SLAM II™ TUTORIAL

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

In 1979, the state-of-the-art in simulation languages was extended with the introduction of SLAM™, the first language that provided three different modeling viewpoints in a single integrated framework.(5) SLAM permits discrete event, continuous, and network modeling perspectives and/or any combination of the three to be implemented in a single model. SLAM represented a significant breakthrough in simulation methods development, as it provided the flexibility to use the most appropriate world view for the system being studied. This improved upon the more traditional situation in which simulation modelers were restricted to the modeling perspective embodied in the language they were using.

The success of this new approach was readily apparent. SLAM has been installed in more than 300 industrial, academic, and governmental organizations. This response by the simulation community is evidence that SLAM meets the needs of practicing simulation modelers by:

* making it possible to model a wider variety of systems using the most effective modeling perspective;

* allowing rapid model development using network modeling concepts; and

* supporting models that combine modeling perspectives through well-defined, carefully designed interfaces.

Figures 1, 2, and 3 present the organization for discrete event modeling, the organization for continuous modeling and the network nodes used in SLAM. Complete discussions of these concepts are presented in references 3 and 4.

As experience with the use of SLAM increased, Pritsker & Associates has refined and expanded SLAM capabilities to produce SLAM II VERSION 2.0. SLAM's modeling power and flexibility have been enhanced by adding the following functional capabilities:

* blocking and balking at AWAIT nodes

* user-provided resource allocation strategies at AWAIT nodes

* state conditions based on threshold crossings by discrete as well as continuous variables

* user-provided SELECT criteria for queues and servers when SLAM II selection rules do not suffice

* additional statistical output reporting on gate status, resource availability, the event calendar, and histograms

* variable batch sizes at an ACCUMULATE node

* variable probability for branch selection on ACTIVITY statements

* variable statistics numbers on ACTIVITY statements

* increased flexibility for arithmetic statements at ASSIGN nodes

* variable resource specification at ALTER, AWAIT, FREE, and PREEMPT nodes

* variable gate specification at AWAIT, CLOSE, and OPEN nodes

* variable file numbers at AWAIT, QUEUE, and PREEMPT nodes

* multiple SELECT or MATCH nodes associated with a QUEUE node

In addition to the modeling enhancements listed above, SLAM II VERSION 2.0 allows for the tailoring of trace reports showing the operation of a model. This is accomplished through the use of a MONTR statement which initiates the calling of the user-written subroutine UMONT. Functions for accessing the current system status including the location of the current entity in a network are now provided. Furthermore, additional functions for accessing statistics on activities, resources, and gates are included.

In order to facilitate the rerunning of network models, SLAM II VERSION 2.0 provides an option to save and reload a decoded network description by adding a parameter to the NETWORK statement. In

CH1953-9/83/0000-0321 $01.00 © 1983 IEEE

this way, a network can be run without re-inputting the network description.

SLAM's input error messages have been improved. These improvements expedite the modeling and debugging process and allow the modeler to more quickly move to the task of system evaluation.

SLAM has been upgraded to include search algorithms that improve the efficiency in performing file manipulations for large problems and to reduce the processing time required to detect threshold crossings of continuous variables.

Output reporting flexibility has been increased with the addition of optional 72 column output allowing the convenient use of all computer terminals. In addition, SLAM now provides an automated interface with SIMCHART, an interactive computer graphics package also available from Pritsker & Associates. SLAM will generate simulation data that can be used by SIMCHART to prepare graphics, plots, histograms, pie charts, and pie graphs.(1) SLAM can also be interfaced to standard graphics packages.

In summary, SLAM has been improved to provide additional efficiencies in the simulation modeling process. Ease of model design, speed of implementation, and clarity of outputs have all been enhanced. With these additions, the application of SLAM can result in lower development costs, shorter run times, and lower computing costs.

All improvements made to SLAM have been made so that existing SLAM models are upwards compatible. That is, all models that previously executed in SLAM can be run in SLAM II VERSION 2.0 without any changes. The SLAM user simply adds data inputs or additional user code to invoke the new capabilities. With this design, the existing user can upgrade his or her modeling efficiency with little or no loss in time.(2)

The tutorial presentation will provide examples on the use of SLAM II VERSION 2 and its new capabilities.

