Proceedings of the 1989 Winter Simulation Conference
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

THE GPSS/PC[tm] STUDENT VERSION

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

Abstract

From its inception, GPSS/PC, itself, has always had features designed specifically to facilitate the mastery of the simulation environment by new users. Its basic design goals of visibility, control, and protection from unseen danger fit very nicely into the classroom.

The GPSS/PC Student Version has been available to educational institutions at nominal cost, ever since Version 2 of GPSS/PC was released. It was designed to provide the visual interactive environment of GPSS/PC, including the error prevention and detection features, to students learning the GPSS language and/or simulation methodologies in general. Although the Student Version does not include the FORTRAN interface or the built-in spatial animation, it does include the animated graphics windows and is completely functional with respect to the simulation primitives of its big brother, the commercial version of GPSS/PC. In addition, its use of character mode graphics results in minimal hardware requirements in the laboratory yet retains the graphical user interface.

Several recent improvements to the Student Version have improved its speed and power. Although the Student Version was never designed as a vehicle for advanced projects, users have occasionally requested that its capabilities be expanded. For this reason, in mid 1989, both speed and size limitations were somewhat relaxed.

This paper presents the features and limitations of the current GPSS/PC Student Version, and discusses the advantages of the interactive and protective GPSS/PC simulation environment in the classroom.

Keywords: GPSS, discrete event simulation.

1. Introduction

The main discussion in this paper will center on the advantages of the Student Version of GPSS/PC in an educational role. Several major design improvements distinguish GPSS/PC in this regard. First, the older multiphasic design of the overall simulation process has been replaced by a single, integrated simulation environment combining the functions of editing, compiling, loading, simulation, and debugging. This has a variety of advantages, as discussed below. Second, the user interface makes possible a high level of interaction of the student with his/her simulation. It is possible to modify named values, and even to make structural changes to the model without even having to restart. In addition to considerably reducing model development and debugging time, this provides the power to investigate hunches and to examine the effects of transients in the the simulation. In addition, a mode of operation called "manual simulation" permits the keyboard entry of GPSS block statements as if they were commands. In addition to aiding the learning/teaching of GPSS, manual simulation provides a high level of control over running simulations. Third, a hierarchy of error detection and prevention mechanisms are implemented. The most apparent of these is "Keystroke Error Prevention" which makes it impossible for the student to incorporate syntax errors into the simulation environment. Fourth, a variety of additional usability features are part of the user interface. Fifth, an easily accessible visual interface allows the user to view the dynamics of running simulations. Finally, built-in statistical commands automatically calculate

confidence intervals and perform a first level analysis of variance on the simulation results.

2. The Session

In GPSS/PC, all phases, including compiling, linking, running, and debugging, are combined into a single phase called a session. Once a program is read from a DOS file, in effect, the compilation and link steps are done. From this point it is possible to modify the GPSS model, to save the model, to begin a simulation, or to enter GPSS block statements in manual simulation mode. A set of debugging commands, including STEP, STOP, SHOW, and PLOT are available, as well.

The development of GPSS models is expedited by the combined design of the single phase environment, and the error handling features. As described below, syntax errors cannot enter the environment. Therefore, those problems which must be corrected as a model is under development are logical or semantic in nature. For those problems which remain to be corrected, the student need not leave the session in order to modify the model. As each problem is corrected, the student can proceed immediately to the next. The improvement in model development time resulting from this integrated simulation environment can be dramatic. No longer is it necessary to spend time passing through several debug-edit-compile-link-run phases in order to correct run time problems. Perhaps even more important, the interval between the commission and detection of an error is shortened. This saves time in itself, but it also allows the student to retain a more complete mental context for the problem correction task.

The GPSS/PC session is made much more powerful by the inclusion of features which allow immediate modification to the data structures of the simulation. Without them it would be necessary to leave the single-phase session in order to develop or debug a simulation model. New block statements, originating from the keyboard or from other program files are incorporated incrementally into the existing data structures. Each insertion of a block statement is directed by the line number, which may be a decimal fraction. Any statement is inserted into the ASCII "savable program" in line number sequence, and any GPSS block entity so described is inserted in a corresponding position in the "current model".

Corrections can be made to existing statements by replacing complete lines, or by using the integrated line editor. An EDIT command is available to make minor modifications to existing lines without retyping the whole line. A small set of special editing keys is available for use within the EDIT mode.

