Increased adaptability, versatility, and ease of use are significant advantages offered to the user of the General Purpose Simulation System in this latest version as described in this paper.

Modeling capability is especially enhanced by such improvements as new entities and block types and extended features.

Also discussed are improved input specifications and an output editor.

GPSS/360 — an improved general purpose simulator

by R. L. Gould

The General Purpose Simulation System (GPSS) is an IBM computer program for conducting evaluations of systems, methods, processes, and designs. The latest version of the program, GPSS/360, is compatible with its predecessor GPSS III,[1] and offers significant improvements to the user of this modeling tool. The inclusion of new entities, block types, and expanded features in GPSS/360 increases the adaptability, versatility, and ease of use for a wide range of simulation applications. In addition, output statistics obtained from a simulation run may now be presented in a format specified by the user, and selected statistics may be obtained in graphical form. Knowledge of GPSS,[2,3] and more specifically GPSS III, is assumed throughout the paper. However, we now briefly review some of the basic concepts of GPSS as described by Herscovitch and Schneider.[1]

GPSS features a simple flowchart language for describing the problem or system to be simulated. When this description is fed into a computer, the program automatically carries out the simulation.

The system elements are logically represented by a set of abstract elements called entities, and the logical rules governing a system flow are reduced to a set of standard operations. There are four classes of entities: dynamic, equipment, statistical, and operational.

The dynamic entities are called transactions, representing units of traffic. Equipment entities include such items as facilities, storages, and logic switches that are acted upon by transactions. The statistical entities consist of two types: queues and tables. Blocks constitute the operational entities.

Control and definition cards prepared from a flowchart of the system provide input for the simulation. The cards make up the model in GPSS language. When the system model has been loaded, the GPSS program generates and moves transactions from block to block according to the timing information and logical rules of the individual blocks.

In the GPSS III version, structural changes were made to the basic program. Externally, the most significant change was the elimination of delay-time and selection factor operations from all block types. The modeling capability, versatility, and ease of use of the program were expanded by increasing speed, varying the number of parameters, reallocating storage space, allowing construction of user chains, adding the Attribute Valued function, and adding new blocks and modifying others.

GPSS/360 operates under control of both the SYSTEM/360 Operating System (OS/360) and the SYSTEM/360 Disk Operating System (DOS/360).[4] GPSS/360 allocates entities according to main-storage size, and the REALLOCATE feature enables the analyst to completely tailor the allocation of main storage to fit the needs of his model.

This paper is devoted to descriptions and possible modeling applications of the extended features of GPSS/360.

The significant features incorporated in the GPSS/360 system can be grouped into the following three general categories:

- New entities, block types, and extended features which enhance modeling capability.
- Symbolic entities and features which contribute to ease of use.
- Output editor capabilities which make output statistics more meaningful.

The features included in the first category are the most important from the point of view of simulator power and capability.

Modeling capability enhancement

parameters

In GPSS III, the values that could be contained in the parameters associated with transactions were always considered positive and could range from 0 to $2^{15} - 1$. Modeling became awkward when it was necessary to associate negative quantities or values of magnitude greater than $2^{15}-1$ with transactions. Each of the parameters of a transaction may now be assigned positive or negative integer values. Also, each active transaction may now have either halfword or fullword parameters associated with it. The magnitude of values which may be retained in the parameters are $\pm 2^{15}-1$ for halfword parameters, and $\pm 2^{31}-1$ for fullword parameters. The extension of parameter magnitudes greatly increases the flexibility of GPSS.

savevalues

An improvement along the same line is the provision in GPSS/360 for both fullword and halfword savevalues with maximum capacities of $\pm 2^{31}-1$ and $\pm 2^{15}-1$, respectively. In GPSS, savevalues are storage locations in which statistical information may be saved dynamically during the course of a simulation run. Previous versions of GPSS provided only fullword savevalues (savexes). In many modeling situations, the values to be stored in savevalues are within the range $\pm 2^{15}-1$, and significant main storage savings can be realized by the user of halfword savevalues.

