Pramod Parajuli
Simulation and Modeling, CS-331

Chapter 11
Introduction to GPSS

General Description

- Blocks → activities
- Lines → sequence of activities
- For a choice, more than one line leaves a block and the condition for the choice is stated at the block
- Entities move through the system e.g. messages in a communication system, motor vehicles in transportation ..

Block

- Name of block
- Set of data fields (A,B, C, …)

General Description

Resources

- GPSS Entities
- Block statements
- Reports
- Savevalue
- Expressions
- Logic Switches
- Plot
- Tables
- Queues
- Storages
- Data streams
- User chains
- Math libraries (RNGs, Statistical Analysis, etc.)

GPSS Entities

- Transaction
- Facility
- Logic switch
- Queue
- Save value
- User chain
- Numeric group
- Block
- Function
- Matrix
- Storage
- Table
- Variable
- Transaction group

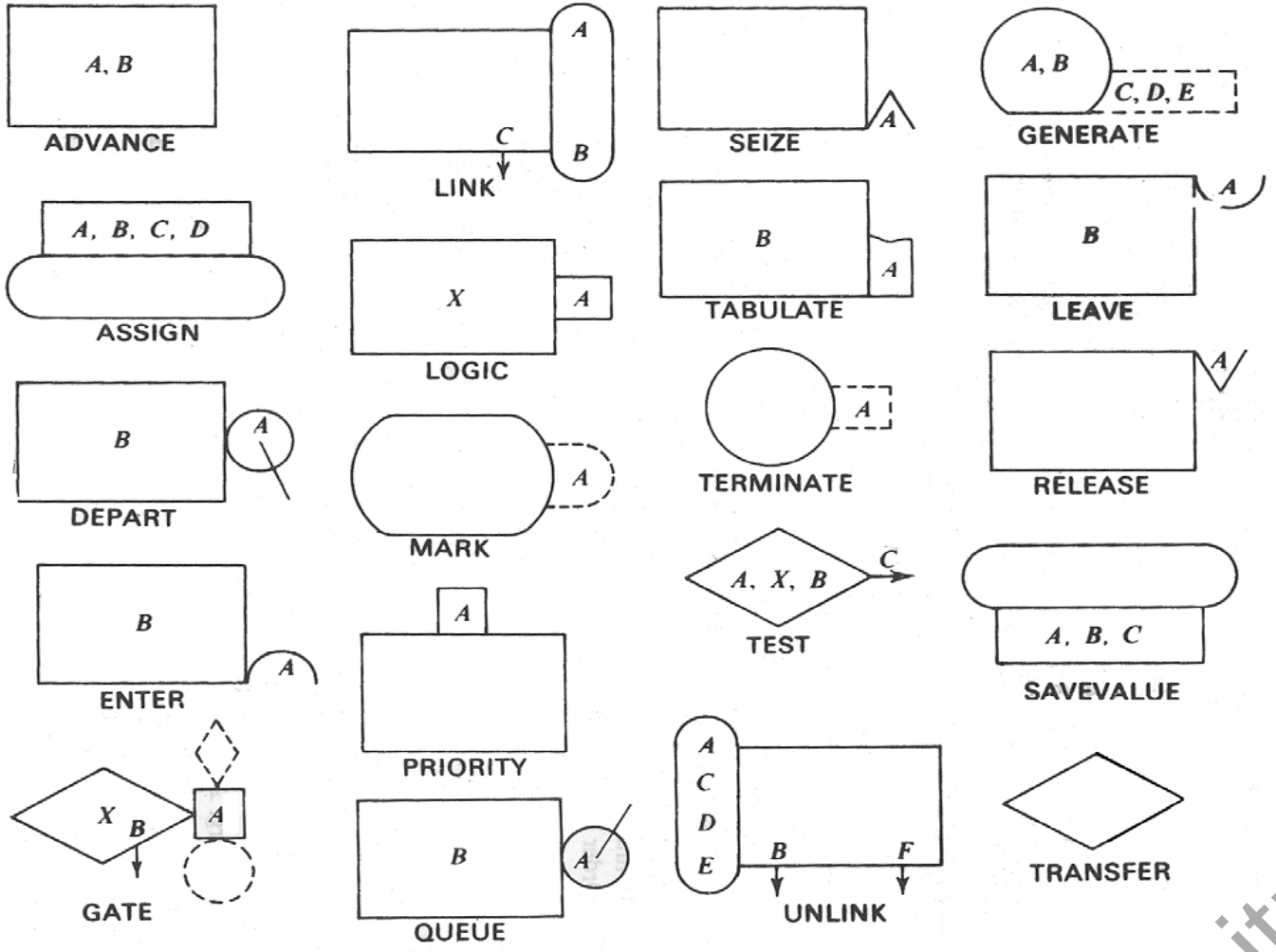
General Description

GPSS – Basic commands

CLEAR	reset statistics and remove transaction
CONTINUE	resume the simulation
EXIT	end the GPSS world session
HALT	stop the simulation and delete all queued commands
INCLUDE	read and translate a secondary model file
INTEGRATE	automatically integrate a time differential in a use variable
REPORT	set the name of the report file or request an immediate report
RESET	reset the statistics of the simulation
SHOW	evaluate and display expression
START	set the termination count and begin a simulation
STEP	attempt a limited number of block entities
STOP	set a stop condition based on block entry attempts
STORAGE	define a storage entity
TABLE	define a table entity
VARIABLE	define a variable entity

