Proceedings of the 1985 Winter Simulation Conference
D. Gantz, G. Blais, S. Solomon (eds.)

CINEMA TUTORIAL

Kevin J. Healy
Vice President
Systems Modeling Corp.
226 Highland Avenue
State College, PA 16801

ABSTRACT

CINEMA is a general purpose, micro-computer based animation system designed to work with the SIMAN simulation language. A sophisticated yet easy to use graphical interface allows users with little or no programming skills to build highly detailed animations of any SIMAN simulation model.

INTRODUCTION

Computer simulation is a powerful tool for use in the design and analysis of manufacturing systems. It allows an analyst to evaluate the consequences of design and operating decisions before those decisions have to be made.

One of simulation's shortcomings has been that outputs from a simulation model typically take the form of summary statistics or simple graphs. Although these outputs are necessary to measure and draw conclusions on the performance of the system, they provide little insight into the dynamic interactions between the components of the system.

Computer animation represents the ideal solution to the problem of determining the dynamic behavior of a simulation model. There are several examples of the use of specialized computer animation for this purpose [1]. In those applications in which animations have been employed, the benefits of "seeing" the system operation have been substantial. The two main benefits which have been cited are the following:

1. Model Verification - An animation is an ideal method for verifying the correct operation of the model. Subtle errors which might not be apparent from standard simulation output become obvious when the system operation is displayed graphically.

2. Selling the Solution - The results from simulation studies are some times difficult to sell to management. Animation is an extremely powerful aid in convincing management that the model does represent the system being modeled.

A typical animation, however,may require months of specialized graphics programming as well as special graphics hardware. For most simulation projects, even the well recognized benefits of animation do not justify the added expense.

These considerations provided the motivation for the development of CINEMA, a general purpose, micro-computer based animation system designed to work with the SIMAN simulation language. A CINEMA animation is a dynamic display of graphical objects that change location , color or shape on a static background to display the dynamic behavior of a SIMAN simulation model as it is executing. The development of a CINEMA animation requires no programming skills. Highly detailed animations which might otherwise require months of programming, can be developed in a few hours with CINEMA.

The key to the CINEMA system is the design of the user interface. The design is based on developments made at the Xerox Palo Alto Research Center (PARC) in the mid-seventies [2]. The user interacts with the CINEMA system by using a mouse-controlled graphics screen cursor to make selections from a hierarchy of "pull-down" menus. By positioning the mouse-controlled cursor on a selection and pressing the left button on the mouse, the selected header will "pull down" revealing a menu of secondary choices. A selection within the menu is made by moving the mouse-controlled cursor to the desired item and again pressing the left button. A copy of an actual screen image in Figure 1 illustrates the pull-down menu concept.

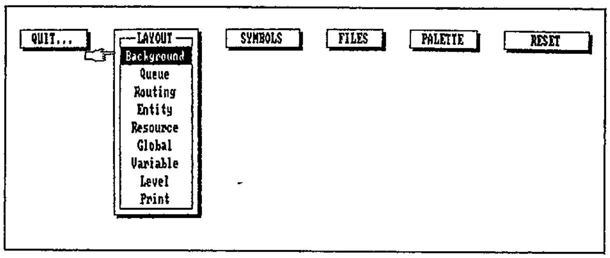

Figure 1: Pull-down Menus

On-line help is available on any menu item by moving the the cursor to the item and pressing the right button on the mouse. CINEMA will overlay a help "window" on top of the current graphics screen with detailed information on the selected item.

Animations are generated by first constructing a SIMAN simulation model of the system. With minor exceptions, the SIMAN model is constructed without special consideration for whether it will be run with an animation. The CINEMA program is then used to construct a corresponding animation layout which is a graphical depiction of the physical components of the system being modeled. The user then executes the SIMAN simulation model in conjunction with the CINEMA layout to generate a graphical animation of the system dynamics.

HARDWARE REQUIREMENTS

The CINEMA system is designed to run on an IBM PC-AT with 640K bytes of memory, an 80287 math co-processor, and the DOS operating system. Graphics specific hardware includes a propriety, high resolution (832 x 624), noninterlaced color graphics board (capable of displaying 16 simultaneous colors from a palette of 4096) and a high resolution, fast scanning, 19-inch diagonal color monitor.

THE ANIMATION LAYOUT

An animation layout is a combination of one or more graphical objects that form a representation of the system being modeled. The objects that comprise a layout are one of two types referred to as static and dynamic. The static objects within a layout form a background and represent the portion of the layout which does not change during the animation. In a simulation of a manufacturing system, this might correspond to a sketch of the walls, aisles, posts, etc., of the facility being modeled. The dynamic objects within a layout are superimposed upon the static background and represent the objects within the system which change location, color, or shape during the execution of the simulation. Workpieces, workers, machines, robots, etc. would be represented as dynamic objects within a layout. Figure 2 contains an example of an animation layout.

