Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

INTRODUCTION TO ARENA™

Michael J. Drevna
Cynthia J. Kasales

Systems Modeling Corporation
The Park Building
504 Beaver Street
Sewickley, Pennsylvania 15143, U.S.A.

ABSTRACT

This paper presents an overview of Arena–an extendible modeling system that is built on SIMAN/Cinema. Arena provides a complete simulation environment that supports all basic steps in a simulation study. The Arena system includes integrated support for input data analysis, model building, interactive execution, animation, and output analysis.

Arena is a graphical modeling/animation system that is based on concepts from object-oriented programming and hierarchical modeling. Arena allows the user to create new modeling modules from basic modeling primitives that consist of SIMAN blocks and elements. These modules can be tailored to a specific problem domain and placed into libraries called templates.

These application-focused templates in areas such as transportation, business process reengineering, high speed manufacturing, health care and many more bring simulation technology much closer to a large class of potential users.

1 INTRODUCTION

The many recent advances in simulation technology have created a greater awareness and use of simulation in industry. Managers are now more aware of the potential benefits of simulation. However, even with the many important advances that have been made over the past several years, there are still many cases where complex systems are being designed and implemented without the benefit of simulation. A very small percentage of systems that could benefit from simulation are actually simulated, and the primary reason for this is the high level of effort required to employ simulation technology successfully. The key to making simulation technology more widely used is to make the tools significantly easier to learn and use without sacrificing modeling power and flexibility.

This paper describes a SIMAN/Cinema-based modeling/animation system, named Arena, developed by Systems Modeling Corporation. This system represents a major advance in simulation technology by combining the modeling power and flexibility of the SIMAN/Cinema system with the ease-of-use of application-focused packages. Arena offers a high level of modeling flexibility across a wide range of problem domains, yet is very simple to learn and use.

The key idea behind Arena is the concept of tailorability to a specific application area. The Arena system is not restricted to a specific set of predefined modeling primitives, but can be easily tailored to a domain-specific application area by means of an application template. Hence a user of Arena who is modeling health care systems could employ a health care template that would contain modeling primitives focused on health care systems (doctors, nurses, beds, X-ray, etc.). Unlike conventional simulation systems that have their modeling primitives "hard-coded" into the software by the vendor, Arena allows the modeling primitives to be "soft-coded" by the end user by means of the template.

The application template concept is fundamental to the flexibility and ease-of-use provided by Arena. This mechanism makes it possible to provide the end user of the product with a tool that closely matches the real system being modeled--hence the user is presented with concepts and terminology that are focused on his or her problem. This dramatically reduces the level of modeling abstraction required by the user. However, the user is not limited to the constructs provided by a single domain-restricted construct set. The user can combine domain-restricted constructs from one or more application-focused templates with the full modeling power of the SIMAN simulation language and thereby avoid the modeling "brick wall" encountered with traditional hard-coded, domain-restricted packages. The modeling power of SIMAN is made available to the user as simply one additional template. Arena is a graphical modeling/animation system that is based on concepts from object-oriented programming and hierarchical modeling. A discussion of these concepts and their relationship to simulation languages and modeling can be found in the 1992 Winter Simulation Conference Proceedings [Pegden, 1992].

2 ARENA BASICS

An entire simulation project may be completed within the Arena system. Arena provides integrated support for input data analysis, model building, interactive execution, animation, execution tracing and verification, and output analysis. Users develop new models in the model window, develop new modeling constructs in the template window, identify data characteristics in the input window, and analyze simulation results in the output window.

2.1 Model Window

The model window is where new models are created, existing models are modified, animations are developed and models are executed. Like all windows in Arena, the model window can be moved, resized, enlarged to full screen, or minimized to an icon. You can also have multiple model windows open in the work area at the same time.

The model window is divided into three main regions: the menu bar region, the panel region, and the workspace region.

The menu bar region provides the user with edit and draw functions similar to those found in popular windowing environments.

The panel bar region is where users attach different groups of application functions referred to as Panels. All Arena model windows contain a Run panel for model execution, reporting and interactive run controller options, a Draw panel for the addition of static graphics or text, and an Animate panel used to add dynamic objects to a model. Users may attach additional panels to their model window. The additional panels are the application-focused templates discussed in Section 3. Each application-focused template contains a number of Modules used for model development. Each Module represents one or more modeling constructs used to represent a component of a system. Modules may include a simple server in a system, transporter characteristics, conveyor length and speeds, arrival processes and a host of system characteristics.

