

Краткое описание языка GPSS

Язык GPSS (GeneralPurposeSimulationSystem), ориентированный на процессы, разработан еще в 1961 г., но продолжает широко использоваться. Язык реализован в ряде программ **имитационного моделирования**, так, версия программы GPSS/PC в среде Windows создана в 2000 г.

Модель (программа) на языке GPSS представляет собой последовательность операторов (их называют блоками), отображающих события, происходящие в **СМО** при перемещениях **транзактов**. Поскольку в интерпретаторах GPSS реализуется **событийный метод** и в СМО может быть одновременно много транзактов, то интерпретатор будет попеременно исполнять разные фрагменты программы, имитируя продвижения транзактов в текущий момент времени до их задержки в некоторых устройствах или очередях.

Операторы (блоки) GPSS имеют следующий формат:

<метка><имя_оператора><поле_операндов> [<комментарий>]

Метка может занимать позиции, начиная со второй, имя оператора — с восьмой, поле операндов — с девятнадцатой, комментарий обязательно отделяется от поля операндов пробелом.

Поле операндов может быть пусто, иметь один или более операндов, обозначаемых ниже при описании блоков символами **A, B, C, ...**. Операндами могут быть идентификаторы устройств, накопителей, служебные слова и **стандартные числовые атрибуты** (СЧА). К СЧА относятся величины, часто встречающиеся в разных задачах. Это, например, такие операнды, как **S** — объем занятой памяти в накопителе, **F** — состояние устройства, **Q** — текущая длина очереди, **P** — параметр транзакта (каждый транзакт может иметь не более **L** параметров, где **L** зависит от интерпретатора), **V** — целочисленная переменная (вещественная и булева переменные обозначаются **FV** и **BV** соответственно), **X** — хранимая переменная (переменная, для которой автоматически подсчитывается статистика), **K** — константа, **AC1** — текущее время, **FN** — функция, **RN** — случайная величина, **RN1** — случайная величина, равномерно распределенная в диапазоне [0, 1] и др. При этом ссылки на СЧА записываются в виде **<СЧА>\$<идентификатор>**. Например, **Q\$ORD** означает очередь ORD или **FN\$COS** — ссылка на функцию COS.

Рассмотрим наиболее часто встречающиеся операторы, сопровождая знакомство с ними простыми примерами моделей.

Источники заявок обычно описываются блоком

GENERATE A, B, C, D, E

Здесь **A** и **B** служат для задания интервалов между появлениями заявок, при этом можно использовать один из следующих вариантов:

- интервал — равномерно распределенная в диапазоне **[A–B, A+B]** случайная величина;
- интервал — значение функции, указанной в **B**, умноженной на **A**;

C — задержка в выработке первого транзакта; **D** — число вырабатываемых источником заявок; **E** — приоритет заявок. Если **D** пусто, то число вырабатываемых транзактов неограничено. Например:

GENERATE 6, FN\$EXP, , 15

Этот оператор описывает источник, который вырабатывает 15 транзактов с интервалами, равными произведению числа 6 и значения функции EXP;

GENERATE 36, 12

Здесь число транзактов неограничено, интервалы между транзактами — случайные числа в диапазоне [24, 48].

Функции, на которые имеются ссылки в операторах, должны быть описаны с помощью блока следующего типа:

M FUNCTION A, B

За ним следует строка, начинающаяся с первой позиции :

X1, Y1/X2, Y2/X3, / . . . /Xn, Yn

Здесь метка **M** — идентификатор функции, **A** — аргумент функции, **B** — тип функции, **Xi** и **Yi** — координаты узловых точек функции, заданной таблично. Например:

EXP FUNCTION RN1, C12

0, 0/.2, .22/.4, .51/.5, .6/.6, .92/.7, 1.2/.8, 1.61/.9, 2.3/.95, 3/.99, 4.6/.999, 6.9/1, 1000

Это описание непрерывной (**C**) функции EXP, заданной таблично 12-ю узловыми точками, аргументом является случайная величина (**RN1**), равномерно распределенная в диапазоне [0, 1]; или:

BVB FUNCTION *4, D6

1, 2/2, 5/3, 11/4, 20/5, 18/6, 12/7, 9

Дискретная (**D**) функция BVB задана 6-ю узловыми точками, аргумент — четвертый параметр транзакта, возбудившего обращение к функции BVB.

Здесь аргумент задан с использованием косвенной адресации, признаком которой является символ *, т.е. запись *4 означает, что аргументом является величина, указанная в 4-м параметре транзакта, вызвавшего функцию (в данном примере можно было бы использовать равноценную запись *p4). В общем случае косвенная адресация выполняется путем записи операнда в виде СЧА*СЧА, например: S*X2, что означает емкость памяти, занятая в накопителе, именем которого является значение переменной X2, или Q*p5 — длина очереди с именем, записанным в параметре 5 транзакта.