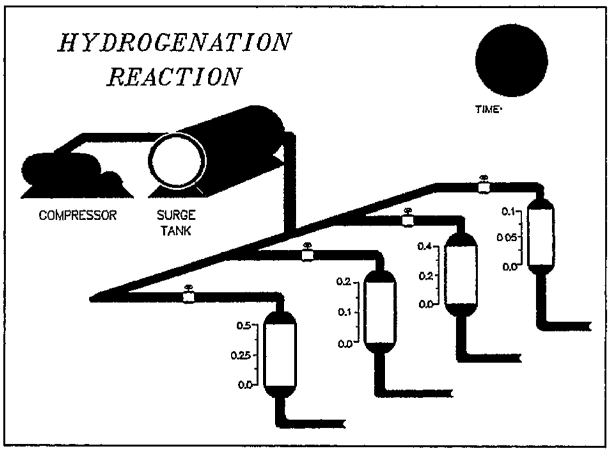

Figure 2: Animation Layout

The Static Component

The static component of the layout is constructed using a set of elementary computer-aided drawing functions which allow the user to easily add both graphics and text to the background. The basic graphic drawing functions include line, box, circle, and arc. Pull-down menus, are used to set attributes of the current drawing function including color, style, and line width. These basic graphics functions are drawn in "rubberband" mode, allowing the user to view the size and orientation of the object before it is actually added to the layout. For example, to add a box to the layout, the user selects the desired line color, style, and width using the pull-down menus. The mouse-controlled graphics cursor is then used to a locate a point on the screen corresponding to one corner of the box. The point is fixed on the screen by pressing the left button on the mouse. The user then sees the box continually stretch and/or shrink as the graphics cursor is moved to locate the opposite corner of the box. Pressing the left button on the mouse again finalizes the size of the box and adds it to the layout.

A sketch function allows the user to enter any free-form curve by simply moving the graphics cursor along the desired path while holding down the left button on the mouse. The curve is drawn on the screen using the current line color, style, and width. A fill function allows the user to color-in any enclosed region with either solids or patterns. Graphics text can be added to the static background in a choice of font, size, color, and orientation that are selected using the pull-down menus. An erase function that operates similar to the sketch function, erases all graphics and text in the path of a mouse-controlled eraser cursor as long as the left button on the mouse is held down. Examples of the various drawing functions are illustrated in Figure 3.

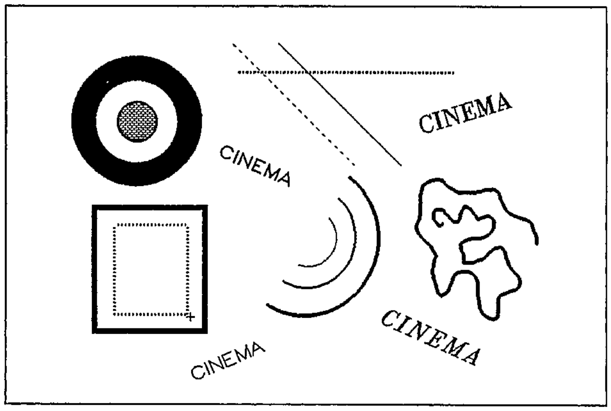

Figure 3: Examples of Static Background Drawing Functions

The Dynamic Component

A CINEMA animation is the dynamic display of objects that change location, color, or shape on a static background in correspondence with changes in a SIMAN simulation model. The changing objects constitute the dynamic component of the animation layout.

The dynamic objects in a CINEMA layout are directly tied to specific modeling constructs within the SIMAN simulation language. When SIMAN executes a simulation of a system, it is continually updating an internal representation of the state of the system. The state of the system is defined by the current value of status variables, the number and location of entities within the system, the values assigned to attributes of the entities, the status of the resources within the system, etc. Each dynamic object in an animation layout is associated with a specific element of the system state as represented by the SIMAN model. For example, one object might be associated with a status variable such as the simulated clock time. Another object might be associated with the status of a SIMAN resource. The dynamic objects in a CINEMA layout are automatically updated as the state of the system changes during the simulation. Following is a discussion of the dynamic objects which can be incorporated into a CINEMA layout and the association between these objects and specific modeling constructs within the SIMAN simulation language.

Entities - In a SIMAN simulation model, entities represent items which move 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 partially assembled car. As the entity moves from workstation to workstation within the SIMAN simulation model, its corresponding symbol is automatically moved across the static background in the CINEMA layout.

Entity symbols are created by coloring-in boxes on a screen sized grid using the mouse-controlled cursor. Each box in the grid corresponds to one picture element (pixel) of the actual symbol image. As the symbol is created on the grid, it is displayed in actual size in the upper left corner of screen (see Figure 4).

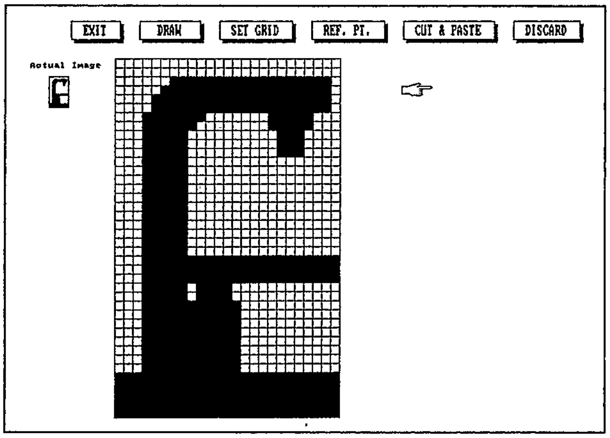

Figure 4: Symbol Drawing Grid

The symbols are created and stored in an **entity symbol library**. Symbol libraries are maintained separately from the animation layout and may be saved (stored on disk) and recalled at will.

