**Моделирование**

1. **Понятие имитационного моделирования. Формальная модель объекта. Типовые математические схемы моделирования**

**Модель** – любое мысленное формальное физическое представление объекта окружающего мира обеспечивающее изучение определённых его св-в. **Моделирование** – это замещение одного объекта (оригинала) другим (моделью) и фиксация или изучение свойств оригинала путем исследования свойств модели. **Система** – группа объектов, объединенных какой-либо формой взаимодействия с целью выполнения опред. задач. Система определяется совокупностью параметров и характеристик. Множество параметров системы отражает ее внутреннее содержание – структуру и принципы функционирования. Характеристики системы – это ее внешние свойства, которые важны при взаимодействии с другими системами. Характеристики системы находятся в функциональной зависимости от ее параметров. **Динамические системы** – состояние с течением времени мен-ся непрерывно.

Методы моделирования: **1) аналитические** – позволяют получить характеристики системы как некоторой функции параметров ее функционирования. Аналитическая модель – система дифф ур-ий. **2) имитационные** - представление динамического поведения с-мы посредством продвижения ее от одного сост-я к др в соотв-вии с определеннымими имитационнымими правилами.

**Имитационное моделирование** – процесс конструирования моделей реальной системы и постановки экспериментов на ней с целью либо понять поведение системы, либо оценить в рамках ограничений, накладываемых критерием или совокупностью критериев, различной стратегии, обеспеч-ие функционир-ия данной системы.

Модель объекта моделирования, т. е. системы ***S***, можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих в общем случае следующие подмножества:

совокупность *входных воздействий* на систему

;

совокупность *воздействий внешней среды*

;

совокупность *внутренних (собственных) параметров* системы

;

совокупность *выходных характеристик* системы

.

Шеннон: **Критерии применения** имитационного моделирования: 1) не сущ-ет законченной математической постановки задачи 2) аналитич. методы имеются, но настолько сложны и трудоемки, что рациональнее использовать имитационное мод-е 3) кроме оценки определенных парам-ов необходимо наблюдение за ходом процесса.

**Проблемы применения имитационного мод-я**: 1) нахождение компромисса м\д сложностью и упрощением модели 2) искусственное воспроизводство случайных воздействий окружающей среды 3) оценка адекватности модели.

**Математические модели систем**

На базе мат модели происходит анализ хар-ик с-мы, при компьютерном мод-ии на основе мат модели созд-ся алгоритм программ для получ-я инф-ии о поведении с-мы - **формальное описание объекта.**

В общем случае мат модель любой динамической с-мы можно представить в виде: 

**x**- совок-сть входных воздействий, **h**- совокупность внутр-х параметров с-мы, **y**- совокупность выходных хар-ик, **t**– время, **F** - закон ф-ния. Процесс функц-я с-мы можно рассм-ать как послед-ную смену сост-ий

:

,

где z- совок начальных состояний.

Т.о, **общую мат модель** с-мы можно представить: .

При построении мат модели можно выделить осн подходы: **1)** непрерывно-детерминиров – использует в качестве мат моделей системы дифф ур-ий (сост-е с-мы изм-ся с течением времени по непрерыв-му строго опред закону: колебания маятника) **2)** дискретно-детерм – система меняет сост-я по опред закону, h – конечное число сост-ий. Реализ-ся с помощью мат аппарата теории автоматов (для упр-я быт тех) **3)** дискретно-стохастический – вероятностный автомат (для анализа надежности ИС) **4)** непрерывно-стох – с-ма мен сост-я с теч вр по случ закону (СМО) **5)** сетевой (сети Петри) **6)** агрегативный подход(универсальный).

1. **Непрерывно-стохастические модели (Q-схемы).**

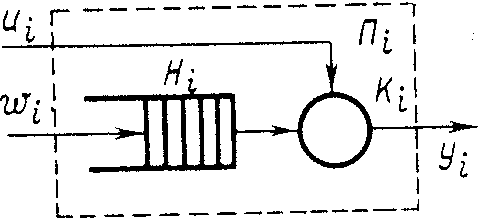
При непрерывно-стохастическом подходе в качестве типовых математических схем применяется система массового обслуживания (англ. queueing system), которые будем называть Q-схемами. Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д.

При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования. Остановимся на основных понятиях массового обслуживания, необходимых для использования Q-схем, как при аналитическом, так и при имитационном.

