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

GPSS/PC™ graphics and animation

Springer W. Cox
Minuteman Software
P. O. Box 171
Stow MA, 01775
(508 897-5662)

Abstract

GPSS/PC is a popular implementation of the discrete event simulation language GPSS, the General Purpose Simulation System. GPSS/PC has interactive graphics and animation tightly integrated into its simulation environment. Its graphics windows allow viewing and manipulation of the simulation via an optional pointing device, and assertion of all simulation primitives. All windows are online, providing for a visualization of model dynamics, and one of the windows allows animations of the simulation to be played out in an essentially unbounded virtual screen.

A CAD based animation postprocessor is now available which can create a detailed 3D animation based on a previously recorded GPSS/PC animation. Using the postprocessor, the animation space can be placed in 3D, and an arbitrary amount of detail can be added to the objects and layout of the animation. A variety of rendering and speedup options are then available.

This tutorial presents the interactive graphics, built-in animation, and CAD based postprocessor animation available with GPSS/PC.

Keywords: GPSS, discrete event simulation, interactive graphics, animation.

1.0 Introduction

Since the very beginnings of the simulation profession, it has always been true that the computing power available to any given simulation study could easily be exhausted. The need for careful monitoring of computer time and expense was impressed on simulation analysts from the very beginning. At first, the high cost of mainframe computation was the dominant force. This motivated the inclusion of Design of Experiments as an essential tool in the arsenal of techniques. One could not blindly execute a full factorial experiment without dire results.

As time has passed, although we find the cost of computation decreasing, our appetite for computer power grows ever more ravenous. Not only is there a clear need to examine more of the variations that would have been ignored in the past, but exciting new techniques are emerging to improve the realism and usefulness of our simulations. From the very beginning, our impressive increases in even stronger needs.

This trend is nowhere more evident than in the case of animation. Completely realistic visual animations tax even our most powerful supercomputers. Even so, there is a clear need to deliver dynamic visual representations of simulations on the smallest of machines. When we add the final requirement of animation in real time, the need for tradeoffs becomes clear.

Against this scenario we must consider that not all animations have the same objectives. A simulation analyst verifying his/her simulation does not need rendering and excessive detail in the debugging animation. On the other hand, a highly symbolic cacophony of flashing lights may be useless in a management presentation.

The present state of animation, then, is that there are several applications with differing objectives. The constraint of computational resources requires a different choice of design tradeoffs for each. This is why we see a wide range of animation features in simulation packages. In the case of GPSS/PC, the online animation, for use primarily by the analyst, is a character based, 2 dimensional representation. Although it is of low resolution, it is extremely fast, is based on low cost hardware, and is completely interactive. On the other hand, a high resolution CAD based animation is available for situations requiring detail. These animations can require much more preparation effort, but can be very convincing.

In creating an animation, one should begin by asserting the objective of the project. It is certainly the case that many simulations should not be animated at all. GPSS/PC offers many visual indicators in its Interactive Graphics Windows that obviate the need for any further animation. These windows allow the dynamics of the simulation to be observed at a single keystroke. Further, all windows are interactive, i.e. the simulation can be manipulated through them as well. If a spatial representation is desired, then the built-in online animation is available in a window called "the Positions Window". And finally, just about any amount of realism can be added by creating a CAD based animation using the optional postprocessor called the GPSS/PC Animator™

Each of these options is discussed further, in the sections that follow.

2.0 GPSS/PC Windows

Version 2 of GPSS/PC now includes 6 interactive graphics windows, each of which allows the user to view, and interact with, a specific GPSS entity type. The user can use a pointing device to select entities, positions, and menu items in the graphics windows. Each window type is updated online and reflects the state of the running simulation. The user can open a window with a single keystroke, even while a simulation is running, or he/she can use the WINDOW command for more precise control. Optionally, the windows can be saved or sent to a hardcopy device.

