Proceedings of the 1989 Winter Simulation Conference
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

AutoMod II: THE SYSTEM BUILDER

Michael B. Thompson
AutoSimulations, Inc.
655 Medical Dr.
Bountiful, UT. 84010

ABSTRACT

AutoMod II (AM II) is an industrial simulation system that combines CAD-like drawing capabilities used to input movement systems with a powerful, engineering oriented, procedural language for the definition of control logic and material flow. AM II's CAD features are used to define the physical geometry of manufacturing, material handling, storage, and distribution systems. Unlike most other simulation packages, AM II's powerful graphical interface accurately captures the physical constraints of distance, size, and space. AM II also provides the user with a set of "expert based" movement systems that have been developed from ASI's real-world experience in industrial automation. As a result, a substantial portion of the underlying model logic is generated for the user from the graphics and the movement system input parameters. Furthermore, once the model is built, the 3-D animation is built as well, with no further work on the modeller's part. AM II's user environment provides unmatched productivity.

AM II is an Engineering Workstation product that runs on Silicon Graphics, Intergraph, SUN, and DEC under the UNIX™ operating system. ASI has recently introduced an entry level version called AutoMod^{2e} that runs on the Macintosh II under the Mac/OS.

1.0 INTRODUCTION

Manufacturing is the process of transforming raw materials into the form of a product. Whether the product is an automobile, a nylon stocking, a box of breakfast cereal, a silicon wafer, or a fan belt, there are activities that occur that are common to all manufacturing systems. Taking a very simple view, you can think of manufacturing as a specific sequence of three basic activities: process, movement, and storage. This applies well to the physical world of manufacturing as raw materials are transformed into products.

Simulation modellers have appliedt general purpose computer languages, general purpose simulation languages, and manufacturing oriented simulation languages to these problems for at least 20 years. In most, if not all cases, these three basic activities of process, movement, and storage have been thought of, and modeled almost exclusively, as time delays such as: product movement from the Wash Station to Central Q.C. takes 5.5 minutes; the facing and turning operation for the gear blank takes 47 seconds per piece. The time delay information for a simulation model is normally calculated from physical parameters and attributes of the facility and the equipment within. In many cases, averages are calculated and assumptions are made as a result of converting from physical parameters such as velocity, size, and distance into time delays.

Besides the activities that directly apply to product materials, there are other activities that apply to industrial systems that are less tangible but are no less important in today's sophisticated manufacturing environment. As these systems become more automated, the necessity to understand and represent information flow and control logic is as important as correct representation of material flow. These activities can have a physical component, in the sense of "where" control messages are generated, but in most cases the control logic is not physical. Thus, AM II's procedural language exists for control logic definition.

AM II 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. AM II also offers advanced features to allow users to simulate complex kinematics and velocity of equipment such as robots, machine tools, transfer lines, and special machinery. All graphics are accurately represented in three-dimensional space with unlimited viewing control including: translation, rotation, scale, light sourced solids, perspective, and continuous motion viewing.

AM II is really 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 runs concurrently. The executable model is a fully interactive stand-alone program. The model can be stopped at any instant in simulated time to view statistics and model status.

The subsequent sections of this paper give a more detailed description of the hardware requirements, model development and run-time environments.

2.0 HARDWARE

ASI supports AutoMod II on the following platforms:

AutoMod II	Silicon Graphics31xx, 4D/xx Intergraph Clipper 3xx, 3xxx Sun-4 DEC VAXstation 3200 and 3500 and microVAX GPX
AutoMod^{2e}	Macintosh II, IIx, IIcx

AutoMod II requires UNIX (Ultrix for DEC), a "C" compiler, a minimum of 8MB of memory, 12 graphics bitplanes, and a color monitor.

AutoMod^{2e} requires the Mac/OS, 4 MB of memory, an 8 bit video card, a color monitor, and a "C" compiler.

3.0 AUTOMOD II MODEL DEVELOPMENT ENVIRONMENT

The world-view of AM II consists of a *model* and one or more systems. A *system* can either be a *process system*, where the control logic is defined, or a *movement system*. Each model must contain one process system and any number of movement systems. *Loads* are units of traffic that move between one or more *processes* in the process system. Processes can contain arbitrarily complex logic to control the flow of either manufacturing materials or control messages. Loads are temporary--they enter the system at some time, contend for resources, are modified, and then leave. Loads can move between processes with or without the use of movement systems. Normally, loads that use movement systems represent manufacturing materials that move directly from one process to another. Loads that do not use movement systems are usually control signals or logical messages. A load within a process, flows through the process logic and has 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 which process it goes to next. All interarrival and event times can be represented by deterministic values or be derived from one of several pseudo-random statistical distributions.

