*Proceedings of the 1996 Winter Simulation Conference*
*ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain*

# AWESIM: THE INTEGRATED SIMULATION SYSTEM

A. Alan B. Pritsker
Jean J. O'Reilly

Pritsker Corporation
8910 Purdue Road, Suite 600
Indianapolis, Indiana 46268, U.S.A.

## ABSTRACT

AweSim is a totally new general-purpose simulation system, released in June of 1996. AweSim takes advantage of the latest in Windows® technology to integrate programs and provide componentware. AweSim includes the Visual SLAM simulation language to build network, discrete event, and continuous models. Network models require no programming yet allow user-coded inserts in Visual Basic or C. Discrete event and continuous models can be created using the object-oriented technology of Visual Basic, C or Visual C++ and can be combined with network models. This tutorial will demonstrate the process of using AweSim's componentware and provide examples of user interfaces that allow integration with other Windows applications both directly and through the AweSim database.

## 1 INTRODUCTION

AweSim is a general purpose simulation tool used to model systems as diverse as manufacturing, transportation, communication/information systems, military operations, health care, and banking. Network (process-oriented) models are built from a set of powerful, flexible building blocks which accomplish functions typically required to simulate any system. Should user-written extensions to a network model be necessary, they are developed in C or Visual Basic and easily linked to the basic system. Continuous variables may also be used whenever this is the most convenient way to represent system elements. This ability to construct models of unlimited size using the approach most suited to the problem gives AweSim utmost flexibility in model input. Flexibility in model output is achieved through AweSim's relational database, which makes it easy to create custom reports using common tools such as Dbase, Access, and Excel.

## 2 MODEL INPUT

An AweSim model can consist of several components: networks, run controls, user-written inserts, data files, animations, and notes for documentation. These components are organized and accessed through the AweSim executive window as shown in Figure 1.

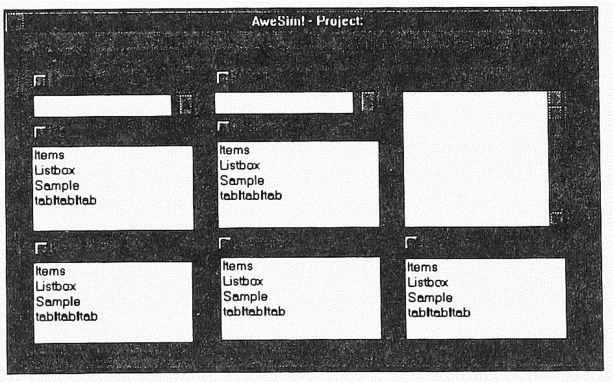


**Figure 1: AweSim Executive Window**

AweSim incorporates the Visual SLAM modeling methodology, which is a new simulator based on the powerful and proven modeling concepts used in the SLAM II modeling language. The basic component of a Visual SLAM model is a network, or flow diagram, which graphically portrays the flow of entities (people, parts or information, for example) through the system. A Visual SLAM network is made up of "nodes" at which processing is performed, connected by "activities" which define the routing of entities

and the time required to perform operations. Visual SLAM 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 status conditions.

The network segment shown in Figure 2, for example, shows a CREATE node, which generates entities according to an exponentially distributed interarrival time, connected by an ACTIVITY (represented by a direct branch) to an ASSIGN node.

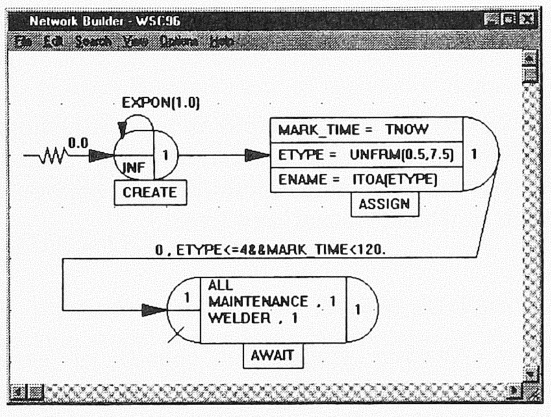


**Figure 2: Graphical Network Builder**

At the ASSIGN node, the newly created entity's attributes (characteristics) are assigned. In this example, the attribute MARK_TIME holds the entity's time of arrival, the ETYPE (entity type) attribute is randomly assigned to be an integer between 0 and 7, and an attribute called ENAME is the character representation of entity type. The ASSIGN node is followed by an AWAIT node at which two resources, MAINTENANCE and WELDER, are allocated to the newly arrived entity. The activity taking the entity from the ASSIGN node to the AWAIT node is taken only if entity type is less than or equal to 4 and arrival time is less than 120. This example is meant to suggest the flexibility provided by a rich set of random variables and intrinsic functions and by the ability to use complex expressions throughout a network model.