Транзакты могут порождаться и оператором размножения:

SPLIT A, B, C

Новые транзакты порождаются, когда в данный блок входит некоторый транзакт. При этом создается семейство транзактов, включающее основной (вошедший в блок) транзакт и **A** его копий. Основной транзакт переходит в следующий по порядку блок, а его копии переходят в блок с меткой **B**. Для различения транзактов параметр **C** основного транзакта увеличивается на 1, а транзактов-копий — на 2, 3, 4, ... и т.д.

Обратное действие — сборка транзактов выполняется операторами:

ASSEMBLE A

GATHER A

Согласно оператору ASSEMBLE первый из вошедших в блок транзактов выйдет из него только после того, как в этот блок придут еще A-1 транзактов того же семейства. Второй оператор отличается от предыдущего тем, что из блока выходят все A транзактов.

Операторы занятия транзактом и освобождения от обслуживания устройства A:

SEIZE A

RELEASE A

Задержка в движении транзакта по СМО описывается оператором:

ADVANCE A, B

A и B имеют тот же смысл, что и в операторе **GENERATE**.

Пример 1

Обслуживание транзакта в устройстве WST продолжительностью α единиц времени, где α — равномерно распределенная в диапазоне [7,11] случайная величина, описывается следующим фрагментом программы

SEIZE WST

ADVANCE 9, 2

RELEASE WST

Аналогично описывается занятие транзактом памяти в накопителе:

ENTER A, B

Здесь помимо имени накопителя (A) указывается объем занимаемой памяти (B).

Освобождение B ячеек памяти в накопителе A выполняется оператором:

LEAVE A, B

Для накопителей в модели нужно задавать общий объем памяти, что делается в следующем описании накопителя:

M STORAGE A

Здесь: M — имя накопителя, A — объем его памяти.

Если транзакт приходит на вход занятого устройства или на вход накопителя с недостаточным объемом свободной памяти, то транзакт задерживается в очереди к этому устройству или накопителю. Слежение за состоянием устройств и очередей выполняет интерпретатор. Но если в модели требуется сослаться на длину очереди или собирать статистику по ее длине, то требуется явное указание этой очереди в модели. Делается это с помощью операторов входа в очередь и выхода из очереди:

QUEUE A, B

DEPART A, B

Согласно этим операторам очередь A увеличивается и уменьшается на B единиц соответственно, если B=1, то поле B можно оставить пустым.

Движение транзактов выполняется по маршруту, заданному последовательностью операторов в модели. Если требуется изменение

естественного порядка, то используется оператор перехода. Оператор условного перехода имеет вид:

TEST XX A,B,C

В соответствии с ним переход к оператору, помеченному меткой C, происходит, если не выполняется условие A XX B, где XX ∈ {E, NE, L, LE, G, GE}; E — равно; NE — неравно; L — меньше; LE — меньше или равно; G — больше; GE — больше или равно (XX всегда размещается в позициях 13 и 14).

Пример 2

Приходящие пользователи ожидают обслуживания, если длина очереди не более 4, иначе от обслуживания отказываются. Соответствующий фрагмент программы

```
TEST LE      Q$STR,4,LBL
           QUEUE      STR
           SEIZE      POINT
           DEPART      STR
           ADVANCE      50,16
RELEASE      POINT
...
LBL      TERMINATE      1
```

В примере 2 использован оператор выхода транзактов из СМО:

TERMINATE A

Согласно этому оператору из итогового счетчика вычитается число A. С помощью итогового счетчика задается длительность моделирования. В начале исполнения программы в счетчик заносится число, указанное в операнде A оператора

START A,B,C

Моделирование прекращается, когда содержимое счетчика будет равно или меньше нуля. Операнд C — шаг вывода статистики на печать. Если B=0 и C=0, то выполняется только стандартная печать по окончании моделирования. В стандартную печать входят собранные за время моделирования статистические данные по основным параметрам модели: средние и максимальные значения длин очередей, объемов занимаемой памяти в накопителях, времени занятого состояния устройств и др. От печати можно отказаться, указав B=NP.

Пример 3

Общая структура программы на GPSS имеет вид

SIMULATE

<описания, в том числе функций, накопителей, массивов и т.п.>

<операторы, моделирующие движение транзактов>

START A,B,C

END.

Оператор безусловного перехода записывается следующим образом:

TRANSFER ,B

Здесь В — метка оператора, к которому следует переход.

Используется ряд других разновидностей оператора **TRANSFER**.
Например:

TRANSFER P,B,C

Переход происходит к оператору с меткой, равной сумме значения параметра В транзакта и числа С.

TRANSFER FN,B,C

То же, но вместо параметра транзакта слагаемым является значение функции В.

