Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

AUTOMOD

Van B. Norman

AutoSimulations
655 Medical Drive
Bountiful, Utah 84011, U.S.A.

1. ABSTRACT

AutoMod is an industrial simulation system that combines CAD-like drawing tools with a powerful, engineering oriented language to model control logic and material flow. Unlike most other simulation languages, AutoMod's powerful graphical interface accurately capture the physical constraints of distance, size, and space, resulting in accurate 3 dimensional detail.

2. INTRODUCTION

AutoMod provides the user with a set of "expert based" movement systems that have been developed from AutoSimulations Inc.'s (ASI) experience in industrial automation. As a result, the underlying model logic is automatically generated for the user from the geometry and the movement system input parameters. The 3-D animation is an automatic part of model development and exactly depicts the logic of the model.

AutoMod differs significantly from other simulation systems because of its ability to deal with the physical elements of a system in physical (graphical) terms and the logical elements of a system in logical terms. AutoMod offers advanced features to allow users to simulate complex movement (kinematics and velocity) of equipment such as robots, machine tools, transfer lines, and special machinery. All graphics are represented in three-dimensional space with unlimited viewing control including: translation, rotation, scale, light-sourced solids, perspective, and continuous motion viewing.

AutoMod consists of two programs. The build portion is for the physical and logical model definition. After the user has defined the physical and logical components of the model, it is then compiled into an executable model, where the simulation and animation run concurrently. The executable model is fully interactive; it can be stopped at any instant in simulated time to examine statistics and model status and to conduct interactive modelling experiments.

3. MODEL DEVELOPMENT ENVIRONMENT

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

Loads that flow through the process logic have the ability to claim and release resources, enter and leave queues, be added and removed from order lists, change the value of variables, counters, and load attributes, create a new load or kill an existing load, read from and write to external files, and determine the next process. All interarrival and event times can be represented by deterministic values or be derived ramdomly 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.

4. PROCESS SYSTEM

The process system is the backbone of AutoMod and provides the general purpose simulation features required for simulation. While material movement is a necessity, it is not the most important element in manufacturing. Value is not added to a product by transporting it around the plant or mill. The value added operations in manufacturing are performed by the machines, processes, and labor that exist in the facility. AutoMod's process system is where the value added manufacturing operations and the control logic are simulated.

AutoMod's process system is both powerful and easy to use. AutoMod's programming procedures provide state-of-the-art compiler technology, with English-like syntax that is manufacturing oriented. There are virtually no bounds to the size and complexity of the logic that can be developed. There is rarely a need to drop down into a lower level language because AutoMod provides the user with the flexibility required to simulate any task required.

Process logic is defined in process procedures. Process procedures use an easy-to-use language that contains:

- if-then-else logic
- while loops
- access to global and load-specific variables
- actions to use, take down or bring up resources
- actions to multiply loads
- actions to choose processes, resources or queues based on their state

```
begin
  move into FixQ
    if FixedYet = 1 then send to die
      else
        begin
          choose a resource from among Picket (1),
          Picker (2)
          whose current loads is minimum
          save as RecIdle
          use RecIdle for uniform 10, 5 min
          set FixedYet to 1
          send to SizeWay
        end
end
```

Figure 1. Sample Procedure Logic

4.1 Loads

Loads are the active elements in AutoMod. They are created either from a creation specification using one of the standard statistical distributions, deterministically from reading data from a file, or based on user defined distributions. Loads are named and may have user defined attributes. These attributes are variables which are unique to the load. For example, a load that represents a car might have load attributes that indicate what color it is, what level of trim it has or whether it receives air conditioning. These attributes change as a result of the model logic in the process procedures. Like most everything else in AutoMod, loads also have 3D shapes and physical dimensions .

4.2 Resources

A resource is a general and flexible entity that can be used to represent a machine, an operator, a fixture, a container, etc. Often several resources are used in a process in a similar fashion. There are two levels in which the resource state is categorized. The first level is whether the resource is Busy or Idle (its state with respect to a load). The second level is the resource's availability, whether it is Up or Down.

