Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

INTEGRATED SOFTWARE FOR MANUFACTURING SIMULATION

Kenneth D. Farnsworth
Van B. Norman
Dr. T. A. Norman
AutoSimulations, Inc.
P.O. Box 307
Bountiful, UT 84010

ABSTRACT

A description of AutoSimulations' simulation tools is given. An example of an AutoMod model is included and a discussion of the integrated environment follows.

1. INTEGRATED SIMULATION SOFTWARE

AutoSimulations, Inc. offers a patented simulation-based computer software system that supports every aspect of facility design from concept through implementation and operation. Each unique software package supports a facility design function offering a variety of automatic model development procedures.

1.1. AUTOMOD

AUTOMOD is a computer language specifically designed to simulate industrial systems such as factories, warehouses, and distribution centers. It combines the ease of objective programming with the power of procedural programming. These non-procedural and procedural elements can be separated so that only the more complex features need be programmed by hand. For example, AUTOMOD programs only need to know the physical dimensions of the movement system, the places along the system that the different types of loads are to be operated on, the time it takes the different resources to process loads, and some information about the speed of the movement system for the AUTOMOD compiler to generate the GPSS code that specifies the complete movement of loads through the system.

It is easy to create AUTOMOD simulations of systems built with conveyors, Automated Guided Vehicle Systems, Automatic Storage/Retrieval Systems, towlines, carousels, and power and free conveyors because AUTOMOD contains standard code for such movement systems given the physical dimensions and vehicle specifications as parameters. Accuracy is obtained by using a library of proven algorithms that operate on files containing the operating characteristics of specific hardware configurations.

To deal efficiently with complex systems, AUTOMOD also allows one to supply detailed procedures that tell the system exactly how, and in what order to perform specific tasks. Various statistics, kept track of automatically, can be used in programmer-written procedures. If necessary, GPSS subroutines can also be used. As a time saving feature, AUTOMOD comes with a report generator. The AUTOMOD graphics package AUTOGRAM allows a designer to see an animated display of the system in operation. The viewing perspective can be changed, and the different components can be viewed individually. The animation can be seen in real time, slow motion, or accelerated modes.

1.2. The AUTOMOD Perspective

All factory and warehousing systems are composed of temporary and permanent elements. In AUTOMOD, the temporary elements are called loads. The permanent elements are of two types: resources, which process loads; and movement systems, which move loads through the system. Resources are people--such as clerks, dock workers, and machine operators; and machines--such as packagers, riveters, assemblers, and robots. Possible components of movement systems are guided vehicles, automatic storage and retrieval cranes, conveyors, bridge cranes, transfer cars, lift trucks, operators and pickers.

Processes serve as a controlling system, they tell the loads which routes to follow, which resources to use, and other information needed to correctly model the system.

1.3. A Simple Program

This section is meant to give a feeling for AUTOMOD. We will explain how to write a simple conveyor program.

1.3.1. A Description of the System.

An AUTOMOD system at its most basic level can be thought of as a series of loads traveling through the system from process to process by means of a movement system. These processes contain instructions for loads to execute, such as: using resources, choosing a next process, managing the movement system, etc. Before writing an AUTOMOD program, the modeler must have a basic understanding of four important program components:

1. The loads that travel through the system
2. The configuration and type of the movement system
3. The different operations performed on the load
4. The number of resources in the system, and the amount of time each resource is required by the various loads.

The names of the movement systems, processes, loads, and resources used in the program must be declared at the top in the section known as the *declarations*. Let's start with a description of the problem.

Our example has only one type of load, called *Box*. One load enters the system about every 60 seconds. The loads can enter as close together as 45 seconds, or as far apart as 75 seconds. The loads' interarrival times (the times between successive arrivals) are uniformly distributed on the interval 45-75 seconds. This is modeled in AUTOMOD by the phrase "create at rate uniform 60+-15 sec." The phrase "uniform 60+-15" specifies the correct interarrival time, the midpoint 60 seconds plus or minus 15 seconds. We know that a maximum of 1000 loads will be created during the period of simulation. The AUTOMOD generation limit statement allows us to easily indicate the total number of each load allowed: "generation limit 1000." AUTOMOD must be told where the created loads are to enter the system. This is done by the statement in the declarations "first process Receiving." One other bit of information is known about Boxes--the first one does not enter the system until four minutes after the simulation starts. This is specified by: "first at 4 min." AUTOMOD has no trouble using minutes, seconds, and hours; or inches and feet in the same program.

