INTRODUCTION TO SLAM II ® AND SLAMSYSTEM ®

Jean J. O'Reilly

Pritsker Corporation
1305 Cumberland Avenue
West Lafayette, Indiana 47906, U.S.A.

ABSTRACT

SLAM II is a general-purpose language which has been used since 1979 to model a wide variety of systems in fields including (but not limited to) manufacturing, transportation, communication, computer systems, military operations, health care and banking. Experience with these applications, plus development in simulation research, have led to the expansion of the language throughout its lifetime. The ease of using the SLAM II language was greatly increased by the introduction of SLAMSYSTEM in 1988; SLAMSYSTEM Version 4.5 is the most recent release, arriving in 1994. SLAMSYSTEM is an integrated simulation system based on the Microsoft Windows interface (under MS-DOS) or the OS/2 Presentation Manager. This tutorial will demonstrate the process of building, analyzing, and presenting SLAM II simulation models using SLAMSYSTEM. A major revision of the product is under development as of this writing and will be described in the tutorial.

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. 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-discrete event-continuous models with interactions between each orientation makes SLAM II an extremely flexible tool for simulation.

Since its introduction, SLAM II has continued to evolve as a result of extensive application. In addition to enhancements to the modeling language itself, the following tools have been integrated with SLAM II:

- A Material Handling Extension (MHEX) provides detailed modeling of special resources (cranes, storage areas, automatic guided vehicles, and guidepaths).
- The SLAM II Interactive Execution Environment (IEE) allows the interruption of a simulation in order to examine system status, reassign variable values, step through events, or call SLAM II support routines for debugging or gaming.
- SLAMSYSTEM includes SLAM II in an integrated simulation system for advanced personal computers.

2   SLAMSYSTEM OVERVIEW

SLAMSYSTEM integrates software supporting the range of tasks performed in a simulation project. All features are accessible through pull-down menus and dialog boxes, and are selected from the SLAMSYSTEM Executive Window, shown in Figure 1.
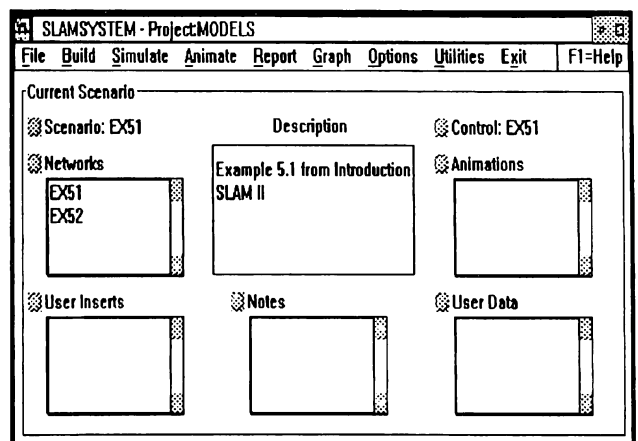

Figure 1: SLAMSYSTEM Executive Window

A SLAMSYSTEM project consists of one or more scenarios, each of which represents a particular system alternative. A scenario is comprised of components such as networks, a control, user inserts and user data, animations, notes for model documentation and model output. The Current Scenario Box in the SLAMSYSTEM Executive Window shows the components associated with the scenario presently being analyzed.

SLAMSYSTEM's project maintainer removes the burden of remembering the procedure necessary to perform the tasks of simulation, animation, and output review. Each time the modeler requests one of these functions, the project maintainer examines the components of the current scenario to determine if any of them have been modified, indicates whether tasks such as model translation and executable creation should be performed to reflect the changes, and allows the user to specify whether these tasks should be done prior to performing the requested function.

Since SLAMSYSTEM is a Windows or OS/2 application, multiple tasks may be performed in parallel while simulations are executed in the background. The simulation modeler can switch between tasks by using a mouse to click in the appropriate window.

3   MODEL BUILDING

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. SLAM II nodes 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.

The process of building a SLAM II network model consists of choosing the symbols which can represent system processes, combining them in a diagram which represents the entity flow, and parameterizing the symbols with model-specific data. A single-server queueing model (representing, for example, a workstation) is shown in Figure 2. The network begins with a CREATE node which

generates the first job arrival at simulated time 0.0 and continues to generate arrivals at a rate drawn from an exponential distribution. A QUEUE node is used to delay arrivals until the station is available. The station, 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 job's arrival time, which was stored in attribute 1. The graphic modeling approach is both quick to use and an effective way to communicate the structure of a model.

