The fundamental concepts of block-diagram structure and transactions have been retained throughout the evolution of GPSS [4]. Since its inception, GPSS has maintained the objective of catering to the analyst. The only major change in language structure increased its power and simplified its use but retained the fundamental block-diagramming concept. Features have been incorporated into the language either to reduce modelling effort without complicating the coding or to increase the scope and details of models. GPSS is particularly well-suited for modelling traffic and queuing systems, situations in which it is necessary to study the interactions of a dynamic situation in detail. Here we refer to GPSS-PC (c) Copyright Minateman software as GPSS.

GPSS has 14 types of entities

I. Basic entities
   1. Blocks
   2. Transactions

II. Equipment entities
   3. Facilities
   4. Storages
   5. Logic switches

III. Computational entities
   6. Arithmetic variables
   7. Boolean Variables
   8. Functions
The General Purpose Simulation System operates by moving transactions from block to block of the simulation model in a manner similar to the way in which the units of traffic they represent progress in the real system. Block diagrams or flow diagrams are widely used to describe the structure of the systems. The units of traffic that move through the system depend upon the system being simulated. Units might be messages in a communication system, vehicles in a traffic system, work items in a production system or any number of other items. These units upon which the system operates in the GPSS program will be called "Transactions". The GPSS program also has various other entities like facilities, storages, queues etc., whose attributes are changed by the movement of the transactions through the various block types.

Statistical variations may be introduced in the block diagram. It is also possible to simulate the interdependence
of variables in the system such as queue lengths and input rates. In order to simulate a system, it must first be represented in terms of these concepts and block types. The program then creates transactions, moves them through the specified blocks, and executes the actions associated with each block. When the transactions are blocked and cannot move at the time they should, the program moves them as soon as the blocking conditions change.

The GPSS program does not simulate the system at each successive interval of time. Instead, it updates the clock to the time at which the next most imminent event is to occur. A GPSS block diagram model can formally be considered as a set of interrelated logical and mathematical symbols which represent those aspects of a system which are of interest. Each model consists of various elemental abstractions, called entities, by which the system is represented.

GPSS has evolved steadily over years and now contains features not contemplated in its earlier releases. Most simulation models do not use all the capabilities of GPSS, therefore useful simulation can be accomplished with a subset of GPSS instructions [4]. The instructions of GPSS are called BLOCKS because they are associated with the blocks of the flowchart of the model. Permanent entities are created by GPSS when first referenced in the program. Temporary entities are created when required by the simulation during program
execution. These entities called the transactions are created by the blocks GENERATE and SPLIT.

A GPSS program consists of blocks logically interconnected. Every block performs a given function on permanent or temporary entities. Blocks are instructions of the GPSS language. The activation of a block and the execution of a GPSS instruction are caused by the arrival of a transaction at the block. The movement of transaction from one block to the next is controlled and executed automatically by the simulator. When a transaction enters a TERMINATE block, it is destroyed and references to it are removed from the model.

THE SIMULATION CLOCK

In real systems time passes as various events occur. Since we are processing the various events sequentially, we perform various activities to be carried out at time "t" and then advance the time to the next earliest event. Some of the features of the clock are as follows:

* The GPSS clock registers only integer values.

* The unit of time which the clock registers is determined by the analyst and never stated explicitly in the model.

* GPSS is a next event simulator i.e., after a model has been fully updated at a given point in simulated
time, the clock is advanced to the nearest time at
which one or more next events are scheduled to occur.

TRANSACTION STRUCTURE:

The meaning of Transaction is determined by the model
builder. Correspondence is established between the
transaction and the elements of the system being modelled.
They are never explicitly declared in the GPSS model. Some
examples of possible analogies between transactions and
elements of real systems could be as follows:

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>ELEMENT (Transactions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highway</td>
<td>Vehicle</td>
</tr>
<tr>
<td>Super market</td>
<td>Customer</td>
</tr>
<tr>
<td>Maintenance Shop</td>
<td>Parts</td>
</tr>
<tr>
<td>Production Process</td>
<td>The product.</td>
</tr>
<tr>
<td>Port</td>
<td>Ship</td>
</tr>
</tbody>
</table>

When a simulation first begins, no transaction exists
in a GPSS model. As simulation proceeds, transactions enter
the model at certain times according to logical requirements
of the system being modelled.

Transactions provide facilities to hold the information
like priority, mark time, and various other attributes of the
transactions while traveling in the model.
GENERATE BLOCK:

The GENERATE Blocks can be thought of as a door through which transactions enter a model. Transactions can be made to enter the model at different points in time. The time between two consecutive transaction arrivals at a given GENERATE block is called inter-arrival time. The block to which the transactions move is the next sequential block following the GENERATE block. There can be any number of generate blocks in a model.

When the GPSS program encounters a GENERATE block one transaction is immediately created. The set of four basic fields for the new transaction is obtained from the internal chain of inactive transactions, which is a stack.

TERMINATE BLOCK

This block always accepts transactions and transactions are removed from a model whenever they move into a TERMINATE block. There may be any number of terminate blocks in the model. The A operand of the terminate block is called the termination counter. The termination counter is a special counter in which a positive integer value is stored at the time a simulation is begun. As the simulation proceeds, when the transactions move into a terminate block with the A operand, this counter is decremented. When this counter reaches a value of zero or less, the simulation stops. Though
there may be more than one termination block, there is only one termination counter and this termination counter will be decremented whenever a transaction moves into any terminate block in a model. The termination counter is supplied with its initial value at the time simulation begins. It uses the A operand of the START blocks as the initial value for the termination counter.

