DOCUMENT RESUME

ED 076 057 | EM 011 060

AUTHOR	Vivekananthan, Pathe S.
TITLE	How to Build a Simulation Model; A Computer Language Approach.
INSTITUTION	Virginia Polytechnic Inst. and State Univ., Blacksburg.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO	BR-7-0348
PUB DATE	Mar 73
GRANT	OEG-2-7-070348-2698
NOTE	25p.; Paper presented at the American Educat nal Research Association Annual Meeting (New Orleans, La., February 25 - March 1, 1973)
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Computer Programs; Flow Charts; *Models; *Programing Languages; Research Tools; *Simulation; Systems Approach
IDENTIFIERS	*General Purpose Simulation System; GPSS

ABSTRACT

The purpose of this project was to develop a "meta model"--a plan which could assist investigators in building models--using a computer simulation language. It is based on the assumption that there is a one-to-one correspondence between using a simulation language and building a model. The language selected is the General Purpose Simulation System (GPSS), which is built on the modular or block system. Each block has a specific function and can be arranged in such a way that the arrangement imitates the system under study. There are three steps involved in construction of a model: 1) delineation of system entities; 2) flow charting and selection of GPSS blocks, and 3) simulation. The definition of these blocks and selection of appropriate blocks for a problem are explained. (JK)

HOW TO BUILD A SIMULATION MODEL[1] A COMPUTER LANGUAGE APPROACH

Pathe S. Vivekananthan
Virginia Polytechnic Institute and State University

Introduction

When a researcher proposes an innovative educational practice he needs to demonstrate accuracy, reliability and validity of the new practice. An inexpensive way to do formative evaluation of the practice before implementing in a real situation is via simulation. A necessary ingredient for simulation is a dynamic model. A dynamic model realistically imitates a system being studied. It indicates not only what changes occur in part of the system as a result of a change in one part but also how changes take place. Brownell (1968) states: "It is no' :irely clear whether the difficulty in developing such theories or models is attributable to the present stage of thinking in the field of education or whether it is attributable to the inability of particular teams of workers to think through a dynamic model or theory which they can use as a first approximation of the phenomenon they are studying." One way to alleviate the present situation is to provide a plan--'meta model-''which could assist investigators in building models.

The approach taken in developing the meta-model is using a computer simulation language. It is based on the assumption that there is one to one correspondence between using a simulation language and building a model. That is, the process of 'loading' a simulation language with information is similar

[1]This article represents a part of research in the author's doctoral dissertation and was supported by the USOE grant to the Center for Occupational Education (Project No. BR-7-0348; Grant No. OEG-2-7-070348-2698; Transaction No. 71-0-E3002). The author wishes to thank his advisor Donald W. Drewes for his assistance in the study. Presented at the 1973 AERA meeting, New Orleans.

to constructing a model. The language selected for the purpose is the General Purpose Simulation System (GPSS).

GPSS Terminology

The main reason for selection of the GPSS is for its versatility. The GPSS is based on the modular concept. The modules are "blocks." Each block has a specific function and can be arranged in such a way that the arrangement imitates the system under study. These blocks are essentially equivalent to the components of a system. The principal entities in GPSS are described as Transactions, Facilities, Storages, and Logic. The transactions are dynamic in nature in that they are the particles that flow through the system according to the sequential arrangement of blocks. They are created and destroyed as required during the simulation. The principal attributes of the transactions are referred to as parameters which are used to define the properties of the transactions. The parametric values are assigned by the functions specified by the model builder. These value functions can be considered as the constraints imposed by the environment of the system. The model-builder obtains the functions from the systems analysis, i.e., through research.

The facilities and storages can be considered the resources of the system. These entities are representative of the equipment in the system used in the processing of transactions. Facilities are time-shared, and each facility can be considered as a service unit. A facility provides service to one transaction at a time. To obtain service from a facility, transactions have to wait in line, if the facility is in use. On the other hand storages are space-shared. For example, a storage can be a classroom. Since more than one transaction can occupy a storage, many transactions can obtain the service of a storage simultaneously. Counseling serves as an example of the logic entities.

Although counselors perform managerial functions they can also be considered resources available in the school system. The counselor helps to decide what course of action a student should take such as the field of study. Counseling helps to make decisions based on the student's attributes, i.e., his interests, his achievements, and other parametric constraints.

