Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

SOFTWARE FOR SIMULATION

Jerry Banks

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0205

ABSTRACT

This tutorial describes computer languages and other software packages that support discrete-event simulation.

1 INTRODUCTION

The first and second sections describe general purpose and manufacturing oriented software, respectively. Usually, a distinction is made between simulation languages and simulators. (At the extreme, a pure simulator requires no programming. It is a special purpose simulation tool that is totally data driven.) However, many simulation language vendors are modifying and extending their products to have the capabilities of simulators, and vice versa. Hence, the classification into these two categories is no longer sufficient. The third subsection discusses the use of simulation software in planning and scheduling. The fourth subsection describes simulation software that is specific to some other application area such as communications or networking. Next, several packages for supporting simulation are discussed.

The fourth major section describes simulation environments. An environment performs all of the support services, eliminating the need for third-party software. For example, there is an input procedure that eliminates the need for a data analyzer or text editor, there is an output analyzer, animation is conducted within the environment, and models, or portions thereof, can be reused for other purposes. Of many variations in environments, one of these is described. Animation of the simulation has become a very important tool. Vendors implement animation differently. One software package available for animation is discussed which can be used with many different simulation packages and languages. This enables a user to contemplate the construction of a custom made environment. The last section describes two of the many packages that support simulation.

2 SIMULATION MODELING SOFTWARE

Applications of simulation exist in many arenas as indicated by the papers presented at the Winter Simulation Conference (WSC). The 1992 WSC included applications in the following areas: Manufacturing, material handling, health services, military decision support, natural resources, public services, transportation, and communications, to mention a few.

These simulation applications are usually accomplished with the use of specially developed simulation software. This tutorial describes the software in four subsections. The first of these subsections describes software for general purposes. This type of software can solve almost any discrete simulation problem. In this section, five products, GPSS/H™, GPSS/World™, SIMAN V®, SIMSCRIPT II.5®, and SLAMSYSTEM®, will be discussed to provide a basic familiarity with this type of software.

Similarly, there are many special purpose simulation packages that support manufacturing. A subsection is devoted to this topic with discussion of five of the many packages available including AutoMod™, FACTOR®/AIM™, ProModel® for Windows™, SIMFACTORY®, and WITNESS®. The third subsection is devoted to software for planning and scheduling. Two products are discussed, FACTOR and AutoSched™.

Eight of the many other special-purpose simulation packages are briefly described in one subsection including MODSIM II®, an object oriented language; COMNET II.5®, a network capacity planning and performance prediction package; LANNET II.5™, for local area planning and performance prediction; NETWORK II.5® for computer or communications network analysis; MedModel™ for medical care systems; ServiceModel™ for the service industry; MOGUL™ for modeling computer systems and communications networks; and ISI as an interface for SIMAN.

2.1 General Purpose Software

GPSS/H is a product of Wolverine Software Corporation, Annandale, VA (Smith, Brunner, and Crain, 1992). It provides improvements over GPSS V that had been released by IBM many years earlier. These enhancements include built-in file and screen I/O, use of an arithmetic expression as a block operand, interactive debugger, faster execution, expanded control statement availability, and ampervariables that allow the arithmetic combinations of values used in the simulation. The latest release of GPSS/H is version 2.0. It added a floating point clock, built-in math functions, and built-in random variate generators. Options available include Student GPSS/H, Personal GPSS/H within the 640K memory limit, and GPSS/H 386 providing unlimited model size.

GPSS World™ is a complete redesign of GPSS/PC™ (Cox, 1991). There are five areas where the redesign is evident: Windowing, virtual memory, the modeling language, animation, and hierarchical modeling. In the limited space available in this tutorial, only a fraction of the changes can be described.

1. The windowing features allow an arbitrary number of windows, of various types, to be opened on one or more simulations. Animations and windows may exist side-by-side.
2. Virtual memory essentially removes model and entity size restrictions.
3. The modeling language provides a set of built-in probability distributions and a range of system functions such as animation control. User-defined subroutines can be incorporated into a program. Numerical integration is provided opening the way to continuous simulation.
4. The animation capabilities are enhanced by a separate program Simulation Studio™ (Cox, 1992). It is basically a draw program allowing flexibility in zooming and panning, and in the transformation of objects.
5. The hierarchical modeling feature of GPSS World provides an ability to define submodels, which can then be treated as if they were user-defined GPSS blocks. These abstractions can also be expanded in the other direction. This will promote either top-down or bottom-up modeling.

