USERF function sets that condition false if no entities remain to be read.

The second interface for non-standard processing is shown at the AWAIT node, used to allocate scarce resources. Two distinct resources are included in this model, MACH1 and MACH2. Which one should actually be assigned to process an entity depends upon complex system conditions, and must be determined dynamically. Therefore, instead of naming a specific resource at the AWAIT node, a user-written subroutine ALLOC is invoked to determine the resource assignment. This routine also determines the service time based upon the allocation and stores it in ATRIB(2) (the second attribute) of the entity being processed. In addition, ATRIB(3) is set to the resource type assigned so that it may be referenced at the FREE node, which releases and attempts to reallocate a resource.

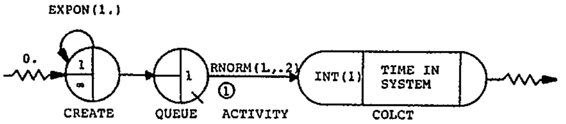

Figure 2: A single-server queuing example

```
CREATE,EXPON(1.),0.,1;           GENERATE ARRIVALS
QUEUE(1);                        WAIT FOR SERVICE
ACTIVITY(1),RNORM(1.,.2);        PROCESS
COLCT,INT(1),TIME IN SYSTEM;     COLLECT STATISTICS
ENDNETWORK;
```

Figure 3: Example model input

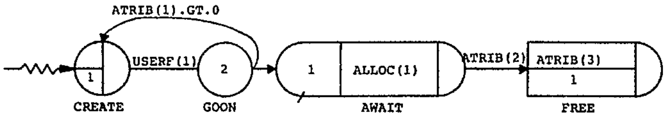

Figure 4: Illustration of user-written interfaces

## 4. CONTINUOUS MODELING

In a continuous simulation model the state of the system is represented by variables that change continuously over time. The modeler specifies equations which determine the values of state variables and the "step size", or time increment, between the updating of variable values. These equations may be differential equations, in which case the simulator uses a numerical integration algorithm to obtain new variable values from the derivative values.

For continuous simulations, SLAM II provides a set of special storage arrays which the modeler uses to encode the equations of continuous variables. In combined models these variables (known as SS and DD variables) may be affected by the occurrence of discrete events as well as by their defining equations.

Continuous variables have proven to be an efficient way to model high-speed, high volume systems such as packaging lines (O'Reilly, 1985). In such a system, a buffer area between two machines may contain several hundred items - too many to be modeled individually. The population of such a buffer is conveniently modeled as a continuous variable which increases at the production rate of the feeding machine and decreases at the production rate of the following machine (Figure 5). The equations defining the rates of change for continuous (SS) variables are written in a FORTRAN subroutine. SLAM II updates the variable values at prescribed time intervals and monitors those variables against any threshold values defined. One threshold value, for example, would be the capacity of a buffer. When it is crossed, the feeding machine would need to cease production until the buffer level decreased enough to accept more production.

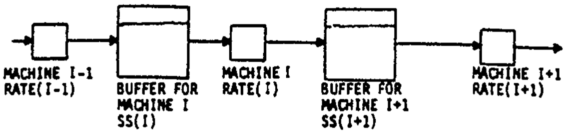

Figure 5: Modeling with continuous variables

## 5. MATERIAL HANDLING MOVEMENTS

Among the most complicated elements to incorporate in a simulation model are automated devices which follow fixed paths. These include overhead cranes, stacker cranes, and AGVS (automated guided vehicle systems). Movements of such devices must be modeled in detail if one is to take into account interference among devices which share a common path. When contention occurs, some way must be found to assign priority to the vehicle which will be allowed to proceed. A Material Handling Extension (MHEX) to SLAM II, first available in 1986, provides constructs for simulating these complexities (Pritsker, 1986). Its concepts were derived from several simulation models developed at P&A which required detailed material handling logic.