The indicators of the system's performance are provided by the statistical entities. The statistical entities store the required information and help to retrieve the stored information. The information that can be stored includes the status of a storage at a given time, the rate of service by a facility, total waiting time for a transaction to obtain services, average waiting time to obtain services, present state of the system, etc.

The operational entities are the processing blocks. They perform operations on the parametric values of the transaction. They change the values, assign new values, and also route the transactions. Operational entities provide the logic of the system by instructing the transactions where to go and what to do next. This type of handling of the transactions performs operations similar to system management functions.

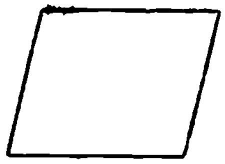
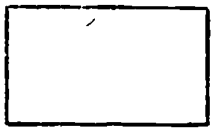
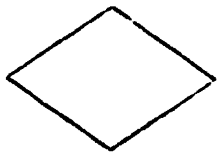
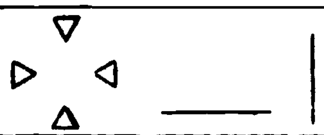
Model Construction

There are three steps involved in construction of a model. They are: (1) Delineation of system entities, (2) Flow-charting and selection of GPSS blocks, and (3) Simulation.

Delineation of System Entities. An initial activity in model construction is identification and delineation of system entities. An analysis of the system under study would give data on storages, facilities, performance, throughputs and processes the throughputs undergo. The data include information on the number of storages and facilities in the system, location of decision points, attributes and attribute changes of the transactions, etc. Once the general characteristics of the system are understood, flow-charting processes involved in the system can be attempted.

Flow-Charting. Flow-charting involves temporal or sequential arrangement of system activities. For convenience, the flow-charts are broken down into two categories: a) Static and b) Dynamic. A static flow-chart is an arrangement of basic standardized symbols given in Table 1 to outline basic procedures and to provide a guide for selection of GPSS blocks.

Table 1. Basic Flow-Charting Symbols and Description

Symbols	Description
	Input/Output Symbol - To indicate input or output operation. Information available for generating transactions, or recording of processed information such as system states.
	Processing Function Symbol - Defines operation/s causing change in value, form or location of information.
	Decision Function Symbol - It determines of a number of alternative paths followed. Used when decision has to be made as of which path to take.
	Direction Symbol - To indicate operations sequence and transaction flow direction. (lines and arrowheads)
	Termination Symbol - A terminal point in a flow-chart. To indicate termination of sequence.

In a static flow-chart, for example, the flow of a throughput is indicated via the arrangement of the symbols. This gives a general description of the system under study. The static flow-chart is not specific. For example, a specific flow-chart would indicate what happens to a transaction rather than what attribute of the transaction has been changed and how much. A static flow-chart would aid in drawing up a dynamic flow-chart of a block diagram. A dynamic flow-chart is specific in the process. For example it would indicate what attributes of the throughput changed, how much at what stage in the system. The GPSS blocks forms a viable tool for this purpose.

System operations are indicated by the blocks. Operational blocks are classified into six categories: (1) Input Operator, (2) Handling Operators, (3) Modifiers, (4) Information Storing, (5) Information Retrieving, and (6) Output Operator. There are one or more blocks for specific operations under each category.

Table 2. Classifications of GPSS blocks under six categories

Input	Handling	Modifier	Infor. Stor.	Infor. Ret.	Output
Generate	Split	Assign	Save Value	Tabulate	Terminate
	Assemble	Index			
	Match	Mark			
	Queue	Priority			
	Advance	Alter			
	Gather				
	Gate				
	Transfer				
	Test				
	Select				
	Leave				
	Loop				
	Seize				
	Release				
	Enter				
	Preempt				
	Return				
	Depart				
	Logic				

The model builder selects appropriate block/s to meet his needs. The input operator contains a block to generate transactions. This block is used to initiate the system activity. The handling group contains 19 blocks. They move transactions from one place to another. Their activity is essentially flowing the transaction. The model builder selects appropriate blocks for proper functions. For example, if a transaction needs services simultaneously from two facilities, the transaction can be broken into two using a 'split' block, and the two halves of the transaction can be reformed as one by the 'assemble' block. Five blocks are under the modifier group. Their essential function is to assign or modify attributes of transactions. For example, the 'assign' block assigns attribute values to a transaction. If the attributes have to be changed, the 'alter' block will do the necessary operation. The information storage block monitors system status and keeps a record of it. It records such information as number of transactions in a queue, average waiting time for service, etc. The information retrieval block would retrieve the information stored by the storage block. The block of output category removes transactions from the system thus terminating the processing of the transactions. The termination is the last system activity.