There is an enhanced memory version of GPSS/PC that is also available. It allows access of up to 32 mb of memory.

SIMSCRIPT II.5, from CACI Products Company, is a language that allows models to be constructed that are either process oriented or event oriented (Russell, 1992). The microcomputer and workstation versions include the SIMGRAPHICS® animation and graphics package. SIMSCRIPT can be used to produce both dynamic and static presentation quality graphics such as histograms, pie charts, bar charts, levels of meters and dials and time plots of variables. Animation of the simulation output is also constructed using SIMGRAPHICS. SIMGRAPHICS can be used also to produce interactive graphical front-ends or forms for entering model input data. An input form may include such graphical elements as menu bars with pull-down menus, text or data boxes, and buttons that are clicked on with a mouse to select an alternative. The graphical model front-end allows for a certain set of modifications to the model to be made without programming, facilitating model use by those that are not programmers.

SIMAN V, a product of Systems Modeling Corporation, is a general-purpose program for modeling discrete and/or continuous systems (Banks, Burnette, Jones, and Kozloski, forthcoming). The program distinguishes between the system model and the experiment frame. The system model defines components of the environment such as machines, queues, transporters and their interrelationships. The experiment frame describes the conditions under which the simulation is conducted including machine capacities and speeds, and types of statistics to be collected. "What-if" questions can usually be asked through changing the experiment frame rather than by changing the model definition. Some important aspects of SIMAN V are as follows:

1. Special features that are useful in modeling manufacturing systems including the ability to describe environments as workcenters (stations) and the ability to define a sequence for moving entities through the system.
2. Constructs that enable the modeling of material handling systems including accumulating and non-accumulating conveyors, transporters, and guided vehicles.
3. An interactive run controller that permits breakpoints, watches, and other execution control procedures.
4. The Arena environment that includes menu-driven point-and-click procedures for constructing the SIMAN V model and experiment, animation of the model using Cinema, the Input Processor that assists in fitting distributions to data, and the Output Processor that can be used to obtain confidence intervals, histograms,

correlograms, and so on. (More aspects of the Arena environment are discussed later in this tutorial.)

5. Portability of the model across all types of computers.

SLAMSYSTEM, a product of Pritsker Corporation, is an integrated simulation system for PCs based on the Microsoft® Windows™ interface (under MS-DOS) or the OS/2® Presentation Manager™ (O'Reilly and Ryan, 1992). All features are accessible through pull-down menus and dialog boxes, and are selected from the SLAMSYSTEM Executive Window. A SLAMSYSTEM project consists of one or more scenarios, each of which represents an alternative system configuration. A 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 should be performed, and allows the user to accomplish these tasks before the next function is requested. Since SLAMSYSTEM is a Windows application, multiple tasks may be performed in parallel while the simulation is operating in the background.

Some of the features of SLAMSYSTEM are as follows:

1. Models may be built using a graphical network builder or text editor. Using the text editor a network symbol is selected with the mouse, then a form is completed specifying the parameters for that symbol. The Windows clipboard allows many other operations such as grouping one or more symbols and placing them elsewhere on the network.

2. Output analysis includes a "report browser" that allows alternative text outputs to be compared side-by-side. Output may be viewed in the form of bar charts, histograms, pie charts, and plots. Output from multiple scenarios can be displayed at the same time in bar chart form. Using the Windows environment, multiple output windows can be opened simultaneously.

3. Animations are created under Windows using the Facility Builder to design the static background and the Script Builder to specify which animation actions should occur when a particular simulation event occurs. Animations can be performed either concurrently or in a post-processing mode. Two screens can be updated simultaneously and up to 225 screens can be swapped into memory during an animation.

4. SLAMSYSTEM was designed to be used in an integrated manner. For example, historic data may be read to drive the simulation. CAD drawings may be loaded. Output charts and plots created by SLAMSYSTEM may be exported via the Windows or Presentation Manager clipboard to other applications.

