ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

ОГЛАВЛЕНИЕ

1. КОРОТКО О ГЛАВНОМ 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 равномерно распределенная величина в диапазоне [0, 1]; или:

```
BBB FUNCTION(RN1), *4,D6
1,2/2,5/3,11/4,20/5,18/6,12/7,9
```

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

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: **Q*p5** – длина очереди с именем, записанным в парамтере 5 транзакта.

Тразакты 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**[1].

ПРИМЕРЫ в **ПРИЛОЖЕНИИ 1.**

1**A** *и* **B** *задают интервал между появлениями заявок, юзаем один из следующих вариантов:*

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