Proceedings of the 1988 Winter Simulation Conference
M. Abrams, P. Haigh, and J. Comfort (eds.)

Animation with CINEMA

Trevor Miles
Randall P. Sadowski
Barbara M. Werner
Systems Modeling Corporation
The Park Building
504 Beaver Street
Sewickley, PA 15143

ABSTRACT

Cinema is a general purpose animation package designed to work intimately with the SIMAN simulation language. Cinema consists of two parts. The first, called CINEMA, is used to define the graphical images used in the animation. The second, called CSIMAN, is used to execute the animation. Both programs have a user-friendly graphical interface which does not require any programming. Cinema is available on microcomputers as well as Sun, VAX, and Apollo workstations.

1. INTRODUCTION

Animation has become very popular in the discrete-event simulation industry over the last few years. One reason for this increased interest is that numerical summary statistics give little information about the dynamic interactions of the components of a system.

While summary statistics are extremely important in the performance evaluation of a simulated system, they do not necessarily point out the cause of poor performance. Without animation, the task of the analyst is often tedious and time consuming.

The three main benefits of animation may be summarized as follows:

1. Model Verification - An animation provides the analyst with a visual method to rapidly and easily verify that the model is behaving like the actual system. By watching entities move and fixed resources change state, the analyst can check complex part routing, decision logic and process interactions that would be nearly impossible to verify with only summary statistics.
2. Bottleneck analysis - Although statistics may identify the location of a problem or bottleneck in the system, only through animation is it possible to easily capture the system status that caused that situation to occur.
3. Presentation & Communication - The animation provides the analyst with a powerful vehicle to clearly explain complex models and results to people unfamiliar with simulation and programming. It also provides a way for these nontechnical people to actively participate in the debugging and analysis of the system. The results obtained from animated simulations gain in credibility since viewers can watch a demonstration, see the critical situations evolve and draw their own conclusions instead of blindly trusting reams of numerical calculations.

The process of building a detailed, realistic simulation is complex and time consuming. Cinema allows the user to add animation to a model with a minimum of effort. The building and execution of a Cinema animation requires no modeling skills. Consequently, anyone with a knowledge of the real system being modeled, can develop the animation with some input from the modeler.

The user interface to Cinema is a key element of the product. The design is based on the specifications published by the Xerox Palo Alto Research Center. The user interacts with Cinema by using a mouse-controlled graphics cursor to pick items from a hierarchy of menus as shown in figure 1. Detailed on-line help is available at all levels of the menu system. The left button on the mouse activates the function pointed to by the graphics cursor while the right button displays the on-line help on a text screen.

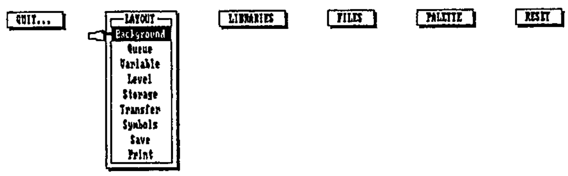

Figure 1 : CINEMA Main Menu

There are two files needed to run an animated simulation; a SIMAN program file and a Cinema layout file. The program file is a regular SIMAN model that describes the process flow, logic and parameters of the real system. The program file can be developed using Systems Modeling's graphical input programs, BLOCKS and ELEMENTS, or by using any text editor. The Cinema layout file contains the graphical images that will be viewed during execution. The layout file is constructed using the menu-driven CINEMA program and does not require any programming. There are no SIMAN blocks specific to animation, therefore, there are no modeling changes required to add animation to an existing model.

2. HARDWARE REQUIREMENTS

Since the first release of Cinema, many new platforms have been added. The primary platform is still an IBM PC-AT (or compatible) with 640K of memory, a math coprocessor, and the DOS operating system. There are currently two graphics cards available for the PC-AT/DOS configuration, namely the IBM EGA card (or compatible) or a high-resolution card purchased through Systems Modeling. The EGA has a pixel resolution of 640x350 while the high-resolution card (HGA) has a pixel resolution of 1024x768. The functionality of the two configurations is identical.