Loads use resources for specified processing times, which can be based on a standard time with variations based on several random number distributions that are built-in to AutoMod, or on custom distributions. The processing times can be general for all loads or specific to each product type.

Resources can have downtimes. During the downtime period, the resource accumulates statistics in the down state. When the resource becomes available, it continues to work on the preempted load for the remaining processing time. Mean-Time-Between-Failure (MTBF) and Mean-Time-To-Repair (MTTR) are included and can be based on the same built-in statistical functions or on custom curves. The standard MTBF is based on model simulated time. AutoMod can easily accommodate MTBF based on machine run-time or machine cycles.

4.3 Queues and Order Lists

When loads are modeled they must always reside in a physical space. AutoMod uses two types of physical space: movement systems and queues. Queues have capacities which can range from 1 to infinity. If a queue is full (it has reached its capacity), the next load must wait until there is room. Loads within queues can be sorted and sequenced by using order lists. The queue contents can be shown dynamically during the animation.

Loads may be sorted and delayed at a process or queue until they are explicitly ordered to leave. A load can be directed to place itself on an order list. An order list is not a physical element but a way of sorting loads that are delayed for some reason. Once on an order list, the load still remains in the process and physical territory it was in prior to the order list inclusion.

The load can be ordered from that list by another load or from an order action in another part of the model. The load which has been ordered can be sent to another process by the ordering action or it may continue on its way.

Order lists are not attached to specific processes. Many processes may place loads on the same order list. Likewise, when a load is ordered to a process, that process has no control of where it is coming from.

4.4 Variables and Counters

AutoMod provides a number of ways of storing values during the simulation period. Variables are data structures that can change as a result of a load executing the appropriate statement in a process procedure. Variables can be used in calculations or logically compared to other variables for any means useful to the modeler. In addition to integer and real numeric values, variables can also represent:

- Process Names
- Queue Names
- Resource Names

- Order List Names
- Counter Names
- Load Names

By loading the name of an AutoMod entity into the variable, extensive if-then-else logic can be avoided, for example, by sending a load to a variable that has the process name loaded into it.

Counters are similar to variables but are positive integers only and use a maximum capacity. A load trying to increment a counter already at its maximum capacity will be stopped until it can successfully perform the increment. Statistics are kept on counters throughout the simulated period.

4.5 Blocks and Traffic Limits

AutoMod has powerful means of controlling the number of loads that can be either in processes or within physical space in the facility. Traffic limits prevent too many loads within a process, while blocks provide the same utility for physical space. Blocks are often used to control such things as AGV collisions.

Blocks are like counters in that they use a set limit—they cannot be incremented beyond the limit. Normally the limit is one so only one entity can occupy the physical space at a time. Blocks are commonly used as AGV intersections to prevent vehicles from colliding.

AGV's automatically increment/decrement blocks when entering/leaving blocks. Blocks can also be incremented and decremented from process procedures to prevent, for example, having a bridge crane and an AGV collide.

4.6 Run Control

By using run control features, various experimental runs can be compared. Run control defines the length of the simulation (in simulated time units), when reports are printed to a file, and when statistics are reset. Resetting the statistics allow the model to run for an initialization or priming period prior to running the model in a "steady state" period from which statistics are again gathered.

5. PICTURE CONSTRUCTION IN 3-D

Both dynamic and static objects can be displayed during model execution. Dynamic objects represent loads, resources, queues, and statistics.

The static layout is the background graphics of the plant. It may contain column lines, aisle markings, and walls. Labels can identify specific areas in the facility.

There are several ways to create a layout of the system that is 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 also has an optional utility called IGES/Sim. The acronym "IGES" stands for the Initial Graphics Exchange Standard. IGES is an industry standard exchange format for translating the graphic data from one CAD system to another. Any IGES file of a plant layout that was created from a CAD system can be easily imported into AutoMod through ASI's IGES/Sim utility. IGES/Sim can also export AUTOMOD graphics files to the IGES format.