queuing

Recognition of the need to measure intermediate delay times as well as overall delays resulted in an expansion to the statistical gathering capability related to GPSS queues. GPSS/360 permits a transaction to be a member of multiple queues simultaneously, and also allows processing time (ADVANCE blocks) to be included in queue statistics. Formerly, accurate queue statistics were not obtained under the conditions just described. These capabilities increase the queuing situations which can easily be modeled using GPSS.

random numbers

Eight random-number generators in GPSS/360 provide the user with independent sources of random numbers. Parametric studies often require that specific delays based on a random number in certain paths of a model be identical from run to run. The user now has the ability to reference unique random-number generators in such paths and can be assured that changes in other parts of the model will not affect the consistent progress of transactions in the desired paths. The eight random-number generators normally all produce the same sequence of random numbers; however, the user has the ability to specify different initial multipliers for any or all generators to vary the sequences produced. Also, the random-number generators may be reset or the multipliers changed at CLEAR time.

standard numerical attributes

The provision for additional Standard Numerical Attributes increases the versatility of GPSS/360 and enhances the logical power of the simulator by permitting the user access to more dynamic information concerning his model. Among the new Standard Numerical Attributes which may be specified anywhere in a model are: the priority of a transaction, the utilization and number of entries for facilities and storages, and other dynamic statistics relating to tables, facilities, storages, queues, and user chains. Thus, for example, the analyst may now decide on a path for a transaction based on its priority and may choose a queue based on the average time transactions have spent in the queue up to this point in the simulation run.

variables

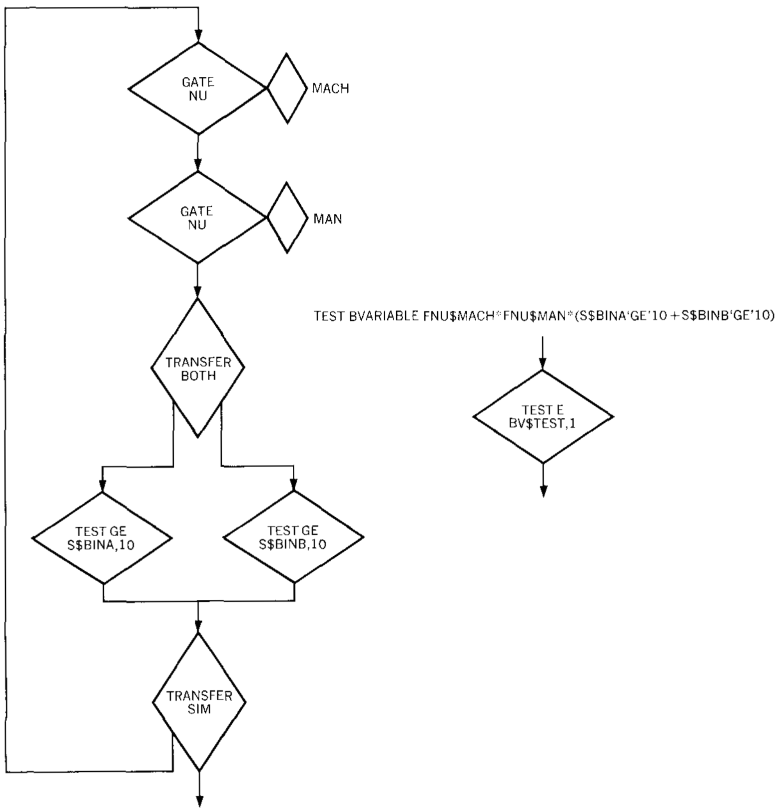
The arithmetic capability of variable statements has been expanded by provision for the use of parenthetical expressions. Up to five sets of parentheses may be specified in any variable statement. Elements within the parentheses are evaluated before other arithmetic operations outside the parentheses take place (similar to FORTRAN). Also, greater precision in the computation of variable expressions may now be obtained by the use of *floating* *point arithmetic variables.* Whereas the results of intermediate arithmetic operations are truncated during the computation of variables, no truncation takes place in the evaluation of floating variables until the final result has been determined.

A new entity, the *Boolean variable,* increases the logical power and capabilities of GPSS/360. This entity makes it possible to make decisions at a single GPSS block based on the status and value of many GPSS entities. Boolean variables are Boolean combinations of various Standard Numerical Attributes. The result of the evaluation of a Boolean variable is one if the conditions specified are met or zero if the conditions are not met.