The single phase design greatly improves the immediacy of the simulation system. When, and if, the student reaches the point where the simulation environment feels like an extension of self, it becomes relatively easy to refine his/her intuition by interacting with the simulation environment. It then becomes possible to develop a "feel" for those changes which will be effective in modifying the behavior of the target system under study. The reason that this is of importance is that the quality of a simulation study is often determined by the range of alternate designs under consideration. That design set is best specified when the simulation analyst possesses an accurate intuition with respect to the behavior of the system under study.

3. The Interactive User Interface

GPSS/PC is distinguished by its interactive user interface. It allows the student to make online modifications to a simulation that run deep into the structure of the model. In addition, GPSS block statements assume a new interactive function, called "manual simulation", where they are treated as keyboard commands. A temporary GPSS block entity is created by this action, and the active transaction at that instant attempts to enter it. The results of the interaction can be observed immediately. This gives the student a powerful level of control over running simulations.

Any named value may be changed at any time in the simulation and the results may be observed immediately, without restarting the simulation. Such observations can be visualized dynamically in one of the graphics windows, or in one of up to four ad hoc Microwindows. Alternately, they can be made by plotting the value of one or two SNAs. The plots can be observed as the simulation runs or the simulation can be interrupted at any point and values of

various SNAs displayed. By changing named values, it is easy to explore a variety of alternatives in a single simulation.

GPSS blocks may be inserted, replaced, or deleted in the middle of a simulation. Transactions destined for a deleted block are rescheduled for the "next sequential block" following the deleted one. Transactions destined for a replaced block are rescheduled for the replacement. The placement of the GPSS block is made according to line number sequence. If a block statement is entered with a line number identical to that of an existing block statement, the original statement is replaced with the new one, and the original GPSS block entity is replaced by the newly defined one. Similarly, if a block statement is entered with a line number that does not yet exist, the statement is inserted in line number sequence and the newly defined GPSS block statement is inserted in a corresponding place among the existing ones.

Finally, the manual simulation feature places all the simulation primitives on the user interface. This means that any GPSS block statement can be entered through the keyboard in the middle of the simulation. If a block statement has no line number, it is a manual simulation statement. This causes a temporary GPSS block entity to be created which the active transaction attempts to enter. In this manner, any action of any existing block in the model can be initiated by an EXECUTE statement, or a new block statement can be typed. ASSIGN blocks can alter transaction parameters; TRANSFER blocks can move transactions; GENERATE and SPLIT blocks can create new transactions. In fact, the whole simulation could be done manually. In essence, it allows actions occurring deep within the simulation to be controlled from the keyboard. In addition, manual simulation allows one to initiate a block action and then to explore the results immediately.

Manual simulation is complimented by the windows and the debugging commands: STOP, STEP, SHOW, and PLOT. These commands allow one to find an specific condition in a simulation so that the environment may be explored and manipulated. The PLOT and SHOW commands are available for observing state variables during a simulation. The PLOT command allows one to plot any SNA, and to observe its value as the simulation progresses. If exceptional conditions are detected, it is easy to interrupt the simulation, explore with the debugging commands, and even introduce transients or make corrections. All of this is possible without leaving the session.

There are two major objectives for manual simulation: first, in program development for the correction of errors and the exploration of the simulation; second, for learning and teaching.

4. Error Detection and Prevention

It is highly desirable to dispose of user errors as soon as possible. Each additional delay increases the chance that the student will be distracted. This would be inefficient because much of the student's effort of modeling is in creating a mental context where the implications of statements in the modelling language are obvious. Since this context must be recreated in order for problems to be fixed, it is desirable for the simulation development environment to detect errors as close as possible to the time when they are committed.

GPSS/PC possesses a feature called "Keystroke Error Prevention" to provide for 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. It is impossible for syntax errors to enter the simulation environment. Further, the mental context of the student is still established for choosing the correct alternative. The net result of the error prevention is significantly decreased model development times.

The mutability of the GPSS/PC simulation environment is intended to serve the same function for semantic and functional errors. Although these errors are not prevented, they may be corrected and retried without leaving the session. As described above, the single phase design coupled with the the ability to make drastic changes to the model, keeps the correction of errors as close as possible to the time of commission. When one error is corrected, the student may then proceed to the next without leaving the session.

