# Московский Инженерно-Физический Институт

(технический университет)

Реферат по курсу Моделирование Вычислительных Систем

Тема: «Механизм генерации транзактов в модели.

Формирование цепей текущих и будущих событий.

Основные атрибуты транзактов.

Управление движением транзактов.»

# Студент группы К8-123 Киреев Анатолий

Москва 2000

## ВВЕДЕНИЕ

Процессы функционирования различных систем могут быть представлены той или иной совокупностью систем массового обслуживания (СМО) - стохастических, динамических, дискретно-непрерывных математических моделей. Исследование характеристик таких моделей может проводиться либо аналитическими методами, либо путем имитационного моделирования. Имитационная модель отображает стохастический процесс смены дискретных состояний СМО в непрерывном времени в форме моделирующего алгоритма. При его реализации на ЭВМ производится накопление статистических данных по тем атрибутам модели, характеристики которых являются предметом исследований. По окончании моделирования накопленная статистика обрабатывается, и результаты моделирования получаются в виде выборочных распределений исследуемых величин или их выборочных моментов. Таким образом, при имитационном моделировании систем массового обслуживания речь всегда идет о статистическом имитационном моделировании.

Одним из наиболее эффективных и распространенных языков моделирования сложных дискретных систем является в настоящее время язык GPSS. Он может быть с наибольшим успехом использован для моделирования систем, формализуемых в виде систем массового обслуживания. В качестве объектов языка используются аналоги таких стандартных компонентов СМО, как заявки, обслуживающие приборы, очереди и т.п. Достаточный набор подобных компонентов позволяет конструировать сложные имитационные модели, сохраняя привычную терминологию СМО.

## Транзакты

Динамические объекты, соответствующие заявкам в системах массового обслуживания, называются в GPSS/PC транзактами. Они "создаются" и "уничтожаются" так, как это необходимо по логике модели в процессе моделирования. С каждым транзактом может быть связано произвольное число параметров, несущих в себе необходимую информацию об этом транзакте. Кроме того, транзакты могут иметь различные приоритеты. В начальный момент времени в GPSS-модели нет ни одного транзакта. В процессе моделирования симулятор генерирует транзакты в определенные моменты времени в соответствии с теми логическими потребностями, которые возникают в моделируемой системе. Подобным же образом транзакты покидают модель в определенные моменты времени в зависимости от специфики моделируемой системы. В общем случае в модели одновременно существует большое число транзактов, однако в каждый момент времени симулятор осуществляет продвижение только какого-либо одного транзакта. Если транзакт начал свое движение, он перемещается от блока к блоку по пути, предписанному блок-схемой. В тот момент, когда транзакт входит в некоторый блок, на исполнение вызывается подпрограмма симулятора, соответствующая типу этого блока, а после ее выполнения, при котором реализуется функция данного блока, транзакт "пытается" войти в следующий блок. Такое продвижение транзакта продолжается до тех пор, пока не произойдет одно из следующих возможных событий:

1. транзакт входит в блок, функцией которого является удаление транзакта из модели;
2. транзакт входит в блок, функцией которого является задержка транзакта на некоторое определенное в модели время;
3. транзакт "пытается" войти в следующий блок, однако блок "отказывается" принять его. В этом случае транзакт остается в том блоке, где находился, и позднее будет повторять свою попытку войти в следующий блок. Когда условия в модели изменятся, такая попытка может оказаться успешной, и транзакт сможет продолжить свое перемещение по блок-схеме.

Если возникло одно из описанных выше условий, обработка данного транзакта прекращается, и начинается перемещение другого транзакта. Таким образом, выполнение моделирования симулятором продолжается постоянно. Проходя через блоки модели, каждый транзакт вносит вклад в содержимое счетчиков блоков. Значения этих счетчиков доступны программисту через СЧА блоков: **W** - текущее содержимое блока и **N** – общее количество входов в блок. Каждое продвижение транзакта в модели является событием, которое должно произойти в определенный момент модельного времени. Для того чтобы поддерживать правильную временную последовательность событий, симулятор имеет таймер модельного времени, который автоматически корректируется в соответствии с логикой, предписанной моделью.