3.1 USER INTERFACE

The user interface is mouse driven, utilizing card stack pop-up menus, multiple windows that can be moved and re-sized, dialog boxes, select lists, and a high performance editor for developing process procedures. AM II uses a three-button mouse. The right button activates the pop-up menus, the center button is used for selecting or choosing options, and the left button is for zooming and graphics view position.

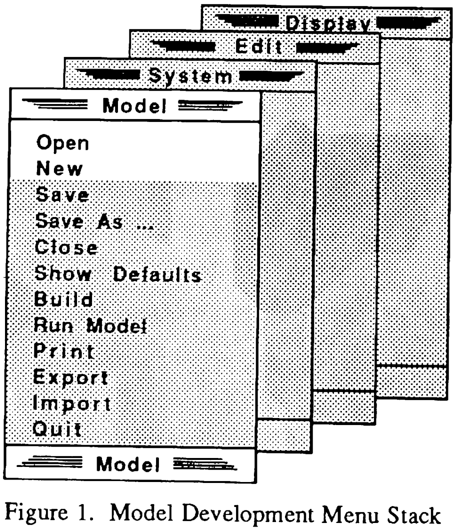


Figure 1. Model Development Menu Stack

3.2 THE LAYOUT

There are several ways to create a layout of the system that is to be modeled. AM II comes with a utility called ACE (AutoSimulations' Creation Editor). ACE is a three-dimensional graphics editor that allows the user to construct objects from standard graphics primitives. Cone, Box, Hemisphere, Trapezoid, Frustrum, Cylinder, Arc, Vector (list), Set, Text, and Triad are primitives that can be selected, placed, and scaled to create the static layout of the facility. The user has the ability to specify drawing attributes such as line thickness, color, solid fill, or wire frame display. Dynamic entities such as vehicles, loads, robots, machine tools, bridge cranes, and SRMs can be defined in ACE such that AM II can simulate and animate their kinematics.

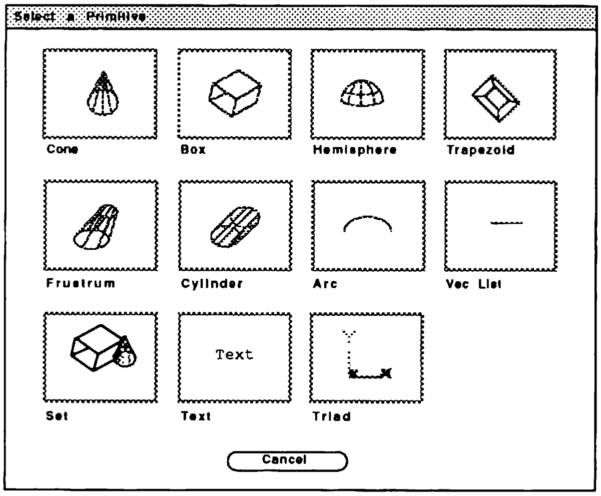


Figure 2. ACE Primitives

AM II has an optional utility called IGES/Sim. The acronym "IGES" stands for the Initial Graphics Exchange Standard. This standard was designed as an exchange format for translating the graphics representation from one CAD system to another. Any IGES file of a plant layout that was created from a CAD system can be imported into AM II through AutoSimulaitons' IGES/Sim utility. IGES/Sim can also export graphics files in the IGES format.

3.3 MOVEMENT SYSTEMS

A movement system in AM II is a special purpose system that moves loads between processes. AutoSimulations has invested man decades of effort to accurately represent the behavior of industrial movement systems. Movement systems are not procedural. The user defines the geometry of the movement system using AM II's CAD functions. The user is prompted to "fill in the blanks" for the appropriate parameters for each movement system. AM II's expert-system logic for each system creates the correct model logic to accurately represent the system behavior based on the users specifications. AM II offers the following movement systems:

- Conveyor
- AGVS
- ASRS
- Bridge Crane
- Power and Free Conveyor
- The Kinematic Extension

Conveyor. A conveyor system consists of one or more conveyor sections which loads move along according to section-specific attributes. Loads move from section to section at transfer points. Some of the rules controlling transfers depend on the attributes of the section the load is moving to or from; other transfer rules are dependent on the values of system wide defaults. AM II correctly represents both accumulating and non-accumulating conveyor. The leading and trailing edges are tracked based on the user-defined load size. AM II's CAD features offer numerous ways of defining the geometry of conveyor sections. Stations (points at which the conveyor physically interfaces with a process) can be placed anywhere on a conveyor system.