There are two configurations for users on the IBM PS/2 machines running DOS which correspond to the PC-AT EGA and HGA configurations. The EGA configuration is emulated by running the Cinema on the VGA under EGA emulation mode. The HGA resolution is achieved by using the IBM 8514 board and monitor system. The PS/2 Model 80 is recommended.

There is also a version of Cinema which runs in protected mode on the 80287 DOS machines. This allows the analyst to construct models of large systems. Cinema is also available on OS/2 machines. This version also allows simulation of large systems.

Cinema is available on the Digital Equipment Corporation (DEC) VAXstation series of workstations, which includes microVAX workstations with the GPX upgrade. Other workstations available are the Sun Microsystems Sun-3, Sun-4, and Sun-386i series and the Apollo DN3000 series.

3. THE CINEMA LAYOUT

The Cinema layout is a graphical representation of the system being simulated. The layout consists of graphical objects grouped into two categories, the static component and the dynamic component. The static component of the layout does not change during the execution of the simulation model, while the dynamic component consists of graphical images which do change color, location or shape.

3.1 The Static Component

The static component of a layout represents the physical structure or background environment in which the simulated system exists. As an example, in an animation of a factory floor, the static component would consist of the walls, aisles, storage bins, offices, bulletin boards, etc.. That is, the static component includes anything that helps to pictorially define the scene, yet does not need to move or change during model execution.

The static component or Background can be generated in a variety of ways; the first is by simply using the drawing facilities provided in the CINEMA program (See *Background* menu, figure 2). Under the *Draw* option there are functions available for *line, box, circle, bar* (a box filled with a solid color or a hatch pattern), *arc*, and freehand *sketch*. All of these options can be modified by selecting a different color and/or line thickness. Text can be placed on the screen at any location and in any orientation. The *Text* option allows the user to add lettering in any combination of six different *sizes*, with six different *fonts*, and 16 active *colors*. *Explode* allows the user to zoom for fine detailing of small portions of the drawing, while *Cut* provides a means of *copying* or *moving* rectangular regions of the background.

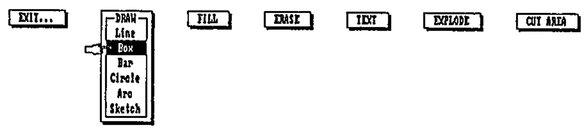

Figure 2 : CINEMA Background Menu

The drawing functions within CINEMA are activated by pointing to them with the mouse and clicking the left button. The attributes of the drawing function, such as line *thickness* and *color*, can be set by picking from pulldown menus. *Line, box, bar, circle,* and *arc* are all performed in "rubberband" mode where the user places, for example, the center of a circle somewhere on the screen and then can choose the radius of the circle by moving the mouse away from the center. As the mouse is moved, the circle will continuously stretch or shrink depending on whether the mouse is being moved away from or toward the center of the circle. When the user clicks the left button, a permanent circle is drawn with the current settings for color, width and radius. Similar steps are available for the other functions listed above.

Sketch, fill and *erase* are different in that no rubberbanding is available. Once the user picks the seed point for the fill, the fill will progress until all unbounded regions joining the seed point are filled with the current pattern and color.

The *sketch* and *erase* functions are very similar in that erase can be thought of as sketching in black. The user moves the graphics cursor to the desired location and depresses the left button of the mouse. If the user keeps the left button depressed while moving the mouse, a black line will be traced following the path of the mouse.

The second method for generating a background is to start with a DXF file. A DXF file is generated by many of the computer-aided design (CAD) packages available, such as AutoCAD, PC-CAD, etc.. These packages have more sophisticated drawing functions, especially for three-dimensional images, allowing the user to develop a more complex background. Any DXF formatted drawing can be input by Cinema's utility program called DXF2LAY. This program converts the drawing into the static component of a layout. An Example of a layout generated from a DXF file is illustrated in figure 3.