A network is built interactively in AweSim by selecting symbols from a graphical palette and locating them with the mouse. The symbol's parameters are specified by filling out a form, as illustrated by the ACTIVITY form shown in Figure 3. On-line error checking is performed upon completion of the form so that input errors can be corrected immediately. AweSim also facilitates model building by providing context-sensitive help and search capabilities (Figure 3).

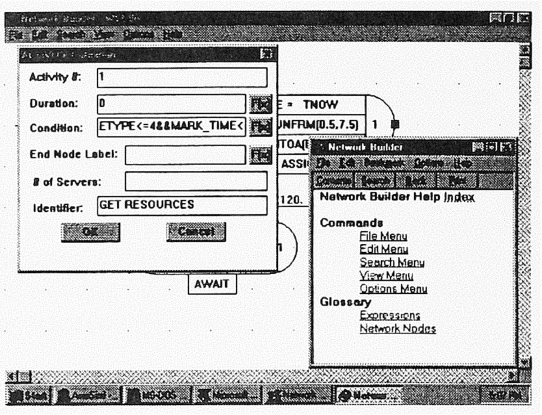


**Figure 3: Network Builder dialog box and Help window**

In addition to the Network Builder, AweSim includes a forms-based builder for developing run controls (simulation run length, output options, when to clear statistics, queue ranking rules, etc.) and text builders for User Data files and Notes for documentation. The User Insert builder is either a text builder, if AweSim is installed for use with user inserts written in C, or the Microsoft Visual Basic development environment.

## 3 MODEL OUTPUT

AweSim provides the ability to compare output from various scenarios both graphically and textually. A report "browser" allows alternative textual outputs to be compare side by side. Graphically, output may be viewed within AweSim in the form of bar charts, histograms, pie charts, and plots. Bar charts can be used to display the value of a statistic across 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 the AweSim database can be exported to other Windows packages such as EXCEL or Word for additional analysis and for documentation.

