Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

AUTOMOD PRODUCT SUITE TUTORIAL
(AutoMod, Simulator, AutoStat)
BY AUTOSIMULATIONS

Matthew Rohrer

AutoSimulations, Inc.
655 Medical Drive
Bountiful, UT 84010, U.S.A.

ABSTRACT

Whether designing a new system or modifying an existing one, engineers want to take the guesswork out of finding the best possible solution. While there are many analysis methods for designing industrial systems, simulation remains the method that gives the highest level of confidence a system will work. A well-written simulation model can be a valuable tool in the design, analysis, and operation of manufacturing and other complex systems.

The AutoMod™ Product Suite from AutoSimulations has been used on 1000s of projects to help engineers and managers make the best decisions possible. AutoMod combines the ease of use of a simulator-type tool with the power and flexibility of a simulation language. As shown in Figure 1, 3-D graphics have been an integral part of the AutoMod product since its inception in 1984.

The main focus of the AutoMod simulation product is on manufacturing and material handling systems. AutoMod is flexible enough, though, to use in other non-manufacturing applications, for example, fast food restaurants, rental car lots, airport ticket counters, and container loading in ports.

With the release of AutoMod version 9.0 in 1999, AutoMod now supports continuous and discrete models. The new Tanks and Pipes module allows users in the process industry to model the details of fluid and bulk material flow.

The AutoMod Product Suite includes:

- AutoMod – discrete and continuous simulation
- Simulator – manufacturing-oriented interface for fast and easy modeling
- AutoStat – statistical analysis including optimization technology
- AutoView – dynamic walk-though of animation with support for AVI file creation
- Model Communications Module (MCM) – for linking models to other programs, like control systems.

This tutorial gives an overview of the AutoMod Product Suite, explaining the technical details of the products and how they interrelate to each other.

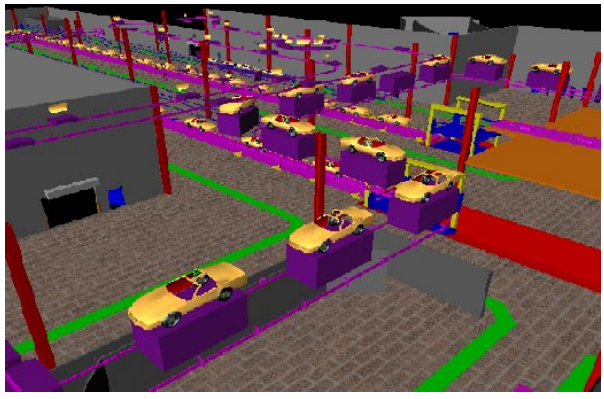

Figure 1: AutoMod Model Graphics in an Automotive Application

1 INTRODUCTION

AutoMod's strength is that it combines the ease of use of a "simulator" with the power and flexibility of a simulation language. Simulators usually address specific real-world problems with predefined constructs, making model definition quick and easy. *AutoMod's* movement systems aid users in defining the movement of material either manually or by automated equipment. Material movement systems include:

- Path Mover (path/vehicle systems such as lift trucks, AGVS, and human movers)
- Conveyors (including belt and roller types)

- Automated Storage and Retrieval Systems (ASRS)
- Robots
- Bridge Cranes
- Power and Free Chain Conveyors

To define material movement systems, the user simply defines movement elements, such as paths and stations, then inputs the operating parameters, such as velocity and acceleration. *AutoMod* then automatically creates the corresponding model logic. 3-D animation is created automatically as well, providing a realistic picture of how a facility will look and operate. Model animation can be viewed from any angle or perspective in real time, providing visualization capabilities unmatched in other simulation tools.

A big difference between *AutoMod* and other simulation software is that users deal with geometry in a graphical way and create logic using a fourth-generation simulation language. This approach is intuitive to engineers who understand how their facilities operate but who do not have experience with simulation tools. Model output includes 3-D animation, automatically collected statistics, and business graphs. *AutoMod* provides users with a rich environment that facilitates thorough understanding and analysis of their systems.