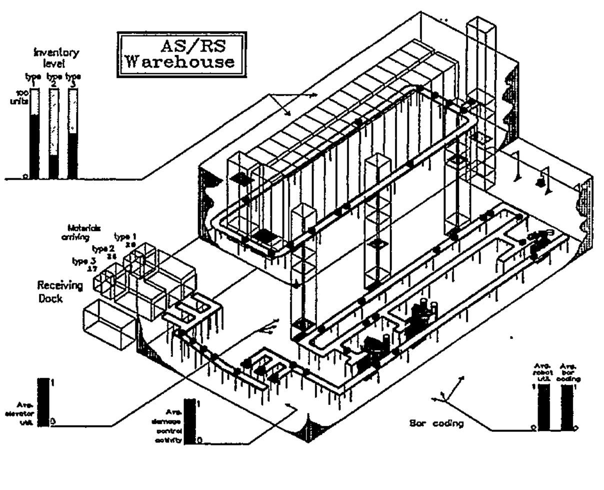

Figure 3: Cinema Layout Generated with DXF2LAY

The third method is to use an image scanner, such as the Hewlett Packard Scanjet. Using a scanning utility, the user can scan a blueprint, a photograph, etc. and place each scanned image on the background at any location. The user may scan as many images as desired to form the background. This scanning utility is not a part of the Cinema package, but is a separate product available through Systems Modeling.

The CINEMA drawing functions can be used to "pretty up" the backgrounds generated from either the DXF2LAY or scanning utilities. For example, AutoCAD v9 does not allow the user to fill certain faces of a particular image. In this case, the user can generate a layout file using the DXF2LAY utility and later fill the desired regions with patterns or colors using the CINEMA drawing *fill* feature.

3.2 The Dynamic Component

The dynamic component of the layout consists of graphical images which change shape, color, size, or location in response to a change in the SIMAN model. The dynamic objects in a Cinema layout are directly tied to SIMAN modeling constructs. As the states of the constructs change during execution, the dynamic objects will change to reflect the state of the SIMAN construct.

The user may choose to animate as much or as little of the SIMAN model as is useful and necessary to convey pertinent information. If the user omits a certain section of the model from the animation, the model will continue to run unhindered. If the user places an extra dynamic object in the layout which is not used in the model, the unassociated dynamic object will not change when the model is run, but will also not stop the animation from running.

There any many dynamic objects available; each is discussed in some detail below.

Entities - In a SIMAN simulation model, entities represent the items that are being processed or flow through the system. An entity is represented in a Cinema layout as a moving and/or changing symbol. The symbol could be a sketch of a workpiece or a packet of information moving down a communications line. As the entity moves from work area to workarea within the SIMAN simulation model, its corresponding symbol is automatically moved across the static background in the Cinema layout.

Entity symbols are created by drawing the icon on a blown-up grid, using the mouse-controlled cursor. The same drawing functions available in *Background* are provided through pull-down menus. Each box in the grid corresponds to one picture element (pixel) of the actual symbol image, and therefore allows tremendous flexibility for detailing. As the *symbol* is created on this grid, it is simultaneously displayed in actual size in the upper left corner of the screen, as shown in figure 4.

The entity icons are created and stored in an entity symbols library (see figure 5). *Symbol libraries* are maintained separately from the animation layout and may be saved (stored on disk) and recalled at will. This allows a user to save time by reusing symbols previously created.

To establish the association between a graphical entity symbol and a specific entity within the SIMAN model, the user must reserve one of the general purpose entity attributes in the SIMAN model. The modeler is responsible for using that attribute to assign an entity number to entities in the SIMAN model. In CINEMA, a specific symbol from an entity library is then associated with a particular entity number. If the designated entity attribute changes values in the SIMAN simulation model, the corresponding entity symbol changes in the animation. Consider, for example, a simulation of an automotive assembly plant. The designated attribute of an entity arriving to an assembly station might be set to a value that corresponds to a symbol of a car body without doors. After leaving the workstation, the associated symbol could be changed to a car body with doors simply by assigning a new entity number to the designated attribute in the SIMAN model.