General Description

GPSS – Blocks



General Description

GPSS – Blocks

- GPSS block diagram consists of many blocks
- An identification number called a '***location***' is given to each block
- Movement of transactions is usually from one block to the block with the next highest location
- Blocks can be given names for the identification if required

Transaction

- Transaction move from block to block
- Transaction attempting block to block is called '***active transaction***'
- If a transaction fails to find favorable conditions while entering a block, it may come to rest. Then, another transaction is chosen to begin
- Transactions are numbered sequentially throughout a session starting with 1
- CLEAR statement begins the numbering of transaction at 1
- Behavior of transaction is determined by its attributes;

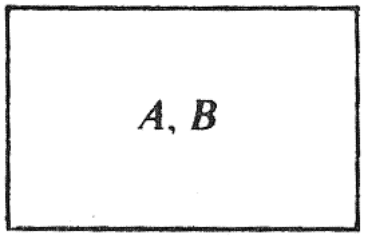
  GPSS World Reference Manual – Chapter 4 (r4.html), page 3
- ASSIGN, MARK, TRANSFER SUB, SELECT, SPLIT, COUNT blocks create a transaction parameter

Transaction

Active transaction

- Active transaction go as far as it can
- When it can not move further, another transaction is chosen to be active
- There can be no more than one active transaction

GPSS Blocks



ADVANCE

An ADVANCE Block delays the progress of a Transaction for a specified amount of simulated time.

ADVANCE A,B

Operands

A - The mean time increment. Required. The operand must be *Name*, *Number*, *String*, *ParenthesizedExpression, SNA* or *SNA*Parameter*.

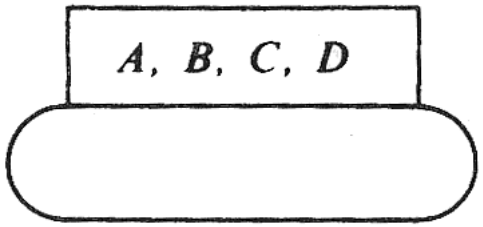
B - The time half-range or, if a function, the function modifier. Optional. The operand must be *Null*, *Name, Number*, *String*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

ADVANCE 101.6, 50.3

This example creates a Block which chooses a random number between 51.3 and 151.9, inclusively (i.e. 101.6 plus or minus 50.3), and delays the entering Transaction that amount of simulated time.

GPSS Blocks



ASSIGN

ASSIGN Blocks are used to place or modify a value in a Transaction Parameter.

ASSIGN A,B,C

Operands

A - Parameter number of the Active Transaction. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA*, or *SNA*Parameter*, followed by +, -, or *Null*.

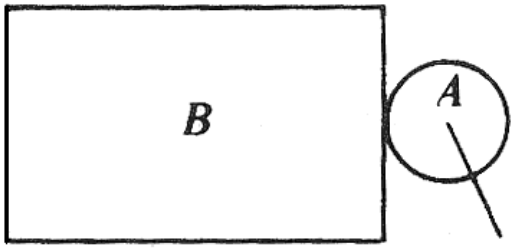
B - Value. Required. the operand must be *Name*, *Number*, *String*, *ParenthesizedExpression*, SNA, or SNA*Parameter.

C - Function number. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression*, *SNA* or *SNA*Parameter*.

Examples

ASSIGN 2000, 150.6

This is the simplest way to use the ASSIGN Block. The value 150.6 is assigned to Parameter number 2000 of the entering Transaction. If no such Parameter exists, it is created.



DEPART

A DEPART Block registers statistics which indicate a reduction in the content of a Queue Entity.

DEPART A,B

Operands

A - Queue Entity name or number. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA* or *SNA*Parameter*.

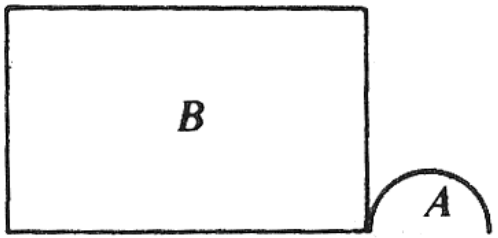
B - Number of units by which to decrease content of the Queue Entity. Default value is 1. Optional. The operand must be *Null*, *Name*, *PosInteger*, *String*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

DEPART WaitingLine

In this example the content of the Queue Entity named WaitingLine is reduced by one and the associated statistics accumulators are updated.