The animation provided in *AutoMod* models is concurrent, meaning that the graphic pictures are running real-time with the simulation model. The model execution environment is very interactive, allowing the user to stop and start the simulation, run without animation in an accelerated time scale, and select objects from the animation screen to gather more-detailed statistics about the system being modeled. Statistics can be viewed at any time during a simulation run. These model execution features make it easier to verify and validate models of complex systems and provide a forum for communication about system performance among project team members.

2 AUTOMOD INTERFACE

An *AutoMod* model consists of one or more systems. A system can either be a process system, in which flow and control logic is defined, or a movement system. Each model must contain one process system and may contain any number of movement systems. Processes can contain complex logic to control the flow of either manufacturing materials or control messages, to contend for resources, or to wait for user-specified times. Loads can move between processes with or without using movement systems.

All inter-arrival and event times can be represented by deterministic values or be derived randomly from one of several statistical distributions. *AutoMod*'s interface is window-oriented, utilizing pop-up and pull-down menus, dialog boxes, selection lists, and a mouse-based editor for developing process logic.

3 AutoMod's World View

Any number of movement systems can be defined in an AutoMod model, and a process system connects the movement systems to the logical flow of products. In the process system, loads (products, parts, etc.) move between processes (locations) and compete for resources (equipment, operators, and queues). The load is the active entity, executing action statements that are connected to the processes. Typical action statements give users the ability to:

- Use machines/operators
- Move into queues
- Clone new loads
- Change load types
- Wait on user-defined delay lists
- Increment/decrement counters
- Set variable values
- Read from data files
- Send to other processes
- Make conditional tests

Figure 2 shows the process system pallet for AutoMod. The pallet is organized in a "top-down" manner with the most commonly defined elements at the top.

Figure 2: AutoMod's Process System Pallet

3.1 Processes

The process system is the backbone of an *AutoMod* model, providing the general-purpose simulation features required to model a wide range of real-world problems. While material movement is important, it is not always critical in manufacturing. No value is added to product while it is being moved around a manufacturing facility. Value-added operations are performed by machines and people. The *AutoMod* process system is where the value-added operations and control logic are defined.

AutoMod's process system includes a simulation language based on action statements. Action statements combine the power of a structured language with the ease of use of English-like, manufacturing-oriented syntax. *AutoMod* models are not limited in any way, so model logic can be of any size and complexity. The power and flexibility of the *AutoMod* language makes is easy to model almost any real-world situation.

3.2 Loads

Loads are the active entities in *AutoMod* and can be created in many ways, including deterministic or probabilistic generation. Their interarrival rate can be read from an external data file or attached to a statistical distribution. *AutoMods has* predefined random distributions that can be used to fit most real-world random events. Loads are given types, such as "RedCar" or "PartA," and they can have attributes such as color, stock keeping unit (SKU), priority, and time in the system. Attributes can be accessed and modified in *AutoMod's* action statements. Loads have 3-D shapes and dimensions like most other entities in *AutoMod*.

3.3 Resources

Resources in *AutoMod* are used to represent machines, operators, fixtures, containers, and any other finite capacity objects. There are two default categories for a resource's state. The first is the working category (busy or idle), and the second is the availability category (up or down). During the animation, state colors indicate the status of each resource. Statistics are automatically collected for every resource in a model.

Loads use resources for specified processing times. These times can be deterministic or probabilistic, using *AutoMod's* built-in statistical distributions to simulate randomness. Processing times are either part-specific or are applied to all parts using a particular resource. Resources can also be preempted if a higher priority load needs immediate attention.

Resources can have downtimes, which are defined by MTBF (mean time between failures) and MTTR (mean time to repair). These times can use *AutoMod's* statistical distributions to represent random failures. Though MTBF is calculated by model-simulated time by default, it can also be based on parts processed or machine running time.