The newest release of SLAMSYSTEM is Version 4.0. Some of its unique features include the following:

1. Multiple networks in a single scenario. Networks can be constructed in sections and combined at run time. The sections can be reused in future models.

2. New output graphics. These graphics support 3-D, X-Y grids and displaying of point plot data.

3. Direct interface to SimStat (discussed later in this tutorial). These files may be loaded for advanced statistical analysis.

4. OS/2 metafiles for graphics. The OS/2 metafile format can be read for animation backgrounds or icons.

2.2 Manufacturing Oriented Software

The software discussed in this section is limited to those associated with manufacturing and further to only five within that category including SIMFACTORY II.5, ProModel\Windows, AutoMod, FACTOR/AIM, and WITNESS. References for these software packages include the following: Goble (1991) for SIMFACTORY II.5, Harrell and Tumay (1992) for ProModel, Norman (1992) for AutoMod, Lilegdon (1992) for FACTOR/AIM, and Murgiano (1991) for WITNESS.

SIMFACTORY II.5, from CACI Products Company, is a factory simulator written in SIMSCRIPT II.5 for engineers who are not full time simulation analysts. The current version is Release 6.0 operating under Microsoft 3.1. A system amenable to SIMFACTORY II.5 can be modeled rapidly. A model is best constructed in stages by first defining the layout consisting of processing stations, buffers, receiving areas and transportation paths, defining the products, then the resources, next the transporters, and finally the interruptions. The animation automatically follows from the definition of the model. These model elements are pulled from a pallet rather than a menu bar. The resulting model may be changed with a text editor. Flexible flow modeling is supported. For example, OR logic may be used (as in Request Part A OR Part B).

The layout is created by positioning icons, selected from a library, on the screen. As each icon is positioned, characteristics describing it are entered. The products are defined by process plans that define the operations performed on each part and the duration of that operation.

Resources are added to the model in two steps. First, the resource is defined and its quantity is set. Second, the stations requiring specified resources are identified. While resources are moving, simulation time can elapse. Resource requirements are flexible, for example, one unit of Resource A and two units of Resource B can be requested.

Transporters may be batch movers such as fork lifts or they may be conveyors. Characteristics of a transporter are specified (pickup speed, delivery speed, load time, unload time, and capacity of a fork lift, as an example). The transporter path is identified on the screen. Transporters can avoid each other by collision detection and they can carry resources.

Any interruption, planned or unplanned, can be applied to any model element or group of elements (e.g., conveyors, queues, resources, and transporters). Interruptions can require any combination of resources.

Reports are available concerning equipment utilization, throughput, product makespan, and buffer utilization. Multiple business graphics (pie charts, histograms and plots) can be compared at the same time. Data can be compared across multiple runs. Text reports can be customized, and specific statistics can be collected on the elements of interest. A summary report of all replications provides means, standard deviations, and confidence intervals on the model output.

An optional interface to SIMFACTORY II.5 is SIMPROCESS®. It allows easier modeling of business re-engineering processes.

ProModel for Windows, a product of PROMODEL Corporation, has programming features within the environment, and the capability to add C or PASCAL type subroutines to a model. Some of the features of ProModel for Windows are as follows:

1. Models are created using a point and click approach. Intuitive interfaces, interactive dialogs and online help are provided. An auto-build feature guides the user through the model building process.

2. The software operates in the Windows and OS/2 environments taking advantage of memory management techniques, synchronized windowing and data exchange. Windows fonts, printer drivers, cooperative multitasking, and the Dynamic Link Library are available.

3. Virtually unlimited model size is offered.

4. The simulator offers a 2-D graphics editor with scaling, rotating, and so on. Icons can be defined using either vector based or pixel graphics. These icons are saved as bitmaps at runtime for fast animation during the simulation.

5. CAD drawings can be imported as well as process information and schedules. Customized output reports and spreadsheet files can be produced. If the data is another Windows application, cutting and pasting can be accomplished.