The blocks and information that is required in order to use the blocks in the block diagram are described briefly in the appendix. Parametric requirements for each block are given in form of questions. Answers to these questions would require extensive research. Arrangement of blocks with their parametric values (answers to the questions) in appropriate sequence forms the block diagram. A block diagram is essentially the model of the system. When the blocks with necessary information are arranged in an appropriate sequence they form a simulation model of the system. The models can be used to check out the system activities and their effects on the attributes of transactions. To do so one uses simulation technique.

<u>Simulation</u>. Simulation of the system would help to understand the intricacies of the model. A computer is a viable tool to do simulation. The simulation would show the effect of the interactive relationship of variables on attributes of transactions. Data on system status may reveal changes needed in the system. For example, it may furnish information on a number of additional resources (facilities and storages) needed to cut down waiting time. Similarly results may also indicate whether desired changes have taken place on the attributes of the transactions.

Conclusion

Construction of a model of a system requires great effort, resources and time. The skill of a model-builder plays an important role. Though a model may not solve problems, the process of constructing a model would enhance the thinking process. Aptly, Bloom (1968) stated, "Perhaps (~~educated~~ researchers) need to be reminded repeatedly that impractical theory or model building may be (their) most practical work."

References

Bloom, Benjamin S., 1968. Research and development centers: promise and fulfillment. Journal of Research and Development in Education, 1 (4): 181-189.

Brownell, S. M., 1968. Research and development centers and schools: a reaction to progress reports. Journal of Research and Development in Education, 1 (4): 173-180.

Appendix

A Description of the GPSS Blocks

GPSS Blocks

Generate. For an input-output model, the inputs provide the impetus for system activity. Without the inputs, the system is idle. Therefore, the first activity of the system entails a generation of inputs (through-puts). This is similar to a recruitment function.

The recruits (inputs) come from various populations whose characteristics are different from one another. Each recruit carries his population characteristics. Since some populations are more attractive to the system than others, the recruits hold different priority values according to their population. Each recruit also possesses his own idiosyncratic characteristics or personal attributes.

The rate of arrival of inputs may be constant or may vary from time to time. Similarly, the interarrival rate between two transactions (recruits) may be a constant or a variable. All transactions may not come from one population. It may be necessary to allocate or restrict the sample size for a population.

The GPSS block which serves the above mentioned purposes is called the GENERATE block. The GENERATE block creates the transactions that are inputed to the system. The generation of transactions follow specified conditions (statistical distributions). Activation of this block creates transactions in a given temporal sequence (interarrival times) with certain priority attributes and characteristics. The following information is necessary to use this block.

Questions

1. What is the average time between arrival of two transactions?

2. What is the spread (similar to standard deviation of the arrival rate)?

3. Is the spread a constant or a variable? If it is a constant, what is the value? If it is a variable, can it be expressed as a mathematical function?

4. What is the time that elapses before the first recruit arrives?

5. What is the sample size? How many transactions come from this population?

6. What is the priority level to be assigned to each of the transactions being generated from that population?

7. How many parameters (attributes) are assigned to each new transaction from this population?

Assign. Each transaction can carry up to one hundred different attributes with it. It should be remembered that all attributes are numerical in value. When a transaction undergoes a process or an operation, changes may take place in one or more of its attributes. For example, a student's attributes may change as a result of a learning process or learning experience. In other words, attributes of throughputs are changed by some operation. The ASSIGN block represents the system activity that changes or modifies an attribute of a transaction. This block is the principal means of establishing the initial values of the parameter fields (attributes) of each transaction from a derived population.

Questions

1. Which of the attributes of the entering transaction is to be altered by this operation?

2. Is a value to be added to the current value, subtracted from the current value, or to replace the current value with a new value?

3. What is the function used to alter the specified parameter?

Index. The INDEX block is similar to the ASSIGN block in that it modifies the attribute value of a transaction but only to the extent that the first attribute is modified. No other attribute other than the first can be modified by the INDEX block. This block is very convenient to substitute or replace the first attribute value with some other attribute. The new value of the first parameter can be the value of another parameter or the result of a constant added to another parameter value. Thus, the new value of the first parameter, for example, can be the value of the third parameter plus a constant.