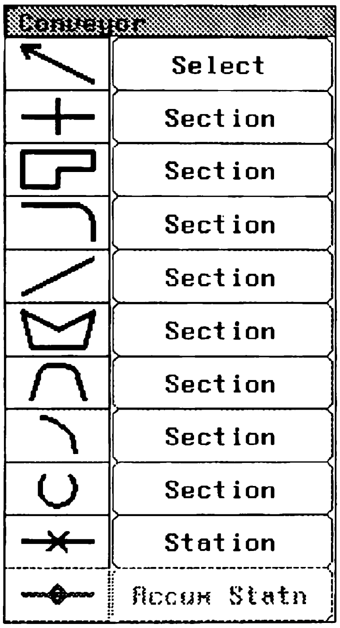


Figure 3. Conveyor Palette

AM II also represents load windows. A window is the total amount of space that a load needs on a conveyor. In the real world, rarely is the window size identical to the load size. The window is generally composed of three lengths:

a) The distance the conveyor moves while a load is transferring on the conveyor.
b) The load size.
c) the distance the conveyor moves while the load transfers off.

Figure 4 shows an example of a load waiting to transfer onto a section. It will have to wait until a large enough space between the loads (window) is available before the load can make the transfer.

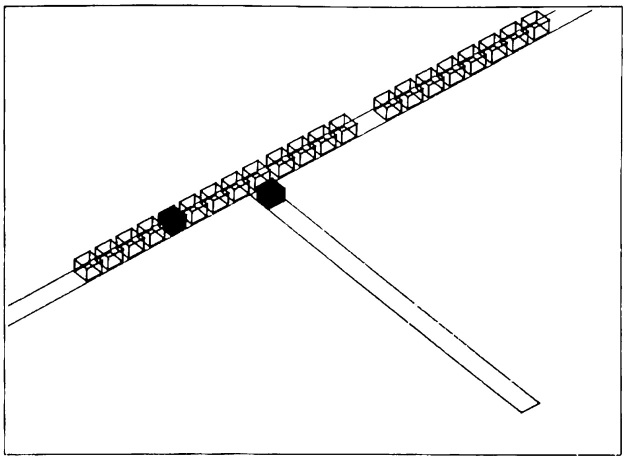
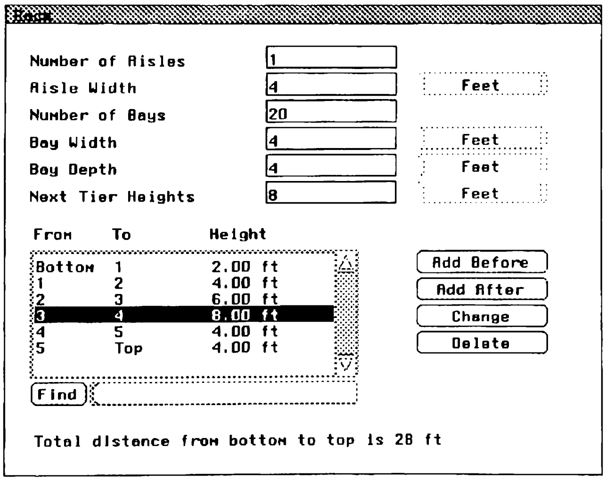


Figure 4. Conveyor Window

AGVS. An Automatic Guided Vehicle System (AGVS) is a movement system in which driverless computer controlled vehicles move along a guidepath carrying loads between pickup and delivery stands as they are directed by the host computer. When developing a simulation model of an AGV system, the modeller is concerned with movement control, collision avoidance, and assigning tasks to empty vehicles. AM II provides powerful but easy to use features to describe guidepath, control points, work search algorithms, and park rules. AM II correctly represents vehicle acceleration and deceleration, it automatically calculates the shortest path between control points, and provides very sophisticated blocking and collision avoidance features. No other package simulates AGVs with the skill and accuracy of AM II. The majority of the major AGV vendors in the United States use ASI technology to simulate AGV systems.