SLAMSYSTEM is used to build SLAM II models with the assistance of the graphical network builder (Figure 2) and the forms-oriented control builder. These builders use the mouse and a forms-based approach to aid the modeler in the entry of the system description. They remove the need to remember syntax and field definitions, and allow the modeler to concentrate on constructing models.
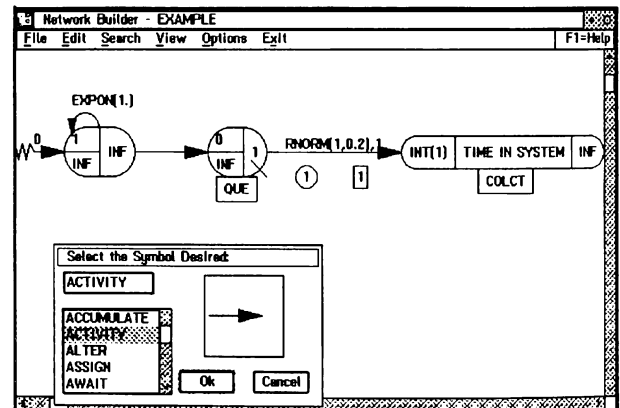

Figure 2: Graphical Network Builder

Network symbols are selected from a graphical palette and located with the mouse. The symbol's parameter values are specified by filling out a form of information applicable to the particular symbol. On-line error checking is performed upon completion of the form so that input errors can be corrected immediately. The Windows or OS/2 "Clipboard" can be used for copying a set of symbols to other positions in the network or to other networks or applications. Symbols may be repositioned by selecting and relocating them with a mouse. The Network Builder also facilitates model building by providing context-sensitive help, searching capabilities, and options for placing symbols at grid points, selecting symbol colors, and flowcharting models by defaulting symbol parameters.

A model may be decomposed into multiple networks to facilitate project management. The networks are associated with a scenario and are concatenated at run time.

4   RUN CONTROL

Control statements are used to define experimental conditions such as the number of runs to make, run length, and when to clear statistics after a warmup period. Other control functions include the ability to change default random number seeds, queueing priorities, and initial queue populations. A model's control statements are defined in the SLAMSYSTEM Control Builder (Figure 3). The modeler selects a statement, then fills in a form defining the statement parameters. As with network symbols, most fields have default values, prompts are supplied, and on-line error-checking is immediate.
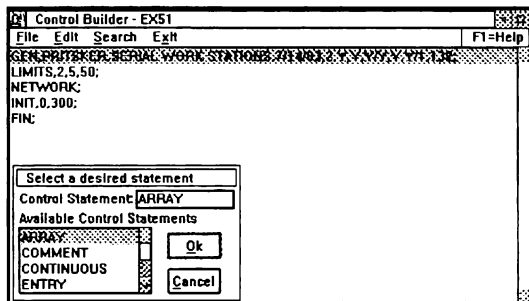

Figure 3: Control Builder

5   USER INSERTS

As a model grows in detail, some aspects may become difficult to model within the network constructs provided. For this reason, SLAM II provides easy access to user-written subroutines for more complex discrete event processing. From FORTRAN or C, the modeler may call upon a library of subprograms which perform common simulation functions: event scheduling, storage and retrieval, collecting and accessing statistics, and generating random samples. Other subprograms allow the modeler to access information about the status of network operations and alter that status.

In some cases, system variables do not change values at discrete, identifiable points in time but are continuously changing as a function of time. Such variables might include the level of fuel in a storage tank, the temperature of a heat-treated item, or the position of a moving target.

Continuous aspects of a system are modeled in SLAM II by coding the equations defining continuous variables in a user-written subprogram. When the model includes differential equations, SLAM II automatically integrates them to maintain the values of continuous variables over time. Thresholds may be defined so that when a variable reaches a critical level, new processes may be initiated.

6   OUTPUT ANALYSIS AND PRESENTATION

SLAMSYSTEM provides the capability for comparing simulation outputs from various scenarios both graphically and textually. A report "browser" allows alternative textual outputs to be compared side by side. Graphically, output may be viewed in the form of bar charts, histograms, pie charts, and plots. Bar charts can be used to display the value of a statistic across as many as 10 scenarios. It is possible to view multiple windows of graphical output at the same time, as shown in Figure 4. Graphical and textual information from SLAMSYSTEM graphs and reports can be exported to other Windows or OS/2 packages for additional analysis and for documentation.
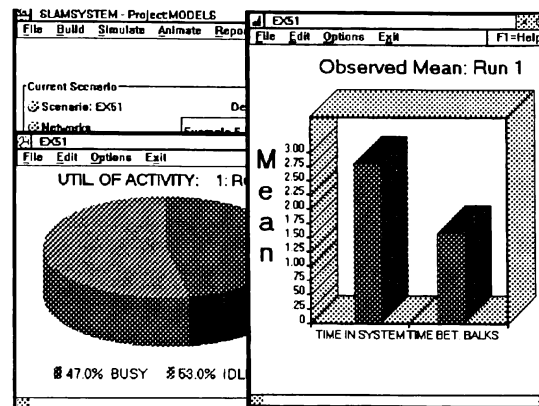

