Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

THE INTERACTIVE GRAPHICS AND ANIMATION OF GPSS/PC™

Springer W. Cox
Minuteman Software
P. O. Box 171
Stow MA, 01775
(617 897-5662)

ABSTRACT

GPSS/PC is an implementation of GPSS, the General Purpose Simulation System. It was created primarily as an interactive simulation environment for personal computers and dedicated workstations.

GPSS/PC is distinguished by its visibility and interactivity. With the introduction of Version 2, it has added interactive graphics and animation to its simulation environment. Its graphics windows allow manipulation via an optional pointing device, and each window retains full controllability over the simulation. In addition, a CAD based animation post-processor option lends considerable realism to previously recorded animations, allowing panning, zooming, and a changeable 3D viewpoint.

This tutorial is an introduction to the GPSS/PC simulation environment, and includes a discussion of the Collision Prevention Mode of animation.

1. INTRODUCTION

We begin with GPSS/PC's integration, interactivity, error prevention, and model development aids. We then turn to the interactive graphics and animation provided in Version 2 of GPSS/PC, and end by considering, in detail, the use of the Automatic Collision Prevention mode of animations. The reader is referred to Schriber (1987), in these proceedings, for an introduction to the GPSS language.

The development of GPSS/PC was begun in 1981. It was designed from the beginning to provide a controllable simulation environment, and to relieve the Simulationist from the distracting minutiae required by many existing systems. In order to do this, a restructuring of the the model building process was required.

The Session. In GPSS/PC, all phases, including compilation, link, run, and debug, are combined into a single phase called a session. This eliminates the need for a reedit/recompilation/relink/ rerun/ redebug cycle every time a correction or other change is made to the model. Once a program is read from a file or entered through the keyboard, in effect, the compilation and link phases are done. GPSS/PC builds the appropriate data structures in the memory of the computer when each statement is scanned. New block statements, originating from the keyboard or from other program files are automatically incorporated into the existing data structures.

The GPSS/PC session is made more powerful by the inclusion of features which allow immediate modification to the data structures of the simulation. First, any named value may be changed at any time and the results may be observed immediately, without restarting the simulation. Second, GPSS blocks may be inserted, replaced, or deleted. Third, the manual simulation feature places all GPSS block statements on the user interface. This means that any GPSS block statement can be entered through the keyboard in the middle of simulation. This causes a temporary GPSS block entity to be created which the active transaction attempts to enter. In this manner, the complete range of statements is available for keyboard entry. In addition, using the interactive graphics windows, discussed below, much of the interaction can be directed with an optional pointing device such as a mouse, or light pen.

To the Modeler, the single phase design greatly improves the immediacy of the simulation environment. No longer is it necessary to leave the run time phase, passing through edit, compile, link, and debug phases in order to correct a problem. The GPSS/PC session allows corrections to be made immediately, and the simulation to be continued without the need for a restart. In model development, one proceeds from one problem to the next, without the intervening overhead steps.

User Oriented Features. Many features of GPSS/PC were designed to relieve the Modeler of tedious error-prone activities. In fact, a hierarchy of anti-error defenses is designed into the user interface. Not only are modifications effective immediately, but errors are detected and treated as soon as possible. GPSS/PC introduces a new feature call "keystroke error prevention" to push the immediacy of error detection. The GPSS/PC statement parser is directed by the grammar of GPSS and at any instant will refuse keystrokes which cannot possibly lead to correct syntax. This makes it impossible for the Modeler to enter syntax errors through the keyboard.

Although semantic and logic errors are not prevented, they can be corrected interactively, with no need for compilation. In addition, the visibility of the simulation through the graphics windows, discussed below, is an important aid to problem detection and diagnosis. Rather than study only the results of a simulation, the Modeler can take an active part in exploring the running simulation. Not only does this aid in testing and verification, but it tends to convey an intuitive feel for the behavior of the system being simulated. It is often the case that the understanding gained by the Modeler during a simulation project is more valuable than the simulation results, per se.