There are three types of operators associated with Boolean variables. Logical operators enable the analyst to determine the status of the equipment entities—facilities, storages, and logic switches. Conditional operators make comparisons between Standard Numerical Attributes. Boolean operators are logical connectives between elements of the Boolean statement—"and" represented by *, "or" represented by +.

Figure 1 represents a processing step in a job-shop model. For the job to be processed at this point, the following conditions must be met simultaneously: a machine must be free, the operator

Figure 1 Processing step in job-shop model



must be free, and 10 parts must be available from either Bin A or Bin B. Section A of the figure represents a block sequence which determines when the job can be processed; Section B shows a block which serves the same purpose using a Boolean variable.

preemption based on priority

Experience has shown that in most modeling situations the ability to obtain an item of equipment is based primarily on the priorities of the transactions in contention for the equipment. GPSS/360 provides a possible 128 levels of priority for transactions and facilitates the modeling of use of equipment by providing for multiple-level preempts. The preempting of facilities may now be determined by the priorities of the transactions attempting to use the equipment.

The operation of the PREEMPT block has been expanded to take priorities into account and to provide other options. In the priority mode of operation, the PREEMPT block allows the entering transaction to preempt a seizing or preempting transaction if its priority is higher than that of the current user. A maximum of 127 levels of preempts are possible on any facility at any one time. The analyst may also specify a block to which the preempted transaction is sent, a number of a parameter in which the amount of processing time remaining at the time of preemption is placed, and whether or not the preempted transaction remains in contention for the facility. Figure 2 illustrates a case where a facility is preempted according to the priorities of the entering transactions. The preempted transactions are removed from contention for the facility, their remaining processing times are tabulated in a distribution table, and they use an alternate machine.

matrix savevalues

The *matrix savevalue* entity provides the ability to define and reference a two-dimensional array of main-storage locations. A primary use for a matrix savevalue is to associate many attributes with a GPSS entity, and to be able to easily reference any attribute. For example, a matrix might be defined that represents a number of machines and their associated attributes: processing time, number of failures, average down time, last job type processed, etc. Figure 3 lists such machines and the matrix savevalue representing them.

A matrix of this type normally contains elements that remain constant throughout the simulation and elements that change dynamically. The constant elements, such as processing time and average down time, could be initialized at the beginning of the run and easily referenced throughout the run. The dynamic elements, such as number of failures and last job type processed, would be saved during the run by use of the MSAVEVALUE block. For example, the block sequence in Figure 4 could be used to represent a machine failure.

Matrix savevalues have many applications where information related to an entity is to be stored. They also can serve as a convenient branching table in a job-scheduling environment where the sequence of operations is dependent upon job type. The matrix in Figure 5 could be referenced at the end of each operation to

Figure 3 Matrix savevalue and machines it represents

MACH MATRIX X,4,5.

ROW	MACHINE NO. 1	2	3	4	5	
1	15	25	50	30	30	PROCESSING TIME
2						NO. OF FAILURES
3	500	420	500	500	650	AVERAGE DOWN TIME
4						LAST JOB TYPE PROCESSED