GPSS Blocks



ENTER

When a Transaction attempts to enter an ENTER Block, it either takes or waits for a specified number of storage units.

ENTER A,B

Operands

A - Storage Entity name or number. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA* or *SNA*Parameter*.

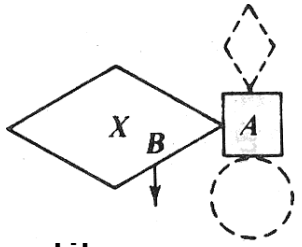
B - Number of units by which to decrease the available storage capacity. Default value is 1. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

ENTER Toolkit, 2

In this example the Active Transaction demands 2 storage units from the storage units available at the Storage Entity named Toolkit. If there are not enough storage units remaining in the Storage Entity, the Transaction comes to rest on the Delay Chain of the Storage Entity.

GPSS Blocks



GATE

A GATE Block alters Transaction flow based on the state of an entity.

GATE O A,B

Operands

O - Conditional operator. Condition required of entity to be tested for successful test. Required. The operator must be FNV, FV, I, LS, LR, M, NI, NM, NU, SE, SF, SNE, SNF, SNV, SV, or U.

A - Entity name or number to be tested. The entity type is implied by the conditional operator. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

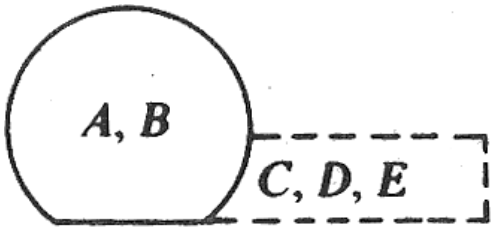
B - Destination Block number when test is unsuccessful. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, *SNA*Parameter*.

Examples

GATE SNF MotorPool

In this example of a "Refuse Mode" GATE Block, the Active Transaction enters the GATE Block if the Storage Entity named MotorPool is not full (i. e. if at least 1 unit of storage is available). If the Storage is full, the Active Transaction is blocked until 1 or more storage units become available.

GPSS Blocks



GENERATE

A GENERATE Block creates Transactions for future entry into the simulation.

GENERATE A,B,C,D,E

Operands

A - Mean inter generation time. Optional. The operand must be *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, or *DirectSNA*. You may not use Transaction Parameters.

B - Inter generation time half-range or Function Modifier. Optional. The operand must be *Null*, *Name*, *Number, String, ParenthesizedExpression*, or *DirectSNA*. You may not use Transaction Parameters.

C - Start delay time. Time increment for the first Transaction. Optional. The operand must be *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression*, or *DirectSNA*. You may not use Transaction Parameters.

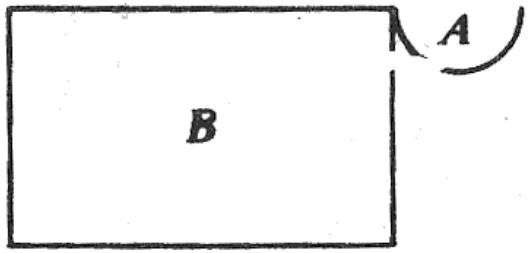
D - Creation limit. The default is no limit. Optional. The operand must be *Null*, *Name*, *PosInteger*, *String*, *ParenthesizedExpression*, or *DirectSNA*. You may not use Transaction Parameters.

E - Priority level. Optional. Zero is the default. The operand must be *Null*, *Name*, *integer*, *String*, *ParenthesizedExpression*, or *DirectSNA*. You may not use Transaction Parameters.

Example

GENERATE 0.1

This is the simplest way to use the GENERATE Block. This Block causes a priority zero Transaction to enter the simulation every tenth of a time unit.



LEAVE

A LEAVE Block increases the accessible storage units at a Storage Entity.

LEAVE A,B

Operands

A - Storage Entity name or number. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

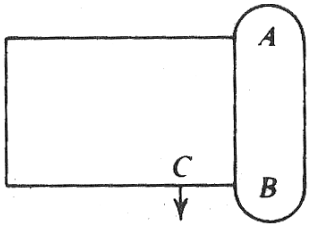
B - Number of storage units. The default is 1. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

LEAVE RepairMen,10

In this example, when a Transaction enters the LEAVE Block, the available storage units at the Storage Entity named RepairMen is increased by 10.

GPSS Blocks



LINK

A LINK Block controls the placement of the Active Transaction on the User Chain of a Userchain Entity.