6. The static and dynamic elements of the animation are developed while defining the model.

7. Business output graphics may be produced and sent to a laser printer.

8. Only standard hardware is required (IBM or compatible with VGA graphics). No special graphics card, monitor, or math coprocessor chip is needed.

9. Preprogrammed constructs are provided. This allows for fast modeling of multi-unit and multi-capacity locations, shared and mobile resources, downtimes, shifts, and so on.

10. Automatic statistics are available.

11. Submodels allow the creation of a library of templates of work steps, activities, or sub-processes that can be reused. This allows for model construction to be performed by a team with later merger of submodels into one model.

12. A free runtime capability is provided.

A model is constructed by defining a route for a part or parts, defining the capacities of each of the locations along the route, defining additional resources such as operators or fixtures, defining the transporters, scheduling the part arrivals, and specifying the simulation parameters. The software then prompts the user to define the layout and the dynamic elements in the simulation.

AutoMod, from AutoSimulations, Inc., contains general programming features including the specification of processes, resources, loads, queues, and variables. Processes are specified in terms of traffic limits, input and output connections, and itineraries. Resources are specified in terms of their capacity, processing time, MTBF, and MTTR. Loads are defined by their shape and size, their attributes, generation rates, generation limits, and start times, as well as their priority.

The simulator is very powerful in its description of material handling systems. AGVs, conveyors, bridge cranes, AS/RSs, and power and free devices can be defined. The range of definition is extensive. For example, an AGV can be defined in terms of the following: Multiple vehicle types, multiple capacity vehicles, path options (unidirectional or bidirectional), variable speed paths, control points, flexible control and scheduling rules, arbitrary blocking geometries, automatic shortest-distance routing, and vehicle procedures.

Numerous control statements are available. For example, process control statements include If-Then-Else, While-Do, Do-Until, Wait-Until, and Wait For. Load control, resource control and other statements are also available. C functions may be defined by the user. Attributes and variables may be specified.

The animation capabilities include true 3-D graphics, rotation and tilting, to mention a few. A CAD-like drawing utility is used to construct the model. Business graphics can be generated.

The latest release, Version 7.0, contains a simulator within AutoMod. Features of the simulator include its spreadsheet interface. This eliminates the need for programming in building basic models. The spreadsheet interface also allows the definition of models outside of AutoMod.

WITNESS, a product of AT&T ISTEL, contains many elements for discrete-part manufacturing. For example, machines can be single, batch, production, assembly, multi-station, or multi-cycle. Conveyors can be accumulating or non-accumulating. Options exist for labor, vehicles, tracks, and shifts. WITNESS also contains elements for continuous processing including processors, tanks, and pipes.

Variables and attributes may be specified. Part arrivals may be scheduled using a file. Distributions and functions can be used for specifying operation times and for other purposes. Machine downtime can be scheduled on the basis of operations, busy time, or available time. Labor is a resource that can be pre-empted, use a priority system, and be scheduled based on current model conditions.

Track and vehicle logic allow requests for certain types of jobs, vehicle acceleration and deceleration, park when idle, and change destinations dynamically. Many types of routing logic are possible in addition to the standard push and pull. For example, If-Then-Else conditions may be specified.

Simulation actions may employ programming constructs such as For-Next, While-End, and GoTo-Label. The user can look at an element at any time and determine the status of a part.

Reporting capabilities include dynamic on-screen information about machines and elements. Reports may be exported to spreadsheet software.

Tools within the language include access to the model database through C language, arithmetic and logical operators, save current status of model, built-in status functions and many others. In addition, all of the above mentioned features can be enhanced through the use of C language.

An animation is constructed along with the model definition. This animation and statistical feedback can be turned on or off during any run. Many changes to the model may be made at any time.

The latest release of the software is Version 5. Its features include tool bars for commonly used menu selections. There is a new prototype window to which an element can be saved. Whenever the element is needed in the model, a click on the element in the prototype window allows it to be placed in the model at the dsired location. CAD files may be imported. Models may be commented on-screen using a new layering feature. There is also a capability to change terminology from manufacturing to other environments.