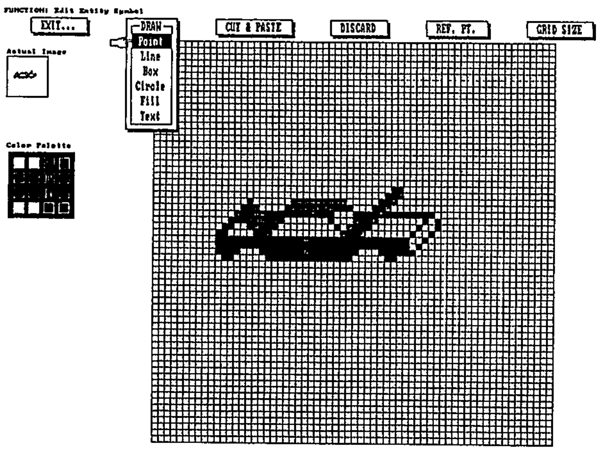

Figure 4 : Creating an Entity Symbol

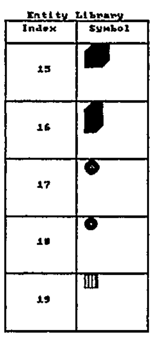

Figure 5 : Entity Symbol Library

Resources - *Resources* are used in SIMAN to model limited items in a system, such as machines and workers. Entities compete for the limited number of units of a resource and incur queueing delays when enough units are not available. In a SIMAN model, each resource assumes one of four possible states: idle, busy, inactive, or preempted. The resource is in a busy state when it has been allocated to an entity at a *SEIZE* block in SIMAN. The resource remains busy until it is released by the entity at a RELEASE block. Once released, its status changes to idle (making it available to be re-seized by other entities). Units of a resource can be removed or made unavailable by using an *ALTER* block. This changes the status of those units of the resource to inactive until they are placed back in service using the *ALTER* block again. When a resource is allocated to an entity at a *PREEMPT* block in SIMAN its state is preempted until it is released by the entity at a *RELEASE* block.

Resource status changes within a SIMAN model are displayed in a Cinema animation by using resource symbols that represent the resource in each of the possible states. Like entity symbols, resource symbols are created by drawing icons on the enlarged grid, with the mouse cursor. They are stored in a resource symbol library file that can be recalled and used with other models. An example of a resource symbol library is shown in figure 6.

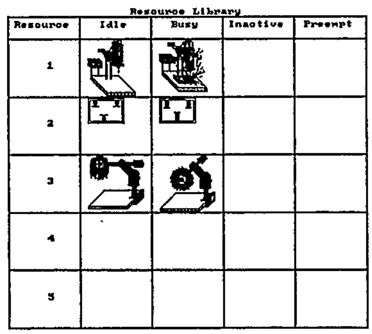

Figure 6 : Resource Symbol Library

For all libraries, there are functions to manipulate the symbols within the library or between different libraries as shown in figure 7. For example, to create a set of machines that only differ in color, the user does not have to create each one from scratch; he can use the Copy function to make the machines, then go back and simply edit each one to change the color. The Fetch option allows the user to copy a symbol from another library, for example if a computer is an entity in one model and a resource in another.

Figure 7 : Library Symbols Menu

A resource is added to the layout by selecting an idle symbol from the library and positioning it on the static background, using the mouse. At that time the user must also associate a SIMAN resource number with that symbol (as used in the experiment frame of the SIMAN model). When the status of a resource changes in the SIMAN model, the associated Cinema resource symbol (busy, idle, inactive, or preempted) is displayed at the designated location on the screen.