С-ма меняет свои сост-я с течением времени по случ закону (СМО).

Типовой мат схемой мод-я таких с-м явл **Q-схемы**. В обслуживании можно выделить 2 составляющие: ожидание обсл-я и обсл-е, а в любой СМО можно выделить элементарный прибор. В нем выделяют накопитель, некоторой емкостью; канал; потоки событий: поток заявок на обслуж-е wi, хар-щийся моментами поступления и атрибутами заявок, и поток обслуж-я ui, хар-щийся моментами начала и окончания обслуж-я заявок.



На каждый элемент прибора обслуживания *Пi*, поступают потоки событий: в накопитель *Hi* — поток заявок *wi* на канал *Ki* — поток обслуживании *ui*.

***Потоком событий*** называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. *Различают* потоки *однородных* и *неоднородных* событий.

**Поток однородных** событий (хар-ся только моментами поступления этих событий и задается послед-стью {tn }={0<= t1<= t2 ... <= tn <=...}, где tn - момент наступления n-го события. **Потоком неоднородных** событий (последовательность {tn,fn }, где fn - набор признаков события).

**Допущения**

**Ординарный поток** P>1(t, ) << P1(t, ). Для любого интервала  верно следующее:

P0(t, )+ P1(t, )+ P>1(t, )=1

**Стационарный поток** - поток, для кот вер-ть появления того или иного числа событий зав от длины интервала и не зав от того, где на оси времени взят этот интервал.

Среднее число событий, наступающих на интервале врем :

0·P0(t, )+1·P1(t, ) = P1(t, )

Тогда ср число событий орд-го потока в ед времени (**интенсивность потока**):



Для стационар потока a(t)=a= const/

Процесс функционирования прибора можно представить как процесс изм-я сост-ий его элем-ов во времени z(t). Вектор сост-ий д\прибора имеет вид , где zн – сост-е накопителя, zк – сост-е канала.

Неоднородность заявок, отражающая процесс в реальной с-ме, учитывается с помощью классов приоритетов. Различают статические (назначаются заранее) и динамические (возникают при мод-ии). **Относительный приоритет** - заявка с более высоким приоритетом, поступившая в Н, ожидает окончания обслуж-я предшествующей заявки и только после этого занимает канал. **Абсолютный приоритет** - заявка с более выс приоритетом, поступившая в Н, прерывает обслуж-е заявки с более низким приоритетом и занимает канал.

1. **Сущность метода статистического моделирования. Примеры использования.**

**Метод статистического моделирования** на ЭВМ - основной метод получения результатов с помощью имитационных моделей стохастических систем, использующий в качестве теоретической базы предельные теоремы теории вероятностей.

**Основа** - метод статистических испытаний Монте-Карло, который базируется на использовании случайных чисел, то есть возможных значений некоторой случайной величины с заданным распределением вероятностей. Статистическое моделирование представляет собой метод получения с помощью ЭВМ статистических данных о процессах, происходящих в моделируемой системе.

Сущность метода: построение для процесса функционирования исследуемой системы S некоторого моделирующего алгоритма, имитирующего поведение и взаимодействие элементов системы с учетом случайных входных воздействий и воздействий внешней среды и реализации этого алгоритма с использованием программно-технических средств ЭВМ.

2 области применения: 1) изучение стохастических систем; 2) решение детерминированных задач.

**Рез-тат статист-го мод-я** - серия частных знач-ий искомых величин или ф-ий, их статистическая обраб-ка. Если кол-во реализаций N → ∞, рез-ты устойчивы и достаточно точны.

**Пример1** (детерм). Необходимо найти площадь фигуры.





**Пример 2**. Проводится s=10 независимых выстрелов по мишени, причем вероятность попадания при одном выстреле задана и равна p. Требуется оценить вероятность того, что число попаданий в мишень будет четным.