An irregular user interface tends to be unpredictable in practice, and requires a relatively long educational period. Each exceptional rule or condition represents yet another item that must be learned by the user. One approach to reducing the "knowledge load" of the Modeler is to remove exceptional conditions and unnecessary data types, whenever possible. GPSS/PC uses unlimited precision integers for internal values. As numbers grow in size, additional memory is allocated for them. Unlimited numerical precision relieves the Modeler of the worry about overflows and underflows in the clock, and of overflows in the statistics accumulators. These exceptional events do not occur.

The unlimited precision feature of GPSS/PC also eliminates the need for a choice of data types. Not only are multiple data types unnecessary, they are now undesirable. The Modeler no longer must make the choice when a GPSS statement is written—a further reduction in the programming load imposed on the Modeler. Also, the need to worry about unexpected overflows and underflows is removed.

The unlimited precision arithmetic in GPSS/PC has some advantages over systems using floating point arithmetic. When using microseconds as time units, even with double precision, floating point accumulators for calculations of standard deviations can begin to lose accuracy after less than an hour of simulated time. In GPSS/PC, on the other hand, the Modeler can make the time granularity as fine as desired without worrying about clock overflows or underflows.

GPSS/PC has features which automate many of the trivial tasks required during a simulation project. These usability features include:

- Command recognition. If the space bar is pressed when a command is partially, but uniquely determined, GPSS/PC™ finishes the command word automatically.

- Keystroke error prevention. You CANNOT enter invalid syntax into an operand field.

- Assignable function keys. All statements and commands can be saved by a single keystroke and thereafter be re-entered with a single keystroke.

- Automatic spacing. You need no longer be concerned with aligning statement fields with column numbers. This occurs automatically when a valid delimiter key is pressed.

- Cursor prompting. Each field is prompted by a distinct cursor which serves as a reminder.

- Fractional line numbers. It is easy to insert lines without renumbering using decimal fractions. A RENUMBER command is available also.

- Integrated line editor. A powerful line editor is available for small changes to line-numbered (saved) statements.

- Automatic numeric precision. There is no need to declare data types. All have unlimited value and precision.

- Run Time Control. Powerful START-STEP-STOP commands are available for manipulating simulations. In addition, TRACE blocks may be inserted into the simulation, of may be used in manual simulation mode, in order to trace specific transactions. All commands and statements may be assigned to a function key. Thereafter, a single keystroke recalls the complete statement. Also, much of the debug phase can be directed by a pointing device.

- Integrated Online Help. By pressing the ? key at any point, a context sensitive help message will appear, listing the options available for the current item.

2. GPSS/PC WINDOWS

Version 2 of GPSS/PC now includes 6 interactive graphics windows, each of which allows the Modeler to view, and interact with, a specific GPSS entity type. The Modeler can use a pointing device to select entities, positions, and menu items in the graphics windows. Figures 1 and 2 show the use of a mouse or light pen as the optional pointing device.

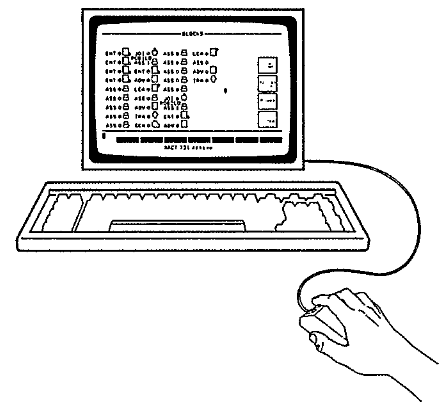

Figure 1. Using a Mouse as the Pointing Device

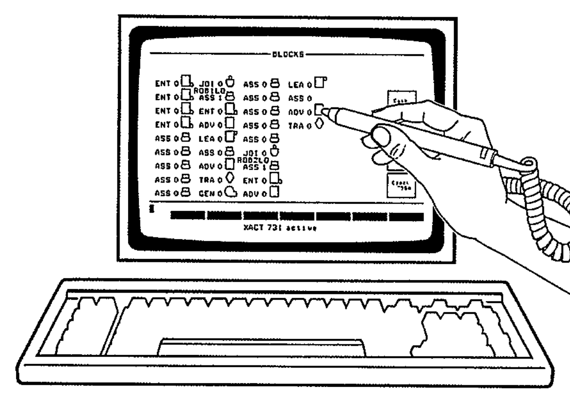

Figure 2. Using a Light Pen as the Pointing Device

Each window type is updated online and reflects the state of the running simulation. The Modeler 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 post-processor 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 Modeler wants 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.