## 6. MODELING CRANES

An example involving stacker cranes is shown in Figure 6. The schematic depicts a local ASRS system with two stacker cranes serving a lathe and a mill; storage is maintained in nine racks along the crane runway.

The MHEX software takes into account the following complications in this system:

1. Movement times are dependent on crane velocity, distance from destination, and interference with the companion crane.
2. Competing requests for a crane must be prioritized.
3. Storage is limited, and the amount to be allocated depends on the size of an item.
4. If alternative storage locations are possible, selection may be based upon both proximity and material type.

These interactions are illustrated in the network segment shown in Figure 7. It begins with a GWAIT (generalized AWAIT) node which requests an available rack and crane. Knowing from the RACK definition (not shown) the capacities and locations of the storage areas and where to find the size of the item to be moved, the software will allocate the closest storage location having sufficient space.

Knowing from the CRANE definitions the velocity, acceleration and deceleration of the equipment, and keeping track of both cranes' positions, the software will release the item only when an available crane (CR1 or CR2) can reach the pickup point.

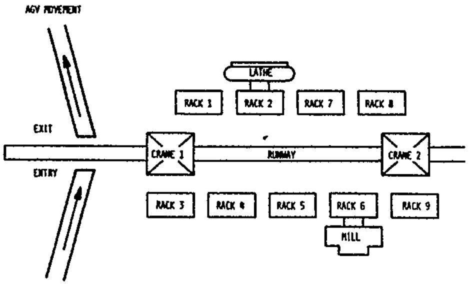

Figure 6: Schematic Diagram of an ASRS System

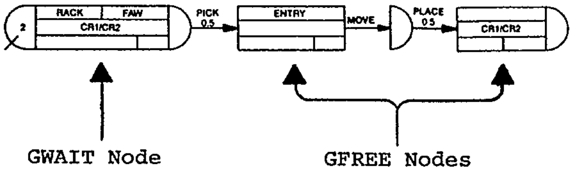

Figure 7: Movement From Entry Point to Storage

After the item is loaded on the crane, taking 0.5 minutes, a GFREE node releases the pickup point and initiates the crane move. Movement time is calculated internally and is based on equipment speed, distance between the ENTRY and RACK locations, and any interference encountered dynamically. Following transport, 0.5 minutes are required to remove the item from the material handling equipment, and a second GFREE node releases whichever crane was assigned.

## 7. MODELING AN AGVS

An Automatic Guided Vehicle System (AGVS) consists of a fleet of vehicles, a guidepath, and a computer control system which determines how a vehicle is selected and routed to a job request. Unlike cranes on runways, AGV's on guidepaths may turn corners, greatly complicating the logic required to deal with interference and possible alternate routes.

MHEX includes constructs for defining an AGV fleet (number of vehicles, their sizes and speeds) and guidepaths (number of control points, length of each segment, and direction of travel). Once these elements are defined, three node types are used to model the control logic of the system by allocating a vehicle, initiating a move, and releasing a vehicle for reallocation.

Figure 8 depicts an example AGVS serving a machining cell. The guidepath consists of 15 segments and 13 control points. It is described with a set of 15 statements similar to this one:

```
VSGMENT, 9, 7, 8, 35, UNI;
```

This defines the segment between control points 7 and 8 (MACH1 and MACH2), which is 35 feet long and accepts unidirectional travel. The vehicle fleet is described with a single statement specifying two vehicles, each four feet in length, with speeds of 4.5 feet/minute empty and 4.0 feet/minute loaded, as follows.

```
VFLEET,AGV1,2,4.5,4.0,,,4.0,4.5,,7/CLOSEST,
STOP(4),4(2,4);
```

Also defined on this statement are the primary allocation rules for the vehicles (closest to requesting point), the action taken with idle vehicles (stop at control point 4), and initial vehicle location.