Figure 1 describes the conveyor system that our model uses. Note that although our model uses only one type of movement system, any number can be used in a single AUTOMOD model. The process Receiving puts loads on the conveyor at the beginning of section K. They travel down conveyors A, B, and D, and are removed at the end of conveyor D to be packed. Process Packaging packs every three loads into one load, and then places that load onto the movement system at the beginning of conveyor F. The packaged loads then move down conveyors F, B, C, and L. The process Shipping removes the loads, where they leave the system.

We have three resources in our system:

1. Two receiving clerks. It takes them 50 seconds to receive a Box.
2. One packaging clerk. It takes 3 seconds to package a load.
3. One shipping clerk. This job takes 50 seconds as well.

To summarize the information:

1 load type: Box
3 processes:
- A. Receiving--performs 2 functions:
  1. Receives loads from outside of the system
  2. Puts loads onto the movement system
- B. Packaging--performs 3 functions:
  1. Takes loads off conveyor at end of section D
  2. Packages loads
  3. Puts packaged loads back onto the conveyor
- C. Shipping--performs 2 functions:
  1. Takes loads off the conveyor at end of section L
  2. Ships loads, that is, takes them out of the system.

1 movement system: a conveyor called Mover
3 resources: Rec_clerk, Ship_clerk and Pack_clerk.

The first line of a program is the declaration of a model name

```
model: FirstExample
```

The word "model" and the colon appear in all AUTOMOD programs. The name "FirstExample" is the name of this program. All declarations start with the name of the item being declared which is immediately followed by a colon. The end of each individual declaration is signaled by a semi-colon. The declarations follow.

```
movement systems: Mover
;
processes: Shipping Receiving Packaging
;
loads: type Box create at rate uniform 60+-15 sec
            generation limit 1000
            first at 4 min
            first process Receiving
;
resources: Rec_clerk capacity 2 processing time 50 sec
           Ship_clerk capacity 1 processing time 50 sec
           Pack_clerk capacity 1 processing time 3 sec
;
```

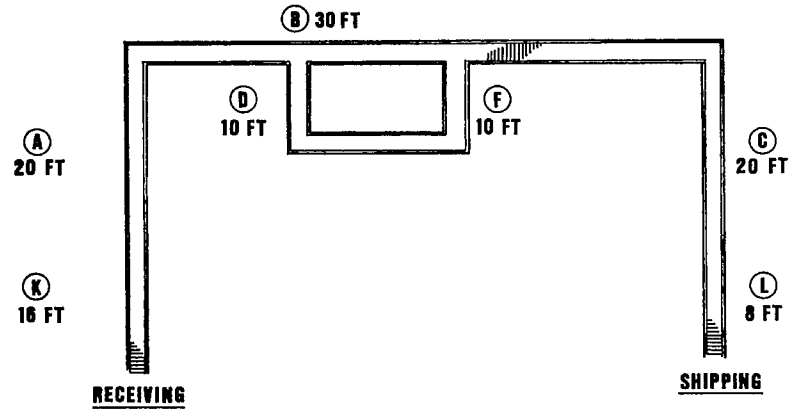

Figure 1. Simple Conveyor System

The Movement System. Next, we describe the conveyor. Four items must be included for each conveyor section:

1. The name of the section
2. The length of the section
3. The type of conveyor
4. The locations this section connects with other sections.

Physical information such as the velocity and width of the conveyor can also be included at the top of the conveyor description. If different sections of the conveyor have different physical characteristics they can be specified in the individual section definitions.

AUTOMOD understands several types of conveyor systems: queuing, accumulation, and transport. It is important to use the correct type because each is modeled differently by AUTOMOD. This example uses transport and accumulation conveyors.