Таймер GPSS/PC имеет следующие особенности:

1. регистрируются только целые значения (все временные интервалы в модели изображаются целыми числами);
2. единица модельного времени определяется разработчиком модели, который задает все временные интервалы в одних и тех же, выбранных им единицах;
3. симулятор не анализирует состояние модели в каждый следующий момент модельного времени (отстоящий от текущего на единицу модельного времени), а продвигает таймер к моменту времени, когда происходит ближайшее следующее событие.

Значения таймера доступны программисту через системные СЧА C1 (относительное время) и AC1 (абсолютное время). Центральной задачей, выполняемой симулятором, является определение того, какой транзакт надо выбрать следующим для продвижения в модели, когда его предшественник прекратил свое продвижение. С этой целью симулятор рассматривает каждый транзакт как элемент некоторого списка. В относительно простых моделях используются лишь два основных списка: список текущих событий и список будущих событий.

Список текущих событий включает в себя те транзакты, планируемое время продвижения которых равно или меньше текущего модельного времени (к последним относятся транзакты, движение которых было заблокировано ранее). Он организуется в порядке убывания приоритетов транзактов, а в пределах каждого уровня приоритета - в порядке поступления транзактов.

Список будущих событий включает в себя транзакты, планируемое время продвижения которых больше текущего времени, т.е. события, связанные с продвижением этих транзактов, должны произойти в будущем. Этот список организуется в порядке возрастания планируемого времени продвижения транзактов.

Симулятор GPSS/PC помещает транзакты в зависимости от условий в модели в тот или иной список и переносит транзакты из списка в список, просматривает списки, выбирая следующий транзакт для обработки, корректирует таймер модельного времени после обработки всех транзактов в списке текущих событий.

## Блоки, связанные с транзактами

С транзактами связаны блоки создания, уничтожения, задержки транзактов, изменения их атрибутов и создания копий транзактов.

Для создания транзактов, входящих в модель, служит блок GENERATE (генерировать), имеющий следующий формат: **имя GENERATE A,B,C,D,E**

В поле A задается среднее значение интервала времени между моментами поступления в модель двух последовательных транзактов. Если этот интервал постоянен, то поле B не используется. Если же интервал поступления является случайной величиной, то в поле B указывается модификатор среднего значения, который может быть задан в виде модификатора-интервала или модификатора-функции. Модификатор-интервал используется, когда интервал поступления транзактов является случайной величиной с равномерным законом распределения вероятностей. В этом случае в поле B может быть задан любой СЧА, кроме ссылки на функцию, а диапазон изменения интервала поступления имеет границы A-B, A+B.

Например, блок **GENERATE 100,40** создает транзакты через случайные интервалы времени, равномерно распределенные на отрезке [60;140]. Модификатор-функция используется, если закон распределения интервала поступления отличен от равномерного. В этом случае в поле B должна быть записана ссылка на функцию, описывающую этот закон, и случайный интервал поступления определяется, как целая часть произведения поля A (среднего значения) на вычисленное значение функции. В поле C задается момент поступления в модель первого транзакта. Если это поле пусто или равно 0, то момент появления первого транзакта определяется операндами A и B. Поле D задает общее число транзактов, которое должно быть создано блоком GENERATE. Если это поле пусто, то блок генерирует неограниченное число транзактов до завершения моделирования. В поле E задается приоритет, присваиваемый генерируемым транзактам. Число уровней приоритетов неограниченно, причем самый низкий приоритет - нулевой. Если поле E пусто, то генерируемые транзакты имеют нулевой приоритет. Транзакты имеют ряд стандартных числовых атрибутов. Например, СЧА с названием PR позволяет ссылаться на приоритет транзакта. СЧА с названием M1 содержит так называемое резидентное время транзакта, т.е. время, прошедшее с момента входа транзакта в модель через блок **GENERATE**. СЧА с названием XN1 содержит внутренний номер транзакта, который является уникальным и позволяет всегда отличить один транзакт от другого. В отличие от СЧА других объектов, СЧА транзактов не содержат ссылки на имя или номер транзакта. Ссылка на СЧА транзакта всегда относится к активному транзакту, т.е. транзакту, обрабатываемому в данный момент симулятором. Важными стандартными числовыми атрибутами транзактов являются значения их параметров. Любой транзакт может иметь неограниченное число параметров, содержащих те или иные числовые значения. Ссылка на этот СЧА транзактов всегда относится к активному транзакту и имеет вид Pj или Р$имя, где j и имя - номер и имя параметра соответственно. Такая ссылка возможна только в том случае, если параметр с указанным номером или именем существует, т.е. в него занесено какое-либо значение.