With respect to GPSS/PC, the phrase online animation is restricted to the Positions window and its associated features; the animation postprocessor is a separate option. The other five online graphics windows (Blocks, Storages, Facilities, Matrices, and Tables) provide alternate viewpoints of simulation dynamics based on specific entity types which are provided by GPSS. They, with the Positions Window, are collectively called the interactive graphics windows.

The Data Window is not considered to be a graphics window, and does not enjoy online update, even though it is part of the integrated environment. This is an advantage in that the Data Window is the fastest of all windows. If the user want to "skip ahead" or complete a simulation as quickly as possible, he/she should open the Data Window.

2.1 Microwindows

A Microwindow is a small window which shows the current value of any GPSS state variable (SNA) and an optional title. Up to 4 Microwindows may be opened within any major graphics window.

Microwindows are opened and closed by the MICROWINDOW command. They are visible at the right side of each of the graphics windows and are updated as the simulation runs. Using Microwindows, the user can view his/her choice of state variables, regardless of which major graphics window is open.

2.2 Interactions through the Major Graphics Windows

In line with GPSS/PC's design objectives, a high level of interactivity is available to the user. This level far exceeds that available in compiled languages. A pointing device can be used to interact with entities visible in any of the windows. This level of manipulation can run deep into the structure of the simulation. For example, during execution of the simulation, icons can be moved or modified in the Positions Window, and blocks can be insert, changed, or deleted in the Blocks Window. The general procedure is to select an entity on the screen with the pointing device and then to select the menu item which initiates the intended action.

All windows allow the user to single step or to continue an interrupted simulation using the pointing device. However, a much more powerful set of controls is available through standard keyboard entry. GPSS/PC allows the user to type any GPSS blocks statement through the keyboard. This mode, called Manual Simulation, creates a temporary block and attempts to pass the Active Transaction through that block. In this manner, the full range of simulation primitives is raised to the interactive user interface.

The Blocks Window and the Positions Window have, in addition, several powerful menu functions which allow online manipulation of animations with the pointing device. These windows are discussed next.

2.2.1 The Blocks Window

The most powerful control of the simulation is available through the Blocks Window. This window maintains a one to one correspondence with GPSS block statements in the source code. The user may develop the program working either with GPSS text in the Data Window or inserting blocks in the Blocks Window. The combination of breakpointing, stepping, and continuation, allows much of the interaction during the testing/verification phase to be done without typing commands, i.e. using only the pointing device.

Blocks are arranged in the window top-to-bottom, left-to right. The Blocks Window shows the flow of transactions from block to block, and it has indicators for the occupancy counts of transactions in blocks. An alternate view of the Blocks Window shows the accumulated count of transaction entries for each block. This represents a simple "history" of the simulation and is useful in verification and problem determination.

The color of a block is determined by the number of transactions it contains. This draws attention to sources of congestion in the simulation. If the Blocks Window is open while the simulation runs, each transaction block entry causes the block representation to flash a high intensity version of its current color.

The user can interact with the Blocks Window using the pointing device. Several of the menu items in the Blocks Window require that the user select a block before selecting the menu item. For example, to EDIT a block, the user first selects the block on the screen and then select the menu item, EDIT.

The Blocks Window Menu

The Blocks Window menu has 7 items which can be selected with the pointing device:

CONTINUE - Resume the simulation until a stop or end condition is detected.

STEP - Attempt 1 block entry, then stop.

STOP - Set a stop condition on the last block selected.

UNSTOP - Remove all stop conditions.

EDIT - Edit the GPSS statement associated with the last block selected.

INSERT - Prepare to insert a block immediately after the last block selected.

DELETE - Delete the last block selected, from both the current model and the savable program.

2.2.2 The Other Interactive Graphics Windows

With the exception of the Positions Window, the other interactive graphics windows are shown in Figures 6 through 9. They are used to observe and interact with GPSS facilities, matrices, storages, and tables. All windows can show the changing state of the running simulation. For example, the convergence of frequency histograms toward parent distributions can be observed visually in the Tables Window.

These Window menus have 2 items which can be selected with the pointing device:

CONTINUE - Resume the simulation until a stop or end condition is detected.