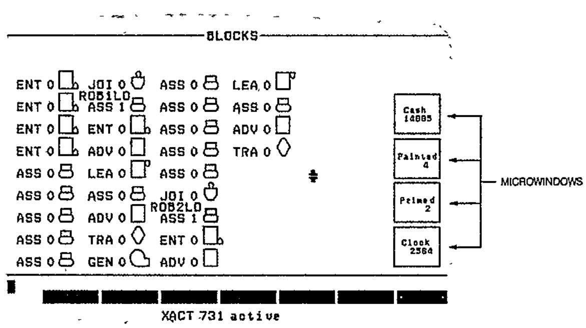

Figure 3. GPSS/PC Microwindows

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 Modeler 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 Modeler. 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 inserted, 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 Modeler 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 Modeler to type any GPSS block 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 actions can be initiated from the keyboard.

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.

The Blocks Window. The most powerful control of the simulation is available through the Blocks Window (See Figure 4). This window maintains a one to one correspondence with GPSS block statements in the source code. The Modeler may develop the program working either with GPSS text in the Data Window or by selecting block insertion points 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.

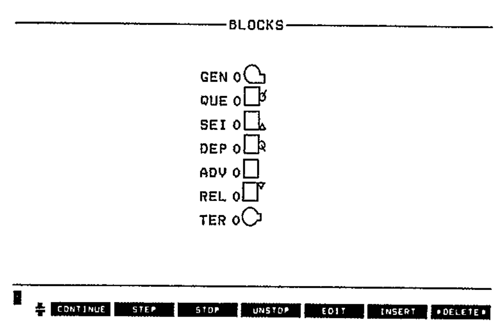

Figure 4. The Blocks Window

Figure 4 shows an example of the Blocks Window where the current block diagram of the simulation may be viewed. 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.

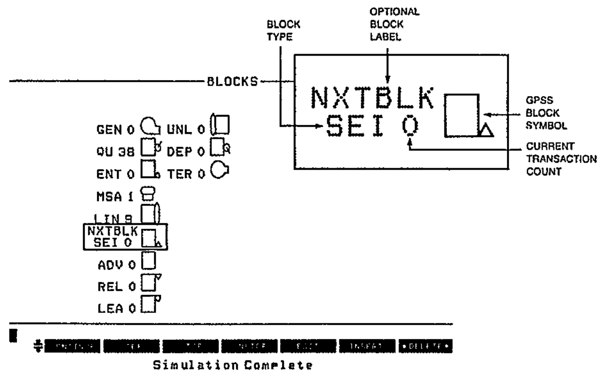

Figure 5. Information Cells in the Blocks Window

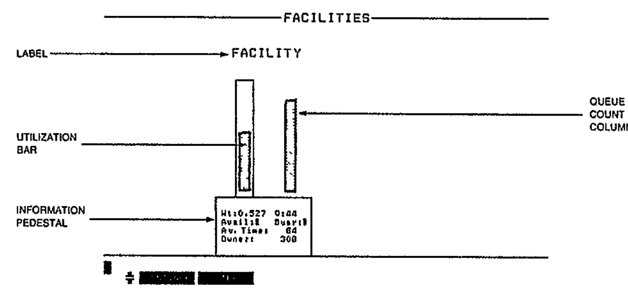

Figure 6. The Facilities Window

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

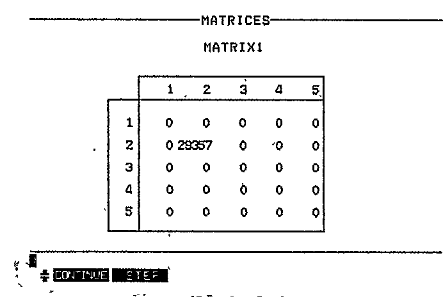

Figure 7. The Matrices Window

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.