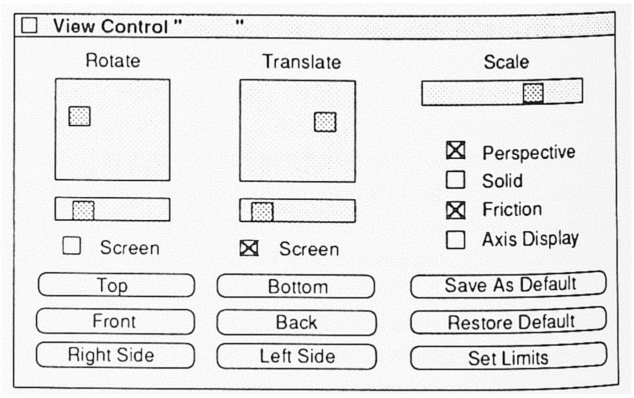
6. RUN TIME ENVIRONMENT

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 still performs all simulation calculations. The user can suspend the simulation at any instant and review statistics through pop-up windows, take resources down, set break points or alarms, and 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.

Figure 2 shows the view control window. This



□ Screen

☒ Screen

☒ Perspective

□ Solid

☒ Friction

□ Axis Display

Figure 2 - View Control Window

window provides the user with the ability to position the graphics of the model in any orientation desired. The View Control allows rotation, translation, and scale to be dynamically changed with respect to all three axes (X, Y, and Z). The animation picture can be shown in solid mode with all the correct Z sorting, so that hidden lines and surfaces are accurately represented. The animation picture can be shown in either perspective mode or orthographic mode. The friction toggle in View Control allows the user to spin or translate the model picture continuously. This is a helpful feature when the model's animation is being video taped or filmed.

6.2 USER INTERACTION

AutoMod provides advanced debug and trace facilities. The model can be single stepped at any time during the animation. Also, the ability to set breakpoints and alarms allow 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 model build.

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.

Reports can be sorted alphabetically or numerically for easier analysis. The user can also develop and generate custom reports from within process procedures.

7. SUMMARY

AutoMod is a industrial oriented simulation system that provides the ability to define the physical elements of a system using CAD-like graphics and to define the logical portion of the system using a powerful procedural language. The results are that a typical user can be between 3 to 10 times more productive using AutoMod in comparison to using any other simulation language. The accuracy and degree of detail with respect to movement systems is unapproached.

AutoMod allows the construction of very large, complex models. In fact, AutoMod's structured language 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, so all entities are shown in the correct relation to one another on the screen.

Enhancing AutoMod's already robust capabilities are the following extensions and utilities:

- AutoSched - Provides a powerful, fully featured finite-capacity planning and scheduling system.
- AutoView - AutoMods's post process animation package that allow you to create a directed "walk through of the model by panning, zooming and moving back and forth in thime and space.
- AutoStat - Provides enhanced analysis of the statis tics generated by AutoMod by calculating mini mums, maximums, confidence intervals and steady state information.

REFERENCES

AutoSimulations, Inc. 1989, *AutoMod User's Manual.*

AutoSimulations, Inc. 1989, *AutoMod Lessons Guide.*

AutoSimulations, Inc. 1989, *AutoMod II with Kinematics User's Manual.*

AUTHOR BIOGRAPHY

VAN B. NORMAN, President of AutoSimulations, Inc. received a B.S. in mathmatics at the University of Utah in 1969. He spent 6 years at Eaton-Kenway, where he implemented the first simulation animator. In 1982, using his experience in factory automation and simulation, he co-founded AutoSimulations, Inc., where he co-authored AutoMod, ASI's first graphic simulation software. He has authored papers on the application and the future of simulation in manufacturing. His interests are in world-class manufacturing operations and simulation research.