Questions

1. What is the other parameter that is used to make changes on the first parameter?

2. What is the constant which when added to the parameter value is assigned to the first parameter?

Priority. When transactions are competing for facilities or storage spaces, some transactions need to use the equipment prior to other transactions because of the priority level. Thus, some transactions may hold higher priority and some other transactions may hold low priority. The PRIORITY block is used to set the priority value for a transaction. The priority values range from 0 to 127. Higher numerical values indicate higher priorities.

Question

1. What is the priority level (number) assigned to a transaction?

Seize and Release. A facility is a resource unit which can serve only one transaction at a time. A transa[illegible]ts getting serviced by a facility when the transaction enters [illegible]IZE block. When a facility is serving a transaction, another transaction cannot obtain the same facility. The transaction which is currently using the facility must RELEASE the facility in order for another transaction to use it. The SEIZE block records the use of the facility by the entering transaction in such a way that the facility remains in use until the seizing transaction enters the corresponding RELEASE block. A transaction can use more than one facility at a time.

A copy of a book is an example of a facility. Only one student can use the book at a time. The book is in the student's hand as soon as he seizes it. He has to give up the book (release) in order that another student can use it.

Question

1. What is the facility that is to be used by the entering transaction? There should be a corresponding RELEASE block which will indicate the release of facility by the entering transaction which had previously seized the facility.

Preempt and Return. The PREEMPT block is used in order to free a facility which is currently in use. The transaction which enters the PREEMPT block suspends the progress of the transaction which is currently using the facility and seizes the facility. However if the facility is processing a transaction which has already preempted, the entering transaction cannot free the facility. That is, if the facility is already in the preempt condition the PREEMPT block can refuse entry to another transaction. However, there is an exception when the PREEMPT operates in the priority mode. If the PREEMPT block is operating in the priority mode, the transaction which has the higher priority value can use the facility. Thus, if the facility is already operating in the preempt condition the transaction which seeks the facility can use the facility if it holds a higher priority value than the transaction which is currently using the facility. The preempted transaction (the transaction which has lost [illegible]trol of the facility) always contends for the facility. As soon as the preempting transaction (which temporarily suspended the operation of the facility) releases the facility by entering the RELEASE block, the preempted transaction will seize the facility. The preempted transaction can do one of two things during the preempt condition: (1) wait for the facility or (2) perform some other activity.

The PREEMPT block temporarily stops the progress of the transaction which is currently using the facility and permits the preempting transaction to obtain the control of the facility. The RETURN block serves the function of ending the state of preempt condition of the facility.

Questions

1. What is the facility number which is sought by the preempting transaction?

2. Does the PREEMPT block operate in priority mode?

3. Where does the preempted transaction go?

4. If the preempted transaction is to wait, which of its attributes records the waiting time?

5. Does the preempted transaction need to contend for the facility?

Enter and Leave. A storage is one of the resources available to the system. A storage can serve more than one facility at a time. The ENTER block records the usage of a storage by a transaction. By entering the ENTER block a transaction can obtain the services of the storage. A transaction can occupy more than one space (unit) of the storage. Thus, a transaction can use several units of storage. A library can serve as an example of a storage. It contains a limited number of books. An entering student can borrow any number of books that he wants. The ENTER block keeps the record of the library activity. If the storage is full the transaction can depend upon the number of units that would be occupied by the transaction which seeks entry. If the entering transaction demands more storage units than the unused storage units, entry will be denied to the transaction.

The LEAVE block is the opposite of the ENTER block. The LEAVE block removes a number of units from the contents of the storage. A transaction need not remove the same number of units that is added to the storage. However, as many units should ultimately be removed as were previously added. In the library example, if a student borrows ten books, he may return five books after some time, then three books, and two books at a later time. However, he must return all the ten books that he borrowed initially.

Questions

1. What is the storage that the entering transaction occupies?

2. How many storage units does the transaction utilize?

3. When the transaction leaves the storage how many storage units does it free?

Queue. When a transaction is denied entry to a block it is forced to wait. This causes delay in the progress of the transaction. The QUEUE block is used to measure the delay in the flow of transactions. This block automatically gathers statistical information on the delay of the transaction, such as average delay time, number of transactions in the waiting line (queue), and average number of contents in the queue.