Аналитическое решение этой задачи:



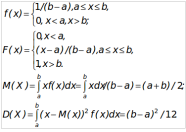
1. **Генерация случайных чисел. Генерация последовательностей псевдослучайных чисел. Требования к генератору псевдослучайных чисел. Улучшение качества последовательностей.**

Существует **три** способа генерации случайных чисел:

* **Аппаратный** - в основе лежит какой-либо физический эффект (например, шумы в электронных устройствах, случайные числа вырабатываются с помощью специального датчика. Этот способ не гарантирует качество последовательности случайных чисел непосредственно во время моделирования. С помощью этого способа нельзя получать одинаковые последовательности. Используется редко.
* **Табличные** - случайные числа оформлены в виде таблицы в оперативной памяти или на внешнем носителе. При этом способе запас чисел ограничен, вычислительные ресурсы используются неэффективно. Используется редко.
* **Программный** (алгоритмический) - случайные числа формируются с помощью специальных программ. Каждое случайное число вычисляется с помощью соответствующей программы по мере возникновения потребностей при моделировании системы на ЭВМ. Этот способ наиболее распространен.

Программная имитация случайных воздействий сводится к генерированию некоторых стандартных (базовых) процессов и к их последующему функциональному преобразованию. Чаще всего в качестве базовой последовательности используют независимые случайные величины, равномерно распределенные на интервале (0,1).

Непрерывная случайная величина имеет равномерное распределение в интервале (a,b), если ее функции плотности и распределения соответственно примут вид:



Для получения случ чисел на ЭВМ исп-ся алг-мы, поэт такие последоввт-сти, являющиеся по сути детерминированными, назыв **псевдослучайными**.

ЭВМ оперирует n-разрядными числами, поэтому вместо непрерывной совок-ти равномерных случ чисел интервала (0,1) исп-ют дискретную послед-ть 2n случ чисел того же интервала - закон распред-я такой дискретной послед-ти назыв **квазиравномерным распред-ем.**

**Требования к идеальному генератору:**

1) Последовательность должна состоять из квазиравномерно распределенных чисел. 2) Числа должны быть независимыми. 3) Последовательности случайных чисел должны быть воспроизводимыми. 4) Последовательности должны иметь неповторяющиеся числа. 5) Последовательности должны получаться с минимальными затратами вычислительных ресурсов.

Наиб применение на ЭВМ для генерации послед-тей псевдослучайных чисел находят алг-мы вида: xi+1=Ф(xi), представляющие собой рекуррентные соотношения 1го порядка.

«-» наличие коррелляции м\д числами послед-ти, иногда случ-сть отсутствует.

Прим-ся конгруэнтные процедуры генерации псевдослуч-х послед-тей.

2 целых числа конгруэнтны (сравнимы). Основано на формуле: 

где  - неотрицат целые числа.

**Методы улучшения качества:**

1)Исп-е рекуррентных формул большего порядка r:



2)Метод возмущающих ф-ций:



1. **Моделирование случайных воздействий (моделирование случайных событий). Моделирование дискретной случайной величины.**

1. **Необходимо реализовать случайное событие А**, наступающее с заданной вероятностью p. Определим А как событие, состоящее в том, что выбранное значение xi равномерно распределенной на интервале (0,1) случайной величины удовлетворяет неравенству:

xi =<p.

Тогда вероятность события А будет  Противоположное событие состоит в том, что xi >p, его вероятность равна 1-р.

2**. Рассмотрим группу событий**. Пусть А1, А2 ,..., Аs - полная группа событий, наступающих с вероятностями p1, p2 ,..., ps соответственно. Определим событие Аm как событие, состоящее в том, что выбранное значение xi случайной величины удовлетворяет неравенству

,

где 

Процедура моделирования испытаний в этом случае состоит в последовательном сравнении случайных чисел xi со значениями *lr*. Если условие выполняется, исходом испытания оказывается событие Аm .

3. **Рассмотрим независимые события А и В** с вероятностями наступления рА и рВ. Возможными исходами совместных испытаний в этом случае будут события АВ,  с вероятностями рАрВ, (1-рА)рВ, рА(1-рВ), (1-рА)(1-рВ). Для моделирования совместных испытаний можно использовать два варианта процедуры:

* Последовательное выполнение процедуры, рассмотренной в п.1.
* Определение одного из исходов АВ,  по жребию с соответствующими вероятностями, т.е. процедура, рассмотренная в п.2.

Первый вариант потребует двух чисел xi и двух сравнений. При втором варианте можно обойтись одним числом xi , но сравнений может потребоваться больше. С точки зрения удобства построения моделирующего алгоритма и экономии количества операций и памяти ЭВМ более предпочтителен первый вариант.