STEP - Attempt 1 block entry, then stop.

3.0 Animation

The Positions Window provides for the coupling of a visual animation to the simulation. This feature allows the user to define layouts and icons, then to control their behavior in a dynamic pictorial representation.

According to GPSS/PC nomenclature, the word animation is reserved for the representation of a simulation in the Positions Window by user chosen shapes moving, at user control, with at least two degrees of freedom. Moving bar charts, digital readouts, and other dynamic screen indicators are available in the other interactive graphics windows but are not considered true animation.

Animation in GPSS/PC has been tightly integrated into the GPSS language. It can be used as a pictorial representation of the state of running simulations, it can impose additional constraints upon objects in the simulation, and it can provide a vehicle for manipulating the simulation itself. Consistent with the design goal of inexpensive hardware requirements, Version 2 of GPSS/PC runs its graphics in character mode at the same time allowing users to create shapes and layouts from bitmaps. The result is a reasonable speed of animation and yet few hardware and software requirements, thereby keeping the cost of the total system to a minimum.

3.1 The Positions Window

The Positions Window permits a true animation to be played out over an unbounded two dimensional matrix called the "Virtual Screen". Each view of a segment of the Virtual Screen represents a submatrix which is addressed by row and column coordinates.

As with the other interactive graphics windows, simulation objects may be manipulated in this window using the pointing device. Icons associated with GPSS transactions may be moved or redirected, or they may be set up for the Manual Simulation mode of GPSS/PC, which allows the application of any GPSS block statement to that transaction. Any such changes are immediately effective throughout the simulation environment.

The control of an animation is extremely straight-forward. With Version 2 of GPSS/PC, GPSS transactions can now have the attributes of POSITION, SHAPE, and COLOR. By proper assignment of attributes the user can create a pictorial representation of the state of the system, or can animate a simulation using his/her own defined shapes.

The Virtual Screen is different from other matrices in that row and column coordinates can be zero or negative. Due to the unlimited precision of GPSS/PC numeric values, the Virtual Screen is essentially unbounded. The Positions Window, however, is limited to a 17 by 80 submatrix (17 by 70 if a Microwindow is open) view of the Virtual Screen. The Positions Window is like a camera. It can be moved anywhere on the Virtual Screen as long as no coordinate exceeds 10^{16} in magnitude. The user can move the Positions Window with paging keys, and can open the Positions Window to any specific place using the WINDOW command.

GPSS/PC can accommodate extremely large background layouts. The user can define one or more background layouts anywhere within a 10^{16} by 10^{16} segment of the Virtual Screen. With such an arrangement it is possible to run several animations independently of each other in remote parts of the Virtual Screen

The user can interact with the Positions Window using the pointing device. Several of the menu items in the Positions Window require that the user first select a transaction or position, and then a menu item.

The user can select a transaction by pointing to its icon and triggering a Selection Event. This causes an audio signal, the icon to change to a diamond and to be highlighted, and the transaction number to appear in the Status Line of the Command Window. For example, to RELOCATE a transaction, the user first selects the transaction's icon on the screen, then selects the new position, and then selects the menu item, RELOCATE. GPSS/PC immediately moves the icon on the screen and updates all the internal data within the simulation to reflect the change.

The Positions Window menu has 7 items which can be selected with the pointing device:

CONTINUE - Resume the simulation until a stop or end condition is detected.

STEP - Attempt 1 block entry, then stop.

STOP - Set a stop condition on the last transaction selected.

UNSTOP - Remove all stop conditions.

PULL - Assign a new desired position to the selected transaction. This is used in Collision Prevention Mode.

RELOCATE - Assign a new current position to the selected transaction. This causes an immediate move to the new position.

3.2 Controlling the Animation

The programming required for animation is quite straight forward. GPSS transactions now have attributes of shape, color, position, and--in the case of Collision Prevention Mode--intended position. The animation is controlled by maintaining the proper transaction attributes within the GPSS simulation.