The workspace region is where models and their associated animations are developed. Models are developed by placing and interconnecting modules in the workspace. For example, a model of a simple service system might be built by interconnecting modules from a general purpose modeling template representing an arrival process, service process, and departure process. Modelers may mix modules from multiple application templates to describe the characteristics of a system. Additional animation symbols, graphs and plots may be added into the workspace region by adding objects found in the Animate panel. Background symbols or text are added by selecting objects from the Draw panel. Users move freely between these panels throughout the model development process.

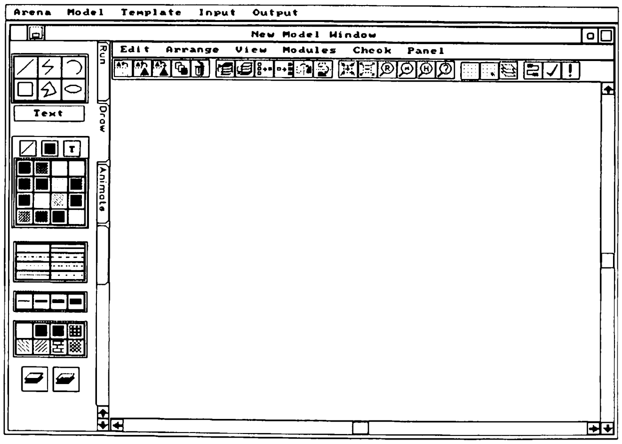

Figure 1: New Model Window

2.2 Template Window

The template window is where new modeling constructs (modules) are created, and existing modules are modified. A template panel contains modules collected into a file and intended to be presented as a self-contained group. A template consists of a panel or set of panels that encompass modeling constructs for a particular application.

A module's definition is created by working with five windows: operand, logic, user view, switch, and panel icon. The template window provides a base from which these five module definition windows are opened.

Three aspects of the module definition are visible to the user of the module: the panel icon, the user view, and the module's dialog boxes and operands. When a template is attached to a model window, the panel icons are displayed. The panel icon window is used to define the icon for a module definition.

After a module has been selected and placed in the model window by the user, an instance is formed and the module's user view is displayed. The user view consists of a module handle (the name of the module, displayed as a text object that opens the module's main dialog box when the modeler double-clicks on it), and may contain connection points, static drawing graphics, and/or animation objects. This user view for a module definition is designed by the template developer from within the user view window.

After a module is placed in a model window, its operands may be edited by the user. The module designer decides which operands are to be presented to the modelers, the default values, and their organization into one or more dialog boxes for the module. A module definition's dialog boxes and the operands contained in them are defined using the operand window.

The two final aspects of a module are hidden from the modeler: the module logic and the definitions of module switches. The logic underlying an Arena module definition is created by building an Arena "submodel" from within the logic window.

Finally in an Arena module definition, individual objects in the user view, operand, and logic windows may be selected to be included in an instance only if a particular condition is true. To define this behavior, objects called switches are created in the module definition. The definition of a switch is based on conditions involving the values of operands. The switch is "attached" to an object in one of the three other definition windows (user view, operand, or logic.)

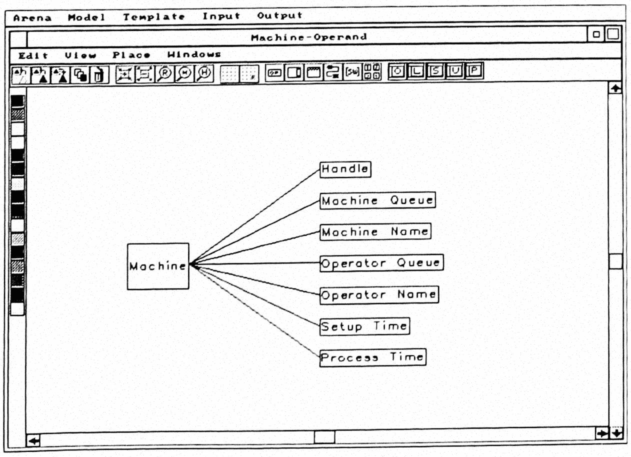

Figure 2: Template Operand Window

2.3 Input Processor

The input processor provides users with the capability of fitting process or performance data to statistical distributions. Users evaluate data downloaded from a file and may display the input data in the form of a histogram. The data may then be fit to one of 14 commonly referenced distributions. Users unfamiliar with the data characteristics may select a Best Fit option that will perform statistical tests and choose the distribution that has the minimum mean square error. Additionally, the expression required as input to the Arena model will be provided. Other options provided allow the user to modify parameters related to the input distribution to identify alternative input scenarios.

