Proceedings of the 1983
Winter Simulation Conference
S. Roberts, J. Banks, B. Schmeiser (eds.)

AN INTRODUCTION TO SIMULATION SUPPORT SOFTWARE

Charles R. Standridge, Ph.D.
Steven A. Walker
Pritsker & Associates, Inc.
P.O. Box 2413
West Lafayette, IN 47906

INTRODUCTION

A wide variety of simulation languages have served analysts well for conceiving models and implementing these models on a computer. However, many other activities in addition to model conception and implementation are necessary parts of performing a simulation study. Only recently have extensions to simulation languages been provided to deal with these other aspects of performing a simulation project.

These other activities may be partitioned in to six categories:

1. Model building. Once a model has been conceived, it must be constructed in a form that is acceptable to a simulation language processor. Typically, these forms have been computer programs or statements which describe the model in terms of fixed sets of predefined modeling elements such as queues, activities and resources. Recently, methods for describing models in response to interactive prompts or by interactively building graphical representations have been developed.

2. Program building. Programs are built by adding run specifications to a model. These run specifications contain information such as what data to collect during the simulation run, the time the simulation run should begin and end, and the initial conditions for the simulation run. In a similar way to models, run specifications have been traditionally given by the user as a file of statements. Furthermore, often these statements have been intermixed, or at least contained in the same file as, the description of the model. Some languages allow these statements to be input in a separate file from statements pertaining to the model. Still others allow the statements to be built interactively by responding to prompts.

3. User data management. User data can include information required by a model as input such as order books, a schedule of maintenance for equipment, or the routes of products through a shop as well as any other data the user deems relevant to a particular study. Mechanisms for creating or editing such data and making them available during simulation runs are required. Multiple versions of the same set of user data must be maintained to aid in the execution of different simulation scenarios.

4. Scenario execution. A simulation program run with a particular set of user data is a scenario. A simulation study may consist of multiple scenarios. These scenarios are generated either through variations in user data, modifications to the run specifications, or changes in the model.

   During the execution of each scenario, simulation results should be automatically collected. These results include statistical summaries, histograms and individual observations. In addition, a user should be able to define and collect non-standard statistical summaries, histograms and individual observations. The collected data are preserved for future analysis and presentation.

5. Data analysis. Both user data and simulation results can be the objects of statistical analysis. Typically, the computation of basic statistics (mean, standard deviation, minimum, maximum and count) and histograms, perhaps broken down by time interval or attribute values, are desired. There are advantages to analyzing data independently of the simulation run. For example, multiple analyses of the same data can be performed, subsets of the collected data can be selected for analysis, and all the desired analyses need not be known prior to making the simulation run.

6. Data presentation. The presentation of both user data and simulation results is a vital part of a simulation study. Presentation techniques fall into two categories, reports and graphics. Reports can be generated either in formats determined by the report generator or specified by the user. Graphics such as plots, histograms, bar charts, pie charts and stat charts (the graphical display of means, minimums and maximums) are useful in displaying the dynamics of queue sizes, resource utilization, and time delays. Futhermore, graphics can be used to animate the entity movements occurring in a simulation.

In order to be most useful, software support in each category must be supplied in a single integrated software system. This system must allow the user to move between commands in each category at will, using the same user interface for all work. It must allow access to all relevant data whenever desired. It must shield the user from being a computer technician, that is having to spend a significant amount of time managing files, making sure data are in correct formats, remembering which of a set of procedures do what functions and the like.

An Integrated Simulation Support System

Pritsker and Associates has developed an integrated simulation support system conforming to the characteristics discussed in the previous section. This product is being integrated with both the SLAM II V2.0 and MAP/1 simulation languages. This tutorial will discuss the SLAM II V2.0 implementation of the support system.

Integration of the system components is accomplished using relational data base management principles. Specifically, all data are kept in a single data base. These data elements are network models, run specifications, programs, user data, scenarios, statistics, histograms, individual observations from simulation runs, display sets (parameters for drawing graphical displays), and pictures (graphical displays preserved for future recall). Furthermore, all references to data are made in terms of these data elements by name. The data base manager automatically and transparently transforms references made in terms of data element names into a relational data model. A single high-level, user interface is provided by the PAQ (Pritsker and Associates Query) Language. All system commands can be expressed in this language. The PAQ also provides entry into the model building and the prompting/editing subsystems. A FORTRAN subroutine interface is also provided for data storage and retrieval.