Figure 4: Multiple windows

7   ANIMATING MODELS

The SLAMSYSTEM OS/2 animator (Figure 5) utilizes special features of the OS/2 operating system. These special features include preemptive multitasking and a windowed user interface. Preemptive multitasking allows multiple animations of a single simulation model to be displayed concurrently. The windowed interface is used in the graphical animation builder to increase the animation developer's productivity. In addition, advanced animation constructs dramatically reduce the amount of time required to animate complex situations.

With the SLAMSYSTEM OS/2 animator, one may develop and display multiple animations of a single simulation. For example, the modeler can create one animation of a system at an aggregate level and another at a department level, side by side in separate windows. The two views may then be displayed by selecting those animations through the SLAMSYSTEM executive and running the simulation.

Animation constructs under the OS/2 animator are called actions. Many of these actions directly correspond to elements in a SLAM II network model. These direct mappings are provided for activities, resources, queues, global variables (either as counters, plots, or bar graphs), AGV segments and cranes. For example, the ACTIVITY action shows movement of a symbol. It requires that the modeler define a symbol, a graphical path location where movement of the symbol will be shown, and the number of an activity in the SLAM II model to which to tie the movement.

Advanced animation actions are provided for modeling more complex system constructs. One of these actions can be used to accurately represent the movement of parts along a conveyor. The modeler identifies a graphical path where movement will appear, a symbol or symbols to move, and the simulation event or events which trigger the action. The symbols may be made to appear instantly on the path or they may be moved from the beginning of the path to a buffer area in a specified time. When a symbol is removed from the buffer, the remaining set of entities moves forward just as packages on a conveyor behave when the lead package is removed.

The symbols manipulated by the animator are of two types: graphical items one wants to display or move, and the background on which they will appear. These symbols are stored in standard OS/2 bitmap format. This allows

them to be exchanged between programs using the OS/2 clipboard. For example, the modeler may have a facility layout drawn in AutoCad with OS/2. This layout can be copied to the clipboard and pasted into the SLAMSYSTEM animator as the background for the animation. In addition, a utility is provided with the SLAMSYSTEM animator which converts any portion of the OS/2 Presentation Manager (PM) screen to a bitmap. This allows the modeler to import images from OS/2 PM programs which do not support the clipboard. The SLAMSYSTEM animator under Windows is similar to though less powerful than the OS/2 module.

8 INTEGRATION OF SLAMSYSTEM WITH OTHER SOFTWARE

The SLAMSYSTEM user will typically use SLAMSYSTEM with other software programs for systems analysis. SLAMSYSTEM was designed to be used in an integrated manner. Using the Windows or Presentation Manager standards it is easy to move data into or out of SLAMSYSTEM.