AIM (Analyzer for Improving Manufacturing), a product of Pritsker Corporation, is one of three applications in the **FACTOR** Finite Capacity Management System. The other two applications are for schedule development and schedule management. The current release of AIM is version 5.3 All three applications use the same data base. (The schedule development application is discussed later.)

AIM models are OS/2 based and built graphically with icons that represent machines, operators, conveyors, and so on, placed directly on the screen. The animations are created in a virtual window. During a simulation, the model can be stopped to check its status or add other components, then continue the simulation. Performance data is dynamically updated and displayed while the simulation is running. A dynamic Gantt chart is provided for tracking machine and operator status. Inventory levels and material handling utilization can also be graphed dynamically. Outputs include bar charts, pie charts, and plots of inventory levels. Alternately, information can be transferred to other software for development of presentation graphics.

Features of AIM include the following:

1. Manufacturing representation.

Manufacturing specific modeling components can represent a variety of discrete manufacturing processes. Customized rules may be written to extend the logic of these processes.

2. Integrated with scheduling applications. Models written with AIM can be used with other FACTOR applications providing support for capacity, logistics, production scheduling, and schedule management.

3. Manufacturing data. AIM is built around a relational database that stores the manufacturing operation and simulation output. Part descriptions, process plans, order release schedules, machine locations and schedules, shift schedules, and so on can be transferred from other data sources to the AIM database.

4. Animation support. AIM models are built graphically and are animated automatically.

5. Interactive model building and simulation. Components are located on a scaled facility background. Intelligent defaults are provided for all components. Components are customized by completing forms. During execution, the modeler can change the status of a component and observe the simulated impact on the manufacturing system.

6. Comparison of alternatives. The AIM project framework organizes all aspects of a manufacturing simulation project. Alternative models of the manufacturing process are stored. Comparison reports show model performance data to identify differences between alternatives.

2.3 Planning and Scheduling

A fertile market for simulation-based software is the area of finite capacity planning and scheduling. These products enhance shop-floor performance by increasing the effectiveness of resources and meeting delivery dates. The products are typically integrated with the MRP and/or other CIM systems. The developers of these products are very careful in "hand-shaking" with existing databases. Available scheduling rules can be tried, or tailored rules may be developed.

FACTOR was the first simulation based scheduling product marketed by any of the major vendors. To use FACTOR, a model is built by combining components consisting of order characteristics, shop floor status, production calendar, shift schedules, resources, functional resource groupings, resource maintenance, parts, materials, and process plans (Liligdon, 1992). A simulation of selected alternatives is conducted with results stored in a database which then may become input to the manufacturing system.

One of the major accomplishments of the software is its data integration capabilities. There are two standard interfaces, one from the scheduler's perspective, and the other from the modeler's perspective. Both of these interfaces are tailorable to the user's environment.

In practice, FACTOR is used to schedule operations at regular intervals and to handle unexpected events. At the start of a scheduling interval, a shift for example, status information is transferred into the FACTOR database. The scheduler executes the simulation and reviews a summary of the performance anticipated from the schedule. The scheduler can insert adjustments to the FACTOR generated schedule as desired. The revised model is then simulated, and comparisons are made. This process is continued until a satisfactory schedule results.

If an interruption occurs (arrival of a rush order, breakdown of a major resource, etc.) a new schedule can be generated to meet the new conditions. Schedule adjustment can be accomplished interactively.

AutoSched uses AutoMod as its engine. The following information is required to operate AutoSched:

1. Production resources consisting of workstations, storage locations, operators and tools.

2. Products consisting of parts to be manufactured along with their routings.

3. Production requirements consisting of orders and lots.

4. Operating rules including how tasks are selected and calendars that specify when workstations are unavailable. Much of the information required by AutoSched may already exist in a database such as an MRP. If it does exist, AutoSched can import it.

AutoSched allows task selection rules for each resource. More than one rule (logic filter) is allowed. Although optimal schedules are not guaranteed using these sophisticated rules, significant performance can be achieved.

AutoSched performs a simulation of the shop floor according to the task selection rules provided by the user. Two outputs are provided, graphical and statistical. Business graphs can be created to track any statistic(s). These graphs are updated dynamically. AutoSched also contains an interactive Gantt chart from which an event can be selected for detailing to include the quantity of orders in a workstation's queue when an order was selected, the quantity of orders in the next workstation's queue, etc. Statistical or historical reports are also available. These include the master schedule file, performance report, workstation report, and other user-defined reports.