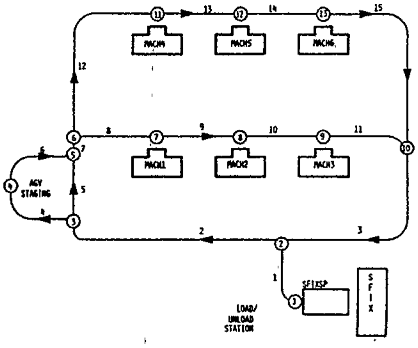

Figure 8: Schematic Diagram of Manufacturing Cells with an AGVS

Once the system is thus defined, the flow of jobs through the system is modeled with a network of only thirty nodes and activities. Figure 9 depicts the part of this network which accomplishes the delivery of a job to an available machine.

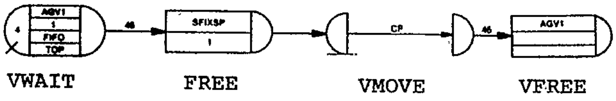

Figure 9: An AGV Movement

This network segment begins with a VWAIT node which allocates the AGV closest to control point 1 to the first job waiting at that location. Once an AGV is available, the requesting job will continue to wait at the VWAIT node until the vehicle can actually arrive at the request point. It takes 45 seconds to load the vehicle, after which the loading station is released and the move to control point CP can begin. This control point, located at an available machine, was assigned to the job prior to the AGV request. The time required for the move is calculated internally, based upon vehicle speed, distance to CP, and interference (if any) with the other AGV. After arrival at the control point and a 45-second delay for unloading, the AGV is released and either sent to the staging area or reallocated to a waiting job. A similar network segment accomplishes movement from the machining cell back to the load/unload station.

## 8. SUMMARY

SLAM II is distributed by Pritsker & Associates, Inc. It is written in ANSI Standard FORTRAN, allowing it to be installed on any supermini or mainframe computer having a standard FORTRAN compiler. Annual revisions throughout the life of the program have significantly increased its flexibility and efficiency. SLAM II's FORTRAN base makes it easy to have a model include interfaces to other functions such as database management, statistical analysis, or linear optimization.

In 1984 a microcomputer version of SLAM II was made available for the IBM PC and compatible micros. The PC version produces, in addition to standard output reports, DIF files for use with other microcomputer software such as graphics and spreadsheets. Also in 1984, Pritsker & Associates announced the release of TESS™(The Extended Simulation System), a support system for simulation projects using SLAM II. Using a TESS database, manipulated with an easy-to-use command language, one may interactively build SLAM II models and prepare output reports from simulation data in a variety of graphical formats, including animation. The most recent development to SLAM II is the Material Handling Extension, released in 1986.

SLAM II has been used for hundreds of simulation projects and as the basis for simulation courses in many colleges and universities. Published applications (see references) describe models dealing with problems in manufacturing, transportation, material handling, staffing, experimental design, and many more. In other applications, SLAM II has been used to simulate computer and communications systems.

## REFERENCES

Dessouky, M.M., F.H. Grant and D. Gauthier (1985). "Simulation of an Injector Plunger Production Line". In: Proceedings of the 1985 Winter Simulation Conference, San Francisco, California.

Doring, B. and A. Knauper, (1983). "A Simulation Study with a Combined Network and Production Systems Model of Pilot Behavior on an ILS-Approach". In: Automatica, Vol. 19.

Duket, S.D., and C.R. Standridge (1983). "Applications of Simulation: Combined Models". In: Modeling, Issue No. 19.

Erdbruegger, D., D. Starks & C. Farris, (1982). "SLAM II Model of the Rose Bowl Staffing Plans". Presented at 1982 Winter Simulation Conference, San Diego, California.

Felder, R.M., P.M. Kester & J.M. McConney, (1983). "Simulation/Optimization of a Specialities Plant". In: Chemical Engineering Progress, July 1983, 25-35.

Graber, Andre and Francois E. Cellier (1982). "On the Usefulness of SLAM II for the Modeling and Simulation of Large Transport Systems". Presented at the IMACS Conference, Montreal, Canada, August 1982.