Resource cycles can be created and attached to specific resources to indicate when or how often events such as random failures or PM's occur for a resource.

States

States other than the default states can be defined by the user. These states might be used to represent conditions such as blocked, starved, PM, offline, etc. The state of a resource can then be changed in *AutoMod*'s procedural code so that statistics can be tracked for each state. In addition, state monitors can be defined and used to track states for entities other then resources, such as vehicles, conveyors, or particular areas of a facility.

3.4 Queues and Order Lists

Queues in *AutoMod* are both graphical and statistical elements, and they can have capacities ranging from one to infinity. When a queue has reached its capacity, the next load trying to enter that queue must wait until there is space available. Queue contents can be shown dynamically in the animation, and loads can be "stacked" in any direction in the queue.

Loads that are at a queue or process may be sorted or delayed until they are explicitly ordered to leave. To determine which action should take place next, the loads must place themselves on order lists. An order list is not a physical entity like a queue, but a logical element that provides a way to sort loads that have been delayed for any reason. To remove a load from an order list, another load must execute an order action. Loads can be ordered to move to another process or order list, or to simply continue where they left off in their processing. Order lists can be sorted by load priority or other load attributes, either in ascending or descending order.

3.6 Blocks

Blocks control the number of entities occupying a physical space, making them very useful for controlling path mover vehicles. Blocks may have any capacity from one to infinity. Path mover vehicles (lift trucks, AGVs, Electrified Monorails, etc.) and loads increment blocks automatically when moving through the physical space defined by the block. Loads can also claim blocks in process logic, as directed by the user. Blocks can have any shape, including combinations of the AutoMod-supported shapes (discussed in section 5).

3.7 Variables and Counters

Data may be stored in an *AutoMod* model using variables. Variable values are changed using the "set" action as follows:

```
set Setup_time to 123.456
```

Variables can be used in calculations or can be compared to other variables. In addition to integer, real, and string types, variables can also be used to store references to other process system entities, such as processes, queues, resources, order lists, counters, loads, and locations. Storing these references gives the users more power and flexibility to expand and extend models to match modifications of the actual system. *AutoMod* also supports the notion of arrayed entities, making it easier to model real-world systems that have some dimensionality, for example a warehouse with similar operations on two floors.

Counters are similar to variables, except they can have finite capacity and they can only be positive integer values. Counters have a maximum capacity, making them useful for traffic control. When a load tries to increment a counter that is at its capacity, the load will be delayed until another load decrements the counter. Statistics for counters are collected automatically.

3.8 Functions, Subroutines, and Source Files

Functions and subroutines are used to create modular models. This allows the models to be extended more easily. Users can define their own functions in the AutoMod or C languages, and these functions can be called from anywhere in the AutoMod model. Subroutines help eliminate duplication of action statements in the model.

Source files contain the logic for the model. Any number of source files may be defined, and they are processed differently depending on their file extension. For example, ".m" files contain AutoMod logic and will be checked by AutoMod for correctness when edited. Files with a ".c" extension will be compiled with the model. Users may also define other source files to contain input data or documentation for the model.

3.9 Labels

AutoMod provides the ability to use text labels to enhance model understanding. Labels may be added to any location in the model's physical space, and they can either be static or dynamic during the simulation. Labels can rotate when the animation view changes or can be attached to a fixed position on the screen.

3.10 Tables

Tables in *AutoMod* supply the user with the means to collect statistics on any model entity and to classify those statistics for better understanding of their distribution. Tables automatically provide average, standard deviation, maximum, and minimum for all values entered in the table. Users can define the number of "bins" or categories and *AutoMod* adds values to the appropriate bin when directed by the user in the tabulate action statement.

3.11 Types and Random Streams

Types in *AutoMod* provide users with a powerful tool for creating, among other things, lists of entities such as resources or stations, and then making complex decisions based on the contents of the lists. Some of the default types are integer, real, resource, location, etc. But with user-defined types, it is possible to create, for example, a variable of type ResourceList, whose data can then be manipulated in any number of ways to facilitate complex decision making.