Для присваивания параметрам начальных значений или изменения этих значений служит блок ASSIGN, имеющий следующий формат: **имя ASSIGN A,B,C**

В поле A указывается номер или имя параметра, в который заносится значение операнда B. Если в поле A после имени (номера) параметра стоит знак + или -, то значение операнда B добавляется или вычитается из текущего содержимого параметра. В поле С может быть указано имя или номер функции-модификатора, действующей аналогично функции-модификатору в поле B блока **GENERATE**.

Например, блок **ASSIGN 5,0** записывает в параметр с номером 5 значение 0,

а блок **ASSIGN COUNT+,1** добавляет 1 к текущему значению параметра с именем COUNT.

Для записи текущего модельного времени в заданный параметр транзакта служит блок **MARK**, имеющий следующий формат:

**имя MARK A**

В поле A указывается номер или имя параметра транзакта, в который заносится текущее модельное время при входе этого транзакта в блок MARK. Содержимое этого параметра может быть позднее использовано для определения транзитного времени 0пребывания транзакта в какой-то части модели с помощью СЧА с названием MP.

Например, если на входе участка модели поместить блок **MARK MARKER,** то на выходе этого участка СЧА MP$MARKER будет содержать разность между текущим модельным временем и временем, занесенным в параметр MARKER блоком MARK. Если поле A блока MARK пусто, то текущее время заносится на место отметки времени входа транзакта в модель, используемой при определении резидентного времени транзакта с помощью СЧА M1.

Для изменения приоритета транзакта служит блок **PRIORITY**, имеющий следующий формат: **имя PRIORITY A,B**

В поле A записывается новый приоритет транзакта. В поле B может содержаться ключевое слово BU, при наличии которого транзакт, вошедший в блок, помещается в списке текущих событий после всех остальных транзактов новой приоритетной группы, и список текущих событий просматривается с начала. Использование такой возможности будет рассмотрено ниже.

Для удаления транзактов из модели служит блок **TERMINATE**, имеющий следующий формат: **имя TERMINATE A**

Значение поля A указывает, на сколько единиц уменьшается содержимое так называемого счетчика завершений при входе транзакта в данный блок TERMINATE. Если поле A не определено, то оно считается равным 0, и транзакты, проходящие через такой блок, не уменьшают содержимого счетчика завершений.

Начальное значение счетчика завершений устанавливается управляющим оператором **START**, предназначенным для запуска прогона модели. Поле A этого оператора содержит начальное значение счетчика завершений. Прогон модели заканчивается, когда содержимое счетчика завершений обращается в 0. Таким образом, в модели должен быть хотя бы один блок TERMINATE с непустым полем A, иначе процесс моделирования никогда не завершится. Текущее значение счетчика завершений доступно программисту через системный СЧА TG1.

Участок блок-схемы модели, связанный с парой блоков **GENERATE-ТERMINATE**, называется сегментом. Простые модели могут состоять из одного сегмента, в сложных моделях может быть несколько сегментов.

Например, простейший сегмент модели, состоящий всего из двух блоков **GENERATE** и **TERMINATE,** в совокупности с управляющим оператором **START** моделирует процесс создания случайного потока транзактов, поступающих в модель со средним интервалом в 100 единиц модельного времени, и уничтожения этих транзактов

**GENERATE 100,40**

**TERMINATE 1**

**START 1000**

Если необходимо управлять продолжительностью прогона по модельному времени, то в модели используется специальный сегмент, называемый сегментом таймера.