Questions

1. What is the queue where the information should be stored?

2. How many time units are added to the content of the queue by the delayed transaction?

Depart. The DEPART is similar to the LEAVE block. This block is used to remove a transaction from a queue block. Thus, when a transaction encounters this block, computation of the delay time (waiting) is immediately stopped. Each QUEUE must have a corresponding DEPART block. All the contents of a QUEUE block are removed subsequently by one or more DEPART blocks.

Question

1. When a transaction is removed from the queue how many units are reduced from the content of the queue?

Logic. A transaction may make a binary decision. For example, a student passes or fails in a course. Accordingly a record is kept. A logic switch is the record where set (pass) or reset (fail) condition is kept. At later points the condition on the logic switch can be used to make decisions. The LOGIC block is used to determine logic switch condition. The entering transaction in the LOGIC block fixes one of three conditions on a specified logic switch. The three conditions are set, reset, or invert.

Questions

1. What is the change made by the entering transaction?

2. What is the logic switch where the change is made?

Gate. The GATE block is a junction block where a decision is made on the route of a transaction. At this block the path of the transaction is altered. Decisions depend on the system status. The route of the transaction is decided on the basis of the system status and prescribed decision rules. Thus, the GATE block serves a decision point where the status of the system is checked and
transaction is made depending upon the specified alternatives.

The GATE block operates on two modes: a "refusal" or "conditional" entry mode and a "transfer" or "unconditional entry" mode. The conditional mode does not allow entry to a transaction if specified conditions are not met. Thus, the conditional mode would allow a transaction to enter only if the specified condition is satisfied. The transaction is delayed until the specified or desired condition is accomplished.

In the transfer mode, when the desired condition is not satisfied, the route of the transaction is altered. Thus, when the system status is not in a desirable condition, the transaction does not flow in a sequential order, but takes an alternate path.

There are twelve system status conditions, of which one is used for decision-making. There are twelve mnemonics associated with the twelve conditions. They are:

NU = Facility not in use
U - Facility in use
NI = Facility not in preempt state
I = Facility in preempt state
SE = Storage empty
SNE = Storage not empty
SE = Storage full
SNF = Storage not full
LR = Logic switch in RESET state
LS = Logic switch in SET state
M = Check for MATCH condition
NM = Check for no MATCH condition

In order to use the GATE block, the following information is required:

Questions

1. What is the system condition to be tested to make decisions? Select a mnemonic code.

2. If the desired condition is not met, what does happen to the transaction?

Transfer. A transaction flows through a system, going from one block to another. The flow of the transaction is dictated by the system attributes and/or transaction attributes. Normally a transaction moves sequentially, flowing from one block to the next sequentially numbered block; however, this is not always so. The flow may be modified due to system attributes and/or transaction attributes. Changes in the flow of the transaction are made at a decision junction, such as the TRANSFER block.

The change of flow is done at the TRANSFER block. The modification is done in one of four ways: unconditionally, conditionally, statistically, or logically. The choice is specified by a mnemonic selection mode. The mnemonic code blank stands for unconditional transfer which would transfer the transaction to the specified block. The conditional transfer has three options--BOTH, ALL, and SIM. Using the BOTH code one can indicate the next two blocks where the transaction would move to one of the two blocks. If the transaction cannot find entry to the first indicated block, it will try to enter the second block. If the second block also does not allow entry, then the transaction will enter one of the two blocks which becomes available. The ALL code allows the testing of many blocks. The first block and the last path are specified. There is also a provision to test only selected blocks between ranges of the first and last blocks. The SIM condition tests for the condition where all specified blocks are simultaneously free. If free, the transaction moves to the next sequentially numbered block. If the condition is not met, the transaction is delayed until the condition is satisfied.

The statistical transfer refers to the condition where the transaction can take one of two paths depending upon the specified probability value.

Questions

1. What is the mnemonic code?
2. What is the next path that the transaction is to take?
3. If statistical condition is used, what is the probability value associated with each path?

Advance. A transaction moves through the system and operations take place on the transaction. Operations take time to work on the transaction. In other words, the progress of the transaction is delayed. The delay may be due to the process time. For example, a student takes time to learn. The ADVANCE block is used to indicate the time taken in processing the transaction. Thus, this block is used to indicate time to perform an operation in the system. The time is indicated by arbitrary time units, called action time.