TRANSFER PICK,B,C

Это оператор равновероятного перехода к операторам, метки которых находятся в интервале [В,С].

Важное место в СМО занимает переход по вероятности:

TRANSFER A,B,C

Здесь А — вероятность перехода к оператору с меткой С, переход к оператору с меткой В будет происходить с вероятностью $1 - A$.

Оператор перехода в циклических процедурах выглядит так:

LOOP A,B

Здесь А — номер параметра транзакта, в котором содержится число повторений (витков) цикла, В — метка оператора, с которого начинается повторяющаяся часть.

Пример 4

Заказы, поступающие в СМО в случайные моменты времени в диапазоне [20,40], выполняет сначала бригада WGR1, затем параллельно работают бригады WGR2 и WGR3, каждая над своей частью заказа. Заданы экспоненциальные законы для времен выполнения работ бригадами WGR1, WGR2 и WGR3 с интенсивностями 0,05, 0,1 и 0,125 соответственно. Моделирование нужно выполнить на временном отрезке, соответствующем выполнению 1000 заказов.

Программа:

SIMULATE

EXP FUNCTION RN1,C12

0,0/.2,.22/.4,.51/.5,.6/.6,.92/.7,1.2/.8,1.61/.9,2.3/.95,3/.99,4
.6/.999,6.9/1,1000

GENERATE 30,10

SEIZE WGR1

ADVANCE 20, FN\$EXP

RELEASE WGR1

SPLIT 1,MET1

SEIZE WGR2

ADVANCE 10, FN\$EXP

RELEASE WGR2

TRANSFER ,MET2

MET1 SEIZE WGR3

```

ADVANCE      8 , FN$EXP
RELEASE      WGR3
MET2 ASSEMBLE 2
TERMINATE    1
START        1000
END .

```

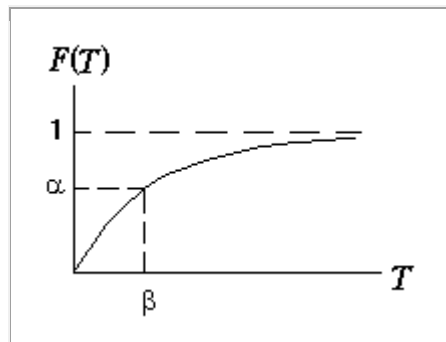


Рис. 1. Функция экспоненциального закона распределения

В этом примере использован экспоненциальный закон распределения с плотностью

$$P(t) = \lambda \exp(-\lambda t),$$

где λ — интенсивность. Функция распределения экспоненциального закона

$$F(T) = \int_0^T p(t) dt = 1 - \exp(-\lambda T).$$

Из рис. 1 ясно, что поскольку искомыми являются значения β случайной величины T , то, задавая значение α , как равномерно распределенной в диапазоне $[0, 1]$ случайной величины, по формуле

$$\beta = \frac{1}{\lambda} \ln \frac{1}{1-\alpha} \quad (1)$$

находим искомое значение. Именно в соответствии с (1) в операторах **ADVANCE** множителями были значения $\frac{1}{\lambda}$.

Приведем еще несколько операторов языка GPSS.

Оператор изменения параметров транзактов:

```

ASSIGN      A, B

```

Здесь A — номер параметра транзакта, B — присваиваемое ему значение.

В следующих операторах параметр A увеличивается (уменьшается) на значение B:

```

ASSIGN      A+, B

```

```

ASSIGN      A-, B

```

Оператор фиксации времени события

```

MARK       A

```

его выполнение заключается в запоминании значения текущего модельного времени в параметре вошедшего в оператор MARK транзакта, в поле A указывается номер параметра.

Пример 5

На вход производственной линии поступают и проходят обработку на станке TOOL1 детали типов X и Y. Далее детали типа X обрабатываются на станке TOOL2, а детали типа Y — на станке TOOL3. Интервал моделирования соответствует обработке 600 деталей (ниже у операторов **GENERATE** и **ADVANCE** значения операндов не конкретизированы):

SIMULATE

```

      GENERATE   A,B
      ASSIGN     1,LBL4
      TRANSFER   ,LBL1
      GENERATE   A,B
      ASSIGN     1,LBL2
LBL1   SEIZE     TOOL1
      ADVANCE    A,B
      RELEASE    TOOL1
      TRANSFER   P,1
LBL4   SEIZE     TOOL2
      ADVANCE    A,B
      RELEASE    TOOL2
LBL3   TERMINATE 1
LBL2   SEIZE     TOOL3
      ADVANCE    A,B
      RELEASE    TOOL3
      TRANSFER   ,LBL3
START   600
      END.
```

Расширение возможностей управления движением транзактов достигается благодаря операторам, реализующим механизм семафора:

LOGIC X A

```
      GATE XX    A,B
```

Первый оператор при $X = S$ устанавливает переключатель A в единичное состояние, при $X = R$ сбрасывает его в нулевое состояние, при $X = I$ инвертирует значение состояния. Второй оператор при $XX = LR$ и значении переключателя, указанного в A, равном 1, или при $XX = LS$ и состоянии переключателя 0 передает транзакт оператору с меткой B (или задерживает его в блоке **GATE**, если поле B пусто), а при других сочетаниях XX и значений переключателя — направляет к следующему оператору.