ASRS. An Automated Storage and Retrieval System (ASRS) is a set of one or more aisles of storage racks in which Storage/Retrieval Machines (SRMs) pick-up and retrieve loads. Storage racks are arranged along parallel aisles with each aisle being served by an SRM. The SRM



Rack

Number of Aisles 1
Aisle Width 4 Feet
Number of Bays 20
Bay Width 4 Feet
Bay Depth 4 Feet
Next Tier Heights 8 Feet

From	To	Height
Bottom	1	2.00 ft
1	2	4.00 ft
2	3	6.00 ft
3	4	8.00 ft
4	5	4.00 ft
5	Top	4.00 ft

Total distance from bottom to top is 28 ft

Figure 5. ASRS Parameters

can move vertically and horizontally at the same time. AM II treats an ASRS as a movement system. It accurately represents the ability of the SRMs to move loads to and from zones of bins and to and from pick-up and delivery stands. Coupled with processes, an AM II ASRS system can represent both the movement and the rack utilization of the ASRS system. Much like the parameters that have been explained for the previous movement systems, the ASRS definition is divided into the CAD features for the physical description of the rack, as well as a fill-in-the-blank approach for the parameters such as acceleration, deceleration, velocity, creep speed, creep distance, fine position time, pick-up time, and set-down time.

Bridge Cranes. A Bridge Crane is a set of rails on which a crane moves over a series of user defined pick-up and delivery areas. The area bounded by the rails and the travel of the bridge is divided into bays and tiers--they define the locations where the crane can pick up and deliver loads. The modeller builds a Bridge Crane movement system by graphically placing the bridge and specifying its attributes. Crane movements are specified with a detailed battery of parameters that include separate X and Y values for accelerations, decelerations, velocities, creep speeds, creep distances, and fin position times, as well as pick up time and set down time.

Power and Free Conveyor. A Power and Free system is similar to both the AGV and the conveyor systems. The track (called a "chain") is laid out just like conveyor path, while the carriers behave similarly to AGV vehicles. Carriers travel around the chain on "dogs" that hang down at set intervals. When a carrier stops in the track, the carriers behind stop too. The chain continues to move. Carriers carry loads around the system. In fact, a load is not considered to be in the power and free system until it is on a carrier. Dogs are always moving at the same speed as the chain, and therefore carriers are instantly traveling the chain speed when they are engaged (picked up by the dogs). There is a certain amount of time required for a carrier to transfer to a new chain. This transfer time is modelled in AM II as the time required for a carrier to drop off a dog and be moved into position to pick up the next available dog on the destination chain.

Carriers cannot pass one another. Whenever a carrier is stopped along a power and free section, the carriers behind stop as they run into the already-stopped carriers based on the accumulation spacing rules defined in AM II. The chain does not stop when individual carriers stop.

As with other movement systems, the definition of a power and free system is a combination of inputting graphics and specifying parameters which include section geometry, dog spacing, chain velocity, transfer times, transfer spaces, and carrier accumulation rules, to name a few.

The Kinematic Extension. The Kinematic Extension is an optional movement system within AM II. It lets you simulate any device that has motion. Using ACE (see section 3.2) a user can define three dimensional components and can inter-connect them with joints--either rotational or translational. These joints allow a user to create machines that have vertical traverses (movement up and down), radial traverses (movement in and out), rotational traverses (circular movement), as well as linear traverses (movement along a track). A joint (motion axis) is defined by its joint speed, motion limits (how far can this rotational joint pivot--90°? 180°?) and other dynamic parameters. This means that machines move at the same speeds and with the same motions that they have in the real world. They are not just a pretty picture.

The Kinematic extension system enhances the power and flexibility of AM II by adding the ability to integrate machine tasks directly into a system model. Because of the way kinematic objects are built, they take the same time to perform a task in the simulation as they would in the real world. Integration is not just graphical--it is also functional. These moving shapes can also enhance other movement systems. For example, an AGV could be built that has a multi-axis robot arm on the payload deck. The robot arm and the AGV can be synchronized so that the AGV correctly positions itself, the robot arm reaches out and picks up a load off of the dock, the robot arm retracts, and then the AGV takes off for the destination.