**GENERATE 100,40**

**TERMINATE**

**GENERATE 100000**

**TERMINATE 1**

**START 1**

Например, в модели из двух сегментов, приведенной на рис. 2, первый (основной) сегмент выполняет те же функции, что и в предыдущем примере. Заметим, однако, что полеA блока **TERMINATE** в первом сегменте пусто, т.е. уничтожаемые транзакты не уменьшают содержимого счетчика завершений. Во втором сегменте блок **GENERATE** создаст первый транзакт в момент модельного времени, равный 100000. Но этот транзакт окажется и последним в данном сегменте, так как, войдя в блок **TERMINATE**, он обратит в 0 содержимое счетчика завершений, установленное оператором **START** равным 1. Таким образом, в этой модели гарантируется завершение прогона в определенный момент модельного времени, а точное количество транзактов, прошедших через модель, непредсказуемо. В приведенных примерах транзакты, входящие в модель через блок **GENERATE**, в тот же момент модельного времени уничтожались в блоке **TERMINATE**. В моделях систем массового обслуживания заявки обслуживаются приборами (каналами) СМО в течение некоторого промежутка времени прежде, чем покинуть СМО. Для моделирования такого обслуживания, т.е. для задержки транзактов на определенный отрезок модельного времени, служит блок **ADVANCE** (задержать), имеющий следующий формат:

**имя ADVANCE A,B**

Операнды в полях A и B имеют тот же смысл, что и в соответствующих полях блока GENERATE. Следует отметить, что транзакты, входящие в блок **ADVANCE**, переводятся из списка текущих событий в список будущих событий, а по истечении вычисленного времени задержки возвращаются назад, в список текущих событий, и их продвижение по блок-схеме продолжается. Если вычисленное время задержки равно 0, то транзакт в тот же момент модельного времени переходит в следующий блок, оставаясь в списке текущих событий. В приведенном ниже сегменте транзакты, поступающие в модель из блока **GENERATE** через случайные интервалы времени, имеющие равномерное распределение на отрезке [60;140], попадают в блок **ADVANCE**. Здесь определяется случайное время задержки транзакта, имеющее равномерное распределение на отрезке [30;130], и транзакт переводится в список будущих событий. По истечении времени задержки транзакт возвращается в список текущих событий и входит в блок **TERMINATE**, где уничтожается. Заметим, что в списке будущих событий, а значит и в блоке **ADVANCE** может одновременно находиться произвольное количество транзактов.

**GENERATE 100,40**

**ADVANCE 80,50**

**TERMINATE 1**

В рассмотренных выше примерах случайные интервалы времени подчинялись равномерному закону распределения вероятностей. Для получения случайных величин с другими распределениями в GPSS/PC используются вычислительные объекты: переменные и функции.

Транзакты могут входить в модель не только через блок GENERATE, но и путем создания копий уже существующих транзактов в блоке **SPLIT**, имеющем следующий формат: **имя SPLIT A,B,C**

В поле A задается число создаваемых копий исходного транзакта (родителя), входящего в блок SPLIT. После выхода из блока **SPLIT** транзакт-родитель направляется в следующий блок, а все транзакты-потомки поступают в блок, указанный в поле B. Если поле B пусто, то все копии поступают в следующий блок. Транзакт-родитель и его потомки, выходящие из блока **SPLIT**, могут быть пронумерованы в параметре, имя или номер которого указаны в поле C. Если у транзакта-родителя значение этого параметра при входе в блок SPLIT было равно k, то при выходе из блока оно станет равным k+1, а значения этого параметра у транзактов-потомков окажутся равными k+2, k+3 и т.д.

Например, блок **SPLIT 5,MET1,NUM** создает пять копий исходного транзакта и направляет их в блок с именем MET1. Транзакт-родитель и потомки нумеруются в параметре с именем NUM. Если, например, перед входом в блок значение этого параметра у транзакта-родителя было равно 0, то при выходе из блока оно станет равным 1, а у транзактов-потомков значения параметра NUM будут равны 2, 3, 4, 5 и 6.