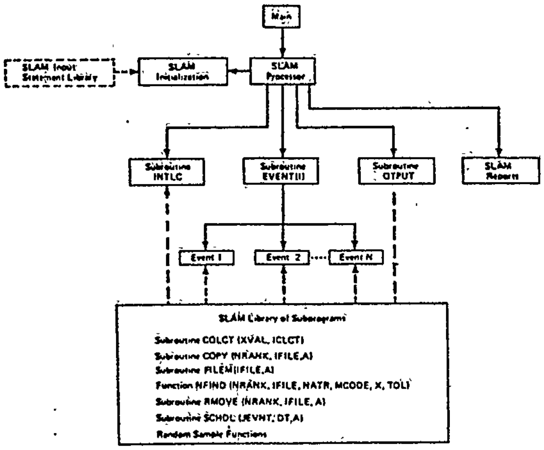

Main
SLAM Input Statement Library
SLAM Initialization
SLAM Processor
Subroutine INTLC
Subroutine EVENT(I)
Subroutine OTPUT
SLAM Reports
Event 1
Event 2
Event N
SLAM Library of Subprograms
Subroutine COLCT (XVAL, ICLCT)
Subroutine COPY (NRANK, IFILE,A)
Subroutine FILEM(IFILE,A)
Function NFIND (NRANK, IFILE, NATR, MCODE, X, TOL)
Subroutine RMOVE (NRANK, IFILE, A)
Subroutine SCHDL (JEVNT, DT,A)
Random Sample Functions

Figure 1. SLAM Organization for Discrete Event Modeling

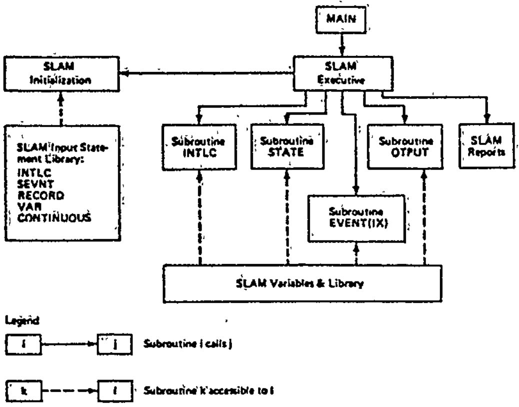


Figure 2. SLAM Organization for Continuous Modeling

REFERENCES

Duket, S.D, A.F. Hixson, and L. Rolston (1981), The SIMCHART User's Manual, West Lafayette, Indiana: Pritsker & Associates, Inc.

Duket, S.D., J.J. O'Reilly, and R.J. Hannan (1981), SLAM II Enhanced Simulation Capabilities, West Lafayette, Indiana: Pritsker & Associates, Inc.

O'Reilly, J.J., (1983), SLAM II User's Manual, West Lafayette, Indiana: Pritsker & Associates Inc.

Pritsker, A.A.B., and C.D. Pegden (1979), Introduction to Simulation and SLAM, West Lafayette, Indiana: Systems Publishing Company.

Pritsker, A.A.B., (1981), Advanced Topics in SLAM, Proceedings of the Winter Simulation Conference pp. 197-198.

SLAM NETWORK SYMBOLS

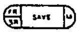
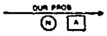
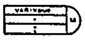
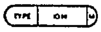
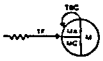
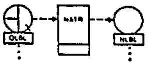
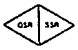
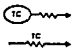
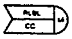
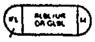
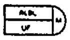
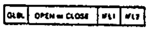
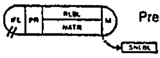
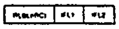
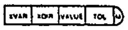
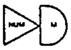
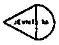
Name	Symbol	Function
ACCUMULATE		Accumulates a set of entities into a single entity
ACTIVITY		Specifies entity routing and delay (operation) times
ASSIGN		Assigns values to attributes or global system variables
COLCT		Collects statistics and histograms on SLAM II variables
CREATE		Creates entities based on a specified arrival pattern
GOON		A continuation or "do nothing" node
MATCH		Delays entities in QUEUE nodes until a match on an attribute is made
QUEUE		Delays entities until a server becomes available
SELECT		Selects among queues and servers based on prescribed rules
TERMINATE		Terminates the routing of entities
ALTER		Changes the available number (capacity) of a resource
AWAIT		Delays entities until a resource is available or a gate is open
CLOSE		Closes a gate
FREE		Makes resources available for reallocation
GATE		Logical switch definition and initial status
OPEN		Opens a gate
PREEMPT		Preempts a resource
RESOURCE		Resource definition and initial capacity
DETECT		Creates (generates) an entity when a variable value reaches a prescribed threshold
ENTER		Entry point for entity insertion from user-written FORTRAN subprogram
EVENT		Transfer of control to user-written FORTRAN subprogram

FIGURE 3. Types of Network Nodes in SLAM