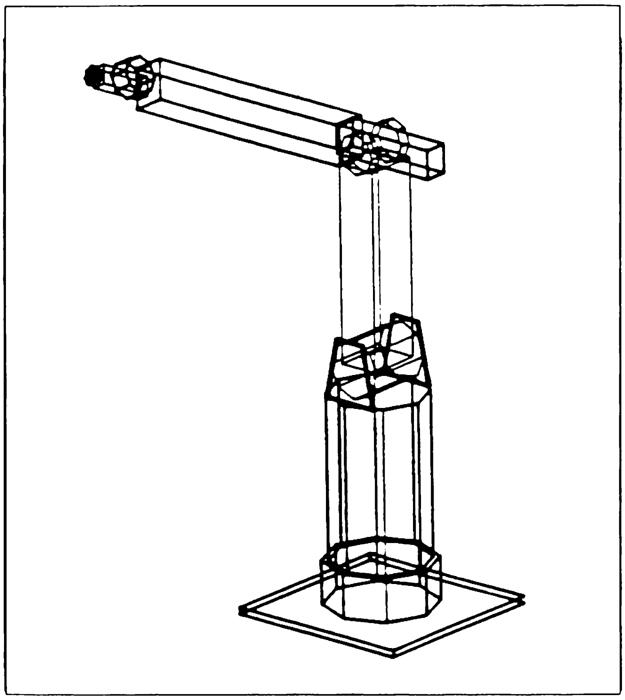


Figure 6. Robot Example

MOVEMENT SYSTEM SUMMARY

AM II's movement systems are unmatched in the simulation software market. At first glance it may seem that AM II's movement systems are specific and inflexible. Not true. ASI acknowledges that there are material movement systems that exist in the real world that do not exactly exist in AM II. However, the six movement systems that do exist are so powerful and flexible that treating a non-explicit movement system as one or a combination of the basic six makes simulating most any movement system very easy. For example: Human driven fork trucks are often represented as AGVs with a flexible guide path (set to the null color in the animation), multiple capacity control points, and virtually no blocking. Carousels are often represented as multi-level conveyor with some additional control logic to mimic the synchronization and bidirectional nature of the device.

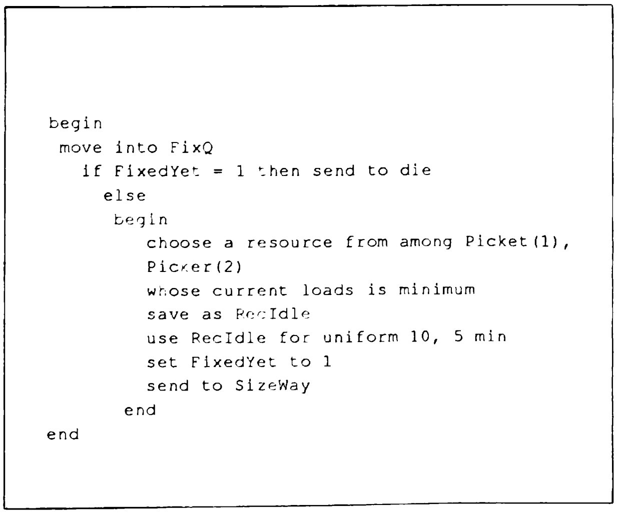
It is far easier for AM II to go from the very accurate and sophisticated simulation detail to the general and simple detail than it is for other simulation systems to use their gross approximations to accurately simulate sophisticated automation.

3.4 PROCESS SYSTEM

The process system is the backbone of AM II. 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. AM II's process system is where the value added manufacturing operations and the control logic is simulated.

The process system consists of one or more processes. A process should not necessarily be thought of as a physical location. A process represents either a location on a movement system, or is used strictly for logical control.