Figure 2 Preemption of facility according to transaction priorities

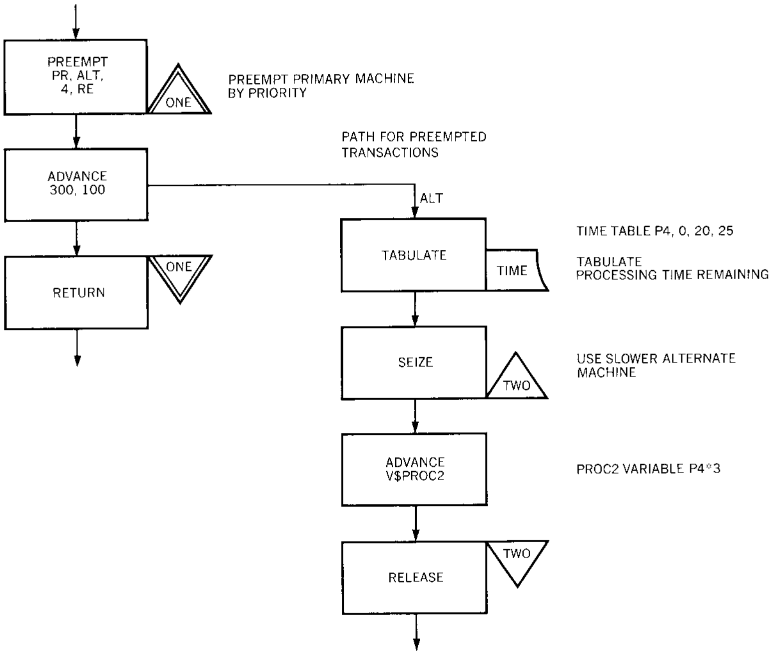
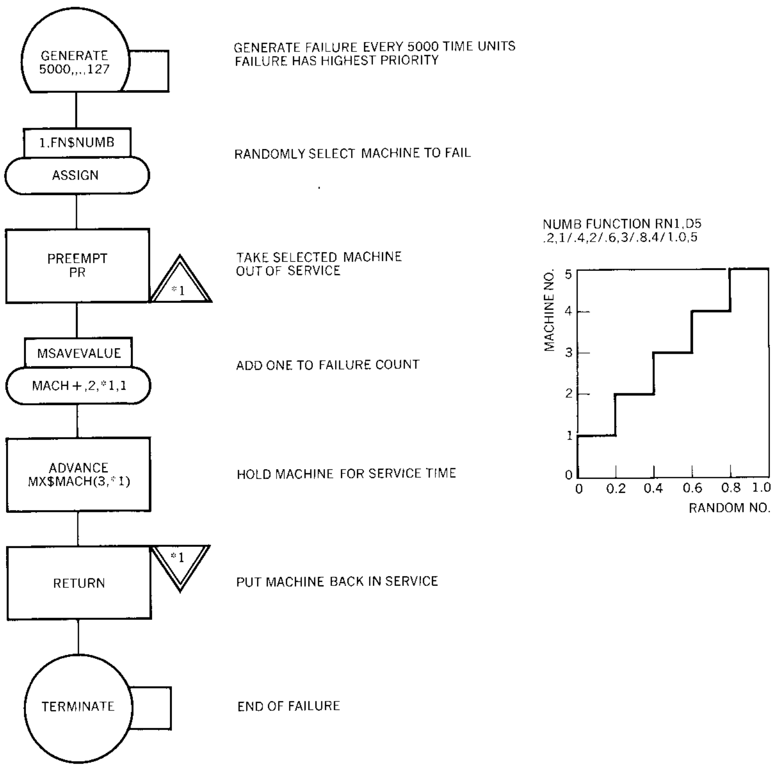


Figure 4 Block sequence representing machine failure



determine what process should be performed next. If parameter 1 of each transaction contained the job type and parameter 2 contained the number of the next sequential job step, the block shown in Figure 5 would route each transaction to its next processing routine (entries in the matrix are block numbers which begin independent processing routines).

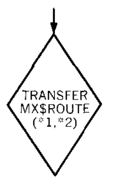
new block types

Two new block types, the COUNT and SELECT blocks, provide a means of determining the status of multiple GPSS entities and either finding the number of entities that meet a specified condition or determining the index number of the first entity that meets a condition. These blocks eliminate the need for complicated block sequences which were formerly required to perform these functions. Both the COUNT and SELECT blocks specify a range of attributes to be examined and a parameter number of the currently active transaction in which to place the result.

Figure 5 Matrix savevalue as a branching table

ROUTE MATRIX X,5,5

JOB TYPE \ JOB STEP	1	2	3	4	5
1	PROC1	PROC2	PROC3	PROC4	END
2	PROC2	PROC3	PROC1	PROC4	END
3	PROC4	PROC1	END		
4	PROC1	PROC2	PROC3	END	
5	PROC3	END			



The COUNT block determines the number of entities that meet a specified condition within a range. For example, the block COUNT SNF 1, 5, 9 examines storages 5 through 9, determines which ones are not full, and places the total number not full in parameter 1 of the entering transaction. Previously, an operation of this type required a series of TEST blocks to determine the status of each storage and an ASSIGN or SAVEVALUE block to accumulate the results.