2.4 Other Simulation Modeling Software

Many additional special purpose software packages exist for simulation modeling. CACI Products Company has a wide array of capabilities, some of which are described in the following paragraphs (Garrison, 1991). PROMODEL Corporation has developed two additional products, MedModel and ServiceModel, and these are also mentioned to broaden the explanation of what is available. Two other products, MOGUL, from High Performance Software Inc., and ISI, from ExTech Ltd., are also described.

MODSIM II is an object-oriented, general purpose simulation language. It is a compiled language that is highly portable. The syntax and structure are based on Modula-2. Built-in, object-oriented constructs include single and multiple inheritance, dynamic binding of objects, polymorphism, encapsulation, data abstraction, and information hiding. The language allows an interface to C so that existing libraries of C source and object code can be included in MODSIM II programs. MODSIM II interfaces with the animator SIMGRAPHICS II.

COMNET II.5 is a capacity planning and performance prediction tool used in the design and ongoing management of LAN, WAN, voice, and data communication networks. The user provides a description of network topology, access protocols, traffic, and routing algorithms. COMNET II.5 simulates the operation of the network, providing measures of performance such as response times, network throughput, node, link, and LAN utilization, end-to-end blocking probabilities, and other traffic statistics. COMNET II.5 does not require programming. An object-oriented, graphical user interface provides an entry into the software. An animated simulation of the network in operation follows the description. Plots and reports are also provided for the analysis of network performance.

LANNET II.5 is a performance prediction and analysis tool for Local Area Networks (LANs). The building blocks of the software are LANs, stations, gateways, and routes. (A station represents a LAN user, i.e., a workstation, a printer. A gateway is a link to other LANs. A route is a list of gateways followed by a destination station.) Traffic is generated by specifications made during the definition of each station building block. Standard LAN protocols such as Ethernet, token ring, and token bus are built-in. User defined protocols can also be modeled. The software has a built-in graphics package. Data is entered interactively through menus and an animated picture of the proposed LAN. In addition to the animation, reports and graphs showing message throughput and utilization statistics for each LAN building block are provided. The traffic pattern and loading statistics are also generated.

NETWORK II.5 is a design tool that accepts a computer or communication system description and provides measures of hardware utilization, software execution, and conflicts. The effects of conflicts and contention are modeled using simulation to investigate the interaction among devices in a system. Text-based and graphics-based menu-driven preproccessors are available. An animated display shows the modeled computer system in operation.

MedModel can be used in a medical environment for patient scheduling, staff planning, personnel scheduling, emergency services planning, and for other purposes. **ServiceModel** is used in the service industry, e.g., banks, insurance companies, schools, offices, and so on.

Built upon GPSS/H as its engine is **MOGUL** (Haigh, 1992). Its purpose is simulation modeling of computer systems and communication networks. MOGUL (MOdel Generator Using Leadthrough) provides an expandable repertoire of processors, peripheral devices, and communication links. The user picks the desired type of object for each processor, device, and link in the model using a series of menus. Activity flow in the model is specified using an interactive editor. A model can be prepared for execution using GPSS/H. After model execution, REGAL™ (REport Generator And Lister) can be used for inspection of selected statistics from the simulated output. Animation of the output is also supported using Proof Animation.

ISI, a product of ExTech Ltd., provides the following:

1. Graphical user interface to SIMAN. ISI assists in, model building, editing of model parameters, experiment definition, code generation, runtime animation, and post processing.

2. Hierarchical modeling capability. Allows the user to develop new functionality based on existing constructs. Higher level building blocks, or macros, can be developed from basic SIMAN

building blocks, which may then be combined to form models.

3. User interaction. A mouse is used for interaction with pull-down menus. The model is synthesized as a network of block functions which are positioned and connected with the mouse. Experiment information is entered in a spreadsheet format.

4. Graphics system. ISI provides a CAD style graphics system with a virtual screen for model building. Zooming and panning are supported.