A process can have logic associated with it within what is called a process procedure. The procedural language used is Pascal-like with special simulation features. It contains if-then-else loops, while loops, both global variables, and variables private to a specific load, actions to capture and free, or take down and bring up resources, actions to multiply or split (clone) loads as well as many items that are internally tracked for comprehensive statistical output. For example, a modeller can check, at any place in the procedure , the number of loads that are currently in any process, any queue, any resource, and so on; the status of a resource, whether it is up or down, busy or idle, and so on.



```
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 7. Example Procedure

There are basic activities that are performed in a process when a load enters the process and executes the process procedure. The following entities will be covered in more detail:

- Loads
- Resources
- Queues
- Order Lists
- Blocks/Traffic Limits
- Variables/Counters
- Run Control

Loads. Loads are the active elements in AM II. They are created either from a creation specification, deterministically from reading data from a file, or they are created as a copy of another load. Loads have attributes-- variables private to an individual load. For example, a load that represents a car body might have load attributes that indicates what color it is, what level of trim it has, whether it is to receive air conditioning, and so on. These attributes change as a result of the model logic. Like most everything else in AM II, loads also have shapes. The default shape is a rectangle with user-defined dimensions. Sophisticated three-dimensional shapes that were created originally in ACE or that are imported through IGES/Sim can also be used as loads.

Resources. A resource is a general and flexible entity that can be used to represent a machine, an operator, tooling, a fixture, a container, etc.. Likewise, there are two levels in which the resource state is categorized. The first level is whether it is Busy or Idle; its state with respect to a load. The second level is the resource's availability (Up or Down). AM II automatically accumulates statistics about resources.

Loads claim resources and use them for either a deterministic or a stochastic amount of time. These times can be calculated, read from an external file or data base, or be provided by AM II as an observation from a statistical distribution.

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.

Queues. Loads are always in a physical space or territory. When a load is not on a movement system it is in a queue. Queues have capacities which range from 1 to infinity. If a queue is full, the next load trying to enter cannot until there is room. Loads within queues can be sequenced within process procedures. The queue contents can be shown dynamically during the animation. Comprehensive statistics are kept automatically on queues.

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

- by another load
- from an order action.

It can be sent to:

- another process by the ordering action

or

- 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.