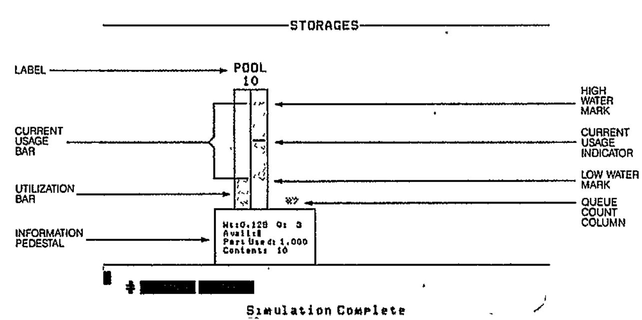

Figure 8. The Storages Window

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.

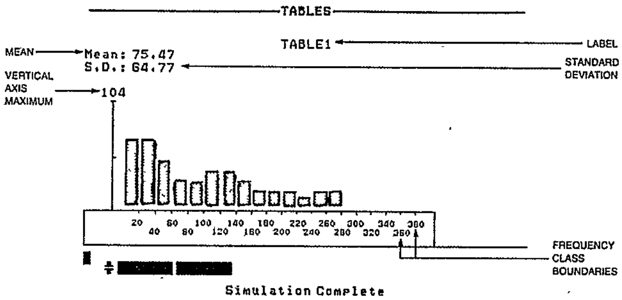

Figure 9. The Tables Window

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, shown in Figure 10, provides for the coupling of a visual animation to the simulation. This feature allows the Modeler to define layouts and icons, then to control their behavior in a dynamic pictorial representation.

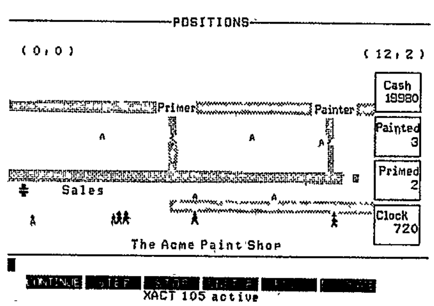

Figure 10. The Positions Window

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 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 Modelers 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. Additional detail can be added to the animation after the simulation by the optional post-processor.

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 Modeler 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, and are effectively unbounded. 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 can be moved like a camera. The Modeler can move the it with paging keys, or to to any specific place using the WINDOW command.

GPSS/PC can accommodate extremely large background layouts. Also, since the Virtual Screen is effectively unbounded, it is possible to run several animations independently of each other in remote parts of the Virtual Screen

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

The Modeler 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 Modeler 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 - Redirect an icon in Collision Prevention Mode.
- RELOCATE - Move an icon.

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 Modeler 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 Modeler to animate simulations.

Collision Prevention Mode. In Collision Prevention Mode, the intended position of icons in the virtual screen is asserted when the Modeler 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 Modeler'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. However, once an icon is in the same row (column) as its destination, the icon does not move out of it.

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

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.

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 row-wise or column-wise to the current position.

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 the 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 Modeler 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.

An Example of Collision Prevention Mode. The movement of icons in the Positions Window is an entirely different matter from the movement of transactions from block to block within the simulation. The real action is occurring in the Blocks Window, where the animation is controlled by the manipulation of transaction parameters in ASSIGN blocks. In Collision Prevention Mode, the Modeler places destination coordinates into the TO_ROW and TO_COLUMN pair of transaction parameters, and can either waits or tests for the current position of the transaction. On every MOVE event thereafter, the system updates the current position of the transaction as indicated by the ROW and COLUMN pair. To get this started, the Modeler specifies the starting position of the transaction by initializing the ROW and COLUMN parameters just once.

A complete GPSS/PC program file, which uses Collision Prevention Mode, is listed in the appendix. In this simulation car bodies are first painted by a priming robot, then painted by a finishing robot, and then sold to customers, who enter from the right. The robots both have a rest period when they are unavailable for painting. All activities are controlled by time durations which can be altered during the simulation. Similarly, you can interact with cars, robots, or customers using menu items or manual simulation. Four Microwindows are opened for viewing state variables: the primed car body inventory, the finished car inventory, and the current cash flow.