Figure 1 shows the integrated architecture of the simulation support system. The PAQ gives access to functions in all six support categories. These store and retrieve data elements (networks, etc.) from the data base as necessary in performing their tasks.

Figure 2 shows the capabilities of the integrated simulation support system for SLAM II V2.0. Major functions in each of the six support categories are listed in bullet form under the appropriate box. In general, these functions provide for building and/or editing and deleting elements such as SLAM II networks, run specifications, programs, user data, scenarios, statistics, histograms, individual observations of simulation variables, display sets and graphic pictures.

There are two version of the support system one graphical and the other alphanumeric. The alphanumeric version includes all of the functions of the graphical version except for the last six bullets under data presentation. Each version has its own prompting/editing function. This capability assists the user in building run specifications, editing user data, building display sets and preparing slides.

Functions in the network building category deal with SLAM II networks. In the graphical version of the support system, a subsystem exists for building or editing networks interactively and graphically. Other commands prepare hard copy of networks on plotters and printers and report selected elements of the network. In the alphanumeric version, this building is accomplished with the prompting and editing subsystem.

Functions in the program building category deal with run specifications and programs. Run specifications can be built, edited, reported and deleted. Furthermore, run specifications can be combined with networks to form programs. Functions in the user data management category handle user defined data. These functions allow the user to describe and define different sets of data, populate these sets by loading data from external sources such as sequential files, edit the data to create new versions using the prompting/editing subsystem, and delete data.

Functions in the scenario building category deal with executing simulations. Scenarios are built by linking programs and versions of appropriate user data. During the simulation of a scenario, user data can be accessed for input. Furthermore, results, including histograms, statistics, and individual observations of variables, are collected either automatically or through subroutine calls made by the user. In the scenario building stage, user written event and continuous code can be integrated with the network part of the model.

Data analysis functions deal with statistical summaries and histograms. These may be collected during simulation runs or computed from user data or individual observations collected during simulation runs. Other functions provide for the deletion of statistics and histograms.

Data presentation functions display user data and simulation results. Central to the presentation functions are display sets. These sets provide the parameters for displaying plots, pie charts, histograms, bar charts, stat charts, and reports. The editing/prompting system is used to build display sets or change parameter values. The other presentation functions accept display sets as input to tell how graphs and reports are to be created. If the user specifies no display set, a default one is used. Still other functions allow the user to produce animations of simulation runs on user designed diagrams. Other functions provide for the creation of materials appropriate for technical presentations such as bullet slides. All graphics including SLAM II networks, can be displayed on transparency material or as 35 mm slides if appropriate hardware is available.

Implementation of the Integrated, Simulation Support System

The integrated simulation support system has been implemented in standard FORTRAN (66 level). The graphics facilities have been coded in the DI 3000 graphics support language. The code is completely portable and will run on any computer with sufficient main or virtually memory and a standard FORTRAN compiler. DI 3000 is device independent, supporting over 50 graphics devices such as terminals, plotters and printers.

Summary

The integrated simulation support system provides software tools for model building, program building, user data management, scenario building, data analysis and data presentation. Functions provide for network building and reporting, collection of data during simulation runs, loading and editing of user data, data analysis and data presentation by report generation, graphic displays and animation of simulation runs. The PAQ language provides a single high-level user interface for the system, while FORTRAN subroutines from within computer programs programs provide for data storage and retrieval. Integration is accomplished by storing all data in a single relational data base which the user accesses through the PAQ by names of appropriate data elements.

BIBLIOGRAPHY

Codd, E. F. (1970), "A Relational Model of Data for Large Shared Data Banks", Communications of the ACM, Vol. 13, No. 6, June.

Ören, T. I., Zeigler, B. P. (1979), "Concepts for Advanced Simulation Methodologies", Simulation, Vol. 32, No. 3, March, pp. 69-82.

Pegden, C. D. and Pritsker, A. A. B. (1982), "The Many in Interfaces of SLAM in Progress in Modeling and Simulation"(F. E. Cellier, ed.) Academic Press, London, England, pp. 247-264.