Order list statistics are kept.

Blocks and Traffic Limits. AM II controls the number of loads that can be either in a process or within physical space in the factory. Traffic limits prevent too many loads to be in a process, while blocks provide the same utility for physical space. Blocks are often used to control vehicle congestion.

Variables and Counters. AM II provides a number of ways to store 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. Calculations can be performed, then logically compared to other variables.

Counters are similar to variables with the exception that counters can have a maximum capacity. When a load tries to increment a counter above its max capacity, that load is stopped until it can successfully perform the increment. Statistics are kept on counters throughout the simulated period.

Run Control. The length of the simulation as well as report options (the frequency of reports printed automatically to a file, and times the statistics are to be reset) are run control options. They are all explicitly defined in the build environment.

PROCESS SUMMARY

AM II's process system is powerful and easy to use. It requires programming to simulate complex control logic. There is virtually no bounds to the size and complexity of the logic that can be developed. It is state-of-the-art compiler technology, with English-like syntax that is manufacturing oriented.

There is no need to drop down into a lower level language because AM II provides the user with the flexibility required to simulate any task required.

4.0 RUN TIME ENVIRONMENT

A summary word for AM II's run time environment is exciting. The user has complete control of the model. You can decide if you want to view the animation or let the model run with animation off. AM II 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 the 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.

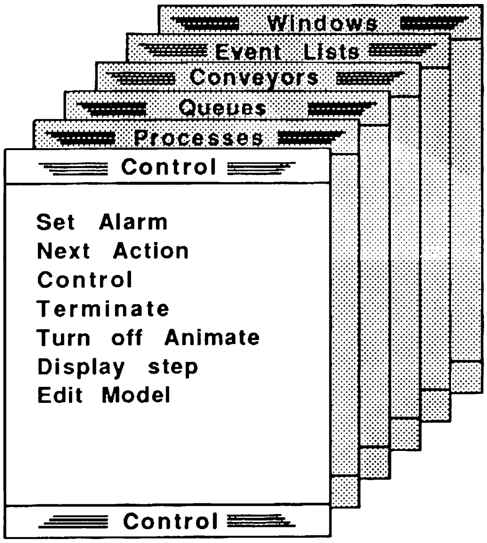


Figure 8.0 Run Time Menu Stack

In the run-time environment the user interacts with the model in the same manner as was explained in the model development environment. There is a pop-up menu hierarchy that allows the user to manipulate the features of the model execution.

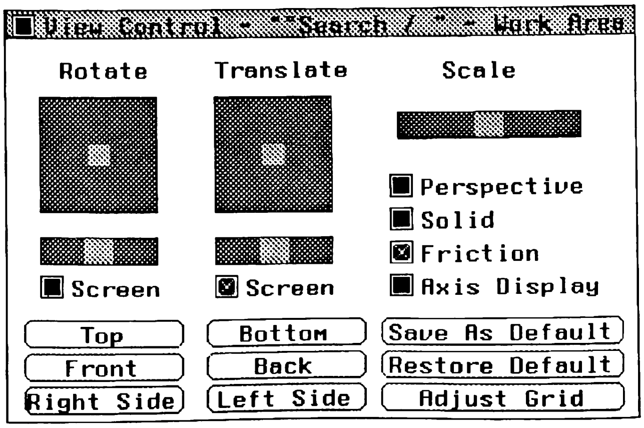
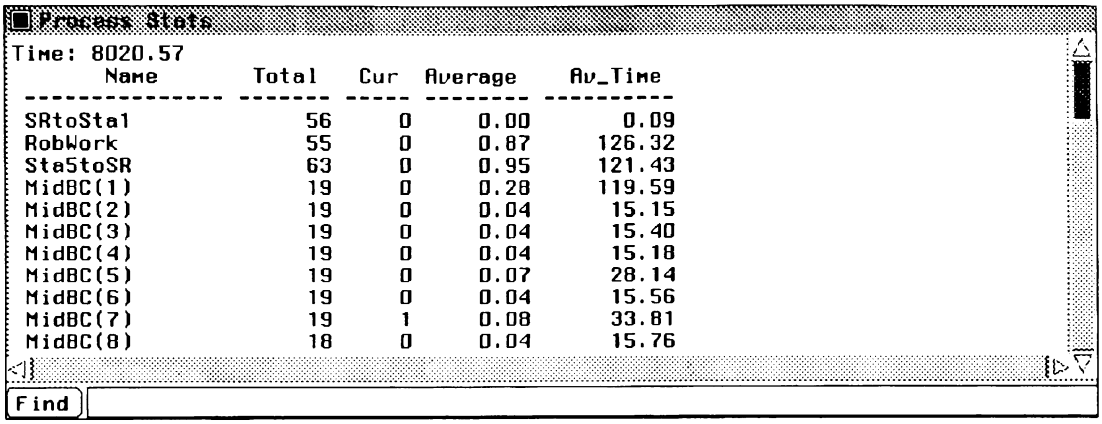


Figure 9. View Control Window

View Control. Figure 9 shows the view control window. This window provides the user the ability to position the graphics of the model in any orientation desired. Simply selecting and dragging the the mouse the rotation, translation, and scale can be dynamically changed with respect to all three axes (X, Y, and Z). The hardware that AM II runs on has the graphics performance to show the view changes dynamically. Different views can be saved and can be recalled as the user wishes. The animation picture can be shown in solid mode with the hidden lines and surfaces shown correctly. The animation picture can be shown in either perspective mode or true orthographic mode. View control also provides the feature of turning friction off allowing the user to spin or translate the model picture continuously. This is helpful when the animation is being video taped or filmed.