Statement number 230 is especially noteworthy. It defines a GPSS variable entity which is nonzero (True) only when the present location of a transaction, as indicated in the system maintained ROW and COLUMN parameters, is the same as the "desired position", as indicated in the Modeler maintained TO_ROW and TO_COLUMN parameters. This variable, named ARRIVE, is used in the program as a signal for an arrival event. Transactions set up a destination and then "go to sleep" awaiting entry into a GPSS Refuse Mode TEST block (see statements 274, 320, 410, 460, 520, 620, and 780). The event of entry into any of these TEST blocks is identical to the event of arrival at the destination.

Transactions can test for the condition of arrival as well as wait for the event of arrival. To test for, but not wait for, the condition of arrival, one can use an alternate exit TEST block specifying the same ARRIVE GPSS variable. Similarly, at any time in the simulation, a transaction can interrogate its current position.

Direct Mode. The Direct Mode of animation in GPSS/PC is different. Using it, the Modeler is responsible for implementing whatever form of collision detection and prevention he/she requires. In Direct Mode, the built-in TO_ROW, TO_COLUMN, COLOR, and SHAPE transaction parameters are set in the Modeler's GPSS program. During the next Move event, the associated icon appears immediately in the proper shape, color, and position. With explicit control of all attributes, Direct Mode offers the Modeler the ability to control individual speeds and accelerations.

The details of Direct mode are not within the scope of this paper. A more extensive treatment may be found in Cox (1987).

Presenting the animation. A whole new range of possibilities is provided by the CAD based animation post-processor option of GPSS/PC Version 2. The Modeler can create extremely detailed 3D layouts and shapes, to be driven in animation by a traced simulation. Such animations can be quite realistic, and they are interactive with respect to altering objects and changing the 3D viewpoint.

The efficient character mode animation provided by the Positions Window is available even without the optional post-processor. It is online, i.e. it operates as the simulation unfolds, and it allows the simulation to be manipulated through the animation window. The CAD based post-processor does not allow this.

Each type of animation has its place. The character based animation offered by the Positions Window, is efficient, interactive, and controllable. The CAD based animation post-processor imposes a heavier CPU load due to its relative realism, and is therefore separated from the simulation itself. Although all the information about the simulation is available online through the animation in the Positions Window, the more realistic post-processor may lead to a better appreciation of the results by colleagues, management, and clients.

Caveats. Modelers should probably not even consider presenting an animation to colleagues until the statistical properties of the simulation are understood. An animation is only a single outcome, and it may be unrepresentative and/or unlikely. On the other hand, many, perhaps most, simulations do not really need to be animated. In the case of GPSS/PC, the interactive graphics windows are available at a single keystroke, with no requirement for additional animation control statements.

4.0 SUMMARY

GPSS/PC has been available since 1984 from Minuteman Software, Stow, Massachusetts. It was designed to provide a highly interactive simulation environment that shielded its users from errors and housekeeping minutiae. Version 2, first introduced in 1986, added interactive graphics and animation to the integrated environment. These features extended the two way communication between the Modeler and the simulation. The CAD based animation post-processor, an new optional feature of GPSS/PC Version 2, allows very detailed 3D animations to be composed.

This brief tutorial has discussed several of the features of the GPSS/PC environment. It presented the interactive graphics windows and animation available in Version 2, and it considered the details of creating an animation in Collision Prevention Mode.

5.0 REFERENCES AND BIBLIOGRAPHY

Cox, S. W. (1984), "The User Interface of GPSS/PC", *Proceedings of the 1984 Winter Simulation Conference,* (S. Sheppard, U. W. Pooch, and C. D. Pegden, eds.). Institute of Electrical and Electronics Engineers, San Francisco, California, 545-552.

Cox, S. W. and Cox, A. J. (1985), "GPSS/PC: A User Oriented Simulation System", *Modeling and Simulation on Microcomputers: 1985,* (R. G. Lavery, ed.). The Society for Computer Simulation, San Diego, California, 48-50.

Cox, S. W. (1987), "Interactive Graphics in GPSS/PC", *Simulation,* **49,** No. 3. The Society for Computer Simulation, San Diego, California.

Schriber, T. J. (1987), "Introduction to GPSS", *Proceedings of the 1987 Winter Simulation Conference,* Institute of Electrical and Electronics Engineers, San Francisco, California.

