Preamble

I found the GPSS tutorial which comes with GPSS World to be quite detailed and so hard to get an overview of the GPSS language. These notes summarise the important bits to form a fast way to get into the language.

The comments work with GPSS World (from Minuteman software), who offer a free student edition which is available for teaching purposes. Go to the minuteman web site for details and download.

Use the help files with the programme for more detailed information, tip: highlight a word and press F1 for context specific help.

Introduction

GPSS is a discrete event simulation language/package which operates not primarily by stepping through time, but by generating transactions (people to serve, jobs to process, etc) and then progresses them through various stages (or BLOCKS to use the correct term). The transactions are tracked and placed on different holding areas for processing dependent on whether they are OK to proceed now (current event chain - CEC), or need to wait for some future event or time (future event chain - FEC). When all the current events are processed, then the time is incremented and the simulation proceeds to work through the new set of CEC. You can look at the CEC and FEC to see the position of transactions using the menu: window > simulation snapshot > CEC/FEC snapshot, as the simulation is running.

A model listing of a simulation is loaded using the menu: file > open, (it has a .gps file extension) or typed into a new model window. It then needs compiling (or whatever happens) with the menu: command > create simulation, at which point you might get some pointers to problems. (You can save it in the pre-compiled state where it has the .sim extension). Next the simulation is run with a command "START n" from the command menu. This runs the model until the number n transactions have left the programme. You can interact with the model as it runs by using halt, step, etc from the command menu or if you have the buttons on the various simulation windows introduced later.

Transactions

A basic simulation needs to have transactions generated, pass through some blocks and then terminated. There can be many different types of transactions in the simulation generated at different points to represent several different things, and waiting in many different “chains” for a certain time or set of events to occur.
Normally transactions represent the flow of work or customers and are processed and then terminated with blocks:

;*****************************************************************************
GENERATE    1 ; produces one transaction every tick of simulation clock
ADVANCE      5 ; waits for 5 before proceeding to next block
TERMINATE    ; removes the transaction from the simulation
;*****************************************************************************

Note that comments (not interpreted by GPSS as commands) have a “;” in front.

A common trick in GPSS is to have a separate set of transactions which represent the flow of simulation time with the real time of the simulation collapsed into this transaction:

;*****************************************************************************
GENERATE    60 ; produces one transaction every 60 ticks of simulation
TERMINATE    1 ; removes the transaction from the simulation
;*****************************************************************************

The last TERMINATE 1 statement also increments the count to meet the number set in the start command, and so will lead towards the end of the simulation. Without the 1, as in the first TERMINATE statement, the run count is not increased and so the simulation here gets no nearer being completed.

Both of the sections of programme above can exist in the same simulation, as the first section generates one set of transactions which do not effect the total runtime, whereas the last section produces one transaction every 60 clicks (which could be a minute if the other transactions are processed in seconds) and so if a START 60 command were used, then the a run representing 1 hour would happen before stopping.

You can watch the simulation running by opening the blocks window using the menu command: window > simulation window > blocks window, either before the simulation starts or whilst it is running. You can halt or step the simulation using the buttons at the top of the window if the simulation is running.
Distributions

GPSS has many ways of producing different distributions, such as uniform probability, triangular, normal, exponential, binomial etc. These are given in the BLOCK such as:

```
GENERATE 60 ; produces one transaction every 60 ticks of simulation
ADVANCE 60 ; do something to the transaction for 60
GENERATE 5,3 ; produces one transaction with equal probability every 5 +/-3 ticks of the simulation
ADVANCE 5,3 ; do something which lasts with equal probability for 5 +/- 3 ticks of simulation
GENERATE (Exponential(1,0,6.5)) ;Create a transaction with an exponential distribution.
```

The more complicated distributions are hard to get to grips with and so try reading more about these in the help files under “distributions.”

Flow of the Programme

Like all programming languages, controlling the flow of the statements is very important. Most significant in GPSS are labels. Many blocks have a label or refer to one, i.e.: give points for transactions to be sent to, or say where the transaction should be sent.

```
;****************************************************************************
TRANSFER ,here ; send to label here (the first comma is because the command has many arguments and if the first one is empty, then the transfer happens immediately
rest of programme here ADVANCE 5 ; transactions can be sent to here and then wait 5
;****************************************************************************
```

Another example:

```
;****************************************************************************
Orders FUNCTION P$Department,L6 1,Order/2,Frame/3,Saddle/4,Handlebars/5,Wheels/6,Pedals
;****************************************************************************
```

is a function which can be called from anywhere in the programme by using the label Orders and actually sets the value P$Department of the current transaction based on the lookup table which follows. These lookup tables can be interpolated, extrapolated, etc. based on the arguments (the L6 above).