В приведенных выше примерах транзакты, выходящие из любого блока, всегда поступали в следующий блок. В более сложных моделях возникает необходимость направления транзактов к другим блокам в зависимости от некоторых условий. Эту возможность обеспечивают блоки изменения маршрутов транзактов.

Блок **TRANSFER** служит для передачи входящих в него транзактов в блоки, отличные от следующего. Блок имеет девять режимов работы, из которых рассмотрим здесь лишь три наиболее часто используемых. В этих трех режимах блок имеет следующий формат: **имя TRANSFER A,B,C** смысл операндов в полях A, B и C зависит от режима работы блока.

В режиме безусловной передачи поля A и C пусты, а в поле B указывается имя блока, к которому безусловным образом направляется транзакт, вошедший в блок TRANSFER. Например: **TRANSFER ,FINAL**

В режиме статистической передачи операнд A определяет вероятность, с которой транзакт направляется в блок, указанный в поле C. С вероятностью 1-A транзакт направляется в блок, указанный в поле B (в следующий, если поле B пусто). Вероятность в поле A может быть задана непосредственно десятичной дробью, начинающейся с точки. Например, блок **TRANSFER .75,THIS,THAT**

 с вероятностью 0,75 направляет транзакты в блок с именем THAT, а с вероятностью 0,25 - в блок с именем THIS. Если же поле A начинается не с десятичной точки и не содержит одного из ключевых слов - признаков других режимов работы блока, то его значение рассматривается как количество тысячных долей в вероятности передачи. Например, предыдущий блок TRANSFER можно записать также в следующем виде: **TRANSFER 750,THIS,THAT**

В режиме логической передачи в поле A записывается ключевое слово BOTH. Транзакт, поступающий в блок TRANSFER, сначала пытается войти в блок, указанный в поле B (или в следующий блок, если поле B пусто), а если это не удается, т.е. блок B отказывает транзакту во входе, то в блок, указанный в поле C. Если и эта попытка неудачна, то транзакт задерживается в блоке TRANSFER до изменения условий в модели, делающего возможным вход в один из блоков B или C, причем при одновременно возникшей возможности предпочтение отдается блоку B. Например: **TRANSFER BOTH,MET1,MET2**

**Блок TEST** (проверить) служит для задержки или изменения маршрутов транзактов в зависимости от соотношения двух СЧА. Он имеет следующий формат: **имя TEST X A,B,C**. Вспомогательный **операнд X** содержит условие проверки соотношения между СЧА и может принимать следующие значения: **L**; **LE**; **E**; **NE**; **GE**; **G**. Поле A содержит первый, а поле B - второй из сравниваемых СЧА. Если проверяемое условие A X B выполняется, то блок TEST пропускает транзакт в следующий блок. Если же это условие не выполняется, то транзакт переходит к блоку, указанному в поле C, а если оно пусто, то задерживается перед блоком TEST. Например, блок **TEST LE P$TIME,C1** не впускает транзакты, у которых значение параметра с именем TIME больше текущего модельного времени. Блок **TEST L Q$LINE,5,OUT** направляет транзакты в блок с именем OUT, если текущая длина очереди LINE больше либо равна 5.

Для задержки или изменения маршрута транзактов в зависимости от состояния аппаратных объектов модели служит блок GATE, имеющий следующий формат: **имя GATE X A,B.** Вспомогательный операнд X содержит код состояния проверяемого аппаратного объекта, а в поле A указывается имя или номер этого объекта. Если проверяемый объект находится в заданном состоянии, то блок GATE пропускает транзакт к следующему блоку. Если же заданное в блоке условие не выполняется, то транзакт переходит к блоку, указанному в поле B, а если это поле пусто, то задерживается перед блоком GATE. Операнд X может принимать следующие значения: **U**; **NU**; **I**; **NI**; **SE**; **SNE**; **SF**; **SNF**; **LS**; **LR**. Например, блок **GATE SNE BUF3** отказывает во входе транзактам, поступающим в моменты, когда в МКУ с именем BUF3 все каналы обслуживания свободны. Блок **GATE LR 4,BLOK2** направляет транзакты в блок с именем BLOK2, если в момент их поступления ЛП с номером 4 включен.