Springer Cox is president of Minuteman Software, of Stow, Massachusetts. He has been involved with computer simulation for over 15 years. He started working at IBM, in 1968, on the prediction of the behavior of computer systems. In 1973, he moved to Xerox to work on the performance evaluation and modelling of a national order-entry network. In 1976, he left Xerox to join the Corporate Research and Development group at DEC, working on the simulation of virtual memory operating systems. In 1982, he founded MINUTEMAN Software for the purpose of developing and supporting GPSS/PCs/tm/, an interactive implementation of the simulation language GPSS.

He took his degrees in Physics and Computer Science at Cornell University, and Syracuse University, respectively. He has published papers in the areas of computer performance evaluation, modelling, and simulation, and has spoken at technical conferences in North America and Europe.

APPENDIX. EXAMPLE OF COLLISION PREVENTION MODE IN GPSS/PC™

```
 ;  GPSS/PC program file REPORT.GPS.  (V 2, # 30104)  09-20-1985 22:06:53
100 ;*********Use ACMEPOS.GPS for the POSITION.GPS file.*********
198 ;
200 ;
202 ;             ENTITY DEFINITIONS
204 ;
206 ;
208 ;
212 CARCOUNT STORAGE       1
214 CUSTOMERCOUNT STORAGE   1
216 CAR_AT_PRIMER   STORAGE  1
218 PRIMER_DONE STORAGE 1
220 CAR_AT_PAINTER STORAGE 1
222 PAINTER_DONE    STORAGE  1
224 CASH      VARIABLE      4995#N$DRIVEOFF
226 PRIMED    VARIABLE      N$PRIBLK-N$PAIBLK
228 PAINTED   VARIABLE      N$PAIBLK-N$SOLDBLK
230 ARRIVE    VARIABLE     (P$ROW'E'P$TO_ROW)'AND'(P$COLUMN'E'P$TO_COLUMN)
232 ;
234 ;
236 ;
238 ;           MODEL BODY
240 ;
242 ;
246 ;
248 ;
250          GENERATE       BODY_INTERARRIVALS
252          JOIN           POSITION
254          ASSIGN         SHAPE,CAR_BODY_RIGHT
256          ASSIGN         COLOR,LIGHT_CYAN
260 ;    Starting position for cars going to painter #1
262          ASSIGN         ROW,5
264          ASSIGN         COLUMN,1
266 ;    Go to Gate A    (7,30)
268          ASSIGN         TO_ROW,7
270          ASSIGN         TO_COLUMN,30
272 ;    Wait for arrival at Primer Painter
274          TEST E         V$ARRIVE,1
276 ;    Engage the primer robot
278          LEAVE          CAR_AT_PRIMER           ;TELL ROBOT TO START
280          ENTER          PRIMER_DONE             ;WAIT FOR ROBOT TO FINISH
282 PRIBLK    ASSIGN        COLOR,LIGHT_RED
290 ; Drive to finishing painter
300          ASSIGN         TO_ROW,7
310          ASSIGN         TO_COLUMN,60
320          TEST E         V$ARRIVE,1
330 ;    Engage the painter robot
332          LEAVE          CAR_AT_PAINTER           ;TELL ROBOT TO START
336          ENTER          PAINTER_DONE             ;WAIT FOR ROBOT TO FINISH
370 PAIBLK  ASSIGN          COLOR,YELLOW
380 ; Drive to intersection A
390          ASSIGN         TO_ROW,7
400          ASSIGN         TO_COLUMN,65
410          TEST E         V$ARRIVE,1
420 ; Change shape and drive to intersection B
430          ASSIGN         SHAPE,CAR_BODY_DOWN
440          ASSIGN         TO_ROW,10
450          ASSIGN         TO_COLUMN,65
460          TEST E         V$ARRIVE,1
470 ; Change shape and drive to customer
480          ASSIGN         SHAPE,CAR_BODY_LEFT
490          ASSIGN         TO_ROW,13
500          ASSIGN         TO_COLUMN,20
```

```
510 ;   Wait for arrival at Customer Pickup