Section K, an accumulation conveyor, is 16 feet long and connects to only one other section: K transfers at the end of the section to the beginning of section A.

```
section K length 16 ft accumulation
  transfer at end to A at beginning
```

The phrase "at end" referring to the "from" section in a transfer list, and the phrase "at beginning" referring to the "to," section are the defaults. They need not be included, but doing so improves the readability of the model. Our next example means exactly the same thing to AUTOMOD as the previous:

```
section K length 16 ft accumulation
   transfer to A
```

The next section, A, transfers to the right rather than transferring straight ahead to its next section, B. This is indicated by including the word "right" in the section definition. If the transfer had been to the left, the keyword "left" would have been used instead.

```
section A length 20 ft transport
   transfer right at end to B at beginning
```

Section B has a transfer in the middle of its section, as well as at the end. Notice that section F is not included in Section B's transfer table. This is because loads cannot travel down conveyor B onto F. The conveyor description accepts only descriptions of transfers in the direction of load flow.

```
section B length 30 ft transport
  transfer right at 10 ft to D at beginning
     transfer right at end to C at beginning
```

1.3.2. Processes. Processes represent the control structures of AUTOMOD. For each process the user must: specify its location on the movement system, indicate the resources involved, specify the load's next process and identify where in the movement system the process outputs the load. These actions can be as complex as desired. Once the configuration of a movement system has been described, the movement system itself knows how to move loads from one process to another, so the programmer need only say where the load is to be placed on the movement system, and what process the load is going to.

Process Receiving fetches the loads from outside of the system using the following statement:

```
input from birth
```

The program already knows from the loads table that the first process that Boxes are expected in is the process Receiving so it need not be specified again. The next line indicates that the resource Rec_clerk is used in Receiving. Since no processing time is given, the time used in the resource table, 50 sec, is automatically used.

```
use resource Rec_clerk
```

If the resource Rec_clerk is used for different jobs which take different amounts of time, the new time can be tacked onto the statement and will automatically override the default processing time listed in the resource declarations.

Example:

```
use resource Rec_clerk processing time 3 min
```

Where does the process Receiving leave the boxes after processing? The specific location of the loads is declared by indicating 1) the specific movement system used, 2) the conveyor section, and 3) the exact location of the loads measured from the beginning of the specific conveyor section:

```
output to Mover at K at 4 ft

/*  Movement system = Mover      *
 *  Conveyor section = K         *
 *  Location on conveyor = 4 ft */
```

The load is placed on the conveyor at 4 feet rather than 0 feet because the "leading edge" of the load; that is, the edge farthest upstream; is at 4 ft. If the load were placed on the conveyor at 0 feet, it would model the load hanging off the front of the conveyor, with only one edge actually touching the movement system.

Finally, the load must be sent to the next process, performed by a send statement.

```
send to Packaging
```

The process Packaging is similar, but it has a more complex send statement. Packaging packs three loads into one. So, only one load leaves for every three loads that enter the process. "Send to nextof Shipping die die" sends one load to process Shipping for every two that are removed from the system. The keyword `die` indicates that the loads referred to leave the system.

Loads leave the system at process Shipping, so all of its loads die. As the loads do not go to a location in the system no output statement is necessary.