Modelers often want to drive their simulation models from historic data. The SLAMSYSTEM project framework includes a user data element as part of the scenario definition. This data element may contain schedule, routing, or facility configuration information. It is read by the
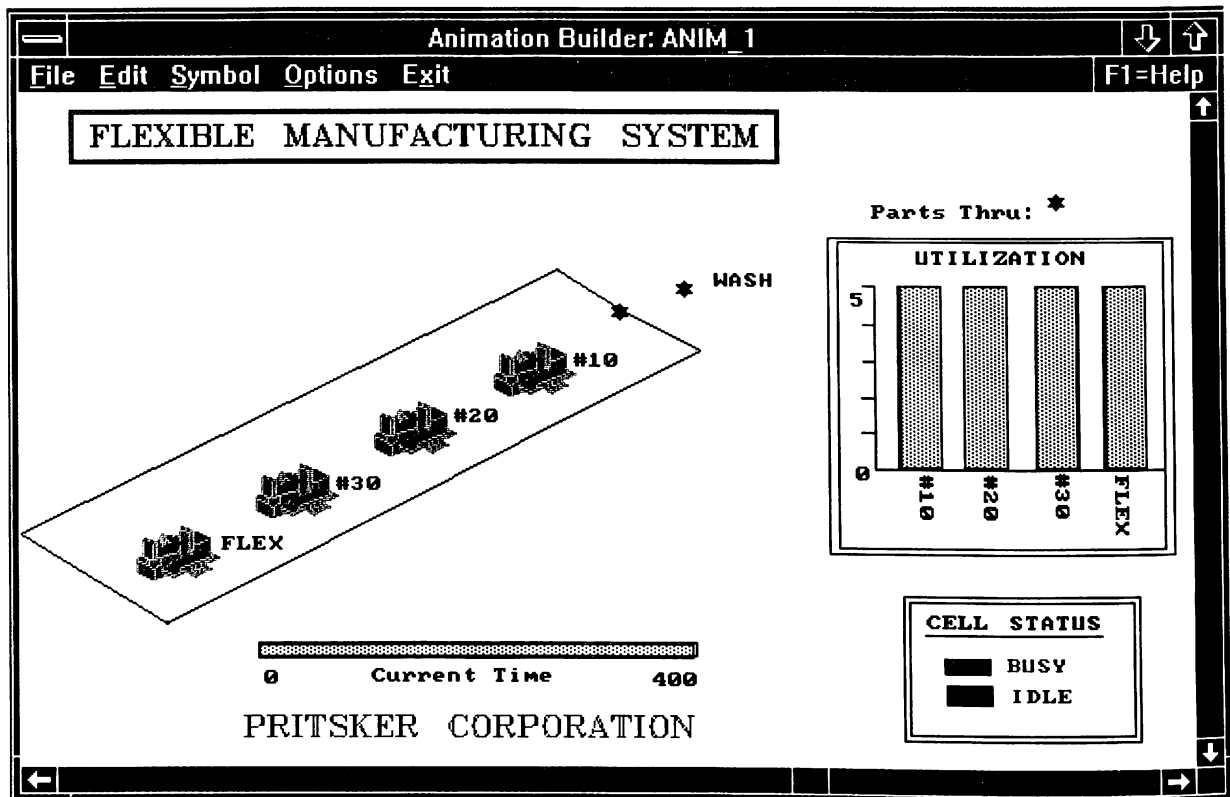

Figure 5: SLAMSYSTEM Animation Builder

SLAMSYSTEM model during the simulation execution. The user can create this data with a spreadsheet or the built in data editor provided with SLAMSYSTEM, saving the information in standard text format. Routines are then added to the model to read the data during the simulation execution. An inerface to Excel spreadsheets allows data to be imported from Excel directly, without programming.

A SLAMSYSTEM animation can use graphics created from other programs. As discussed in the animation section, the graphical elements manipulated by the animation program can be created using CAD, drawing, or paint programs. They can be loaded into SLAMSYSTEM by using the OS/2 bitmap format or, under Windows, a graphical translation program can be used. The output charts and plots created by SLAMSYSTEM can be exported, via the Windows or Presentation Manager clipboard, to other applications. For example, a pie chart created by SLAMSYSTEM may be copied to the clipboard and pasted into a word processing document describing the results of the model.

The standard SLAMSYSTEM output reports are often exported to other packages for communication or analysis purposes. Analysts use sections of the standard SLAMSYSTEM report with a spreadsheet to add cost data to simulation output. The report can be imported into the spreadsheet via the clipboard or by using a standard text file importing feature available in many of the popular spreadsheet programs. An interface to the SIMSTAT statistical analysis package allows output data to be exported and analyzed using the variety of techniques provided by SIMSTAT (confidence level estimation and regression analysis, for example).

9 CONCLUSION

SLAM II is a proven, powerful modeling methodology. It has been used for hundreds of simulation projects and as the basis for simulation courses in many colleges and universities. Published applications describe models dealing with problems in manufacturing, transportation, material handling, staffing, experimental design, communications systems, and many more.

Continuing development of SLAM II and simulation support software has culminated in SLAMSYSTEM, an integrated simulation system for advanced personal computers. While SLAMSYSTEM runs under Windows or OS/2, SLAM II may be run on virtually any workstation or mainframe, and SLAM II/SLAMSYSTEM models are transportable across all platforms. SLAM II and SLAMSYSTEM are distributed by Pritsker Corporation, which offers regularly scheduled training classes as well as applications support.

REFERENCES

Pritsker, A. A. B. 1995. *Introduction to simulation and SLAM II*, 4th Edition. New York: Halsted Press.

Pritsker, A. A. B., C. E. Sigal and R. D. J. Hammesfahr. 1994. *SLAM II network models for decision support*. San Francisco: The Scientific Press.

Pritsker Corporation. 1993. *SLAMSYSTEM user's guide*. Indianapolis, IN.

AUTHOR BIOGRAPHY

JEAN J. O'REILLY is a member of the technical staff at Pritsker Corporation in West Lafayette, Indiana. She holds a Bachelor of Arts degree in Mathematics from St. Mary's College, Notre Dame, Indiana and an M.S. in Applied Mathematics from Purdue University. Since joining Pritsker Corporation 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 technical support, training, and quality assurance for the SLAM product family.