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.

<u>Queues</u> - A CINEMA queue is a dynamic representation of an ordered list of entities residing in a file that is associated with a SIMAN QUEUE block. The entities residing in the file of an associated QUEUE block in the SIMAN model might represent workpieces awaiting the availability of a machine, set-up operator, 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. This graphical representation is then associated with a specific file number in the corresponding SIMAN simulation model.

When an entity enters a queue in the SIMAN simulation model, the entity's symbol is displayed along the corresponding queue symbol at the proper location relative to the other members of the queue. 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 display portion of the queue when preceeding entities exit.

<u>Resources</u> - 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, which changes its status to **idle** (making it available to be re-seized by other entities). Units of a resource can be removed from the system 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 using **resource symbols**. The resource is represented by four distinct symbols called the **idle**, **busy**, **inactive**, and **preempted symbols**. Like entity symbols, resource symbols are created by coloring-in boxes on a screen sized grid and stored in a **resource symbol library** that is maintained separately from the animation layout.

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. When the status of a resource changes in the SIMAN simulation model, the associated CINEMA resource symbol (busy, idle, inactive, or preempted) is displayed at the proper location on the screen to reflect the status change.

<u>Routes</u> - Distinct workstations within a system are modeled within SIMAN as STATION submodels. One method for modeling the movement of entities between STATION submodels is by using the ROUTE block. Dynamic **routes** are used in CINEMA to define a travel path on the static background of a layout that entities follow when they are routed between two SIMAN station submodels.

A route can be placed anywhere on the layout and consists of one or more "rubberbanded" line segments that are connected to form a path. The association with a route in the SIMAN model is established by entering the corresponding beginning and ending SIMAN station numbers. When an entity is routed between two stations in the simulation model, its associated entity symbol is continually redisplayed at new points along the predefined route path to produce the effect of movement.

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 workpieces processed, 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.

A second way to display a status variable is with a feature called **levels** which is an analog representation of the variable's value. Three different level shapes are included in CINEMA: a **box**, **circle**, and **dial**. During an animation, a box or circle 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 a screen sized grid and maintained in a separate **global symbol library**. To incorporate global symbols into a layout, the user associates selected symbols from a library with designated values of a specified system status variable. For example, one symbol could be displayed when the number in a queue is less than or equal to ten, and a second symbol displayed when the queue length exceeds ten.

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, 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.

RUNTIME FEATURES

The CINEMA system requires a special version of the SIMAN simulation language which incorporates the additional code required to dynamically update the graphics screen when a simulation is is executed with an animation. In addition, a series of mouse-controlled pull-down menus are provided for managing the exection of a simulation/animation. Other features in this special version as well as new features in the general release of SIMAN Version 3.0 allow users to interact with a simulation model while it is executing.

The CINEMA specific version of SIMAN includes the capability to switch between two or more animation layouts that might represent different parts of a simulation model that is executing. In addition, snapshots of the system status and graphics screen can be saved (stored on disk) and recalled later to restore the state of the system to a previous point in simulated time. Users can also control the run speed of an animation by specifying values for parameters that are used to scale simulated time to real screen update time. While the simulation/animation is executing, users can zoom-in on the layout by pressing the '+' key and pan to different areas of the zoomed-in layout by pressing one of the arrow keys (up,down,left,right). The "-" key can be pressed to zoom back out.

SIMAN Version 3.0 includes an interactive debugging facility that allows users to interrupt the execution of a model to examine and modify the values of system status variables. For example, machines can be broken down by decreasing the capacity of the correponding SIMAN resource. A CINEMA animation provides immediate visual feedback on the consequences of such changes.

SUMMARY

The development of the CINEMA system represents a significant contribution toward making animation an indispensible yet practical part of every simulation analysis. With minimal instruction and practice, a user with no programming skills can easily construct and execute a highly detailed animation. The mouse/menu interface between CINEMA and the user makes this task simple and efficient.

REFERENCES

1. Healy, K.J., R.V. Kolarik, and D.T. Guidry, "A Simulation and Animation of a Steel-Making Facility", Winter Simulation Conference, Nov. 1983.
2. Lampson, B.W., "Bravo Manual" in Alto Users Handbook, Xerox Palo Alto Research Center, Nov. 1978.

KEVIN J. HEALY is Vice President in Charge of Software Development at Systems Modeling Corporation. Prior to his current position, he was employed by The Timken Company of Canton Ohio as an Operations Research Analyst. He holds a Bachelor of Science in Industrial Engineering and a Master of Science in Industrial Engineering and Operations Research from the Pennsylvania State University.

Systems Modeling Corp.
226 Highland Ave., Suite D
State College, PA 16801
(814) 238-5919