5. Code generation. SIMAN code is generated automatically. Completeness and syntax checks are invoked. Model execution takes place with a single keystroke.

6. Run-time graphics. The graphical network developed during model building is used for animation showing movement of entities and accumulation of materials in queues.

7. Output processor. The menu-driven output processor allows statistics to be displayed in a variety of business graphics formats.

8. Report preparation. The hard copy facility allows any view of the model or animated run, or any of the output plots to be saved to a file for subsequent printing.

9. Help system. Context-sensitive help is provided on ISI menus. Hypertext techniques allow the user to access help on related topics by highlighting areas of the text.

3 SIMULATION ENVIRONMENTS

A simulation environment contains many utilities to conduct a simulation study. These capabilities include input data analysis, model entry support, scenario management, animation, and output data analysis.

Arena, a product of Systems Modeling Corporation, is intended to provide the power of SIMAN to those for whom learning the language is burdensome and enhance the use of tools used by SIMAN modelers (Pegden and Davis, 1992). Assume that a person, other than a simulation analyst, wants to use SIMAN. Currently, he or she must understand the blocks used in the model and the elements used in the experiment frame to proceed. Under Arena, the user could extract a module, place it in its appropriate location, and parameterize it without learning the SIMAN language. For SIMAN language modelers, Arena is intended to increase their functionality, eliminating the need for writing similar code in different models.

SIMAN is the engine for Arena and Cinema is the animation system that is used. Other products included in Arena are an input processor, and an output processor. A shop-floor analysis capability is also being developed and will be supported by Arena. This latter product will be oriented toward scheduling and real-time shop-floor applications of simulation.

The term "modules" is used to represent the building blocks available for creating models. The most fundamental feature of Arena is that a simulation analyst can construct a module definition for use by others. These module definitions may be combined to create other modules. SIMAN Base Modules form the lowest possible level of modules. These correspond to basic SIMAN modeling constructs (blocks and elements). All other modules, called Derived Modules, are built from Base Modules or other Derived Modules. This increases the speed at which models can be built, and aids in understanding by those not familiar with SIMAN blocks and elements. Templates provide modelers with a domain-specific Module Definition set. For example, a Manufacturing Template could be sold by Systems Modeling or by third parties.

A much revised Cinema is contained within Arena. This animation capability is integrated with Arena modules. For example, when adding a module to represent a manufacturing process, a modeler might get both the modeling logic to represent the process, as well as the Cinema components representing work-in-process, and the status of the resource (busy, idle, in repair, etc.).

4 ANIMATORS

Most simulation animators are integrated with the software. However, this is not always the case, and the introduction of general purpose animation packages allows the use of custom made environments.

Proof Animation is a product of Wolverine Software Corporation (Henricksen and Earle, 1992). Any software that can write ASCII data to a file can drive Proof Animation^{TM}. Thus, BASIC, C, FORTRAN, GPSS/H, SIMAN V, and SIMSCRIPT II.5, among others, can serve as drivers. Animation is accomplished by using a static background, the layout file, and a trace file that contains dynamic events. Some of the features of the software are as follows:

1. Drawing takes place on a coordinate grid using mouse-driven primitives.

2. Moving objects are defined internally

by their geometry.

3. Statistics can be displayed dynamically.
4. Animation occurs in a postprocessing mode.
5. Motion is smooth on EGA and VGA PC's.
6. There is a steady ratio of animated (simulation) time to viewing (wall clock) time. This ratio may be varied while the animation is running.
7. Zoom in and zoom out are supported. Maximum resolution is guaranteed at any scale as the drawing is recalculated.
8. Top view can be changed to isometric and back to top view instantly.
9. An option allows the construction of a demo disk.
10. CAD layouts can be imported and exported through an optional utility.

5 SIMULATION SUPPORT SOFTWARE

Two products, among many that are available, are discussed in this section. The first is UNIFIT® II from Averill M. Law and Associates, for input data modeling (Vincent and Law, 1992). The second is SIMSTAT from MC^2 Analysis Systems for input and output data analysis (Blaisdell and Haddock, 1992).