Random streams may be defined to give each element of randomness independence from other elements. The number of different random streams used by AutoMod models is without limit, and AutoMod supports both the Linear Conguential Generator (LCG) and the Tausworth random number generators.

3.12 Run Control

Run control in *AutoMod* allows users to define the warm-up and steady-state periods for the model by resetting time-persistent statistics. Reports can be printed for any run control period, or "snap." Business graph output can be automatically created. Post-processed animation periods used by AutoView, an animation extension (discused in section 4), are also defined in the run control. Run control also provides an entity tracing capability that gives the model builder an event-by-event account of the model run. This information is useful in verifying and validating a model.

3.13 Business Graphics

Graphs in *AutoMod* are easy to define and they update in real time with the animation. Graph types include bar charts, pie charts, and timelines. Figure 3 shows a typical timeline business graph. Any model entity can be attached to a graph, including transporter vehicle velocity, number of loads on a conveyor section, or average utilization of a machine. Graphs can be printed or plotted to a variety of supported output devices, and graph displays can be controlled using the *AutoMod* language.

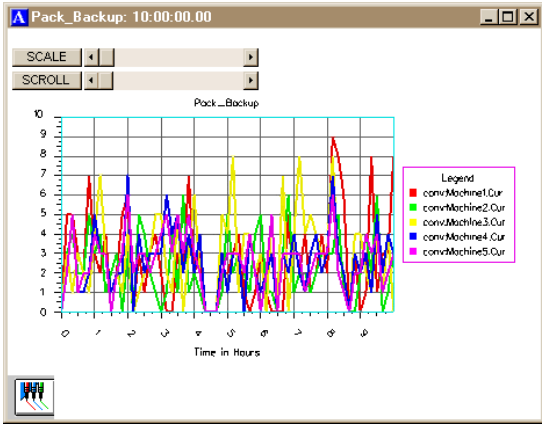

Figure 3: AutoMod Timeline Business Graph

4 AUTOVIEW

AutoSimulations' post-processed animation extension can be used to view the animation records created by running an *AutoMod* model. The AutoView™ product allows the user to pre-define all views and time periods in the model animation, then play them back to generate presentation-quality animation files.

AutoView also has single-frame capture capability to MPEG or AVI format files. This feature allows users to create animations that can be shared with others without the need for any additional software. Also, the single-frame capability helps smooth the animation on larger models where hardware constraints may affect animation performance.

5 GRAPHICS IN 3-D

Both dynamic and static objects can be displayed during model execution. Figure 1 shows a screen shot of a typical AutoMod model during a model run. Dynamic objects represent loads, vehicles, resources, queues, and statistics. The static layout is the background graphics of the plant, such as columns, aisle markings, and walls. Labels can identify specific areas in the facility.

There are several ways to create a layout of the system to be modeled. *AutoMod* comes with a three-dimensional graphics editor that allows the user to construct objects from standard graphics primitives. Cone, Box, Hemisphere, Trapezoid, Frustum, Cylinder, Arc, Vector (list), Set, Text, and Triad are primitives that can be selected, placed, and scaled to create any static entity in the facility.

AutoMod has two utilities: one that allow users to import IGES format files directly into the *AutoMod* model building environment, and another that creates extra graphic elements like legs on conveyors and racks for ASRS systems.

6 RUN-TIME ENVIRONMENT

In keeping with *AutoMod's* interactive features, the user has complete control of the model in the run-time environment. The model can be viewed with the animation on or run with the animation off. *AutoMod* uses concurrent animation; the simulation progresses as the animation picture is being updated. With animation off, the simulation doesn't draw the picture, but it still performs all simulation calculations. The user can suspend the simulation at any instant to review statistics through pop-up windows, to take resources down, to set break points or alarms, or to control the view of the animation without constraint.