Run Control Specifications. The run control section specifies how long the model will execute. In the case of figure 2 the program runs for 4 hours. If the "reset" keyword is used, statistics can be reset and the model run after a steady state has been reached. Any reports generated will reflect only the statistics gathered after the reset.

```
/*   Program FirstExample
 *
 *   This is a sample program utilizing a conveyor system
 *   used to illustrate some of the simple features of
 *   AutoMod
 *
 */

model: FirstExample

/*    Declarations start here                              */
movement systems: Mover
processes: Shipping Receiving Packaging
;
loads: type Box create at rate uniform 60+-15 sec
            first at 4 min
            generation limit 1000
            first process Receiving
;
resources: Rec_clerk capacity 2 processing time 50 sec
           Ship_clerk capacity 1 processing time 50 sec
           Pack_clerk capacity 1 processing time 3 sec
;
/*      Declarations end here                               */
descriptions:
/*      The conveyor's configuration is described          */
conveyor: Mover
      conveyor velocity 1 fps
      conveyor width 4 ft
      transfer cycle time 3 sec
      fixed window size 16 ft
      minimum interload spacing 4 ft

      section K length 16 ft accumulation
           transfer to A
      section A length 20 ft transport
            transfer right at end to B at beginning
      section B length 30 ft transport
            transfer right at 10 ft to D at beginning
            transfer right at end to C at beginning
      section D length 10 ft accumulation
            transfer left at end to E
      section E length 6 ft  transport
            transfer left at end to F at 4 ft
      section F length 10 ft accumulation
            transfer right at end to B at 24 ft
      section C length 16 ft transport
            transfer at end to L at beginning
            transfer right at end to D at beginning
      section L length 8 ft accumulation

;
/*     End of the conveyor description                    */

/*     Beginning of processes                             */
process: Receiving
         input from birth
         output to Mover at K at 4 ft
         use resource Rec_clerk
         send to Packaging
;
process: Packaging
         input from Mover at D at end
         output to Mover at F at beginning
         use resource Pack_clerk processing time 45 sec
         send to nextof Shipping die die
;
process: Shipping
         input from Mover at L at 8 ft
         use resource Ship_clerk
         send to die
;
/*      End of the processes                               */

/*      Beginning of Run Control Specification            */
count 3600
simulate 8

/*      End of Run Control Specification                  */
```

Figure 2. Sample Program FirstExample

Example:

```
count every 3600 sec
simulate for 8 counts
reset
simulate for 24 count
```

These specifications will run a system for 8 hours, reset all statistics, and then run it for 24 more hours. Reports will be printed for the first 8 hour run and the 24 hour run.

The Report Feature. Figures 3 through 5 show the reports AUTOMOD generates for the sample system of figure 1. By studying these, we can verify that our model ran correctly. In the process activity summary, we see that Receiving and Packaging received almost the same number of loads, 473 and 471 respectively. The 2-load difference is no problem, as those loads are on the movement system between processes Receiving and Packaging. Shipping received 145 loads, one-third the number that the other two processes received, as expected.

The conveyor section statistics show that each section had a reasonable number of entries. If there was a build-up of loads, the "max" column would be the same size as the "capacity" column. Notice that section B has more entries than any other section. Loads are counted twice in this section, once when they go from Receiving to Packaging, and then again when they travel from Packaging to Shipping. We would expect B to carry the number of entries on section D plus the number of entries on section F: 471 + 147=618. Every time a load enters a conveyor it is counted.

Since a load enters the system every minute on the average (other than in the first four minutes, when no loads enter the system), in the 8 hours the simulation will run it is expected that approximately 476 loads ((60 X 8)- 4) will be handled by Receiving and Packaging. Similarly, approximately 158 loads should be sent through Shipping (476/3). As the interarrival time of each load is a random number one cannot reasonably expect an exact match between the presumed numbers and the actual numbers, but the two should be close. In the example above, 473 loads enter the system, and 145 leave it, well within expectations. 158 - 145 = 13 loads, the same number that are in the system, either on a conveyor section or in a process.

1.4. AUTOBOTS

AUTOBOTS (developed jointly with Brigham Young University). is a robotic simulator that provides detailed work-station simulation plus integration of those workstations into a total manufacturing system.

1.5. AUTOGRAM

AUTOGRAM is an unequalled visual simulation tool that facilitates development and displays animated-plant operations in three-dimensional color graphics.

```
FirstExample            AUTOMOD REPORT      Tue Jun 24 16:39:29 1986

  RELATIVE CLOCK        28800               ABSOLUTE CLOCK        28800
                    ( Release 1.3.UN6 (20) )

Conveyor Section Statistics for Mover

SECTION UTIL ENTRIES  AVERAGE CAPACITY    CONTENTS          AVERAGE % NOT  DRIVE
                      TIME/ENT          AVE CURRENT  MAX     WAIT  WAITING AVAIL
-------------------------------------------------------------------------------
 K      0.049  474     12.000        4    0.197  0      1     0.00  1.000  0.000
 A      0.550  474     33.388        1    0.550  0      1     0.00  1.000  0.000
 B      0.219  632     20.000        2    0.439  1      2     7.88  0.039  1.000
 D      0.486  474     59.046        2    0.972  2      2     0.00  1.000  0.000
 F      0.056  158     20.354        2    0.112  0      1     0.00  1.000  0.000
 C      0.110  158     19.985        1    0.110  1      1     0.00  1.000  0.000
 L      0.158  157     58.000        2    0.316  0      1     0.00  1.000  0.000
```