There are two distinct modes of operating the animation. In the Direct Mode of operation, the user explicitly sets the appearance and positions of icons in the Virtual Screen. Such changes become visible in the next Move Event, defined below. Alternately, in Collision Prevention Mode, GPSS/PC will handle all of the collision detection and prevention automatically. This minimizes the programming required of the user to animate simulations.

3.2.1 Collision Prevention Mode

In Collision Prevention Mode, the intended position of icons in the virtual screen is asserted when the user loads the TO_ROW and TO_COLUMN attributes (transaction parameters) of the transaction associated with each specific icon. Then on each "Move Event" GPSS/PC moves each icon in the virtual screen no more than one position toward its "intended position". Within the simulation, any transaction can test its current position or wait for arrival at its intended position. This is easy to do because the current position is automatically maintained in the transaction parameters with standard names ROW and COLUMN. Even offscreen positions are maintained and collisions prevented.

A "Move Event" occurs when a transaction enters a MOVE block or (optionally) as the first implicit event in each new simulated time instant. Transactions in the special GPSS transaction group named POSITIONS are scanned repeatedly during a "Move Event" until no icons are able to move. However, no icon moves move than once. In Collision Prevention Mode, the current position coordinates are automatically maintained by the software in the transaction parameters with standard names ROW and COLUMN, and the intended position coordinates are maintained by the user's GPSS program in the transaction parameters TO_ROW and TO_COLUMN. Each Move Event represents another attempt by the software to move the transaction's icon one more step in the direction of the intended position.

Implicit in relinquishing control of collision prevention is the acceptance of several rules of movement. During a Move Event" in Collision Prevention Mode, GPSS/PC attempts to move each icon participating in the animation one position in a direction which decreases the distance to the intended position. The software automatically chooses a direction which avoids collisions with other icons or with the background layout. Diagonal movement is preferred to horizontal, and horizontal movement is preferred to vertical.

Some of the consequences of these simple rules of movement are:

1. When an icon is in the same row (column) as its intended destination, it does not leave it. This leads to orderly queuing when the intended position is aligned rowwise or columnwise to the current position.

2. When an icon is in a different row and column as its intended destination, it first attempts to move diagonally, then horizontally, then vertically. This tends to create bunching.

3. Transactions are scanned in the order they joined the POSITIONS transaction group. This implies a built-in priority of senior group members over newer ones.

Icons can form trains at intersections because of the precedence within the transaction group POSITIONS. This can be changed to alternate access of the intersection by using a GPSS facility to

control intersection. With this method, each transaction must arrive at the position next to the intersection, and then acquire the facility representing the intersection, before moving on and releasing the intersection. If a more detailed level of control is desired, the GPSS LINK and UNLINK blocks can be used.

If the user cannot accept the built-in rules of movement, he/she must explicitly implement collision prevention by asserting GPSS statements. This is done by specifically controlling the positions of icons while operating in Direct Mode instead of Collision Prevention Mode.

3.2.2 Direct Mode Animation Methods

In Direct Mode, the user is responsible for implementing collision detection and prevention. The built-in ROW and COLUMN transaction parameters are set in the user's GPSS program. Similarly, the COLOR and SHAPE parameters are set by the user. During the next Move event, the associated icon appears in the proper shape, color, and position. With explicit control of all attributes, Direct Mode offers the user the ability to control individual speeds and accelerations.

There are many ways to implement a Direct Mode animation, and, in general the best animations result when the user can take advantage of the specifics of his/her problem. Three common ways to implement collision prevention in Direct Mode are:

Grid control. Divide the active part of the Virtual Screen into an imaginary checkerboard, and associate a GPSS facility with each square. Then, require a transaction to acquire ownership of the associated facility before moving into a square. However, the use of multiple general purpose resources to control collisions imposes a limit on the usable extent of the Virtual Screen, because of computer memory requirements.