6.1 View Control

AutoMod provides a comprehensive, flexible, and easy-to-use method of interacting with a model during model execution. If the simulation project is in the experimentation phase where only parameter changes are made and the model needs to be re-run several times, *AutoMod* provides the ability to run in batch mode without animation. Whether the need is for a highly interactive mode or a batch mode, *AutoMod* lets you do it.

6.2 User Interaction

AutoMod provides advanced debugging and trace facilities. A model can be single-stepped at any time during the animation. Also, the ability to set breakpoints and alarms allows the user to suspend the simulation when a certain event occurs or when a specific clock time is reached.

AutoMod also provides comprehensive reports. The reports can be displayed on request at any time during the animation. Printed versions of the reports can also be specified during the Model Development Environment.

AutoMod automatically keeps track of many statistics. These automatic reports are linked to specific entity types, such as:

- Movement systems,
- Processes,
- Queues,
- Resources,
- Order Lists, etc.

Vehicle states are tracked during the entire model run, and reports are generated automatically. Reports can be sorted alphabetically or numerically for easier analysis. The user can also develop and generate custom reports from within process procedures.

7 AUTOMOD'S SIMULATOR INTERFACE

In addition to the standard interface described earlier, AutoMod also includes a manufacturing template called the simulator. The simulator is included with AutoMod, and it can be used to model a certain class of manufacturing problems quickly and easily.

AutoMod's Simulator can be used for classes of problems where:

- Machines are grouped by capability. For example, in a machine shop one might have sets of lathes, mills, and drill presses that can perform the same operation.
- Parts have complex routings, including alternate steps, setups, and batching.
- Product is manufactured to customer order.

Figure 4 shows the main menu for the Simulator.

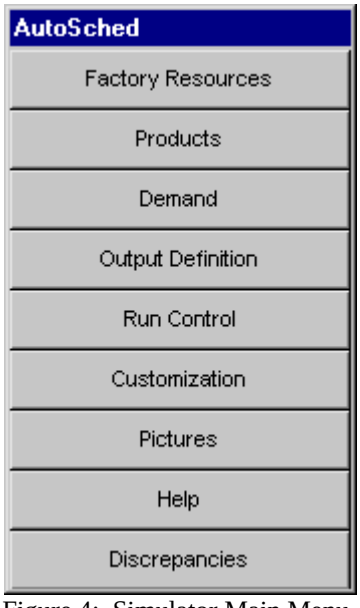

Figure 4: Simulator Main Menu

7.1 Factory Resources

Factory resources include the equipment, operators, fixtures, and any other item required for the manufacturing of products. Machines are grouped into families that can perform similar operations. When a machine finishes the current operation, it looks at all the products available to process, and makes a decision about the next thing to do. This decision is called a scheduling rule. The Simulator allows the users to select from among many standard rules, like first-in-first-out, highest priority, or same setup.

7.2 Products

Products are the items that are being produced in the Simulator model. Products have a routing that defines the operations required to manufacture them. The routing defines families of machines to visit, and can also include alternated steps, setup requirements, processing time, and batching requirements. A bill of materials can also be defined for sub-assemblies, which can be either produced or purchased.

7.3 Demand

Demand in the Simulator represents the customer orders. In manufacturing, many decisions need to be made, including how to allocate resources, what each operation should do next, and when to "launch" new orders into the facility. The demand file in the Simulator defines each customer order, start date and time, and expected due date. Reports from the simulator indicate which orders met their due dates and which were late.

8 AUTOSTAT

The AutoStat™ product is the statistical analysis product in the AutoMod product family. AutoStat does extensive output analysis on an AutoMod model, including:

- Optimization using evolution strategies
- Warmup determination
- Design of experiments
- Confidence intervals
- Sensitivity analysis
- Factor-response analysis

AutoStat also has support for running scenarios across a network of machines. Support for multiple machines and CPUs give users the ability to make many more runs of the simulation than was possible before.