Gross, J.R., S.M. Hare, and Sukumar Roy, (1982). "Simulation Modeling as an Aid to Casting Plant Design for an Aluminum Smelter". Presented at the IMACS Conference, Montreal, Canada, August 1982.

Hoffman, S.E., M.M. Crawford, and J.R. Wilson (1983). "An Integrated Model of Drilling Vessel Operations". Presented at the Winter Simulation Conference, Arlington, Virginia, December 1983.

Lilegdon, W.R., C.H. Kimpel and D.H. Turner (1982). "Application of Simulation and Zero-One Programming for Analysis of Numerically Controlled Machining Operations in the Aerospace Industry". Presented at the Winter Simulation Conference, San Diego, California, December 1982.

Lu, K.H. and L. VanWinkle (1984). "A Critical Evaluation of Some Problems Associated with Clinical Caries Trials by Computer Simulation". In: Journal of Dental Research, May 1984, 796-804.

Martin, David L. (1983). "A Simulation-Optimization Model for Exploration and Expoitation of Exhaustible Mineral Resources". Presented at the SME-AIME Annual Meeting, Atlanta, Georgia, March 1983.

McCallum, J.N. and B.B. Nickey (1984). "Simulation Models for Logistics Managers". In: Logistics Spectrum, Volume 18, No. 4, Winter 1984.

Morris, W.D., T.A. Talay and D.G. Eide (1983). "Operations Simulation for the Design of a Future Space Transportation System". Presented at the AIAA 21st Aerospace Sciences Meeting, Reno, Nevada, January 1983.

Murphy, D.R., S.D. Duket and E. Sigal (1985). "Evaluating Surgical Block Schedules Using Computer Simulation". In: Proceedings of the 1985 Winter Simulation Conference, San Francisco, California.

O'Reilly, J., M. Sale, and D. Martin (1985). "The Use of Continuous/Discrete Event Models in Manufacturing". In: Proceedings of the 1985 Winter Simulation Conference, San Francisco, California.

Prestwood, W.T.. "PATRIOT Air Defense Weapon System Deployment Model Using SLAM". U.S. Army Missile Command Logistics Support Analysis Office, Redstone Arsenal, Alabama.

Pritsker, A.A.B. (1974). The GASP IV Simulation Language, John Wiley, 1974.

Pritsker, A.A.B. (1977). Modeling and Analysis Using Q-GERT Networks, Halsted Press and Pritsker & Associates, 1977.

Pritsker, A.A.B. (1982). "Applications of SLAM", IIE Transactions, March 1982, 70-77.

Pritsker & Associates, (1984). The TESS User's Manual, West Lafayette, Indiana, 1984.

Pritsker, A.A.B. (1986). Introduction to Simulation and SLAM II, 3rd Edition, Systems Publishing Corp., 1986.

Ratcliffe, L.L., B. Vinod and F.T. Sparrow (1984). "Optimal Prepositioning of Empty Freight Cars". In: Simulation, June 1984.

Standridge, C.R., and J.R. Phillips (1983). "Using SLAM and SDL to Assess Space Shuttle Experiments". In: Simulation, July 1983, 25-35.

Wilson, J.R., D.K. Vaughan, E. Naylor, and R.G. Voss (1982). "Analysis of Space Shuttle Ground Operations". In: Simulation, June 1982, 187-203.

## AUTHORS' BIOGRAPHY

JEAN J. O'REILLY is Director of Software Services at Pritsker & Associates, Inc. She holds a Bachelor of Arts degree in Mathematics from St. Mary's College, Notre Dame, Indiana and a Master of Science in Applied Mathematics from Purdue University. Since joining P&A in 1978, Ms. O'Reilly has been involved in software development and in applying SLAM II in various consulting projects. In her current position she is responsible for user support, training, and distribution for all P&A products.

Jean J. O'Reilly
Pritsker & Associates, Inc.
1305 Cumberland Avenue
P.O. Box 2413
West Lafayette, IN 47906
(317) 463-5557