Parallel matrix. Maintain a matrix of positions to reflect the current state of the Virtual Screen. Movements can be tested for feasibility by reference to the state description maintained in the parallel matrix. In GPSS/PC the matrix can be implemented internally or in FORTRAN through the HELP block interface. The FORTRAN implementation can access a matrix of several million elements in the extended memory of IBM AT compatible computers. This method can require a significant amount of computer time to maintain and retrieve state information. In addition, it also imposes a limit to the usable part of the Virtual Screen.

Calculate destinations, and adjust waiting lines. Use GPSS entities to determine the occupancy of queues, for queue entry, etc. and use the GPSS ALTER block to adjust the waiting lines when a transaction leaves.

Direct Mode has other difficulties beyond the more involved programming. Although it allows explicit control of the details of the animation, and often produces a faster simulation, it can also greatly increase the computer memory requirements of the simulation. Each grid element can control many positions, but only one icon can occupy a grid element at any instant. Such limitations are not peculiar to GPSS, however notice that there are no such limits in GPSS/PC in Collision Prevention Mode. The OS/2 version of GPSS/PC relaxes these memory restrictions.

4.0 The GPSS/PC ANIMATOR^{tm}

The GPSS/PC Animator is comprised of the PREPARE program, and "The CAD Animator". The latter is a set of AutoLISP programs to be run with the AutoCAD program from Autodesk, Inc.

GPSS/PC Version 2 now has the ability to trace animations automatically. Special operands have been added to the MOVE block, and the ANITRACE (ANImation TRACE) command has been implemented for detailed control of the trace. By turning on the trace, one or more initial animation trace files are produced, which describe the dynamics of the animation.

The GPSS/PC Animator is a post processing system which provides for a variety of options for viewing the animation. The GPSS/PC Animator, itself, provides facilities which allow one to generate a wireframe animation using a compatible version of AutoCAD.

In addition, the CAD Animator can create a sequence of files which are suitable for rendering and "Movie" creation by the AutoShade and AutoFlix programs, respectively. The wireframe animation is presented by the CAD Animator, in some cases precluding the optional processing by the AutoShade and AutoFlix programs. The AutoCAD program, with AutoLISP, is *required* for operation of the GPSS/PC Animator.

The preparation of rendered movies requires several steps:

1. Creation of a GPSS/PC Animation. (Using built-in GPSS/PC Features)

2. Creating 3-D wireframe layout and moving shapes. (Using AutoCAD)

3. Creating an initial trace of the animation. (GPSS/PC ANITRACE Command)

4. Creating a prepared trace of the animation. (GPSS/PC ANIMATOR's PREPARE program)

5. Generating and editing an initial wireframe animation, and, optionally, creating a sequence of Filmroll, Slide, or Drawing files. (The CAD Animator)

6. Rendering the sequence of Filmroll files. (AutoShade) This creates a sequence of "Rendering" files.

7. Compiling a rendered movie. (AutoFlix) This creates a "Movie" file.

8. Projecting the Rendered Movie. (AutoFlix)

Although the creation of a rendered movie takes several steps and a lot of computer time, users who do not need the additional realism can skip the rendering step. The CAD Animator can generate a sequence of "Slide" files for reading by AutoFlix in the movie compilation step. This bypasses the rendering step, and produces a relatively smooth wireframe movie.

Another option is to stop at the initial wireframe animation generated when the CAD Animator is run. This is the least expensive of the options, in terms of computer time, but still may be sufficient in some cases.

There are several configuration requirements in addition to those imposed by GPSS/PC and AutoCAD. Although the GPSS/PC Animator requires only AutoCAD with AutoLISP, the AutoFlix program imposes additional requirements. At present (Summer 1988), it requires AutoCAD R9 (or compatible), an IBM compatible EGA display adapter with 256K, and an enhanced monitor. The Movie Projection step can be speeded with a fast processor and a lot of RAM memory. A minimum of 1 megabyte is recommended.

4.1 The Initial Animation Trace

Once an initial GPSS/PC animation has been successfully run, the creation of the initial animation trace is a simple matter. One need only use the ANITRACE command in the GPSS/PC Session to cause the automatic creation of the trace file(s). If these files are put on a succession of diskettes, GPSS/PC allows the user to switch media during the simulation.