Queues - A Cinema queue is a dynamic representation of the entities residing in an associated SIMAN *QUEUE* block. These entities might represent workpieces awaiting the availability of a machine, a set-up operator, a space on a conveyor, etc.

A *queue* can be added to the layout at any location, and in any length and orientation. The mouse-controlled graphics cursor, is used to digitize a first point corresponding to the head of the queue. As the graphics cursor is moved to locate a second point corresponding to the tail of the queue, a line segment is rubberbanded between the head and current tail point to show the length and orientation of the queue. Pressing the left button on the mouse digitizes the second point and adds the queue to the layout, as shown in figure 8. This graphical representation is then associated with a specific queue number in the corresponding SIMAN simulation model.

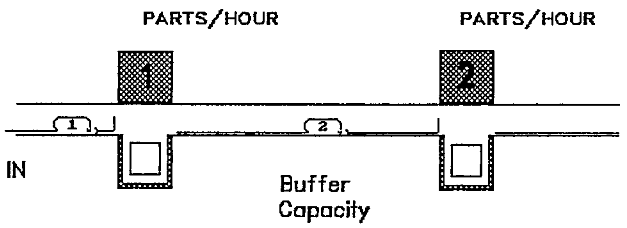

Figure 8 : Adding Queues to a Layout

When an entity enters a queue in the SIMAN model, the entity's symbol is displayed along the corresponding queue symbol at the proper location relative to the other members of the queue, see figure 9. When an entity exits the queue in the model, its associated symbol is removed and all following symbols are moved forward one position. When the queue in the SIMAN model contains more entities than can be displayed along the fixed length of the graphical queue symbol, subsequent arrivals to the queue are not displayed. The entities that are not displayed will eventually become visible as they move forward into the visible portion of the queue as preceding entities exit.

Stations - Distinct workcenters within a system are modeled using the SIMAN *STATION* construct. In CINEMA a *station* symbol is used to designate start and end points for the movement of entities between the workcenters in the SIMAN model. The station symbols, like the queue symbols, do not explicitly show up during the execution of the animation; they are only used to govern the placement of the entity symbols as they move through the system. Station symbols are added to a layout by simply entering the station number and clicking the left mouse button to digitize the desired spot.

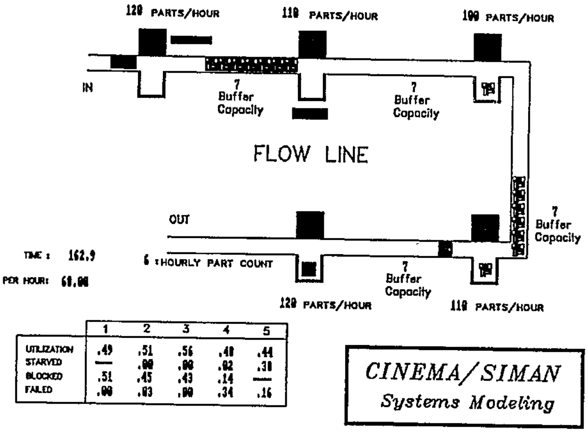

Figure 9: Running Animation With Parts in Queues

Routes - The *Transfer* menu, under the CINEMA Layout menu, is used to define the paths of travel between the stations of the SIMAN model. One method for modeling the movement of entities between *STATIONS* is to use the *ROUTE* block. When an entity reaches a *ROUTE* block in the SIMAN model, its associated entity symbol is continually redisplayed at new points along a predefined CINEMA route path to produce the effect of movement.

A route can be placed anywhere on the layout and consists of one or more rubberbanded line segments that are connected to form the path. To add a route to a layout, as shown in figure 10, the user must digitize the starting station symbol, draw the intermediate path segments and end by digitizing the ending station symbol.