LINK A,B,C

Operands

A - Userchain number. The Userchain Entity which may receive the entering Transaction. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

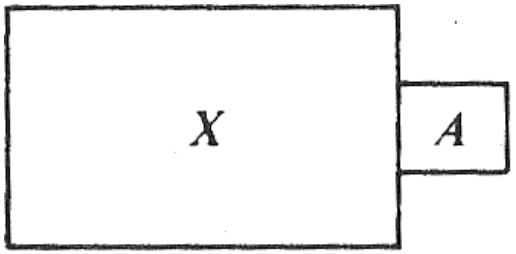
B - Chain ordering. The placement of new Transactions on the Userchain. Required. The operand must be LIFO, FIFO, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

C - Next Block location. The destination Block for Transactions which find the Link Indicator of the Userchain in the off state (reset). Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA* or *SNA*Parameter*.

Example

LINK OnHold,FIFO

In this example, the Active Transaction is placed at the end of the User Chain Entity named OnHold. It is removed from all chains except Transaction Groups and Interrupt Chains. In other words, preemptions are not cleared. The Transaction remains on the User Chain until some other Transaction enters an UNLINK Block and specifically removes it. In the present example, the Transaction is placed at the end of the User Chain named OnHold.



LOGIC

A LOGIC Block changes the state of a Logicswitch entity.

LOGIC O A

Operands

O - Logic operator. Required. The operator must be S, R, or I.

A - Logicswitch Entity number. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

LOGIC S PowerSwitch

In this example, the Logicswitch Entity named PowerSwitch is left in the true or "set" state.

GPSS Blocks

MARK

A MARK Block places an absolute clock time stamp into the Active Transaction or into its Parameter.

MARK A

Operand

A - Parameter number. Parameter to receive value of system clock. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

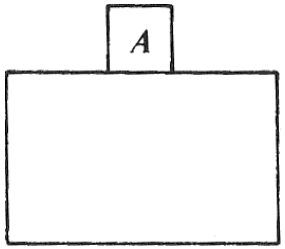
Examples

MARK Beginning

In this example, when a Transaction enters the MARK Block, its Transaction Parameter named Beginning is given a value equal to the value of the absolute system clock, AC1.

MARK

In this example, when a Transaction enters the MARK Block, its Mark Time is set equal to the value of the absolute system clock.



PRIORITY

A PRIORITY Block sets the priority of the Active Transaction.

PRIORITY A,B

Operands

A - New priority value. Required. The operand must be *Name*, *integer*, *String*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

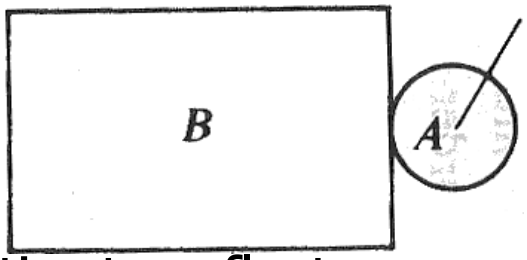
B - Buffer option. Places Active Transaction behind priority peers on CEC. Optional. The operand must be BU or *Null*.

Example

PRIORITY 10

In this example any entering Transaction is made to have a priority of 10.

GPSS Blocks



QUEUE

A QUEUE Block updates Queue Entity statistics to reflect an increase in content.

QUEUE A,B

Operands

A - Queue Entity name or number. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

B - Number of units by which to increase the content of the Queue Entity. Default value is 1. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

QUEUE WaitingLine

In this example the content of the Queue Entity named WaitingLine is increased by one and the associated statistics accumulators are updated.

GPSS Blocks

RELEASE

A RELEASE Block releases ownership of a Facility, or removes a preempted Transaction from contention for a Facility.

RELEASE A

Operand

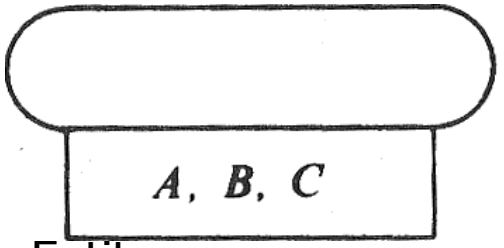
A - Facility number. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

RELEASE Teller1

In this example, when a Transaction which owns the Facility Entity named Teller1 enters the RELEASE Block, it gives up ownership to the Facility.

GPSS Blocks



SAVEVALUE

A SAVEVALUE Block changes the value of a Savevalue Entity.

SAVEVALUE A,B

Operands

A - Savevalue Entity number. Required. May be followed by + or - to indicate addition or subtraction to existing value. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*, followed by +, -, or *Null*.

B - The value to be stored, added, or subtracted. Required. The operand must be *Name, Number*, *String*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Examples

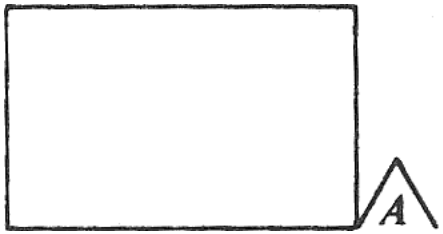
SAVEVALUE Account,99.95

In this example, the Savevalue Entity named Account takes on the value 99.95.