4. **События А и В являются зависимыми** и наступают с вероятностями pА и pВ . Обозначим через pА(В) условную вероятность наступления события В при условии, что событие А произошло. Алгоритм модели подобного случая может быть следующим:

[Пуск] → [Генерация] → <[xi < pA]> → (1+ or 2-) →

(1+) → [KA = KA+1] → [Генерация] → (1a+ or 1b-) →

(1a+) → [KAB=KAB+1] → [выход]

(1b-) → [KANB=KANB+1] → [выход]

(2-) → [KNA=KNA+1] → [генерация] → <[xi < pNA(B)]> → (2a+ or 2b-) →

(2a+) → [KNAB=KNAB+1] → [выход]

(2b-) → [KNANB=KNANB+1] → [выход];

**Моделир-е дискретных случ. величин:**

Дискрет. случ. вел-на Y принимает знач-я y1 ≤ y2 ≤ … ≤ yj ≤ … с вер-стями p1, p2, …, pj,… составляющими дифференциальное распределение вероятностей.

Для получения дискрет. случ. величин можно воспольз-ся методом обратных функций: если X – равномерно распред. на интервале (0,1) случ. величина, то искомую случ. величину получают при выполнении действий:

Если x1 < p1 , то Y= y1 , иначе,

Если x1 < p1 + p2, то Y= y2 , иначе, ……..

Если x1 < ∑[j=1..m]Pj; то Y= ym;

1. **Моделирование непрерывных случайных величин (метод обратных преобразований, показательный закон, треугольный закон распределения).**

Для получения НСВ (непрерывных случайных величин) с заданным законом распределения можно использовать **метод обратной ф-ии**. Если СВ (случайная величина) Y имеет плотность распределения f(y), то распределение СВ: F(y)=∫[0..y]f(y)dy; явл-ся равномерным на интервале (0,1). Чтобы получить число, принадлежащее последоват-ти СЧ (случайных чисел) {yi}, имеющих ф-ю плотности f(y), надо разрешить относительно **yi** ур-е: xi=∫[0..y]f(y)dy;

где xi - число, принадлежащее последовательности СЧ равномерно распределенных на интервале от (0,1).

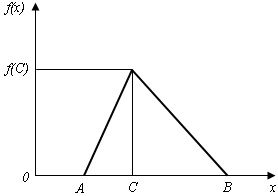
**Показательный закон распределения**: Необходимо получить случайные числа с показательным законом распределения (напр-р, интервалов времени м\ду поступлениями заявок на обслуживание): f(t)=λ∙e-λt; xi=∫[0..t](λ∙e-λt)dt; t=-(1/λ)∙ln(xi);

Этот способ получения случ-ых чисел с заданным законом распределения имеет ограниченную сферу применения, т.к. для многих законов распределения, встречающихся в практических задачах моделирования, интеграл не берется, => приходится прибегать к численным методам реш-я, что увеличивает затраты вычислительных ресурсов на получение каждого числа; Поэтому на практике пользуются приближенными способами преобраз-я СЧ, кот-е делят на:

а) универсальные способы, с пом-ю кот-х можно получать СЧ с законом распределения любого вида; б) неуниверсальные способы, пригодные д\получения случ-ых чисел с конкретным законом распределения.

**Треугольное распределение:** применяется когда о случайной величине ничего неизвестно, кроме наиболее вероятного значения и диапазона возможных значений этой случайной величины

Для получения последовательностей СЧ, подчиненных треугол-му распределению исп-ся метод обратных функций.



1. **Моделирование непрерывных случайных величин (универсальный метод (кусочная аппроксимация), нормальный закон распределения).**

На практике часто пользуются приближенными способами преобразования случайных чисел, которые можно классифицировать следующим образом:

а) универсальные способы, с помощью которых можно получать случайные числа с законом распределения любого вида;

б) неуниверсальные способы, пригодные для получения случайных чисел с конкретным законом распределения.

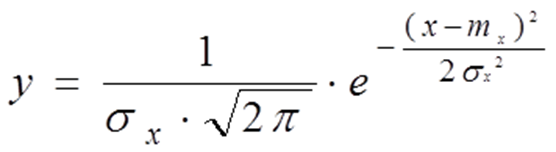
К приближенным универсальным способам получения случайных чисел относятся методы кусочной аппроксимации и усечения.