There are also commands which test and then send to labels, such as TEST and GATE:
Sometimes transactions or orders need to be split up or copied and so can be in worked on different places. For example, if different sub-assemblies are made in different sections of the factory at the same time or if the order from a fast food customer is cooked whilst they themselves go to the pay and pickup steps. The split block does this generation of copies.

SPLIT 1, cook ; generate one copy of the transaction and send to label cook

These can then be reassembled later when all the parts have been through their separate steps:

ASSEMBLE 2 ; proceed when 2 parts of the transaction are present and ; proceed as one transaction

Another statement similar to ASSEMBLE is GATHER which requires a number of transactions to be present, but does not destroy the copies allowing them to be separated out again in the future.

Named Entities

Given the mention of labels above, it can be confusing that some elements of GPSS have names which appear in the model in the same way as labels. For example:

tottime TABLE M1, 30, 30, 20 ; setup a table called tottime which holds M1: ; the time the transaction has been in the system
Tables (which are actually histograms of the distribution captured) can be viewed at the end of the simulation or interactively as it runs using the menu command: window > simulation window > table window.

SNA's

GPSS has many System Numeric Attributes (SNA's) which contain data about the simulation. We saw the SNA M1 above for the length of time the transaction was in the simulation. Others are active transaction number: XN1, clock time: AC1, and remaining termination count: TG1, look at the help system for more. You can put these variables into a window to examine them together in one place as the simulation runs by using the menu: window > simulation window > expressions window, or window > simulation window > plot window, and putting the code into the expression box and clicking view or plot depending on whether you want to list the value or plot its output.

Facilities

As transactions progress, they are likely to make use of limited facilities or staff. These are easily handled by facility entities. These can be viewed at the end of the simulation or interactively as it runs using the menu command: window > simulation window > facilities window. When the facility is in use, transactions needing it are held in the FEC or another user chain until the facility is free.

```
;****************************************************************************
SEIZE crew1 ; use a facility
ADVANCE 10 ; do something for 10 sec
RELEASE crew1 ; give up the facility
;****************************************************************************
```

The facility window and report data show the utilisation of the facilities.

Queues

Queues are, of course, an integral part of the GPSS world and are a simple way of capturing information on the transaction processing times, particularly when awaiting facilities. Data on queue entities are captured automatically and can be examined at the end of a run in the report which is automatically generated, or interactively in the queues window (do the menu command: window > simulation window > queues window). Queues have two parts a transaction being placed on the queue and then removed:

```
QUEUE order,1 ; make a queue called order
SEIZE crew1 ; use a facility
ADVANCE 10 ; do something for 10 sec
DEPART order ; record transaction leaving order queue
RELEASE crew1 ; give up the facility
```
The queues window and report shows the size of the queue and the average time that transactions were in the queue.

Information on the progress of transactions can also be captured easily using the TABLE/TABULATE pairs seen earlier above, and also with SAVEVALUE blocks.

Similar results to QUEUE can be obtained using LINK/UNLINK, which allow different types of rules to be applied to the transactions to sort the order in which they are processed, such as FIFO - first in first out, LIFO - last in first out, by priority of a certain attribute. See the help system for more information.

Storage

A similar result to the restricted resource of facility can be achieved using STORAGE. This is also useful for defining stocks or resources with limited availability and has a structure of STORAGE/ENTER/LEAVE. These set up the amount available in the start of the programme and then are used up and then possibly replaced as the simulation progresses.

```plaintext
;*****************************************************************************
tottime TABLE M1,30,30,20 ; setup table to capture total time in system
crew STORAGE 3 ; make number of crew
;*****************************************************************************
GENERATE 60 ; arrival of customer 1 per minute
ENTER crew ; use a crew
ADVANCE 19,2 ; do something
LEAVE crew ; let a crew go back into stock
TABULATE tottime ; record time in the system
TERMINATE 1 ; customer leaves and
             ; termination count increases
;*****************************************************************************
```

If the storage is not released, then it is a way of setting a finite limit for the simulation, e.g.: when the crew have all been used and not replaced, the simulation will stop.

Variables

There are many ways of dealing with different parameters - SNA's or user defined - which can be set for each of the transactions. These can be used to set attribute values or priorities and are very flexible. These look very complicated if they are not well documented in the programme notes. The help system is not very good at explaining these, but they can be recognised by the way they look something like P$name, or V$name.