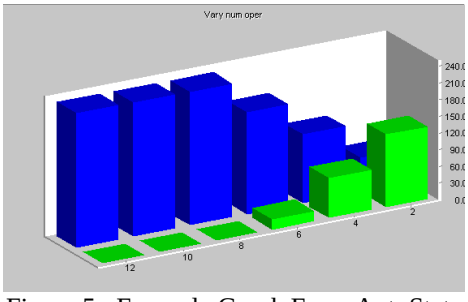

Figure 5: Example Graph From AutoStat

The evolution strategies algorithm used by AutoStat is well suited to finding the optimum solution without getting "trapped" at a local optimal value. Figure 6 shows the

output from an optimization run. Across the X axis are the number of generations required to find the optimum, and the Y axis gives the values for the user-defined fitness function. The fitness function can be defined as any combination of model inputs and outputs, with user-defined weighting factors applied to each value.

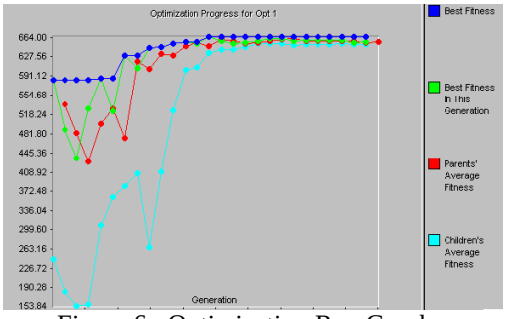

Figure 6: Optimization Run Graph

9 MODEL COMMUNICATIONS MODULE

The Model Communications Module (MCM) is an extension to AutoMod that allows simulation models to communicate with the outside world. Some of the applications of MCM include communications between:

- Two or more AutoMod models
- Control systems and AutoMod models
- Other applications and AutoMod models

Communication between two or more models allows for parallel and distributed simulation. MCM also include Multi-Model Sync (MMS), which keeps the event lists of the linked models synchronized.

Control systems can be linked to models using MCM. This allows control system designers to test the logic using the simulation prior to going into the field. This application is typically called emulation, and can save substantial time in debugging automation controls.

Models may be linked to other applications. Examples of this include Excel spreadsheets, ergonomics programs, or Programmable Logic Controller (PLC) emulators.

10 SUMMARY

AutoMod is an industrial-oriented simulation system that provides the ability to define the physical elements of a system using CAD-like graphics and the logical portion of the system using a powerful procedural language. The results are that a typical user can be three to ten times more productive using *AutoMod* in comparison to using any other simulation language. The accuracy and degree of detail with respect to model development is unequalled.

AutoMod allows the construction of very large, complex models. In fact, *AutoMod* has a structured language that has proven that the larger the project, the more benefits *AutoMod* has over alternative approaches.

AutoMod provides real, three-dimensional graphic animation. There are no limits to the views or the size of the picture to be shown. The degree of animation realism is also unmatched, as *AutoMod* provides light-sourced solid graphics with Z-depth sorting showing all entities in correct relation to one another on the screen.

The AutoMod product family, including the AutoStat, AutoView, and MCM extensions, provides a comprehensive solution to the needs of the simulation user. Through the use of AutoSimulation's technologies, models can become more than design tools, being used to test controls, operate the facility, and plan for expansion.

REFERENCES

AutoSimulations, Inc. 1999. *AutoMod User's Manual*

AutoSimulations, Inc. 1999. *AutoMod Beginning Class Notes*

Royce O Bowden and John D. Hall (1998), "Simulation Optimization Research and Development," In *1998 Winter Simulation Conference Proceedings*, pp. 1693-1698

AUTHOR BIOGRAPHY

MATT ROHRER, Vice President of Simulation, joined AutoSimulations, Inc. in 1988. Over the past 11years, Mr. Rohrer has been a simulation analyst, has managed the AutoSimulations Consulting Group, and has guided the development of AutoMod. He has been active in the simulation community during his career. Mr. Rohrer wrote a chapter in the *Handbook of Simulation* by Dr. Jerry Banks.