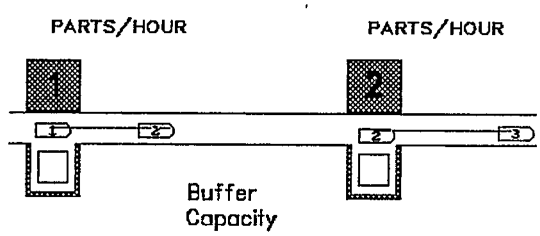

Figure 10 : Adding Routes Between Stations

Movement between stations that involves SIMAN material handling constructs is designated in much the same way as the routes. If entities are transported by a *CONVEYOR*, the user will digitize paths called *segments*, and if the entities are moved by *TRANSPORTERS*, the paths will be called *distances*. All route, segment and distance paths are transparent during model execution, yet the entity symbols will trace these paths as they move.

Status Variables - While a simulation is executing, SIMAN automatically maintains the value of many status *variables* which define the system state. Examples of these status variables are: the value of simulated time, the number of entities in a queue, the number of parts processed, the number of busy machines in a workcell, the cumulative utilization of a worker, etc. A representation of any SIMAN status variable can be incorporated into an animation layout using one of four dynamic features in CINEMA.

A digital display of the numeric value of a status variable can be added to a layout using a feature called dynamic variables. The user selects the variable and defines the format (number of significant figures and places to the right of the decimal), range of values, size, and color using pull-down menus and then positions the variable anywhere on the screen using the mouse. Figure 11 shows an enlargement of the machine statistics for the animation shown in figure 9.

	1	2	3	4	5
UTILIZATION	.49	.51	.56	.48	.44
STARVED	—	.00	.00	.02	.38
BLOCKED	.51	.45	.43	.14	—
FAILED	.00	.03	.00	.34	.16

Figure 11 : Digital Display of Status Variables

A second way to display status variables is with a feature called *levels*. This analog representation of a variable's value, provides a means of displaying dynamic graphs. An example of using levels to display utilization statistics is shown in figure 12. Three different level shapes are included in CINEMA: a *box*, a *circle*, and a *dial*. During execution of an animation, a box or circle level fills and empties in response to changes in the value of the associated status variable. The dial is a circular level with a sweep hand that rotates either clockwise or counter-clockwise. The shape, fill and empty colors, fill direction, size, and location of a level are all specified using pull-down menus and the mouse cursor.

A feature called *global* symbols represents a third way to display the value of a status variable. Like entity and resource symbols, global symbols are drawn on the enlarged grid and maintained in a separate global symbol library. To incorporate global symbols in a layout, the user associates selected symbols from a library with designated values of a specified system status variable. For example, a symbol saying "STARVED" could be displayed when the number in a queue for a machine is zero, and a second symbol saying "BLOCKED" could be displayed when the queue is full to capacity (See figure 13).

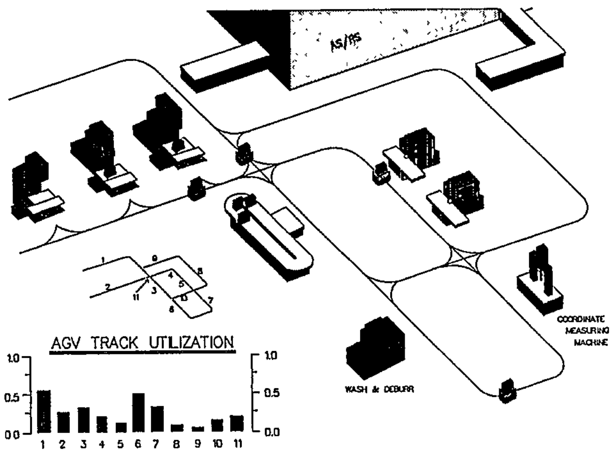

Figure 12 : Utilization Statistics Displayed With Levels

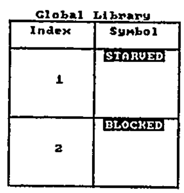

Figure 13 : Global Symbols

Dynamic colors represents a fourth way to display the value of a status variable. The user first associates one of the sixteen CINEMA color indices with a specific status variable. The user then defines one or more different colors for that index (in the form of different combinations of red, green, blue intensities) and associates each new color with a designated value of the status variable. For example, in the simulation of a steel making facility, a status variable that represents the temperature of a furnace could be tied to the color index used to draw the furnace in an animation layout. Specific values of the variable (furnace temperatures) could be associated with different combinations of red, green, and blue intensities that produce colors ranging from gray to red. As the temperature of the furnace increases in the simulation model, we would see the furnace gradually change from gray to red.