SAVEVALUE The_Bard,"A rose by any other name ..."

In this example, the Savevalue Entity named The_Bard is assigned a string. If the Savevalue Entity does not exist, it is created.

GPSS Blocks



SEIZE

When the Active Transaction attempts to enter a SEIZE Block, it waits for or acquires ownership of a Facility Entity.

SEIZE A

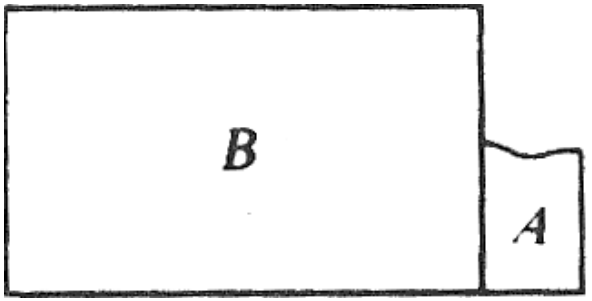
Operand

A - Facility name or number. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

SEIZE Teller1

In this example, when a Transaction attempts to enter the SEIZE Block, the state of the Facility named Teller1 is tested. If it is idle, ownership is given to the Active Transaction, which is allowed to enter the SEIZE Block and proceed to the Next Sequential Block (NSB). If the Facility is busy (owned), the Active Transaction comes to rest on the Delay Chain of the Facility.



TABULATE

A TABULATE Block triggers the collection of a data item in a Table Entity.

TABULATE A,B

Operands

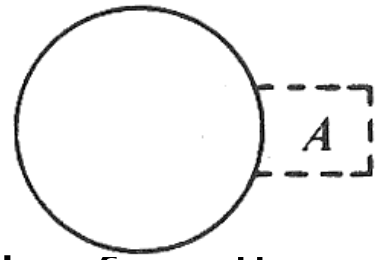
A - Table Entity name or number. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

B - Weighting factor. Optional. The operand must be *Null, Name*, *Number, ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

TABULATE Sales

When a Transaction enters this TABULATE Block, the Table Entity named Sales is found. Sales must have been defined in a TABLE Command. Then the statistics associated with the table are updated with no weighting.



TERMINATE

A TERMINATE Block removes the Active Transaction from the simulation and optionally reduces the Termination Count.

TERMINATE A

Operand

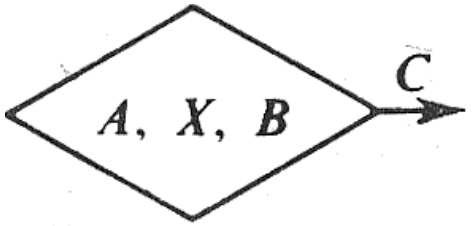
A - Termination Count decrement. Default is 0. Optional. The operand must be *Null, Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

TERMINATE 1

In this example, when a Transaction enters the TERMINATE Block it is removed from the simulation. Also, the Termination Count of the simulation, which is set by a START Command is decremented by 1.

GPSS Blocks



TEST

A TEST Block compares values, normally SNAs, and controls the destination of the Active Transaction based on the result of the comparison.

TEST O A,B,C

Operands

O - Relational operator. Relationship of Operand A to Operand B for a successful test. Required. The operator must be E, G, GE, L, LE, or NE.

A - Test value. Required. The operand must be *Name*, *Number*, *String*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

B - Reference value. Required. The operand must be *Name*, *Number*, *String*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

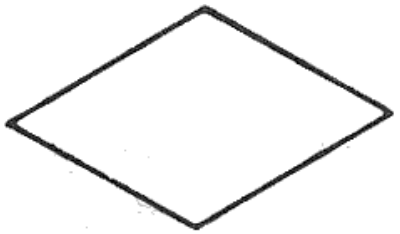
C - Destination Block number. Optional. The operand must be *Null, Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

TEST G C1, 70000

In this example of a "Refuse Mode" TEST Block, the Active Transaction enters the TEST Block if the relative system clock value is greater than 70000. Otherwise, the Transaction is blocked until the test is true.

GPSS Blocks



TRANSFER

A TRANSFER Block causes the Active Transaction to jump to a new Block location.

TRANSFER A,B,C,D

Operands

A - Transfer Block mode. Described below. Optional. The operand must be BOTH, ALL, PICK, FN, P, SBR, SIM, *fraction*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, *SNA*Parameter*, or *Null*.

B - Block number or location. Parameter name or number when in P Mode. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

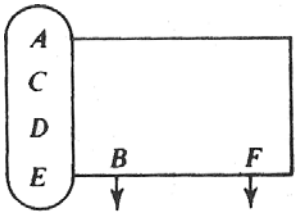
C - Block number or location. Increment value in FN or P Mode. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

D - Block number increment for ALL Mode. Default is 1. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example:

Look into the modes.