```
Process Stats
Time: 8020.57
        Name          Total   Cur  Average    Av_Time
 -------------- ------- ----- -------- ----------
 SRtoStal            56     0     0.00       0.09
 RobWork             55     0     0.87     126.32
 Sta5toSR            63     0     0.95     121.43
 MidBC(1)            19     0     0.28     119.59
 MidBC(2)            19     0     0.04      15.15
 MidBC(3)            19     0     0.04      15.40
 MidBC(4)            19     0     0.04      15.18
 MidBC(5)            19     0     0.07      28.14
 MidBC(6)            19     0     0.04      15.56
 MidBC(7)            19     1     0.08      33.81
 MidBC(8)            18     0     0.04      15.76
Find
```

Figure 10. Example Statistics Window

Debug. AM II 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 are offered to allow the user to suspend the simulation when a certain event occurs or when a specific clock time is reached.

Reports. AM II also provides comprehensive reports. These reports can be displayed on request at any time during the animation. Printed versions of the reports can be requested in the run control as described at the end of section 3.4.

AM II automatically keeps track of many statistics. These reports are linked to specific entity types such as:

- movement systems
- processes
- queues
- resources
- order lists, etc.

The user can also develop custom reports within the process procedures.

RUN TIME SUMMARY

AM II provides a comprehensive, flexible, and easy to use method of interacting with a model when it is executing. There is almost no limitation to what the user can do. If the simulation project is in the experimentation phase where only parameter changes are made and the mod el needs to be re-run several times, AM II provides the ability to run in batch mode without animation. Whether the need is for a highly interactive mode or a batch mode, AM II lets you do it.

5.0 SUMMARY

AM II is a industrial oriented simulation system that provides the user 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 AM II in comparison to using any other simulation language. The accuracy and degree of detail with respect to movement systems is unapproached.

AM II allows the construction of very large, complex models. AM II's structured language has proved that the larger the project, the more benefits AM II has over its competition.

AM II 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 AM II provides light-sourced- solid graphics with Z depth sorting, so all entities are shown in the correct relation to one another on the screen.

AM II is the ULTIMATE simulation system builder!

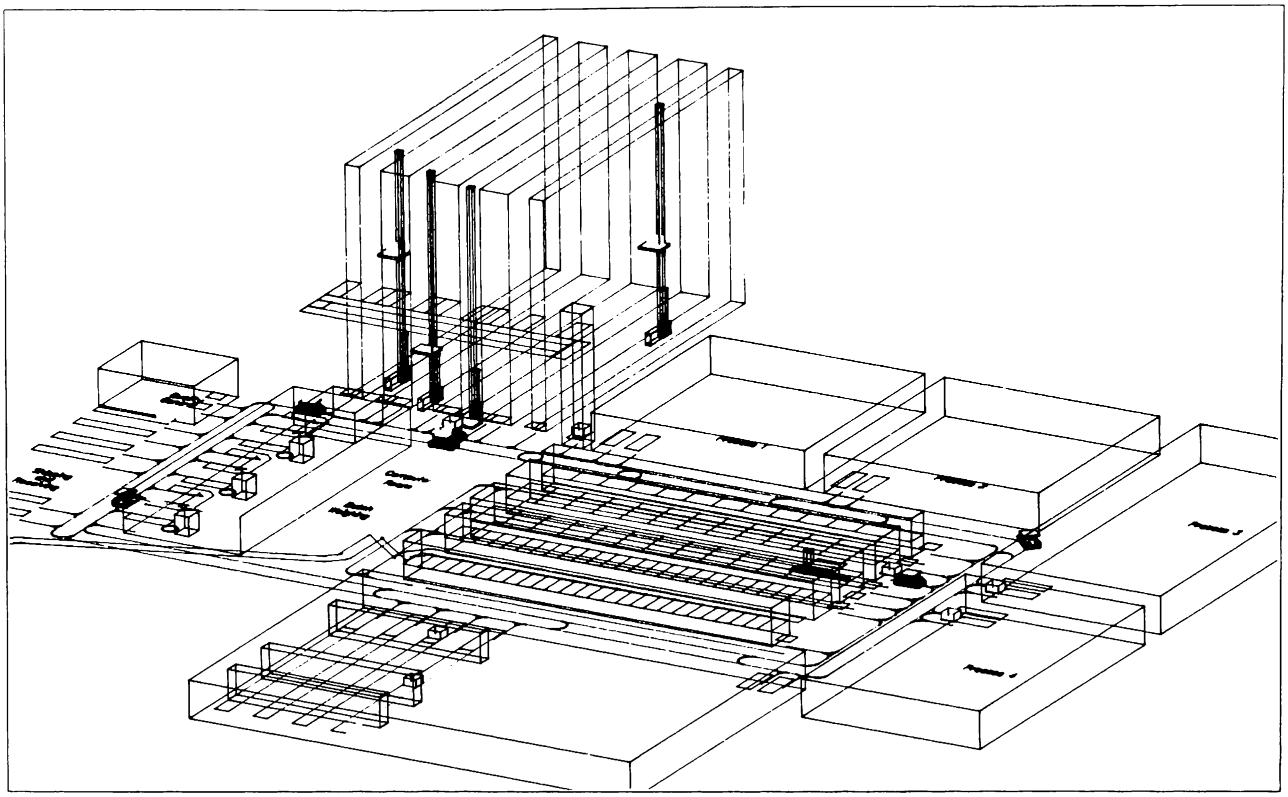


Figure 11. AutoMod II Animation Example

REFERENCES

AutoSimulations, Inc. (1989). AutoMod II Users Manual.

AutoSimulations, Inc. (1989). ACE Users Manual.

AutoSimulations, Inc. (1989). AutoMod II Lessons Guide.

AutoSimulations, Inc. (1989). AutoMod II with Kinematics Users Manual.

AUTHORS' BIOGRAPHY

Michael Thompson is Consulting Division Manager of AutoSimulations. He has also held the positions of Marketing Manager, and Scheduling Products Manager with ASI. He was previously employed by Texas Instruments where he was Manager of Methods and Tooling, Manufacturing Manager, and an Automation Systems Engineer. He has been involved in modeling manufacturing systems for 12 years. He received his M.S. and B.S. degrees from the Department of Engineering Sciences and Technology at Brigham Young University. Michael's research interests are in Manufacturing and Production Systems with emphasis in applied scheduling. He has authored papers on Finite Capacity Scheduling, Simulation Based Scheduling, Generative Process Planning, SQC Methods for Machine Tool Acceptance, and Group Technology. He is a member of the Society of Manufacturing Engineers, American Production and Inventory Control Society, and The Society for Computer Simulation.