Movement of transactions into the TERMINATE block which does not have the A operand does not decrease the value of the termination counter.

START

The GPSS processor starts the simulation when it encounters a START block. It uses the A operand on the START block as the initial value of the termination counter. The TERMINATE block and START block are used in harmony to control the duration of the simulation run.

SEIZE & RELEASE

In GPSS the term "facility" is a synonym for "Server". Just as there can be many servers at different points in a system, there can be many facilities in a GPSS model. Names are given to facilities making it possible to distinguish among them. The names are given by the model builder. When the server is being used the following steps are followed:
1. The transaction waits for its turn, if needed.
2. When its turn comes, it engages the server.
3. It holds the server in a state of capture while the service demands are performed.
4. When the demanded service has been performed, the server is released.

ADVANCE

The ADVANCE block accomplishes the task of freezing a transaction's motion for a prescribed length of time. The information required to describe the applicable service time distribution is expressed through the advance block's A and B operands.

QUEUE & DEPART

The above blocks are to some extent similar to the Facility-Depart blocks. When a transaction moves into the QUEUE block, the event "join a waiting line" is simulated and when a transaction moves into the DEPART block the event "depart a waiting line" is simulated. Statistics describing key features like the following are gathered.

* Total entries to the waiting line.
* Total number of entries with zero waiting time.
* The maximum queue length.
* Average number waiting.
* Average waiting time.
On many occasions, even when waiting is known to occur at certain points in a model, the model builder chooses not to use the queue entity at those points, when he is not interested in gathering statistics at those points. The result of not using the QUEUE-DEPART blocks is a savings in computer time.

TRANSFER

This block is used to divert the transactions to some non-sequential block in a GPSS model. It could be used in a conditional or unconditional or random transfer. Thus it could be used in more than one mode.

ENTER & LEAVE

The parallel servers are represented by the above blocks. The transaction captures the parallel server. The transaction waits, if necessary, then it captures a server, it holds the server in a state of capture over some interval of time and finally it releases the server.

Before making use of the ENTER and LEAVE blocks the statement STORAGE associated with it should be defined. This is the capacity definition statement.
PRIORIT Y

When a transaction enters a model, its priority level is set equal to the value specified by the \textit{E} operand at its \texttt{GENERATE} block. The priority level value influences the chronological sequence in which various transactions move forward in a model. The current transactions priority value can be changed by the \texttt{PRIORIT Y} block.

ASSIGN

Each transaction in a GPSS model possesses a set of a maximum of 100 parameters. As a transaction moves through a model, its parameter values can be assigned and modified as the modeller wishes. The parameter "\textit{n}" is referred to as \textit{Pn} and it is not possible to name them symbolically. The parameters can only be assigned integer values and the meaning of the parameter is determined by the analyst. The initial values of all the parameters are set to zeros.

TEST

The relation between the values of two standard numerical attributes can be examined by the use of the \texttt{TEST} block. The \textit{A} and \textit{B} operands are the names of the two standard numerical attributes involved. An auxiliary operator \textit{X}, indicates the way the two attributes are to be compared against each other.
GATE

The GATE block's A operand states the name of the logic switch to be tested. When the transaction finds the GATE closed, it is held at the block preceding the GATE, contributing to the current count there.

CLEAR

This clears all model statistics and sets them to zero. This also removes all the transactions in the model at the end of the simulation. This also reinitialises the counter.

RESET

In most cases, the initial model consolidations may vary markedly from those when a steady state is reached. In order for the simulation to be realistic, the statistics collected during the initial phase have to be discarded so that the statistics corresponding to the steady stock of the model are available for analysis. This is achieved by use of the RESET block.

A frequent practice is to use the model itself, in experimental fashion, to estimate the duration of simulation required to reach steady state. The RESET card can be used for this experimentation.
FUNCTION SPECIFIER

When it is required in a model to process a sample from non-uniform distributions to determine inter-arrival times and service times, the uniformly distributed random number drawn from 0 - 1 population is taken and converted into non-uniform distribution of interest. It draws a random number from a uniform distribution source, then this value is used to perform a table look up on the cumulative distribution for the population of interest. For this the further details have to be described by a FUNCTION statement.

A continuous GPSS function is initially evaluated in the same way as a discrete function. When the function is called, a number is first drawn from the random number stream used as the function argument. Then a table look up is performed to determine the cumulative probability interval in which the random number falls.

If the function is defined to be continuous, a linear interpolation is performed next between the pair of points at the ends of the cumulative probability interval. The number resulting from the interpolation is returned as the function value. Due to the linear interpolation feature of continuous GPSS function all values in a given cumulative frequency interval have an equally likely chance of occurring.
RMULT

There are eight random number generators associated with GPSS. The value of a given generator's multiplier changes each time the generator is used to produce a random number. The multiplier's initial value can be set by the modeller using the block RMULT. The default value is 1. This Block can be used

* before the first START block to change the default initial value of one or more multipliers before a simulation begins.

* between START blocks, to restore the values of the multipliers to the initial values.

* between START blocks to define an entirely new set of values for the multipliers.