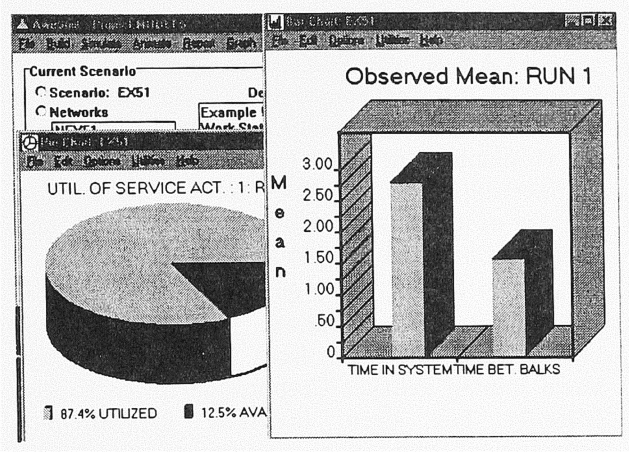


**Figure 4: Multiple report windows**

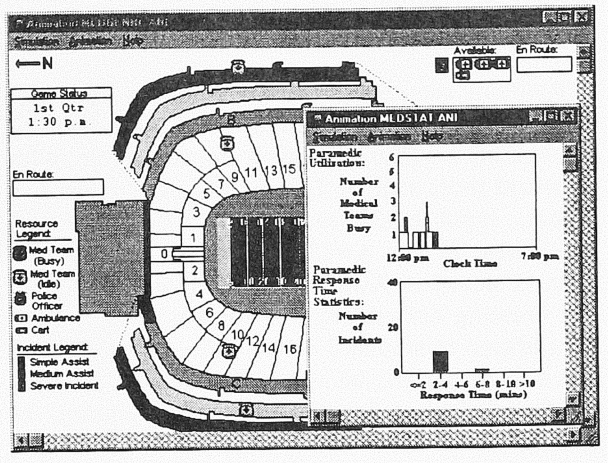


**Figure 5: Animation with multiple windows**

## 4 ANIMATION

With the AweSim animator, one may develop and display multiple animations for a single scenario. 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 associating the animations with the current scenario and running the simulation. Animation constructs, called actions, correspond to elements in an AweSim network model. 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 AweSim model to which to tie the movement. Other actions are provided to display resource status, items in a queue, variable values, and other system status conditions. 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 Windows bitmap format, allowing them to be exchanged between programs using the Windows clipboard.

## 5 INTERACTIVE EXECUTION

The AweSim Interactive Execution Environment (IEE) provides an interface with an executing simulation. The modeler may examine, modify, save, or load the current system status using the IEE. The IEE supports the debugging of a model under construction and verification of a completed model. The analyst may use the IEE to develop and analyze alternative control strategies for a system or to demonstrate the operation of the model to non-modelers. The IEE control panel (Figure 6) lets you advance through the model step-by-step or by setting breakpoints. At any point you may examine variable values, statistics, or queue entries or save the system status in order to reload and experiment with alternative scenarios.

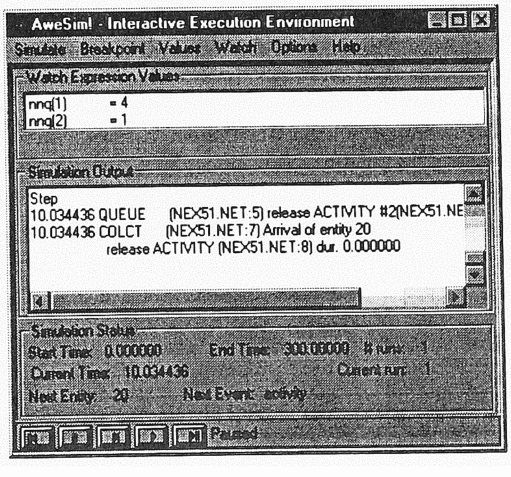


**Figure 6: Interactive Execution Environment**

## 6 INTEGRATION OF AWESIM! WITH OTHER SOFTWARE

AweSim was designed to integrate easily with other Windows applications. AweSim is built on a relational database which is accessible with standard tools such as Dbase, Access, FoxPro and Excel. Input data is easily moved from an Excel worksheet to the AweSim input tables. Output data is stored in AweSim output tables, available for creating custom reports using a favorite tool. In addition to standard output data, raw data from the simulation can be stored in standard database or Excel format for analysis, manipulation, or presentation. Data used to create AweSim output graphics can also be exported "on the fly" to an output file for use in any tool accepting comma-delimited input.

An AweSim animation can use graphics created from other programs. As mentioned in the discussion of animations, the graphical elements manipulated by the animator can be created using CAD, drawing or paint programs and loaded into AweSim by using the Windows bitmap format. The output charts and plots created by AweSim can be exported via the clipboard to other applications. For example, a pie chart created by AweSim may be copied to the clipboard and pasted into a word processing document describing the results of the model.

## 7 SUMMARY

AweSim is a brand new simulation support system which replaced the popular SLAMSYSTEM software in 1996. Written over the last two years, it takes advantage of the latest Windows technology in order to provide a simulation support system able to interface with a variety of familiar tools. It incorporates the Visual SLAM modeling methodology, a modern replacement for the proven SLAM II modeling language. AweSim is distributed by Pritsker Corporation, which offers regularly scheduled training classes as well as applications support.

## AUTHOR BIOGRAPHIES

**A. ALAN B. PRITSKER** is Chairman of the Board of Pritsker corporation. He obtained a Ph.D. from the Ohio State University in 1961. He is a member of the National Academy of Engineering. Dr. Pritsker served twice as a member of the Board of Directors of the WSC and as Board Chairman in 1984 and 1985.

**JEAN O'REILLY** is a member of the technical staff at Pritsker Corporation. She holds 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 a variety of consulting projects. She is currently responsible for technical support, training and quality assurance for the SLAM product family.