Figure 3. The Conveyor Statistics Report

```
Simple1                 AUTOMOD REPORT      Tue Oct  8 13:28:55 1985

  RELATIVE CLOCK        28800               ABSOLUTE CLOCK        28800

RESOURCE STATISTICS

                                                    NUMBER          NUMBER IN USE
NAME          UTIL     ENTRIES  AVE TIME/ENT  AVAILABLE   AVE      CURRENT  MAXIMUM
-------------------------------------------------------------------------------
Rec-clerk     0.410        473        49.975          2   0.821            1        2
Ship-clerk    0.252        145        50.000          1   0.252            0
Pack-clerk    0.736        471        45.000          1   0.736            0
```

Figure 4. Resource Statistics Report

```
Simple1                 AUTOMOD REPORT      Tue Oct  8 13:28:55 1985

  RELATIVE CLOCK        28800               ABSOLUTE CLOCK        28800

PROCESS ACTIVITY SUMMARY

PROCESS NAME     TOT ENTRIES    AVE TIME/ENT          CONTENTS
                                                AVE   CURRENT      MAX
Shipping                 145          50.000  0.25         0        1
Receiving                473          49.975  0.82         1        2
Packaging                471          45.000  0.74         0        1
```

Figure 5. Process Report

Table 1: Explanation of the Conveyor Statistics Report

Section	The name of the conveyor section for which statistics are being reported.
Util	The average amount of the section's capacity occupied by loads; 0.000 if no load ever got on the section, 1.000 if the section was always full.
Entries	The total number of loads to enter or occupy this section.
Average Time/Ent	The average length of time loads stayed in this section, that is, how long it took on average to traverse the section.
Capacity	The maximum number of loads which can be in the section simultaneously.
Contents	Refers to the following three categories
Ave	The average number of loads in the section during the run period. If Util is 1.000 (Full Utilization), Ave Contents should be the same as Capacity.
Cur	The number of loads in the system when the simulation ended.
Max	The maximum number of loads ever in the section at one time.
Average Wait	The average length of time loads had to wait to get into the section.
% Not Waiting	The percentage of the loads entering the section which did not have to wait to enter, i.e., whose wait time was zero.

Table 2: Explanation of the Resource Statistics Report

Name	The name of the specific resource the statistics are for.
Util	The fraction of time the resource was busy.
Entries	The number of loads processed by the resource.
Ave Time Ent	The average length of time it took a load to be processed.
Available	The total number of this resource available. It should always be the same as the capacity declared in the resource declaration.
Number In Use	Refers to the following three categories
Ave	The utilization multiplied by the number of resources. Average number of loads using this resource.
Current	The number of resources being used when the simulation stopped.
Maximum	The greatest number of resources used at one time during the course of the simulation. (Not included if only one available.)

Table 3: Explanation of the Process Statistics Report

Process Name	The name of the specific process that the statistics are for.
Tot Entries	The total number of loads to enter the process.
Ave Time Ent	The average length of time a load was in the process.
Contents	Refers to the following three categories
Ave	The average number of loads simultaneously executing this process during the run.
Current	The number of loads in the process when the simulation ended.
Max	The maximum number of loads in the process over the course of the simulation.

1.6. INTERFASE

INTERFASE is a tool for scheduling modern manufacturing facilities, especially those such as flexible manufacturing systems (FMS) in which many independent work centers are capable of many different operations on many different products. Such flexibility allows an almost infinite number of different sequences (schedules) of the numerous individual operations required to manufacture a given mix of products. **INTERFASE** allows an experienced production scheduler to experiment with several combinations of scheduling rules to select from among the better possible schedules that one which optimizes a user defined measure of factory performance.

The development of **INTERFASE** arose from three needs identified by users of Material Requirements Planning Systems. MRP-generated scheduling techniques exhibit

Outdated and inaccurate forecasts.

Broad lead time generalizations.