GPSS Blocks



UNLINK

An UNLINK Block removes Transactions from the User Chain of a Userchain Entity.

UNLINK O A,B,C,D,E,F

Operands

O - Relational operator. Relationship of D to E for removal to occur. These choices are explained below. Optional. The operator must be *Null*, E, G, GE, L, LE or NE.

A - User Chain number. User Chain from which one or more Transactions will be removed. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

B - Block number. The destination Block for removed Transactions. Required. The operand must be *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

C - Removal limit. The maximum number of Transactions to be removed. If not specified, ALL is used. Optional. The operand must be ALL, *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

D - Test value. The member Transaction Parameter name or number to be tested, a Boolean variable to be tested, or BACK to remove from the tail of the chain. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, *SNA*Parameter* or BACK.

E - Reference value. The value against which the D Operand is compared. Optional. The operand must be *Null*, *Name*, *Number*, *String*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*. Operand E is not used if Operand D is a Boolean Variable.

F - Block number. The alternate destination for the entering Transaction. Optional. The operand must be *Null*, *Name*, *PosInteger*, *ParenthesizedExpression, SNA*, or *SNA*Parameter*.

Example

UNLINK OnHold,Reentry,1

This is the simplest way to use the UNLINK Block. The first Transaction at the head of the Userchain Entity named OnHold, if any, is taken off the chain and is directed to the Block labeled Reentry. It is put on the CEC behind Transactions of the same priority. The Transaction entering the UNLINK Block proceeds to the Next Sequential Block.

GPSS Control Statements

CLEAR

A CLEAR Command returns the simulation to the unused state.

CLEAR A

Operand

A - ON or OFF. If the A Operand is omitted, ON is assumed. Optional. The operand must be ON, OFF or Null.

END

The ENDControl Statement has been replaced by EXIT, which can terminate a Session. END is now a keyword in the PLUS Language.

GPSS Control Statements

FUNCTION

A FUNCTION Command defines the rules for a table lookup.

There are several types of Function Entities. Each has its own rules pertaining to the table lookup. For each, the lookup table is specified in one or more Function Follower Statements. Type C Functions are a special case. They use a table lookup, followed by a linear interpolation.

The use of Function Commands to define probability distributions has been largely supplanted by the built-in distributions in the Procedure Library. This is discussed in Chapter 8. The old Function Types are still supported by GPSS World.

NAME FUNCTION A,B

Label / Operands

NAME - Entity Label this entity. Required. The field must be Name.

A - Function argument. Required. The operand must be Name, PosInteger, String, ParenthesizedExpression, SNA, or SNA*Parameter.

B - Function type (one letter) followed immediately by the number of data pairs in the Function Follower Statements. Required.

INITIAL

An INITIAL Command initializes a Matrix Entity, a Logicswitch Entity, Savevalue Entity, or an element of a Matrix Entity.

```
INITIAL    A,B
```

Operands

A - Logicswitch, Savevalue, or Matrix element specified as SNA, or the name of a Matrix Entity. Operand A must have the form of an LS, X, or MX class SNA, or a Matrix Name. Required. The operand must be Name, LSPosInteger, LS$Name, XPosInteger, X$Name, MXPosInteger(m,n) or MX$ Name(m,n). Coordinates (m,n) must be Name or PosInteger.

B - Value to be assigned, or "UNSPECIFIED" The default is 1. Optional. The operand must be Null, Number, String, Name, or UNSPECIFIED.

GPSS Control Statements

RESET

A RESET Command marks the beginning of a measurement period.

RESET

Operands

None.

GPSS Control Statements

START

A START Command begins a simulation.

```
START    A,B,C,D
```

Operands

A - Termination count. Required. The operand must be PosInteger.

B - Printout operand. NP for "no printout". Default is to print a standard report. Optional. The operand must be NP or Null.

C - Not used. Kept for compatibility with older dialects of GPSS.

D - Chain printout. 1 to include the CEC and FEC in the standard report. Optional. The operand must be Null, or PosInteger.

STORAGE

A STORAGE Command defines a Storage Entity.

NAME STORAGE A

Label / Operand

NAME - Entity Label for this entity. Required. The field must be Name.

A - Total storage capacity. Required. The operand must be PosInteger.

Example

MotorPool STORAGE 20

This Command defines a Storage Entity named MotorPool with a total capacity of 20 units.

Note: Storage name must be defined before the simulation program and name of storage must start with 3 alphabets.

GPSS Control Statements

TABLE

A TABLE Command initializes a frequency distribution table.

NAME TABLE A,B,C,D

Label / Operands

NAME - Entity Label for this entity. Required. The field must be Name. The length of a Table name is limited to 32 characters.