UNIFIT II is used to model input data distributions. The product can be used in conjunction with the major simulation software in producing the necessary code to enter distributions. The product actually augments the built-in distribution capability of most of the software described previously. For example, GPSS/H has four built-in distributions, but UNIFIT increases that number to 21. UNIFIT can be operated in three different modes. Guided selection mode automatically determines the best fitting distribution. Manual selection mode is designed for experienced simulationists allowing them to select the appropriate statistical tools and the order of application in determining an appropriate distribution. Finally, the no-data model selection mode assists in choosing a source of randomness when no corresponding data exist.

SIMSTAT 2.0 is an interactive graphical software tool that performs statistical analysis on simulation input and output data. It is designed to work seamlessly with many simulation packages. The software uses pull down menus and is integrated into the Windows environment. Data is maintained in a spreadsheet format for editing examination, and analysis. SIMSTAT takes advantage of the Windows Clipboard. Some of the many graphical capabilities of SIMSTAT include the fitting of input distributions to data, the determination of initialization bias, and autocorrelation plots.

6 SUMMARY

This tutorial describes software for simulation organized in four categories. The first of these, simulation modeling software, is broken down into four subcategories. An example of a simulation environment and of an animator are provided. Simulation support software is also described.

REFERENCES

Banks, J., B. Burnette, J.D. Rose, and H. Kozloski. Forthcoming. *SIMAN V and CINEMA V*. New York: John Wiley.

Blaisdell, W.E., and J. Haddock. 1992. SIMSTAT: A tool for simulation analysis. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 421-425. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Cox, S.W. 1991. GPSS World: A brief preview. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B.L. Nelson, W.D. Kelton, G.M. Clark, 59-61. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Cox, S.W. 1992. Simulation Studio™. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 347-351. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Garrison, W.J. 1991. NETWORK II.5, LANNET II.5 and COMNET II.5. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B.L. Nelson, W.D. Kelton, G.M. Clark, 72-76. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Goble, J. 1991. Introduction to SIMFACTORY II.5. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B.L. Nelson, W.D. Kelton, G.M. Clark, 77-80. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Haigh, P.L. 1992. Using MOGUL™ 2.0 to

produce simulation models and animations of complex computer systems and networks. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 400-404. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Harrell, C.R. and K. Tumay. 1992. ProModel tutorial. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 405-409. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Henriksen J.O. and N.J. Earle. 1992. Proof animation: The general purpose animator. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 366-370. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Lilegdon, W.R. 1992. Manufacturing decision making with FACTOR. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 361-365. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Murgiano, C. 1991. A tutorial on WITNESS. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B.L. Nelson, W.D. Kelton, G.M. Clark, 177-179. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Norman, V.B. 1992. AutoMod. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 328-331. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

O'Reilly, J.J. and N.K. Ryan. 1992. Introduction to SLAM II and SLAMSYSTEM. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 352-356. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Pegden C.D. and D.A. Davis. 1992. Arena: A SIMAN/Cinema-based hierarchial modeling system. In *Proceedings of the 1992 Winter Simulation Conference* , ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 390-399. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Russell, E.C. 1992. SIMSCRIPT II.5 and SIMGRAPHICS tutorial. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 323-327 Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Smith, D.S., D.T. Brunner and R.C. Crain. 1992. Building a simulator with GPSS. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 357-360. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

Vincent, S.G., and A.M. Law. 1992. Unifit II: Total support for simulation input modeling. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, 371-376. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

AUTHOR BIOGRAPHY

Jerry Banks is a Professor in the School of Industrial and Systems Engineering at the Georgia Institute of Technology, Atlanta, Georgia. He is the co-author of **Discrete-Event Systems Simulation** published in 1984 by Prentice-Hall and **Getting Started with GPSS/H** published by Wolverine Software Corporation in 1989. He is currently completing the co-authored text **Introduction to SIMAN V and Cinema V** to be published by John Wiley. His areas of research include knowledge-based on-line simulation systems for manufacturing and the validation and verification of large-scale simulation models. He is a founding partner in the simulation consulting firm Carson/Banks & Associates, Inc., located in Atlanta, Georgia. He served for eight years as IIE's representative to the Board of WSC, including two years as Board Chair.