Minor modifications have been made to the GPSS language itself in order to make the syntax more uniform and therefore more predictable. Uniformity is very desirable because it reduces the "knowledge load" which

must be carried by the student. The result is that the system is easier to learn and less error prone.

Another approach to reducing the special knowledge necessary to the student is to remove exceptional conditions and unnecessary data types from the simulation environment. GPSS/PC uses unlimited precision integers for internal values. As numbers grow in size, additional memory is allocated for them automatically. Unlimited numerical precision relieves the student of concern about overflows and underflows in the clock, and of overflows in the statistics accumulators. These exceptional events do not occur.

To reduce the level of system knowledge needed to access sophisticated mathematical algorithms, GPSS/PC has integrated a mathematical library into the GPSS language. This alleviates the need to link modules written in other languages, such as FORTRAN, in order to perform complex calculations. In GPSS/PC, expressions can include SNAs (System Numerical Attributes), exponentials, logarithms, trigonometric functions, logical operators, and others. This allows probability functions in closed form to be included as GPSS variable entities. Similarly, GPSS/PC can be used to simulate continuous state systems using any of the commonly used integration techniques.

5. Ease of Use

Several GPSS/PC features are designed to save time in the overall simulation project. Fewer housekeeping tasks required of the student leave more energy to be devoted to mastery of the simulation environment. Not only that, but since tedium tends to increase user error rates, features which remove it are of considerable value to the student. Several features of GPSS/PC are noteworthy in this regard.

The integrated online help feature allows the student to press the ? (question mark) key at any time a statement is being entered. This causes a help message to appear with operand specific syntax information.

The command recognition feature eliminates some of the typing load. If the space bar is pressed when a command is partially, but uniquely determined, GPSS/PC finishes the command word automatically.

All statements and commands can be saved by a single keystroke and thereafter be re-entered with a single keystroke. This is accomplished by use of the assignable function keys found on the keyboard of the PC. By pressing the control key and the selected function key, the previous statement or command will be stored and can be recalled by pressing the same function key, by itself.

Automatic spacing aligns statement fields with column numbers. This occurs when a valid delimiter key is pressed. It is not necessary for the student to know, or even be concerned with, the alignment of statement fields. GPSS/PC will not, however, permit the student to skip a required field.

Each field is prompted by a distinct cursor which serves as a reminder of field identity. When the student ends one field by pressing a delimiter key, GPSS/PC changes the form of the cursor to indicate which field is to be entered next. For example, an inverse video D appears as the cursor when the student may start operand D.

6. GPSS/PC Windows

All versions of GPSS/PC now include at least 5 interactive graphics windows, each of which allows the student to view, and interact with, a specific GPSS entity type. The student 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 student 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.

The Positions Window of the full version of GPSS/PC is not part of the Student Version. However, the other five online graphics windows (Blocks, Storages, Facilities, Matrices, and Tables) provide alternate viewpoints of simulation dynamics based on specific GPSS entity types. They provide the student with a dynamic visualization of the changing state of his/her simulation.

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 student wants to "skip ahead" or complete a simulation as quickly as possible, he/she should open the Data Window.

6.1. Microwindows

A Microwindow is a small window which shows the current value of any GPSS state variable , or user defined variable, 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 student can view his/her choice of state variables, regardless of which major graphics window is open.

Microwindows can be invaluable during the testing and development of simulation models. By opening these tiny windows, the student can follow the changes in the simulation dynamically or one step at a time. When unexpected conditions are detected, they may be trapped and explored with the full set of interactive commands.

6.2. 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. As students develop the program working with GPSS text, the corresponding block diagram can be viewed in the Blocks Window. Alternately, program modifications can be entered by manipulating the block diagram through the Blocks Window. The combination of breakpointing, stepping, and continuation, allows much of the interaction during the testing/verification phase to be done through the Blocks Window 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 can be very 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 global flow of transactions through the model is often obvious from a glance at the Blocks Window.

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

The Blocks Window has, in addition, several powerful menu functions which allow online manipulation of animations with the pointing device. This is discussed next.

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.

6.3. The Other Interactive Graphics Windows

The other interactive graphics windows 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. Students can see for themselves the effects of run length on the regularity of the input and output distributions.