Questions

1. What is the average action time?
2. What is the spread of the action time? Is it a constant or a variable? If it is a variable, what is the mathematical function?

Loop. In some cases a transaction should repeatedly undergo the same set of operations. The LOOP block facilitates doing just this. By using the LOOP block a transaction can undergo the same processes many times. Thus, the LOOP block can control the number of times a transaction can pass through a section of blocks. The number of cycles depends upon the content of a parameter of the transaction. For example, if the content of a parameter is five, then the transaction would cycle five times. Thus, the content of the parameter serves as a counter for cycling process. It is also necessary to indicate the starting cycling block.

Questions

1. What is the starting block of the cycle?

2. What is the parameter whose content will be used to indicate the number of looping process?

Split. In cases where one transaction has to undergo more than one operation simultaneously, duplicates can be created from the transaction. For example, in a given time unit, say, a semester, a student has to take more than one course at a time. For convenience, it can be considered that each course is taken by a different student. To accomplish this purpose the SPLIT block can be used. The SPLIT block serves the function of creating transactions which are not completely new but are offsprings of a parent transaction. There should exist a transaction which can be used to create siblings.

The transactions (siblings) which are created by the SPLIT block hold identical attributes as those of the parent transaction. However, the offspring need not possess all the attributes that the parent holds. For example, if the parent has ten attributes, its offspring can be created such that they possess only the first six parental attributes. The offspring and their parent can be serialized.

Questions

1. How many duplicates (offspring) need to be created?

2. After the creation where do they go next? What is the next operation (block) on the duplicates soon after their creation?

3. How many of the parental attributes do the offspring inherit?

4. If serialized, what is the parameter where the serial number is kept?

Assemble. An offspring (a duplicate) can belong to only one parent. The parent and its offspring (duplicates) are composed under one assembly set, which can contain any number of transactions (parent and its duplicates). Each transaction that is created by the GENERATE block forms an independent assembly set. When a transaction of an assembly set is terminated the content of the assembly set is reduced by one count. However, the assembly set continues to exist until all members are removed from the set.

Offspring (duplicates) are created by the SPLIT block. The duplicates are created in such conditions when a transaction has to perform more than one operation simultaneously. The ASSEMBLY block performs the opposite operation to that of the SPLIT block. When simultaneous operations are over, the duplicates can be recombined into one transaction. The number of duplicates so combined need not be the same as they were generated by a SPLIT block. For example, the SPLIT block may duplicate ten offspring from one transaction, giving eleven members in the assembly set. The ASSEMBLE block may recombine six of these into one, thus leaving an assembly set consisting of seven members. THE ASSEMBLE block is used to re-combine a specified number of members of an assembly set. Thus, the ASSEMBLE block serves the function of collecting the offspring and re-combining them into one transaction.

Question

1. How many offspring are recombined?

Match. The MATCH is used to synchronize the progress of two transactions of an assembly set. This block checks for the condition rather than combining two transactions. The match condition is successful when two siblings enter two MATCH blocks separately, where one MATCH block is the conjugate of the other. When a transaction enters a MATCH block it waits until another transaction of the same assembly set enters the conjugate MATCH block. After the match occurs the two transactions are ready to advance. Thus, the progress of the transactions is withheld until the MATCH occurs.

Question

1. What is the conjugate block?

Gather. When more than two transactions have to be synchronized, the GATHER block can be used. It is similar both to the ASSEMBLE block and the MATCH block. The GATHER block collects more than two transactions like the ASSEMBLE block, but does not destroy any of the transactions. It is similar to the MATCH block since it delays the transactions until necessary conditions are satisfied, then the collected transactions may proceed.

Question

1. What is the number of transactions of the same assembly set that must arrive at the GATHER block before all of the transactions are permitted to proceed?

Mark. A transaction takes time to flow through the system, from one block to another. The time taken by a transaction to flow through from one block to another is called transaction "transit" time. When necessary, the transit time can be computed by using the MARK block. Whenever a transaction enters the MARK block the current clock time (MARK TIME) is recorded in a specified attribute of the transaction. The transit time is equal to the current MARK TIME, minus previous MARK TIME. For example, to measure the transit time of a transaction through a set of processes the MARK TIME is recorded at the beginning of the processes in one of the attributes of the transaction. At the completion of the processes the MARK TIME is recorded on another attribute. The difference between the contents of the two attributes gives the transit time.