Poor interactive capability.

In most manufacturing organizations, the schedule begins with a forecast. However, unless the organization serves a highly predictable marketplace (and few have that luxury), the forecast soon becomes outdated.

Secondly, MRP systems are heavily dependent on the lead times associated with each product. This assumes both equipment availability and accurate estimates. All manufacturers are faced with limited machines, tools, and personnel. In addition, machines break down, tools become dull or lost, processes go out of control, and personnel are absent. The effect of these conditions can only be grossly estimated within an MRP system.

Also, lead times used in MRP scheduling are broad generalizations of what is "supposed" to happen at a given point in time. For example, lead times typically have three "time" components:

Net processing time (includes actual processing time plus planned and unplanned down time).

Material Transportation.

Waiting (Queuing).

Processing times can be developed accurately through the use of the product's process, or production plan. Material transportation is figured less accurately with assumptions such as 10 minutes for intradepartment moves and 45 minutes for interdepartment moves. Waiting, or queuing, time is modeled the least accurately with estimates approaching days for a product to be processed. The problem arises when we are faced with the fact that in many job-shop environments *a part spends less than five percent of its total time on the shop floor being processed.* The balance is spent waiting, and to a lesser extent, moving. The problems in scheduling arise because the variability of process times, material movements, resource failures and other variables are not accurately depicted in the MRP program. The conclusion is that the lead times MRP systems are using—though perhaps sufficient for long-range planning and scheduling—are not very accurate for the short term. Worse, unnecessarily long lead times result in inflated work in process inventory costs.

Finally, most MRP systems do not readily allow production planning personnel to evaluate production schedules - to iteratively determine the effect of alternative resources, schedules, and operating strategies. Such experimentation can make substantial improvement in plant performance using current circumstances. MRP ground rules are fixed and rigid. Changing ground rules in the MRP program requires extensive time and programming skills. Rerunning an MRP "explosion" involves thousands of calculations for the typical factory and likely will require hours of computer CPU time. Planning personnel will consistently be faced with uncertainty in developing schedules. Planners should be able to develop contingency plans by understanding the effects of system variability.

The role of **INTERFASE** is to coexist with MRP systems, or, in small environments, act independently, by providing a management vehicle for "test-driving" schedules in an interactive, highly accurate, environment. A scheduler using **INTERFASE** is essentially trying tomorrow's or next week's schedule on a computerized replica of the factory flow. The key differences between **INTERFASE** and an MRP system is the accuracy of the modeled production operation and the ease-of-use in trying out alternate schedules for the same shop floor.

In large organizations, **INTERFASE** allows the MRP system to develop a long-range schedule. This schedule is then evaluated by **INTERFASE** given the real world status of the shop floor. This status includes machine and operator availability, WIP levels, and current management priorities or policies. The schedule is iteratively evaluated by repeated executions of the **INTERFASE** model. In this manner, the feasibility of the schedule can be determined as well as the effect of operator-induced changes (e.g. shutting down key machines for preventive maintenance, transferring operators between departments, etc.).

In smaller organizations, **INTERFASE** has the capability to act independently of a formal MRP system. **INTERFASE** can be configured to explode bills of material and develop initial schedules based on generally accepted scheduling algorithms.

1.7. IGES/SIM

IGES/SIM processes a properly prepared CAD drawing as input for a simulation model.

Used in various combinations, these fully integrated software products can model the most complex manufacturing/industrial facilities and test both design layouts and scheduling strategies. The system outputs a variety of statistical reports, business-type graphics and animation sequences. Special features allow for previews and even automatic modification of CAD layouts

reflecting the recommended improvements.

This user friendly system allows you to approach a simulation from either an English-like description, a graphics representation or interactive factory scheduling. The AutoSimulations approach has been used on many of the largest integrated industrial system models with impressive results.

2. English Language Model Development

The most general approach to model development is through AutoSimulation's simulation language called AUTOMOD.

Both facility geometry and process functions can be described in AUTOMOD and a simulation compiled from English-Language program statements.

Using AUTOMOD alone you can develop and execute powerful models with results and statistics printed as reports and using business-type graphics.