The SELECT block determines the number of an entity that meets a specified condition within a range. For example, the block SELECT MIN 5, 10, 15, , FR determines which facility within the range of 10 to 15 currently has the minimum utilization and places the facility number in parameter 5 of the entering transaction. This block could be the basis for a decision of which facility to SEIZE or PREEMPT. The operation performed by the above SELECT block was previously impossible because the facility utilization attribute could not be addressed by the user.

group entity

The *group* entity provides a means of categorizing transactions based on common attributes. This entity permits communication among transactions which are members of a common group. Previously, synchronization of transaction flow was possible by use of the Assembly Set; however, this was severely limited because transactions had to be on the Current Events chain and no individual transaction attributes could be referenced. Group membership permits the referencing and altering of transaction attributes at any time during the simulation regardless of where the referenced transactions are in the system, i.e., Current, Future, Interrupt, or user chain. The group entity permits modeling of situations that were previously extremely awkward and sometimes impossible.

In a job-shop model, transactions representing the jobs in process usually carry a numeric representation of job type in a parameter. This permits decision making based on job type. The priority of the transactions reflects the relative need for the job, and parameters carry other attributes such as weight, color, machines needed for processing, etc. The importance of the group entity can be seen in an example of this type. If the transactions are categorized in groups by job type, the blocks associated with the group entity permit the dynamic alteration of the priorities and parameter values of transactions of selected job types. This alteration can be immediately effected on all or a number of jobs of a certain type regardless of where the jobs are in the system. When a particular type of job becomes in short supply or a change in processing sequence is required, the ability to alter priority or parameter values of all types of jobs becomes extremely important. This ability is provided by the group entity and associated blocks.

The group entity may also be used to save a set of numeric quantities without reserving a series of savevalues for this purpose. For example, the numeric entries in a group could represent the storages or facilities which at some time exceed a given utilization, or the user chain numbers which at some time exceed a specified limit. Overall, the group entity greatly extends the power and flexibility of GPSS/360.

Input specification enhancement

The features of GPSS/360 in the second general category pertain to changes to input specifications which result in ease of use of the simulation language. These improvements are now briefly described.

symbolic entity specification

Previous versions of GPSS have relieved the analyst of specifying absolute block numbers by allowing blocks to be assigned symbolic names; however, all other entities had to be designated numerically. GPSS/360 provides the user with the ability to specify all entities symbolically. This feature facilitates coding and debugging of a model as well as making the listing of the model and output statistics more meaningful and easier to interpret. The GPSS/360 assembly program equates the symbolic entities specified by the analyst to numbers and automatically provides a dictionary of symbolics and the entity numbers assigned to each.

This symbolic entity reference capability has been illustrated in the previous figures in this paper. For example, Figure 1 included facilities named MACH and MAN, and storages named BINA and BINB.

macroinstructions

GPSS/360 enables the analyst to define and use macroinstructions. The macroinstruction feature permits parallel or nearly identical parts of models to be coded once and then used at any point in the model by the insertion of a card calling the desired macro-definition.[5] This feature can relieve the analyst of much repetitive coding and keypunching.

Each macroinstruction may have up to 10 arguments associated with it. These arguments enable the user to vary the function of the macro-definition each time it is called within the model. All fields of the blocks within the macro-definition, including the block type itself, may be specified as substitutable arguments and varied each time the macroinstruction is called.

update feature

The update feature of GPSS/360 enables the analyst to place symbolic model(s) on tape or disk and subsequently make modifications to the model by updating the master file. This feature eliminates the need to transport, store, maintain, and submit large symbolic decks each time a simulation run is desired. The update feature provides the ability to create a master file and to update the master (change, replace, and delete cards) to create a new master which may be assembled and executed.

error checking

The assembly phase of GPSS/360 has been expanded to perform more extensive error checking than in previous GPSS versions. Many errors which were not formerly detected until actual execution of the simulation model are now discovered during the initial assembly phase. Also, the diagnostic error messages produced during assembly are superior to those that were given in previous versions. By detecting errors during the assembly phase, this expanded error-checking capability facilitates model debugging and saves considerable machine time.