Question

1. What is the attribute where the MARK TIME is recorded?

Savevalue. The SAVEVALUE block serves the function of a bookkeeping operation. There are several storage units (record books) where information is stored. The SAVEVALUE is used to store information in a specified place. It stores defined values. When a transaction enters this block the specified information is stored in a cited record for future reference.

Questions

1. What is the SAVEVALUE (record) number where the values should be stored?

2. What is the value to be stored?

Tabulate. This is similar to the SAVEVALUE block. It stores statistical information. The TABULATE block serves as an accounting procedure where necessary statistical analysis is made on the values. The necessary statistical analysis is defined by TABLE definition cards. When a transaction enters a TABULATE block it gathers specified information according to the specified table definition card.

Questions

1. What is the TABLE definition card where the information is to be stored?

2. Are the gathered values weighted? If so, by what factor?

Terminate. Each transaction which enters the system should leave the system ultimately. A transaction leaves the system when certain conditions are met. For example, a student leaves the school system when he successfully completes the requirements. A transaction comes out of the system via the TERMINATE block. The block removes transactions from the system.

In a simulation process, a "run termination count" is used to indicate the total number of transactions that should run through the system before printing the final summary statistics of the system variables. When a transaction enters the TERMINATE block, the transaction removes a specified number of termination counts. The total terminal count is reduced by the amount that the transaction carries along with it when the transaction enters the TERMINATE block. When the terminal count reaches zero or less the system shuts down. If this does not happen, the simulation model runs indefinitely. Theoretically it is feasible that some transactions may be stranded within the system when the system shuts down. Unless it is deliberate, caution should be taken that such a condition (stranding transactions in the system) be avoided.

Question

1. How many terminal counts of transactions are removed from the system when a transaction enters the TERMINATE block?

Test. The TEST block serves the function of comparing two arguments, and the resulting condition will dictate the flow of the transaction. The condition is specified in the form of an algebraic comparison between the two arguments. If the desired condition exists the transaction will proceed in the usual manner. If the stated condition does not occur, one of two actions will be taken on the transaction. The transaction will wait until the desired condition is attained or the transaction will be directed next to a specified block. For example, a student's grade can be tested to see if it meets the passing level. If the grade meets the condition, the transaction moves in the usual manner. If it is not so, the student may be asked to repeat the course, or wait until the passing level is changed.

There are six mnemonic codes to specify the desired condition.

L The desired condition is the first argument less than the second argument.

LE The desired condition is that the first argument is less than or equal to the second argument.

E If the first argument equals the second, the desired condition is satisfied.

NE If the first argument is not equal to the second argument, the condition is satisfied.

G The desired condition is that the first is greater than the second.

GE The first argument is greater than or equal to the second argument

Questions

1. What is the desired mnemonic code?

2. What is the first argument?

3. What is the second argument?

4. If the desired condition is not satisfied what happens next to the transaction? What is the next block?

Prior to gathering information for the GPSS blocks, general information about the system is deamed necessary to construct flow-charts of the system. The general information would dictate selection of appropriate blocks whose sequence specified the flow diagram. A procedure is given in this section which would be helpful in gathering information about the system so as to use the GPSS blocks in flow-charting the system.

The collection of information can be encompassed under four categories. Information on the first category deals with the input of the system. The second category pertains to resources available in the system. Information about the system activities forms the third category. The last is concerned with the output information.

Transaction. Inputs to the system are defined as transactions in the GPSS terminology. As the inputs initiate the system activities, they form a good starting point to gather information about the system. Initiation of system activities starts with generating transactions. The following questions facilitate collecting information concerning the transaction.

Questions

1. What are the transactions to the system?

2. Identify the populations from which the transactions are drawn.

3. How many different populations are there? Each population requires a GENERATE block.

4. Is there an upper limit on the number of transactions drawn?

5. If so, how many transactions are drawn from each population? (See GENERATE block).

6. What are the characteristics of each population?

7. Do the characteristics dictate different priority levels on the transactions? If so, what is the priority level? (See GENERATE block.)