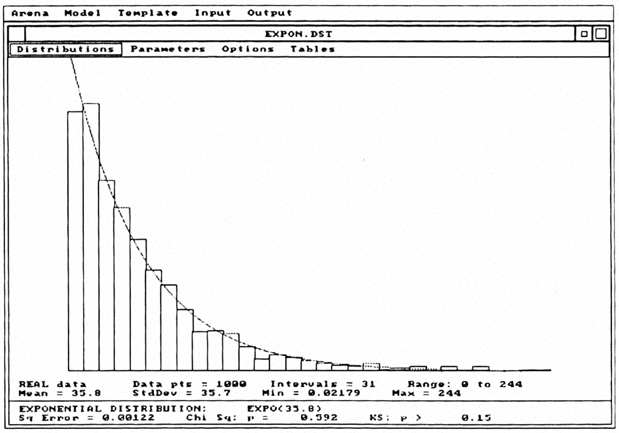

Figure 3: Curve Fitting with the Input Processor

2.4 Output Processor

The output processor provided within Arena allows users to evaluate the statistical reliability of the results generated from their simulation model. Commonly used output data testing procedures such as correlograms, t-tests, data filters, moving averages, analysis of variance, and confidence intervals are included. These tests allow users to define and evaluate system performance or to compare the results from two simulation runs that evaluate different scenarios. The output processor is integrated within Arena such that a user can execute a simulation run, then select output analysis options that will display system performance characteristics.

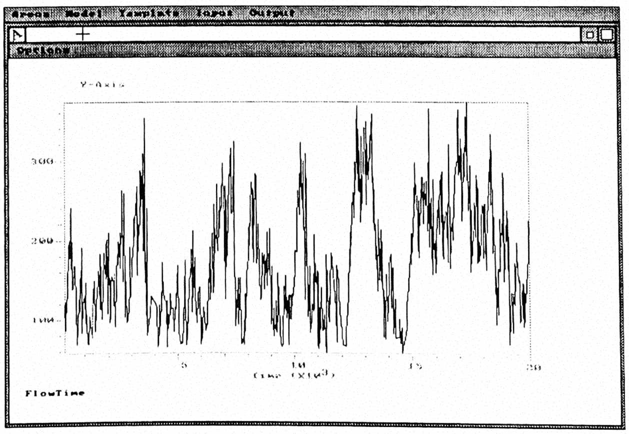

Figure 4: Performance Plot with Output Processor

3 TEMPLATES

The true power of Arena is realized through its support for application-specific templates (ASTs). Numerous application areas have been discussed for possible templates including computer systems, communication systems, traffic flow, airport design, fast food restaurants, high-speed packaging, health care, process industry, package sorting, warehousing, and many more. Each new application template brings simulation technology much closer to a large class of potential users.

3.1 The SIMAN Template

The SIMAN Template contains the base modules that correspond directly to the SIMAN blocks/elements. The SIMAN template contains two panels, the Blocks panel which contains logic modules, and the Elements panel which contains data modules. Users developing a model with the SIMAN Template have full access to the power and flexibility of the SIMAN simulation language. Users develop models by interconnecting modules that relate directly to SIMAN Blocks and Elements. Modules placed in the workspace are detailed by the user who fills in dialog boxes that describe the module characteristics. This method relieves users of FORTRAN or C code-like formatting constructs.

3.2 The Arena Template

The Arena template contains modules that provide general-purpose constructs that are at a higher modeling level than the standard SIMAN blocks/elements. The objective of this template is to provide a comprehensive set of modeling constructs for modeling simple systems across a broad range of applications.

For the new user, the Arena template provides an easier entry into modeling using Arena. By providing the modeler with higher-level modules, the new user can build simple models with less modeling abstraction and, therefore, with greater ease and less learning. Users can combine modules from any of the templates to complete their modeling effort.