Output editor

Another significant feature of GPSS/360 is the output editor which allows the analyst to completely specify the extent and format of output statistics produced by the simulator. This ability to deviate from the standard output format enables the analyst to present output statistics in a manner that is meaningful and easy to understand.

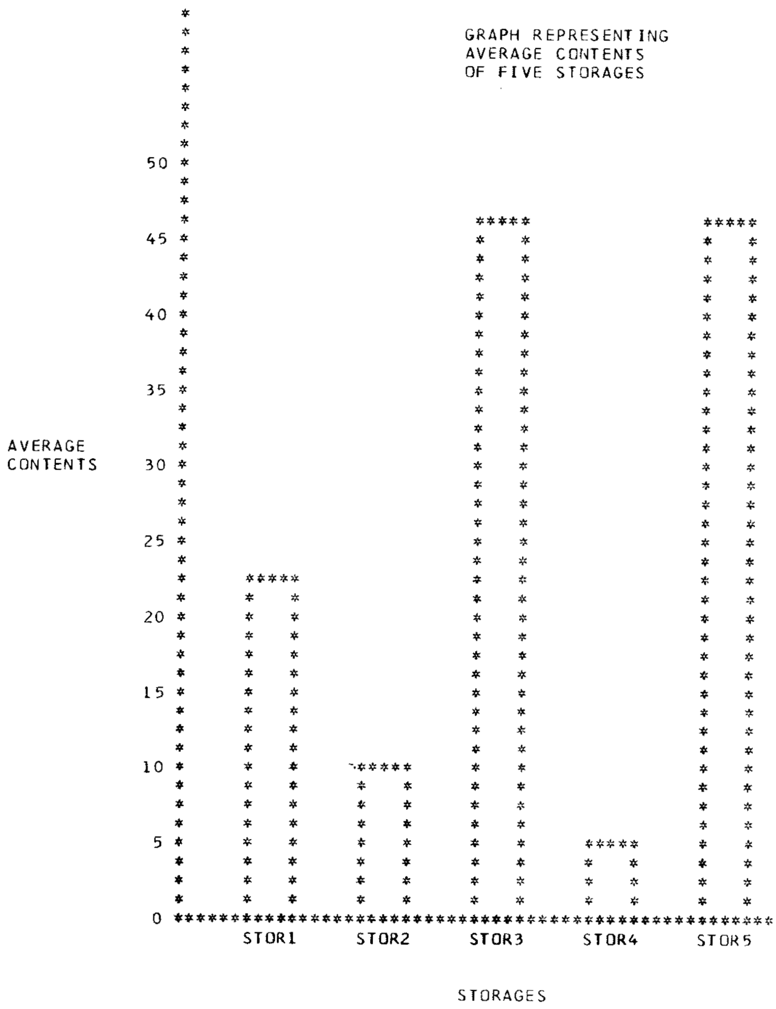
output format

Various options are available when the output editor is employed. Titles may be specified for sections of output statistics by use of the TITLE card, statistics for a user-designated range of entities may be printed by use of the INCLUDE card, and the FORMAT card enables the user to designate selected statistics for various entity types. This makes it possible, for example, to have the average contents and average time per transaction for a range of queues printed in columns adjoining selected statistics for an associated range of facilities. This printing gives a comprehensive, compact picture of in-use and wait statistics for a range of facilities.

The TEXT card enables the user to specify the output in sentence format with statistics produced by the simulation run inserted at appropriate points. For example, a TEXT card could produce the statement "THE AVERAGE UTILIZATION OF THE LATHE WAS XXX.X AND THE NUMBER OF FAILURES DURING THE RUN WAS YYY" where the quantities XXX.X and YYY are filled in by the output editor from statistics produced by the simulation run. Thus the output could be in the form of a report that includes meaningful sentences or phrases to highlight the most important statistics from the simulation.