A simple 2-Dimensional animation presentation called PREVIEW is also available and can be displayed on a low cost graphics terminal or even a personal computer. PREVIEW is especially useful for debugging models.

AUTOMOD model results can then be examined through AutoSimulations' AUTOGRAM software for fully animated graphics if desired.

3. Graphic Model Development

Model development may begin with a design concept CAD drawing of a facility. AutoSimulation's IGES/SIM software may accept inputs from an appropriately prepared drawing using CAD International Graphics Exchange Standard (IGES).

Robotic workstation geometry with path plans and cycle times are established using AUTOBOTS.

Plant layouts, material handling networks, FMS and cell designs can be automatically defined with AUTOGRAM, ASI's 3-D animated graphics software.

Since AUTOBOTS and AUTOGRAM share a common geometry data base, robot path planning and workstation design can be integrated into the system simulation. Performance characteristics of the hardware components being modeled are defined using menu display options. All details are automatically merged into a comprehensive simulation model. In fact, many models can be developed completely using CAD and graphic procedures without a single written program statement.

The graphically defined models, executed by AUTOMOD can be viewed in color 3-D animation. Standard statistical reports can be printed as data or summarized using business-type graphics.

Finally, any design changes that occur through modeling can be redirected back through the IGES/SIM program modifying the original CAD drawing.

This powerful AutoSimulations Graphic Approach lets you:

Input graphic ideas/specifications
Test performance of the modeled system
View output data in animation or as statistical reports
Then update the original CAD drawings

This comprehensive approach tests the output of a facility before it leaves the planning stage.

4. Factory Schedule Model Development

Most large manufacturing system simulation models require the fabrication of many part numbers using many processing steps. The data bases created and the complex decision rules expand most models to an unmanageable size using traditional approaches.

AutoSimulation's INTERFASE (Interactive Factory Schedule Enhancer) provides a remarkably easy technique for simulating complex manufacturing systems from the scheduling point of view.

A factory image is quickly created with INTERFASE representing workstations, operators, materials, tools, scheduling algorithms, decision rules, priorities, etc. From the INTERFASE simulation, the most efficient factory configuration and schedule for resource utilization can be determined.

Among the unique outputs of INTERFASE is a time based sequence of events that would occur in the real physical system if the factory operation were conducted with a particular scenario. The real event sequences can then be transmitted to a detailed AUTOMOD model of the hardware system to produce an accurate and yet easily implemented test of the hardware elements.

This AutoSimulations approach can turn very complex scheduling problems into a form that can conveniently be modeled and be tested by design engineers.

AUTHOR'S BIOGRAPHIES

KENNETH D. FARNSWORTH, manager of software development at AutoSimulations, Inc., received a B.S. in Physics at Brigham Young University in 1983. He has written over 50 industrial simulation models at both Eaton-Kenway Co. (1979-1982), and AutoSimulations, Inc. (1982-present). He has been instrumental in developing several simulation packages including AutoSimulation's AutoMod. He is currently conducting research and development activities on new simulation techniques involving fourth-generation languages and expert systems.

Kenneth D. Farnsworth
AutoSimulations, Inc.
P.O. Box 307
Bountiful, UT 84010
(801) 298-1398

VAN B. NORMAN, vice president of AutoSimulations, Inc., received a B.S. in Mathematics at the University of Utah in 1969. He was a senior systems analyst for Eaton-Kenway Co. and the Utah Board of Education, for which he developed a state-wide computer network. At AutoSimulations, he supervises all simulation modeling and graphics development.

Van B. Norman
AutoSimulations, Inc.
P.O. Box 307
Bountiful, UT 84010
(801) 298-1398

THEODORE A. NORMAN, vice president of AutoSimulations, Inc., was chairman of the Computer Sciences Department at Brigham Young University. He received a B.S. in Mathematics from the University of Utah (1962), an M.S. in Information Science from Washington State University (1968), and a Ph.D. in Information Science at Washington State Univeristy (1970). He was a Systems Engineer for IBM and a consultant in simulation and controls design. He is currently involved in the development of new scheduling tools for manufacturing applications.

Theodore A. Norman
AutoSimulations, Inc.
P.O. Box 307
Bountiful, UT 84010
(801) 298-1398