Experienced users may develop large models more rapidly with the Arena Template. Since a single module in the Arena Template typically corresponds to several base SIMAN Template modules, large models can be built faster and with fewer errors. The high level modules in the Arena Template can be mixed with lower level SIMAN Template modules to provide detailed modeling capability for complex systems.

The Arena Template is made up of three panel files; the Common panel, the Support panel, and the Transfer panel.

3.2.1 The Common Panel

The Common panel contains two types of modules; logic modules and data modules. Logic modules are used to describe the logical process flow of the system, while data modules describe characteristics of individual components of the system.

The Common panel's eight logic modules can be used to model common processes such as entity arrival, service, and departure. The logic modules in the Common panel are:

Arrive	Depart
Server	Inspect
AdvServer	Process
Enter	Leave

As the name "Common" implies, these modules are the typical modules used when modeling with the Arena Template. For many applications, the Common panel logic modules, with some assistance from the Common panel's data modules, include sufficient functionality and flexibility to describe a system being modeled. You may choose to only use modules from the Common panel when building your simulation model.

The Common panel's eight logic modules are all based around the concept of an entity entering a station (the Enter module), an entity being processed at a station (the Process module), and an entity leaving a station (the Leave module). Further, the functionality of the Enter, Process and Leave modules is combined into one module in the Advanced Server (AdvServer) module.

The Arrive module is used to create entities. The Depart module is used to collect statistics and dispose of entities. The Server and Inspect modules are scaled-down versions of the Advanced Server module, except that the Inspect module also determines whether an entity passes or fails inspection based on a user-defined probability.

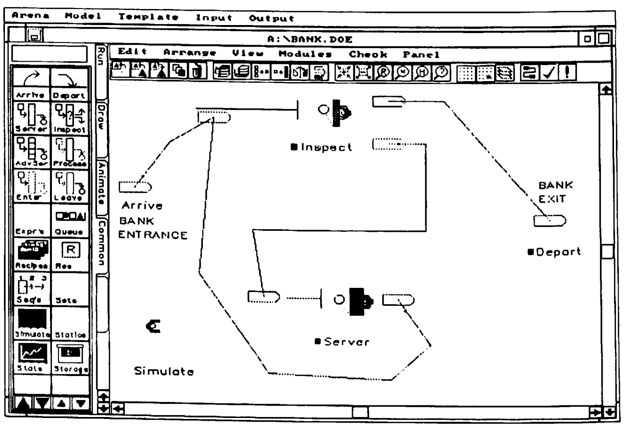

Figure 5: Model Building Using the Common Panel

3.2.2 The Support Panel

All of the modules in the Support panel are logic modules. While the logic modules in the Common panel consist of several functions combined to represent a higher-level process, the modules in the Support panel are a representation of very specific functions. These modules contain the individual components which make up the high level logic modules in the Common panel.

Support panel modules are similar to the modules in the Blocks panel of the SIMAN template except that they provide more functionality. In fact, many of the modules in these two templates have the same name.

3.2.3 The Transfer Panel

The Transfer panel contains modules used to describe the transfer of entities from one area of the system (a station) to another area. There are six data modules and thirteen logic modules in the Transfer panel. Two of the six data modules are used to define transporters and conveyors, respectively. The remaining four modules are used to define data related to the path that a transferred entity follows. All of the logic modules in the Transfer panel are similar to modules in the Blocks panel.

3.3 The Advanced Manufacturing Template

The Advanced Manufacturing Template is a collection of modules that may be combined to describe the process flow of a manufacturing system. The template was designed to support the majority of discrete manufacturing applications--from job shops to flow shops to assembly lines.

The template supports various types of process flow including unconstrained push, constrained push, pull or some combination of the above. The use of production and transfer authorizations enable the template to provide just-in-time (JIT) type pulling capability. The template is also designed to support many types of processes including assembly operations, production operations and batching/unbatching operations.

There are three categories of modules within the Advanced Manufacturing Template: data modules, workcenter modules, and component modules. Data modules allow the user to define specific information related to objects which are referenced in both workcenter modules and component modules (e.g., Machines, Operators, Parts). Workcenter modules describe the logical portions of the manufacturing system. Component modules are a representation of the individual components of a workcenter module. For example there are component modules such as Workcenter entry, Material Handling Unload, Process, and Transfer.