The graph feature of the output editor enables the user to

Figure 6 Average contents of five storages



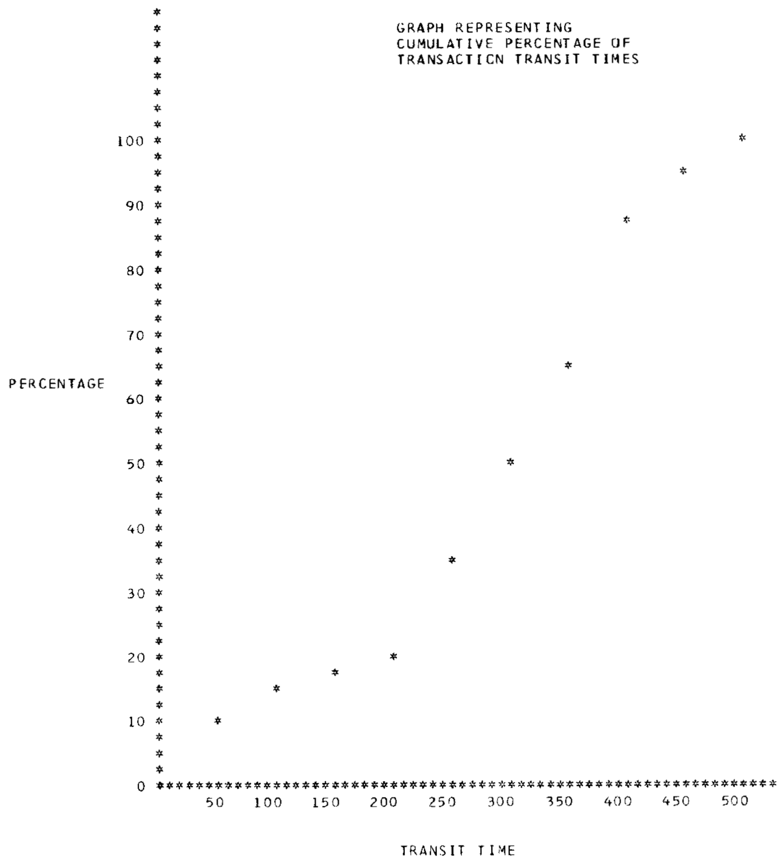
graph feature

obtain selected statistics in graphical form. The Standard Numerical Attribute to be plotted, the increment size and labeling of x and y axes, and appropriate commenting of the graph may be completely controlled by the user. Figure 6 is an example of a bar graph that represents the average contents of a set of five storages. Figure 7 shows a line graph which pictorially represents the cumulative percentage of a distribution of transit times as tabulated during the simulation run.

Summary

The GPSS/360 program offers adaptability, flexibility, and ease of use in a general purpose modeling tool which is applicable to the analysis of a wide range of dynamic systems. The most

Figure 7 Cumulative percentage of transaction transit times



significant extensions and improvements in GPSS/360 in comparison to previous versions of the General Purpose Simulation System can be summarized as follows:

- Signed halfword or fullword parameters and savevalues
- Eight independent random-number generators
- Additional Standard Numerical Attributes
- Boolean variables
- Preemption based on priority
- Matrix savevalues
- COUNT and SELECT blocks
- Group entity and associated blocks
- Symbolic entities
- Macroinstruction capabilities
- Update feature
- Output editor
- Graphic capabilities

GPSS/360 also offers greater versatility by operating under OS/360 and DOS/360.

CITED REFERENCES AND FOOTNOTES

1. H. Herscovitch and T. Schneider, "GPSS III — an expanded general purpose simulator," *IBM Systems Journal* **4**, No. 3, 174–183 (1965).
2. G. Gordon, "A general purpose systems simulator," *IBM Systems Journal* **1**, 18–32 (September 1962).
3. R. Efron and G. Gordon, "A general purpose digital simulator and examples of its application, Part I — description of the simulator," *IBM Systems Journal* **3**, No. 1, 22–34 (1964).
4. The GPSS/360 program is available through IBM branch offices — GPSS/360/OS as Program 360A-CS-17X and GPSS/360/DOS as Program 360A-CS-19X. Related documentation published by the IBM Data Processing Division, White Plains, New York, includes: *Application Description* (H20–0186), *Introductory User's Manual* (H20–0304), *User's Manual* (H20–0326), *OS Operator's Manual* (H20–0311), *DOS Operator's Manual* (H20–0327), and *Education Guide* (R20–4075).
5. The term "macro" serves as an abbreviation for "macroinstruction" when used in combined forms.