520         TEST E        V$ARRIVE,1
530.;   Signal presence of car
540         LEAVE         CARCOUNT
550         ENTER         CUSTOMERCOUNT                 ;TAKE A CUSTOMER
560 ;   Customer drives off
580 ; Drive off to the left
590 SOLDBLK ASSIGN        SHAPE,MAN_IN_CAR_LEFT
600         ASSIGN        TO_ROW,14
610         ASSIGN        TO_COLUMN,0
620         TEST E        V$ARRIVE,1
630 STOPIT  TERMINATE     1
640 ;
650 ;         Create Customers
660 ;
670         GENERATE      CUST_INTERARRIVALS,50
680         JOIN          POSITION
690 ;   All cars look alike
700         ASSIGN        SHAPE,CUSTOMER
705         ASSIGN        COLOR,YELLOW
710 ;   Starting position for customers
720         ASSIGN        ROW,15
730         ASSIGN        COLUMN,80
740 ;  Go to Car pickup
750         ASSIGN        TO_ROW,14
760         ASSIGN        TO_COLUMN,20
770 ;   Wait for arrival at Customer Pickup
780         TEST E        V$ARRIVE,1
790 ;   Signal presence of customer
800         LEAVE         CUSTOMERCOUNT
810         ENTER         CARCOUNT                  ;TAKE A CAR
820 ;   Customer drives off
840 ;  Customer gets into car -- separate icon disappears
850 DRIVEOFF TERMINATE    1
891 ;
892 ;   MOVER Segment
893 ;
894         GENERATE      MOVE_TIME
895         MOVE
896         TERMINATE
900 ;
902 ;  Initializer
903 ;          STORAGES:  AT FIRST, NEITHER ARE AVAILABLE
904 ;  Initial  PLACEMENT FOR ROBOTS
906 ;
908 ;   PRIMER ROBOT CONTROL SEGMENT
910 ;
912         GENERATE      ,,,1
913 ;   Initialize storages to "full" for synchronizing cars and robots/people
914         ENTER         CARCOUNT,1
915         ENTER         CUSTOMERCOUNT,1
916         ENTER         PRIMER_DONE,1
917         ENTER         PAINTER_DONE,1
919         ENTER         CAR_AT_PRIMER
920         ENTER         CAR_AT_PAINTER
921         ASSIGN        ROW,6
922         ASSIGN        COLUMN,30
923         ASSIGN        TO_ROW,6
924         ASSIGN        TO_COLUMN,30
925         ASSIGN        COLOR,BRIGHT_WHITE
926         JOIN          POSITION
930 ROB1LOOP ASSIGN       SHAPE,ROBOT_1
932         ENTER         CAR_AT_PRIMER         ;WAIT FOR THE CAR
934         ADVANCE       PRIMER_TIME           ;PAINT THE CAR
940 ; THE CAR HAS BEEN PAINTED, RELEASE THE CAR
942         LEAVE         PRIMER_DONE
946         ASSIGN        SHAPE,ROBOT_2          ;CHANGE FORM
948         ADVANCE       PRIMER_RESET_TIME     ;PREPARE FOR THE NEXT CAR
950         TRANSFER      ,ROB1LOOP
```

```
952 ;
954 ;   PAINTER ROBOT CONTROL SEGMENT
956 ;
958         GENERATE       ,,,1
959         ASSIGN         COLOR,BRIGHT_WHITE
960         ASSIGN         ROW,6
962         ASSIGN         COLUMN,60
964         ASSIGN         TO_ROW,6
966         ASSIGN         TO_COLUMN,60
967         JOIN           POSITION
968 ROB2LOOP ASSIGN        SHAPE,ROBOT_1          ;THE ROBOT IS READY
970         ENTER          CAR_AT_PAINTER         ;WAIT FOR THE CAR
972         ADVANCE        PAINTER_TIME           ;PAINT THE CAR
974 ; THE CAR HAS BEEN PAINTED, RELEASE THE CAR
976         LEAVE         PAINTER_DONE
980         ASSIGN        SHAPE,ROBOT_2           ;CHANGE FORM
982         ADVANCE       PAINTER_RESET_TIME      ;PREPARE FOR THE NEXT CAR
984         TRANSFER      ,ROB2LOOP
996         WINDOW         POSITIONS
1000        START          1000
```