Experiments: ANOVA Library

GPSS has the ability to run and re-run the simulation many times to gather distributions of the possible results under different conditions or using different
random numbers as the start point. This is managed using PLUS – a language for programmes which access and interact with a GPSS simulation, capturing the results in a MATRIX or SAVEVALUE or tables. These results can be analysed giving the relevant statistics of likelihood using the ANOVA library. See the help system for more information, but I found this confusing as it appeared the help system referred to a different version of the software than I had installed!

**Further Help**

The above represent the very basics of GPSS. To get to grips with it have a look at the tutorial which ships with the software to explore the elements in more depth. The examples in chapter two are particularly useful to show what can be achieved.

**Examples**

Here are a few simple example GPSS programmes I wrote whilst learning the basics and which I think can be used to see quickly some of the strengths of the language. They model the drive and eat simulation from "Using simulation to teach business process design and improvement," M. Seppanen and S. Kumar, Proceedings of the 2002 Winter Simulation Conference, (paper available on the web, look for it through google).

Note however, that the distributions of delays and activities are only approximations to those in the article as I am still learning the details of how to define these entities.

```
;*****************************************************************************
; simple model with facilities for personnel
; drive and eat simulation from: Seppanen and Kumar, 2002 Winter Simulation Conference
; time is seconds
;*****************************************************************************
tottime TABLE M1,30,30,20 ;setup total time in system

;*****************************************************************************
GENERATE 60 ; arrival of customer 1 per minute
QUEUE order,1 ; make a queue to take waiting to order
SEIZE crew1 ; use a crew 1 person
ADVANCE 10 ; go to order point takes 10 sec
DEPART order ; record leaving order queue
ADVANCE 48.3,15 ; place order
RELEASE crew1 ; let the person go
SPLIT 1,cook ; generate the cook order and send to kitchen
QUEUE pay,1 ; wait for the pay window
ADVANCE 10 ; go to pay window
SEIZE crew1 ; use a crew 1 person
DEPART pay ; remove from pay queue
ADVANCE 21.7,10 ; pay
RELEASE crew1 ; let the person go
ADVANCE 10 ; go to pickup window
TRANSFER .pickg ; send customer forward to pick up window

;*****************************************************************************
cook QUEUE cooking,1 ; wait for free agent 2 to do the cooking
```
SEIZE crew2; take up a crew 2 person
DEPART cooking; leave the cooking queue
ADVANCE 45,20; cook the order
RELEASE crew2; let the person go
*************************************************************
QUEUE pickup,1; wait at the pay window
pickg ASSEMBLE 2; proceed when customer and order present
SEIZE crew2; use crew 2 to collect food
DEPART pickup; leave the wait line for pickup
ADVANCE 19,2; hand over the order
RELEASE crew2; let the person go
TABULATE tottime; record time in the system
TERMINATE; customer leaves
*************************************************************
GENERATE 3600; one step is an hour in start command
TERMINATE 1

The next model is the same, but allows the number of staff to be changed easily as they are defined not as facilities, but as storage.

*************************************************************
; more complex model using storage to change easily the number of staff
; drive and eat simulation from: Seppanen and Kumar, 2002 Winter Simulation Conference
; time is seconds
*************************************************************
tottime TABLE M1,30,30,20; setup total time in system
tottime crew1 STORAGE 1; make number of crew1
tottime crew2 STORAGE 1; make number of crew2

GENERATE 60; arrival of customer 1 per minute
QUEUE order,1; make a queue to take waiting to order
ENTER crew1; use a crew 1 person
ADVANCE 10; go to order point takes 10 sec
DEPART order; record leaving order queue
ADVANCE 48.3,15; place order
LEAVE crew1; let the person go
SPLIT 1,cook; generate the cook order and send to kitchen
QUEUE pay,1; wait for the pay window
ADVANCE 10; go to pay window
ENTER crew1; use a crew 1 person
DEPART pay; remove from pay queue
ADVANCE 21.7,10; pay
LEAVE crew1; let the person go
ADVANCE 10; go to pickup window
TRANSFER ,pickw; send customer forward to pick up window
*************************************************************
cook QUEUE cooking,1; wait for free agent 2 to do the cooking
ENTER crew2; take up a crew 2 person
ADVANCE 45,20; cook the order
DEPART cooking; leave the cooking queue
LEAVE crew2; let the person go
*************************************************************
QUEUE pickup,1; wait at the pickup window
pickw ASSEMBLE 2; proceed when customer and order present
ENTER crew2 ; use crew 2 to collect food
DEPART pickup ; leave the wait line for pickup
ADVANCE 19.2 ; hand over the order
LEAVE crew2 ; let the person go
TABULATE totime ; record time in the system
TERMINATE ; customer leaves

******************************************************************************
GENERATE 3600 ; one step is an hour in start command
TERMINATE 1

© Alan Pilkington
18.11.05