8. How many attributes (parameters) are necessary to describe a transaction? (See GENERATE block.)

9. What are the initial attribute values? (See ASSIGN block.)

Resources. Resources refer to the facilities, storages, and the logic switches associated with the system. Resources offer services that the transactions have to use to undergo changes in their attributes. A facility will offer services to only one transaction at a time. Many transactions can use the services of a storage. However, a storage has a finite capacity. A logic switch records the result of binary decision, which is useful in directing the transactions.

Questions

1. How many facilities are available in the system?

2. How many storages are available in the system?

3. What is the capacity of each storage?

4. How many logical switches are there?

Components. System components refer to the activities that a transaction goes through. These activities not only direct transaction flow but also modify the transaction attributes. Thus the system activities can be classified under two groups. The activities under the handling group dictate the flow of a transaction. The second group of activities modify or change the transaction attributes.

Questions (Handling Group)

1. How many activities does a transaction undergo simultaneously? (See SPLIT block.) This indicates the number of parallel activity paths of a transaction.

2. When does a transaction have to perform simultaneously?

3. When does the simultaneous operation stop? This indicates convergence of parallel paths. (See ASSEMBLE block.)

4. What are the processes that delay access to transactions and by so doing form waiting lines? (See SEIZE and ENTER blocks.)

5. Where are the waiting lines located? (See QUEUE block.)

6. Do all transactions have the same priority for processing? (See PRIORITY and PREEMPT blocks.)

7. If extant, are there two or more simultaneous activities that must be satisfied before a transaction can proceed? (See MATCH and GATHER blocks.)

8. Are the system attributes used to modify or delay the transaction flow? (See GATE block.)

9. When is the transaction attribute used to modify the flow? (See TRANSFER block.)

10. When and what are the two arguments that are used to change the transaction flow? (See TEST block.)

11. What are the activities that require a set of repeated operations? (See LOOP block.)

12. When does a transaction leave the waiting line? (See RELEASE, LEAVE, and DEPART blocks.)

13. How much processing time is required at each facility or storage? (See ADVANCE block.)

After obtaining information about the above questions, and using the information as guidelines, answer the following questions.

14. What are the major activities of the system?

15. What are the sequential activity paths taken by the transactions?

<u>Questions</u> (Modifier Group)

1. How many times are transactions attributes altered?

2. When do they change?

3. How are they modified? Can any mathematical functions be developed? If so, what are the functions? (See ASSIGN block.)

4. When is only the first attribute of the transaction altered? (See INDEX block.)

5. When are the priority values assigned to the transactions? (See PRIORITY block.)

6. Is the transit time to be kept? If so, from what initial point to what terminal point in the system? (See MARK block.)

Output. The output information is concerned with bookkeeping pertaining to system and transaction activities. The information comes out in printed forms, when all transactions leave the system. It is necessary, however, to specify the record-keeping format.

Questions

1. When does a transaction leave the system? (See TERMINATE block.)

2. What type of information is collected? (See Table 9.)

3. What is stored in the system memory for future action? (See SAVEVALUE and TABULATE blocks.)

4. How many records (books and tables) are required to keep the information? (See TABULATE and SAVEVALUE blocks.)

5. Is it necessary to calculate the frequency distribution? If so, how many frequency classes and what are the upper limits? (See TABULATE block.)

Table 9. Standard numerical attributes

Entity	Symbol	Meaning
Transactions	P	Parameter, fullword halfword
	PR	Priority
	MI	Transit Time
	MP	Parameter Transit Time
Blocks	N	Total Entry Count
	W	Current Count
Facilities	F	Status of Facility
	FR	Utilization (Parts/Thousand)
	FC	Entry Count
	FT	Average Time/Transaction*
Storages	S	Current Contents of Storage
	R	Remaining Contents
	SR	Utilization (Parts/Thousand)
	SA	Average Contents*
	SM	Maximum Contents
	SC	Entry Count
	ST	Average Time/Transaction*
Queues	Q	Current Length of Queue
	QA	Average Contents*
	QM	Maximum Contents
	QC	Total Entry Count
	QZ	Number of Zero Entries
	QT	Average Time/Transaction*
	QX	Average Time/Transaction Excluding Zero*
Tables	TB	Table Mean*
	TC	Entry Count
	TD	Standard Deviation*
Savevalues	X	Fullword Savevalue
	XH	Halfword Savevalue
Groups	G	Number of Items in Group
Functions	FN	Function

*Truncated to an integer