4. RUNTIME FEATURES

As discussed previously, the Cinema system consists of two components. So far we have discussed the capabilities of the layout generation program, CINEMA. We now turn our attention to CSIMAN which is used to perform the animation.

CSIMAN is a special version of SIMAN which includes the ability to update the dynamic component of the layout interactively during execution as well as control the speed of execution.

The user interface to CSIMAN is identical to that of CINEMA with different menu items, as shown in figure 14. Since Cinema is a real time animation system, as opposed to a post processed system, the CSIMAN menu includes facilities to temporarily stop a run and use the SIMAN interactive debugger features. The debugger allows a user to change variables, look at the entities in a particular queue, look at status variables not displayed on the animation, and monitor practically any variable in the system. A user can turn on the SIMAN trace function so that the animation can be viewed simultaneously with the model execution statements. This feature is very useful for validation and debugging. Associated with the trace capability, is STEP, which halts the model execution after each event is processed. Execution is resumed when the user presses the space bar and continues until the time the next event is processed.

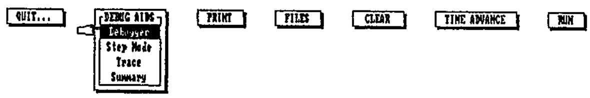

Figure 14 : CSIMAN Menu

The user may associate any number of layouts with a single SIMAN model. These layouts may be recalled from the CSIMAN menus or directly from the keyboard. The user can generate an ASCII file which contains a list of all the files associated with the SIMAN model. By pressing the "1" key during execution, the first file in the list will be read in and the simulation will progress. The "2" key will recall the second file in the list, etc.

Another CSIMAN feature is save SNAPSHOT. This saves a picture of the layout at that instant in simulated time as well as saving the value of all the system variables. The snapshot may be recalled at a later time so that the simulation can progress from the time that the snapshot was saved. This feature is particularly useful in demonstrating and presenting critical situations and comparing variations of the system at the same moment in time.

The speed of the animation is controlled by both menus and the keyboard. The user can choose a scale factor which is an association between the simulation time and real time. The smaller the scale factor, the slower the animation will progress. The dynamic objects are updated as often as possible between changes of the simulation clock. During animation the simulation clock will progress smoothly. Consequently, if the state of the simulation does not change over a long period of time, the graphical image will remain static for a corresponding period of time. The scale factor can be dynamically changed by using the "<" and ">" keys. The ">" increases the scale factor while the "<" key reduces the scale factor. If the time of interest is later on in the simulation, the user may disable the animation and skip ahead to a predetermined time. The simulation will be performed as fast as possible, without updating the animation, so as to greatly reduce the time required to get to the desired simulated time.

There are some features which allow the user to demonstrate several options quickly without having to bother with the mouse and keyboard. When CSIMAN is invoked, the SIMAN program file, Cinema layout file, and the scale factor can be specified on the command line. If all three are specified on the command line, the animation will begin as soon as all the files have been read in. To change to another layout, the user can simply type one of the numeric keys as discussed above. To change to another SIMAN model, the user can press the "Q" key to quit the current simulation quickly and then invoke CSIMAN with a different program file, layout file, and scale factor.

In Cinema/HGA (PC-AT with high resolution graphics board) the user may also zoom and pan during the execution of the simulation. The zoom is invoked by the keypad "+" and "-" keys. The "+" key zooms in (displays less of the total picture on the screen), while the "-" key zooms out. Once the user has zoomed in at least once, the cursor (arrow) keys may be used to move the focal point of image around. To zoom out completely, the user can press the "Home" key.