By incorporating the features of the SIMAN language into the template, material handling capability is strongly supported. Both free path and guided transporters, as well as accumulating and non accumulating conveyors, may be used to move parts from one workcenter to the next.

3.4 The Wafer Fabrication Template

The Wafer Fabrication Template is designed to support modeling of wafer fabrication operations in the semiconductor industry. (This template is based on prototype research performed by D. Phillips, G. Curry, and B. Deuermeyer at Texas A&M University, and it was developed with partial funding from SEMATECH).

Models that are built using this template utilize two types of user input, model data and model rules. Model data definitions include products, technologies and jobsteps (process plans), recipes, workcenters, resources, operators, and production schedules. Model rules define the model logic to be used by a particular workcenter or recipe--i.e., model rules define the logic by which manufacturing lots seize and release sets of resources, are batched and split apart, and undergo processing delays. Given the large number of jobsteps in a typical wafer fabrication model, model rules are defined separately from the model data so that they may be re-used.

A unique aspect of the Wafer Fabrication Template is that is it can generate either a standard simulation model or a special model that is used to perform analytical flow and queue analysis of the system. Either of these models can be generated from the same user description of the system--i.e., the user builds a single model of the system and then selects which form of the model (standard or flow and queue) they would like to execute.

The flow and queue analysis provides the same basic performance measures for the system as does the standard simulation model, but uses queuing approximation formulas to obtain the results. The advantage of the flow and queue analysis is that it executes much faster than a simulation model of the same system. The advantage of the simulation model is that it provides more accurate results.

4 SUMMARY

Arena is a hierarchical modeling system based on SIMAN and Cinema. The key feature of Arena is that it allows users to define new modeling constructs (modules) that can be tailored to a specific problem domain. These domain-focused modules can be placed into libraries called templates. The development of application-specific templates for use with Arena creates the exciting opportunity to bring simulation technology to a large cross section of people who currently do not benefit from this technology. Arena brings simulation technology much closer to the application specialists--and does so across an unlimited range of problem domains.

REFERENCES

Collins, N., and C.M. Watson (1993) "Introduction to Arena" in Proceedings of the 1993 Winter Simulation Conference, IEEE, Piscataway, NJ.

Pegden, C.D., R.E. Shannon, and R.P. Sadowski (1990), *Introduction to Simulation Using SIMAN*, McGraw-Hill, New York, NY.

Pegden, C.D., D. Davis (1992), "*Arena: A SIMAN/Cinema-Based Hierarchical Modeling System*" in Proceedings of the 1992 Winter Simulation Conference, J.J. Swain, D. Goldsman, R.C. Crain, J.R. Wilson, Eds. IEEE, Piscataway, NJ.

Profozich, D., and D. Sturrock (1994) "*Introduction to SIMAN/Cinema*" in Proceedings of the 1994 Winter Simulation Conference, IEEE, Piscataway, NJ.

Systems Modeling Corporation (1994), *Arena Professional Edition Reference Guide*, Sewickley, PA.

Systems Modeling Corporation (1994), *Arena Advanced Manufacturing Template Reference Guide*, Sewickley, PA.

Systems Modeling Corporation (1994), *Arena Template Reference Guide*, Sewickley, PA.

Systems Modeling Corporation (1994), *Arena User's Guide*, Sewickley, PA.

Systems Modeling Corporation (1994), *SIMAN V Reference Guide*, Sewickley, PA.

AUTHOR BIOGRAPHIES

MICHAEL J. DREVNA is a Senior Consultant in the Simulation Services Division of Systems Modeling Corporation. He has provided model development, analysis and training support in the wafer fabrication, consumer goods, transportation, high speed packaging and health care industries. He was previously employed by The BDM Corporation and Ford Aerospace. He received a B.S. in 1982 and an M.S. in 1985 in Industrial and Systems Engineering from Ohio University. He is a member of IIE and Alpha Pi Mu.

CYNTHIA J. KASALES is currently a Consultant in the Simulation Services Division of Systems Modeling Corporation. She has developed models in a wide range of industries including electronics, pharmaceutical, wafer fabrication and consumer products. She received a BS in Industrial Engineering from The Pennsylvania State University. Prior to joining the Simulation Services Division, Ms. Kasales was on the Technical Support Staff where her duties included providing customer support, software testing and porting.