A - Table argument. The data item whose frequency distribution is to be tabulated. Optional. The operand must be Name, Number, String, ParenthesizedExpression, or SNA. Ignored by ANOVA, but must be specified when used by TABULATE Blocks.

B - Upper limit of first frequency class. The maximum argument which causes the first frequency class to be updated. Required. The operand must be Number or String.

C - Size of frequency classes. The difference between the upper limit and lower limit of each frequency class. Required. The operand must be Number or String.

D - Number of frequency classes. Required. The operand must be PosInteger.

Entities

Transaction entities:

GENERATE, SPLIT, TRANSFER, TERMINATE ..

Facilities entities:

SEIZE, RELEASE ..

Queue entities:

QUEUE, DEPART

Storage entities:

ENTER, LEAVE

Action Times

- An interval of time is called an ***action time***
- Represented by an integral number
- All times are in terms of same unit

- GENERATE and ADVANCE blocks employ action times
- GENERATE block controls the interval between successive arrivals of transaction
- ADVANCE is concerned with representation of the expenditure of time
- Action time may be fixed interval or a random variable
- Action time is defined by giving ***mean (A)*** and a ***function modifier (B)***
- If 'B' is zero, the action time is constant equal to the mean

Action Times

- The modifier can be specified as a ***function*** that controls the action time
- The action time is derived by multiplying the mean by the value of the function
- Function might take various parameters

- As seen already, GENERATE block begins creating transactions from zero time and continues
- The 'C' field can be used to specify an offset time as the time when the first transaction will arrive (Ref. 14)
- Transactions have priority level and they can carry items of data called parameters
- 'E' field determines the priority of the transaction

Action Times

- The parameters can exist in four formats
  - Signed integers of fullword
  - Signed integers of halfword
  - Signed integers of byte
  - Signed floating-point numbers
- If not specified, the program creates transactions with 12 halfword parameters.

Succession of Events

- The program maintains records of when each transaction in the system is due to move
- If more than one transaction due to move, transactions are processed according to their priority
- If same priority, then first-come, or first-served basis is applied
- No time is spent to enter into ADVANCE block
- If the transaction can not enter a block due to some conditions, it is monitored and;
  - Another transaction is started from that point or,
  - It enters TERMINATE block and it is removed from the simualtion or,
  - It is kept on a chain
- When processing of a transaction is finished, the program looks into other transactions at that instant

Choice of Paths

- TRANSFER block allows some other location than the next sequential location to be selected
- The choice between two blocks is referred to as next blocks A and B
- The ***selection factor*** in field A of TRANSFER block guides the selection
- It can be set to indicate one of nine choices (coming..)
- Next blocks 'A' and 'B' are placed in fields 'B' and 'C'
- If no choice, the selection factor is left blank
- Random choice can be set by using selection factor 'S'
- Probability of selecting 'A' is then '1-S' and next block 'B' is 'S'
- Setting field 'A' to BOTH, allows a transaction to select an alternate path depending upon existing conditions.

Choice of Paths

- If move is possible, transaction goes to A otherwise to B
- If both are impossible, transaction waits for the first to become possible, giving preference to A in the event of simultaneity

Facilities and Storages

- Facilities and Storages are permanent entities that operates on the transactions
- Facility is defined as an entity that can be engaged by a single transaction at a time
- Storage is defined as an entity that can be occupied by many transactions at a time up to some predefined limit
- A transaction controlling a facility can be interrupted or preempted by another transaction
- Both facilities and storages can be made unavailable if the equipment they represent breaks down and made available again if repaired
- Entities are numbered starting from 1

Facilities and Storages

Example

Type of system	Transaction	Facility	Storage
Communications	Message	Switch	Trunk
Transportation	Car	Toll booth	Road
Data processing	Record	Key punch	Computer memory

In SEIZE, RELEASE, ENTER, LEAVE blocks, field 'A' indicates which facility or storage is intended

- SEIZE block allows the transaction to engage a facility if it is available
- The RELEASE block allows the transaction to disengage the facility
- ENTER block allows a transaction to occupy space in storage
- LEAVE block allows it to give up the space

Facilities and Storages

Example

```
        GENERATE      5
        SEIZE  1
        ADVANCE       4,3
        RELEASE       1
        TRANSFER      0.1,ACC,REJ
ACC     TERMINATE     1
REJ     TERMINATE     1
```

Sample program - 2

Using facility only

Get one inspector

```
myStorage     STORAGE       3

Beg     GENERATE      5
        ENTER         myStorage
        ADVANCE       12,9
        LEAVE         myStorage
        TRANSFER      0.1,ACC,REJ
ACC     TERMINATE     1
REJ     TERMINATE     1
```

Sample program - 3

Using storage

Get three inspectors

Gathering Statistics

- Some blocks are used to gather statistics
- E.g. QUEUE, DEPART, MARK, TABULATE
- These blocks introduce two other entities, queues and tables
- When conditions are not favorable to enter a block, transactions may be kept waiting at a block. When conditions are favorable, they are allowed to move on
- QUEUE block increases and DEPART block decreases the queue numbered in field 'A'
- If field B is blank, the change is a unit change otherwise the value of field B
- The program measures average and maximum queue lengths, distribution of time spent on the queue etc.

