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

SLAM II TUTORIAL

Jean J. O'Reilly
A. Alan B. Pritsker
William R. Lilegdon
Pritsker & Associates, Inc.
West Lafayette, Indiana 47906

SLAM II® provides three different modeling viewpoints in a single integrated framework. The discrete event, continuous, and network modeling perspectives and/or any combination of the three can be implemented in a single model using SLAM®.

The success of this combined approach has been readily apparent. SLAM has been installed in more than 400 industrial, academic, and governmental organizations. This response by the simulation community was evidence that SLAM met 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 network nodes, the organization for discrete event modeling and the organization for continuous modeling used in SLAM. Complete discussions of these concepts are presented in references 1,2, and 3.

As experience with the use of SLAM increases, (4,5) Pritsker & Associates has refined and expanded SLAM capabilities. In SLAM II , 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
* increased flexibility for arithmetic statements at ASSIGN nodes
* parameterized (attribute-based) resource, gate, file and activity specification, allowing macros to be built for common processes
* multiple SELECT or MATCH nodes associated with a QUEUE node
* access to current resource utilization as well as availability
* network access to the STOPA function, allowing easier control of entity flow
* time persistent histograms
* zero-capacity blocking queues
* warning messages providing more information on "fixup" actions taken by the simulator.

In addition to the modeling enhancements listed above, SLAM II 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. Alternatively, the trace may be turned on or off dynamically from user code. 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 provides an option to save and reload a decoded network description by adding a parameter to the NETWORK statement. In 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 II includes algorithms that improve the efficiency in performing file manipulations for large problems and that 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. SLAM II is easily interfaced with TESS™, The Extended Simulation System, available from P&A. Using TESS, SLAM II input is prepared interactively, and SLAM II output is stored in a TESS database as requested by simple data collection control statements. TESS includes extensive data management functions for reporting, analysis, and graphic displays of simulation data. (6,7)

In summary, SLAM II 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 II can result in lower development costs, shorter run times, and lower computing costs.

All improvements made to SLAM II 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 without any changes. The SLAM user simply adds data, statements, 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. (1).

SLAM II is now available for the IBM PC and compatible microcomputers in addition to mini and mainframe computers. The microcomputer version of SLAM II provides all the modeling power of the mainframe version. Models built using the microcomputer version are 100% compatible with existing SLAM II installations on mainframes.

SLAM II for the IBM PC requires 256K RAM to execute models containing only network data statements. With additional memory and the Microsoft FORTRAN compiler, modelers may include user-written FORTRAN subprograms in their models. SLAM II operates entirely under MS-DOS and can be used with most DOS enhancements (hard disk drives, RAM disk drives, any printer supported by MS-DOS, etc.).

SLAM II for the IBM PC may be easily interfaced to any existing software library on the IBM PC. Models may be created using any editor or word processor that creates a standard text file. Output reports from SLAM II are provided in standard text form or DIF (Data Interchange Format) format, for use with spreadsheets or graphics by LOTUS 1-2-3 or Advanced Version VisiCALC to create user formatted reports and graphics.

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

REFERENCES

1. Pritsker, A.A.B. (1984) Introduction to Simulation and SLAM II, Second Edition, West Lafayette, Indiana: Systems Publishing Company and New York, Halsted Press.

2. Pritsker, A.A.B. (1984) Solutions Manual for Introduction to Simulation and SLAM II, Systems Publishing Corporation, West Lafayette, Indiana.

3. Pritsker, A.A.B., (1984), Instructor's Guide for Introduction to Simulation and SLAM II, Systems Publishing Corporation, West Lafayette, Indiana.

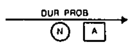
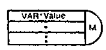
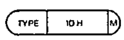
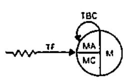
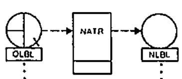
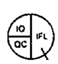
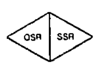
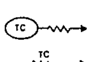
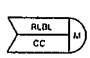
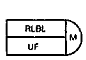
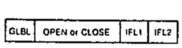
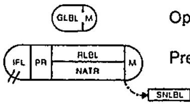
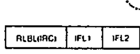
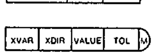
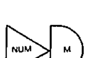
4. Pritsker, A.A.B., "Applications of SLAM", IIE Transactions, Vol. 14, No. 1, March 1982, pp. 70-77.

5. Duket, S.D., and C.R. Standridge, "Applications of Simulation: Combined Models", MODELING, Issue No. 19, December 1983.

6. Standridge, C.R., S.A. Walker, D.K. Vaughan, "TESS Tutorial", Proceedings of the 1984 Winter Simulation Conference, Dallas Texas, November 28-30, 1984.

7. Standridge, C.R., et al, TESS User's Manual, Pritsker & Associates, Inc., West Lafayette, Indiana, 1984.

Figure 1. 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

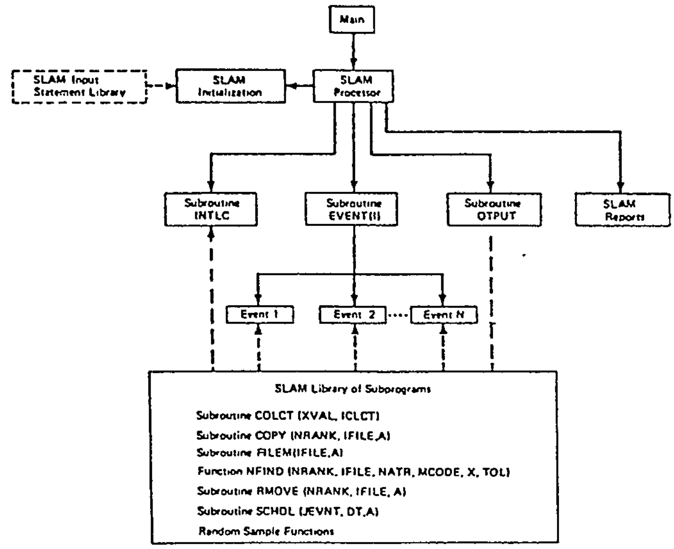


Figure 2. SLAM Organization for Discrete Event Modeling

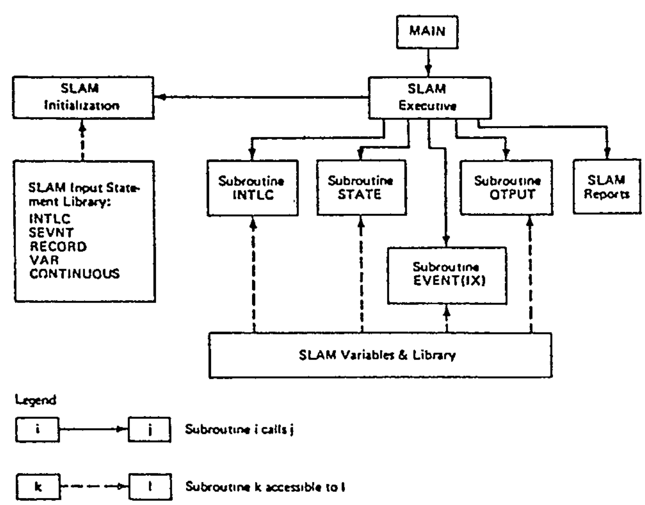


Figure 3. SLAM Organization for Continuous Modeling