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.

6.4. Interactions through the Major Graphics Windows

In line with GPSS/PC's design objectives, a high level of interactivity is available to the student. The keyboard cursor keys or a pointing device, such as a mouse or light pen, can be used to interact with entities visible in the windows. 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 student to single step or to continue an interrupted simulation using the pointing device.

7. ANOVA

The ANOVA command is a significant feature in GPSS/PC. It operates on an accumulation of data in a results database. The student can use it to calculate confidence intervals and easily perform an analysis of variance on experimental results.

In the vein of protecting the user from unforeseen danger, GPSS/PC does not allow a statistically homogeneous simulation to be perturbed by user interactions. This protection is indicated since the interactive nature of GPSS/PC creates a temptation to manipulate the environment right through the final simulations. To prevent such interactions from invalidating the homogeneity assumptions required by the statistical treatment of results, GPSS/PC prevents data from a perturbed environment from being saved in a results data base. This means that the ANOVA command cannot be used on such data without an explicit override by the user. In this manner, the student is guided into the required procedure for a valid statistical analysis.

8. Restrictions

The Student Version of GPSS/PC has several restrictions with respect to the full commercial version. Since it was designed to introduce students to the operation mechanisms of GPSS simulation entities, it contains all the traditional GPSS blocks and control statements of the full version. However, it is restricted in the size and speed of models that can be run. At this writing, models of up to 100 blocks and 20K of RAM memory can be run nearly as fast as the full commercial version when it was introduced on IBM PCs in 1984.

The following features are not available:

1. The Positions Window and CAD based animation Post-processor.

2. Simulation swapout via the GPSS/PC DOS Command.

3. The HELP block and FORTRAN interface.

4. The Session Journal.

Actually, for introductory work, the GPSS/PC Student Version is not restricted very significantly. The Positions Window and the FORTRAN interface are considered to be more advanced features that are not appropriate in a first course in simulation. Statistically, the longer run times of the Student Version may require the student to settle for wider confidence intervals. But even

this tends to draw attention to important statistical considerations which should be addressed in the classroom.

9. Conclusions

The Student Version of GPSS/PC was designed to provide a very visual and interactive environment where it is easy to manipulate, and get an accurate intuitive "feel" for, those factors that determine model behavior. Except for HELP and MOVE, the complete set of GPSS/PC blocks and control statements are implemented. The graphics windows require no programming, and are available at a keystroke for viewing the dynamics of running simulations. A model can be explored and manipulated very easily in order to verify its behavior, or to explore exceptional conditions. The ability to modify simulations in situ, means that small error trapping GPSS segments can be inserted or removed, at will.

In addition, the software goes to great lengths to relieve the student of irrelevancies and tedium which might impede the learning process. For example, the unified environment eliminates the need to learn and deal with external editors, compilers, and linkers. Model development tends to proceed from one problem to the next without intervening edit/compilation/link/run sequences. This means that much time is saved, and that students can keep their attention on the model under development.

The Keystroke Error Prevention feature prevents syntax errors from entering the environment. For a student approaching a new language for the first time, this can be a godsend. Not only does it speed up the learning of the essentials, but it removes the distracting and time consuming recompilations required by old fashioned environments.

The statistical treatment of results often does not receive the attention it really deserves. The Student Version of GPSS/PC includes built-in in dynamic histograms and the ANOVA command, which are within easy reach for the purposes of engendering an appreciation of essential statistical methods.

The Student Version is extremely affordable. It is available under a site license for education institutions, so that a complete computer laboratory may be populated economically. The modest hardware requirements of the software mean that nearly all MS-DOS compatible personal computers can run it without an upgrade.

Finally, recent improvements in speed and model size have made the Student Version of GPSS/PC even more attractive as a vehicle for the instruction of simulation principles at an introductory level.

AUTHOR'S BIOGRAPHY

SPRINGER COX received his degrees in physics and computer science from Cornell University and Syracuse University, respectively, and has completed an advanced study program at MIT. He worked in computer performance evaluation and modeling for IBM and Xerox, and, in 1977, went to the R & D Group at DEC to simulate virtual memory operating systems. In 1982, he founded Minuteman Software for the purpose of creating a microprocessor based interactive simulation environment. He has published over a dozen papers, and has spoken at technical conferences in North America and Europe.