Вычислительный оператор присваивает переменной с номером M значение арифметического выражения A:

```
      M          VARIABLE A
```

Например в следующем операторе переменной номер 3 присваивается разность числа 216 и объема занятой памяти в накопителе MEM2:

XINIT VARIABLE K216-S\$MEM2

Знаки арифметических операций сложения, вычитания, умножения, деления +, -, #, / соответственно. В случае логических выражений имя оператора должно быть BVARIABLE, а знаками операций дизъюнкции и конъюнкции являются + и #. Если операции выполняются над числами типа real, то имя оператора FVARIABLE.

Следующий оператор присваивает хранимой переменной, указанной в A, значение, записанное в B:

SAVEVALUE A,B

Прерывание обслуживания заявки в устройстве A происходит при входе некоторой другой заявки в блок:

PREEMT A

а возобновление прерванного обслуживания заявки — при входе в блок:

RETURN A

Оператор синхронизации, реализующий механизм рандеву, имеет, например, вид:

LBL MATCH NUMB

Приходящий в него транзакт задерживается до тех пор, пока в некоторой другой части модели в сопряженный оператор не войдет транзакт того же семейства. Сопряженный оператор выглядит так:

NUMB MATCH LBL

Часто сведения о некоторых величинах, характеризующих моделируемый процесс, удобно представлять в виде гистограмм. Задание гистограммы выполняют в разделе описаний с помощью оператора:

M TABLE A,B,C,D

Здесь M — имя гистограммы; A — табулируемая величина; B — верхняя граница левого интервала гистограммы; C — ширина интервалов; D — число интервалов. Формирование гистограммы происходит с помощью оператора:

TABULATE A

Выполнение этого оператора увеличивает на единицу число попаданий в i -й интервал гистограммы, имя которой указано в A. При этом i -й интервал соответствует текущему значению переменной, являющейся аргументом для гистограммы.

Пример 6

Требуется разработать модель процессов возникновения и устранения неисправностей в некоторой технической системе, состоящей из множества однотипных блоков; в запасе имеется один исправный блок; известны статистические данные об интенсивностях возникновения отказов и длительностях таких операций, как поиск неисправностей, замена и ремонт отказавшего блока. Поиск и замену отказавшего блока производит бригада TEAM1, а ремонт замененного блока — бригада TEAM2.

SIMULATE

GENERATE A,B моделируется возникновение отказов


```

SEIZE      TEAM1
          ADVANCE      A, Впоискнеисправности
ENTER      MEM,1 получение запасного блока из резерва
ADVANCE    A, Взаменаблока
          RELEASE      TEAM1
          SEIZE        TEAM2
          ADVANCE      A, Времонт
          LEAVE        MEM,1восстановлениерезерва
          RELEASE      TEAM2
          TERMINATE    1
START      1000
          END.

```

Пример 7

Требуется разработать модель сборки изделия из 30 деталей типа A1 и 16 деталей типа A2, поступающих на сборочный участок от независимых экспоненциальных источников с интенсивностями λ , равными 0,1 и 0,04 мин⁻¹ соответственно. Длительность сборочной операции находится в пределах [12,18] мин. Промоделировать выпуск 600 изделий. Табулировать наполнение входного бункера с деталями типа A2 перед началом сборки.

SIMULATE

```

MEM1  STORAGE      30
MEM2  STORAGE      16
TAB   TABLE       MEM2,32,16,6
EXP   FUNCTION      RN1,C12
0,0/.2,.22/.4,.51/.5,.6/.6,.92/.7,1.2/.8,1.61/
.9,2.3/.95,3/.99,4.6/.999,6.9/1,1000
          GENERATE    10, FN$EXP
          ENTER       MEM1,1
          TRANSFER    ,MMM
          GENERATE    25, FN$EXP
          ENTER       MEM2,1
MMM    TEST GE       S$MEM1,30,LLL
          TEST GE     S$MEM2,16,LLL
          TABULATE    TAB
          SEIZE       MONT
          ADVANCE     15,3
          RELEASE     MONT
          TERMINATE   1
LLL    TERMINATE
          START       600

```

END .

Список литературы

1. Томашевский В., Жданова Е. Имитационное моделирование в среде GPSS. — М.: Бестселлер, 2003.
2. GPSS. — <http://www.compmodel.ru/394/>