5. SUMMARY

Cinema is a general purpose animation system which allows for the straight-forward animation of any SIMAN model. Because Cinema utilizes user constructed icons, any type of system ranging from manufacturing, to distribution, to health care, transportation, and communications may be animated.

The mouse oriented, menu-driven user interface allows for the rapid development of animations without the need for programming. The tool of animation aids the analyst in the process of debugging and verification of the SIMAN model as well as in the presentation of the final results.

REFERENCES

Davis, Deborah A. and C. Dennis Pegden, "Introduction to Siman".Proceedings of the 1988 Winter Simulation Conference (M. Abrams, P. Haigh, and J. Comfort, eds.) Institute of Electrical and Electronics Engineers, San Diego, California.

Johnson, M. Eric and Jacob P. Poorte, "A hierarchical approach to computer animation in simulation modeling". SIMULATION 50:1, 30-36. 1988 Simulation Councils, Inc.

Lampon, B. W., "Bravo Manual" in Alto Users Handbook, Xerox Palo Alto Research Center.

Pegden, C. D. (1982). Introduction to SIMAN. Systems Modeling Corporation.

Systems Modeling Corporation (1985). CINEMA User's Manual, State College, PA.

AUTHORS' BIOGRAPHIES

TREVOR MILES is a Senior Project Engineer in Software Development with Systems Modeling Corporation. Since joining Systems Modeling in 1984, he has worked on development of SIMAN, Cinema, and other SM products. He is currently project leader of Systems Modeling's mini and mainframe computer products and of statistical analysis software. He has also ported Systems Modeling products to a number of different workstation environments. He is currently a PhD candidate in the Department of Industrial Engineering at the Pennsylvania State University with research interest in optimization of stochastic systems, on-line scheduling of FMS systems with simulation, and the interface of programmable controllers with simulation systems. Miles received his MSc degree in Engineering from the University of Witwatersrand, Johannesburg, South Africa and his BSc in Chemical Engineering from the University of Cape Town, Cape Town, South Africa.

RANDALL P. SADOWSKI is Vice President of Systems Modeling Corporation in charge of consulting and user-education services. He was previously on the faculty at Purdue University in the School of Industrial Engineering and at the University of Massachusetts. He received his bachelors and Masters degrees from Ohio University and his Ph.D. from Purdue. Dr. Sadowski's research interests are in Manufacturing and Production Systems with emphasis on modeling, control, and applied scheduling. He has authored over 50 technical articles and papers, served as chairman of the Third International Conference of Production Research, and will be the General Chair of the 1990 Winter Simulation Conference. He is a senior member of the Institute of Industrial Engineers, served as editor of a two year series on Computer Integrated Manufacturing Systems for IE Magazine which received the 1987 IIE outstanding publication award and was a co-editor of Computer Integrated Manufacturing Systems: Selected Readings, published by IIE.

BARBARA M. WERNER is an Engineer performing consulting and teaching functions for the Simulation Services Division of Systems Modeling Corporation. She received her Bachelors Degree in Engineering from Cornell University, with a major in Operations Research & Industrial Engineering, and a minor in Interpersonal Communications. Barbara previously worked as a Simulation Project Engineer for General Motors Truck & Bus, Central Operations Engineering. In this capacity she coordinated and directed all simulation efforts at three assembly plants and one metal fabrication plant. She also taught Simulation and Modeling workshops and developed large scale models using SIMAN and Cinema animations. At Systems Modeling, Barbara has developed a menu-driven tutorial package for the SIMAN and Cinema software systems. Barbara is a member of the Institute of Industrial Engineers and has been a Business Consultant for Junior Achievement's Applied Economics program for high school students.

Trevor Miles
Randall P. Sadowski
Barbara M. Werner
Systems Modeling Corporation
The Park Building
504 Beaver Street
Sewickley, PA 15143
(412) 741-3727