Pritsker, A. A. B. and Pedgen, C. D. (1979), "An Introduction to Simulation and SLAM", Halsted Press, New York and System Publishing, West Lafayette, Indiana.

Standridge, C. R. (1981), "Using the Simulation Data Language (SDL)", Simulation 37, Vol. 37, No. 9, pp. 73-81.

Standridge, C. R. (1981), "The Simulation Data Language (SDL): Applications and Examples", Simulation, Vol. 37 No. 10, pp. 119-130.

Standridge, C. R. and Wortman, D. B. (1981), "The Simulation Data Language (SDL): A Data Base Management System for Modelers", Simulation Today in Simulation Vol. 37, No. 8, pp. 55-57.

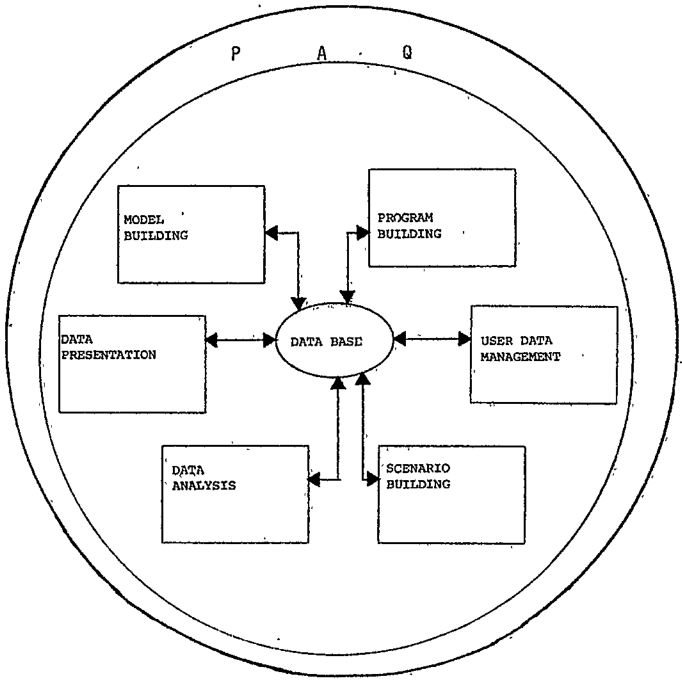

Figure 1. The Integrated Architecture of the Simulation Support System

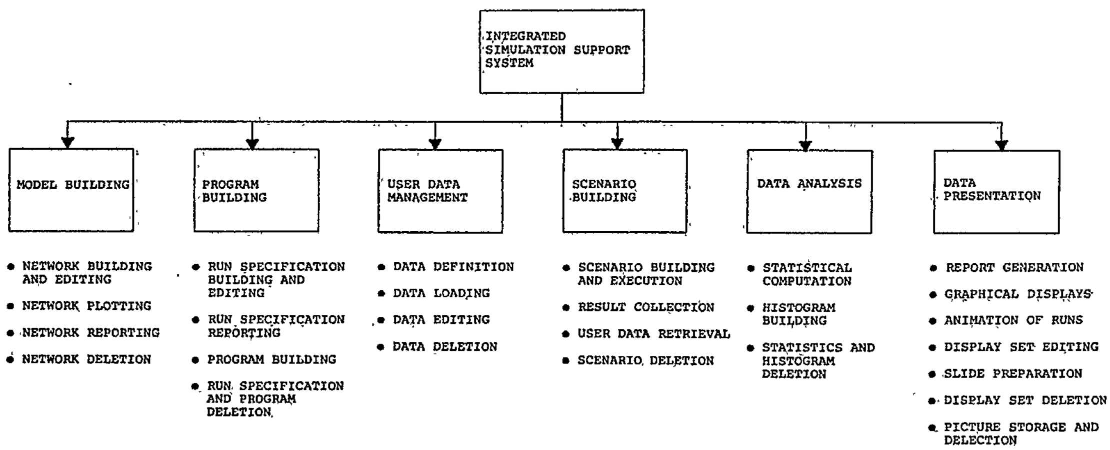

Figure 2. The Integrated Simulation Support Systems for SLAM II V2.0