Gathering Statistics

- To measure the length of time taken by transactions to move through the system or part of the system, MARK and TABULATE blocks are used.
- MARK block notes the time of arrival on the transaction
- TABULATE block subtracts the time noted by a MARK block from the time of arrival at TABULATE block
- The time, '***transit time***', is entered in a table whose number or name is indicated in field 'A' of TABULATE block
- If transaction entering a TABULATE block has not passed through a MARK block, the transit time is derived by using as a base the time at which the transaction was created
- Therefore, MARK block can be considered to reset the transit time to zero

Gathering Statistics

Example

```
myStorage     STORAGE       3
myTable       TABLE  M1,5,5,10

Beg     GENERATE      5
        QUEUE         1
        ENTER         myStorage
        DEPART        1
        MARK
        ADVANCE       12,9
        LEAVE         myStorage
        TABULATE      myTable
        TRANSFER      0.1,ACC,REJ

ACC     TERMINATE     1
REJ     TERMINATE     1
```

TABLE A,B,C,D

A = transit time tabulation

B = lower limit of the table

C = tabulation interval size

D = no. of intervals

Conditional Transfers

- TRANSFER block – conditional, and unconditional

```
myTable       TABLE   M1,5,5,10
Beg     GENERATE      5
        ADVANCE       2
        TRANSFER
        BOTH,,CONV1
        SEIZE  1
        ADVANCE       12,9
        RELEASE       1
TAB     TABULATE      myTable
        TERMINATE     1

CONV1   ADVANCE       2
        TRANSFER
        BOTH,,CONV2
        SEIZE  2
        ADVANCE       12,9
        RELEASE       2
        TRANSFER      ,TAB
```

Gordon – 215

Explain

```
CONV2   ADVANCE       2
        TRANSFER
        BOTH,,CONV2
        SEIZE  3
        ADVANCE       12,9
        RELEASE       3
        TRANSFER      ,TAB

CONV3   TERMINATE
```

Program Control Statements

CLEAR - wipe out statistics and transactions. Re-run starts simulation from beginning. But do not reset random number generator seeds

RESET – clears the statistics gathered, reset relative clock

START – continues with previous statistics if not wiped out

START

CLEAR

START

Given sequence of statements run the same simulation twice but the second run would use a different set of random numbers

JOB – instructs the program to wipe out all preceding model and proceed with the following problem

Program Control Statements

JOB – resets the random number seeds so each simulation finds the program in exactly the same form as does the first problem

Reports

Title

GPSS World Simulation Report - SAMPLE9.1.1

Tuesday June 6, 2000 13:20:07

The title line of the standard report is taken from the name of the Model File that produced the report. The Date and Time of the running of the model is also included.

General Information

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	11359.609	32	3	1

START TIME. The absolute system clock at the beginning of the measurement period. Utilizations and space-time products are based on the START TIME. The START TIME is set equal to the absolute system clock by a RESET or CLEAR statement.

END TIME. The absolute clock time that the termination count became 0.

BLOCKS. The number of Block entities in the simulation at the end of the simulation.

FACILITIES. The number of Facility entities in the simulation at the end of the simulation.

STORAGES. The number of Storage entities in the simulation at the end of the simulation.

Reports

Names

NAME	VALUE
ADDUP	10007.000
CHAIN1	10012.000
COLLECT	10017.000

- NAME. User assigned names used in your GPSS World model since the last Translation.

- VALUE. The numeric value assigned to the name. System assigned numbers start at 10000.

Reports

Blocks

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT-COUNT	RETRY
	1	GENERATE	61	1	0
	2	JOIN	60	0	0
	3	JOIN	60	0	0
	..	..	..	.	.
	11	LINK	51	0	0
NXTBLK	12	SEIZE	51	0	0

LABEL. Alphanumeric name of this Block if given one.

LOC. Numerical position of this Block in the model. "Location".

BLOCK TYPE. The GPSS Block name.

ENTRY COUNT. The number of Transactions to enter this Block since the last RESET or CLEAR statement or since the last Translation.

CURRENT COUNT. The number of Transactions in this Block at the end of the simulation.

RETRY. The number of Transactions waiting for a specific condition depending on the state of this Block Entity.

Reports

Refer to Chapter – 11 of 'Reference Manual'

For

Facilities
Queues
Storages
Tables and Qtables
Userchains
Transaction Groups
Numeric Groups
Logic Switches
Savevalues
Matrix Entities
The Current Events Chain
The Future Events Chain