4.1 Trace Preparation

The PREPARE program is part of The GPSS/PC Animator. It converts an initial animation trace to a "Prepared" trace, which can be read by the CAD Animator. At this time the Virtual Screen of GPSS/PC can be inserted into the 3D CAD space in a variety of ways. It is possible to translate, rotate, scale, and/or shear the original rectilinear space by an easy-to-use 3 point placement method. The PREPARE program has a batch mode of operation which can run without the user interacting with the program, at all.

The Prepared Trace output file(s) are then suitable for reading by the CAD Animator's AutoLISP commands.

4.2 The Initial Wireframe Animation

"The CAD Animator" is the other part of the GPSS/PC Animator. It consists of a set of AutoLISP programs.

After a Prepared Trace is available, and AutoCAD drawing files are available for the layout and for the moving objects, it is time to play out the initial wireframe animation. The CAD Animator reads the Prepared trace, generating a wireframe animation.

The commands available at this step are:

1. CYCLE - Start or Continue the Animation with continual rerun.

2. GO - Start or Continue the animation.

3. FAST FWD - Fast forward to a specified simulated time.

4. RECORD - Create a sequence of Drawing, Filmroll, or Slide files for further processing.

5. REWIND - Go back to the beginning of the Animation Trace.

6. SETDELAY - Insert timed delays between each frame when presenting the wireframe animation on the screen.

7. STOPAT - Cause the animation to stop at a specified simulated time.

8. WHEN - Display the current simulated time.

The initial wireframe animation created by the CAD Animator may produce a satisfactory level of visualization, in some cases. Then, no further processing is required.

If the user chooses to process the animation further, the CAD Animator can create "Filmroll" files for rendering by AutoShade, "Slide" files for direct playback by Autoflix, or "Drawing" files for adjustment and smoothing by AutoFlix's AutoLISP commands.

4.3 Rendering

The optional AutoShade program is able to provide considerable realism to the animation. Such rendering provides surface details, lighting, and

camera positioning to the animation. To use this option, one commands the CAD Animator to create a sequence of "Filmroll" files, one for each frame in the animation.

Rendering takes a lot of computer time. Often, it is best run overnight. The AutoShade program converts each frame in the initial animation into a frame represented in a "Rendering" (RND) file. The rendered frames are then used to compile the movie. The movie compilation step is performed by the Autoflix program.

4.4 Movie Compilation

The AutoFlix program can compile a "Movie" file from a sequence of "Rendering", "Slide", "Drawing", and/or Text files. One can add delays, dissolves, overlays, speed changes, and interactive choices of frame sequences. Further, a musical score can be played in accompaniment. A variety of sequencing commands are available to control the presentation. These include CALL, RETURN, and GOTO. With a little preparation, screen buttons may be inserted, allowing the user to direct the course of the movie when it is projected.

4.5 Movie Projection

The final step is to use the AutoFlix program to play out the movie created in the Movie Compilation step. At this time, the musical score, if any, accompanies the presentation. Optionally, the viewer(s) can control the sequence of scenes by picking user-defined buttons on the screen. In this manner, a reasonably fast animation can be played out.

As with animation systems in general, memory and disk space tend to set practical limits on the duration of the movie. These limits will become looser in the future. Even now, computer projected movies of several minutes are easily possible, and a great deal of realism and flexibility, including special effects, are available to contribute to the effectiveness of the presentation.

4.6 Conclusions

There are many advantages to using a CAD based postprocessor to polish animations intended for presentation to colleagues and management. One advantage is that all of the sophisticated drawing facilities are available for adding as much detail as desired to the background, or to the dynamic elements in the animation. Another is the use of 2 1/2 and 3 D representations add to the realism of the presentation. Not only that, there are additional options for rendering, speed control, and special effects that enhance the overall realism to an even greater level.

The future bodes well for users of animated simulations. In the very near future will see a significant increase in the animation power of both hardware and simulation software. Not only that, but reduction in the costs of animation will make such techniques feasible in even low budget simulation projects.