**Метод кусочной аппроксимации**. Пусть требуется получить последовательность случайных чисел {*xj*} с функцией плотности *fη(x)*, значения которой лежат в интервале (a,b). Дискретизируем заданную функцию - разобъем интервал (a,b) на k интервалов, и будем считать *fη(y)* на каждом интервале постоянной. hi - высота i-ого столбца показывает насколько вероятно выпадение значения в соответствующем интервале. Далее воспользуемся методом моделирования дискретной случайной величины.

**Метод усечения**. Функцию заключают в прямоугольник. На ось X подают случайное равномерно распределенное число r1. На ось Y подают случайное равномерно распределенное число r2. Если точка в пересечении этих двух координат лежит ниже кривой плотности вероятности, то событие X произошло, иначе нет.



**Моделирование нормально распределенных случайных величин.** Нормальный закон распределения встречается в природе весьма часто, поэтому для него разработаны отдельные эффективные методы моделирования. Формула распределения вероятности по нормальному закону имеет вид:



,где mx - математическое ожидание

- среднеквадратичное отклонение

Чтобы получить нормальное число существуют следующие методы.

1. Табличный.

Для этого нормальное число можно взять из справочника в таблице функции Лапласа и получить случайное число по методу взятия обратной функции.

|  |  |  |
| --- | --- | --- |
| X | F |  |
| 0,00 | 0,5 |  |
| ... | ... |  |
|  |  |  |

F - интегральная функция Лапласа.

1. ЦПТ (используя центральную предельную теорему).
2. Метод Мюллера



В современных системах имитационного моделирования обычно используются метод Мюллера.

1. **Управление модельным временем (принцип Δt и принцип δz)**

При создании модели важным явл-ся реализация 2-х функций: 1) корректировка временной координаты состояния сист-ы ("продвижение" времени, организация "часов"); 2) обеспечение согласованности различных блоков и событий в сист-е (синхронизация во времени, координация с др. блоками). Т.о., функционирование модели должно проте­кать в искусственном (не в реальном и не в машинном) времени, обеспечивая появление событий в требуемом логикой работы ис­следуемой системы порядке и с надлежащими временными интерва­лами между ними.

1) **принцип Δt (квантования времени).** Состояние с-мы хар-ся вектором сост-ий Z(t). Организуем счетчик систем-го времени, который в начальный момент показ время t0. Прибавим интервал врем Δt, тогда счетчик будет показывать t1= t0 +Δt. Вычислим знач-я Z(t0 +Δt), затем перейдем к моменту врем t2= t1 +Δt и т.д. Если шаг Δt достаточно мал, то таким путем можно получать приближенные знач-я Z.

**2) принцип δz.** При рассмотрении процессов функц-ния некоторых сист-м можно обнаружить, что для них характерны 2 типа состоя­ний: 1) особые, присущие процессу функционирования системы то­лько в некоторые моменты времени 2) не особые, в кот-ых процесс находится всё остальное время. Особые состояния хар-ны еще и тем, что функции сост-ий Z(t) в эти моменты времени изм-ся скач­ком, а между особыми сост-ми изменение координат Z(t) происходит плавно и непрерывно или не происходит вообще. Для описанного типа сист-м м\б построены моделирующие алг-мы по «принципу особых состо­яний». Обозначим скачкообразное изм-ие сост-я z как δz. Характеристики процесса функционирования таких сист-м оцениваются по инф-ии об особых сост-ях, а неособые состояния при моделировании не рассм-ся.

«Принцип δz*»* дает возможность для ряда систем существенно уменьшить затраты машинного времени на реализа­цию моделирующих алгоритмов по сравнению с «принципом Δt». Логика построения моделирующего алгоритма, реализующего «принцип δz», отличается от рассмотренной для «принципа Δt» только тем, что включает в себя процедуру определения момента времени tδ, соответствующего следующему особому состоянию си­стемы*.* Для исследования процесса функционирования больших систем рационально использование комбинированного принципа построения моделирующих алгоритмов, сочетающего в себе пре­имущества каждого из рассмотренных принципов.

У каждого из этих методов есть свои преимущества с точки зрения адекватного отражения реальных событий в системе и за­трат машинных ресурсов на моделирование. При использовании «принципа δz» события обрабатываются последовательно и время смещается каждый раз вперед до начала следующего события. В модели, построенной по «принципу Δt», обработка событий про­исходит по группам, пакетам или множествам событий.

Для выбора принципа построения машинной модели необходимо знать: цель и назначение модели; требуемую точность результатов моделирования; затраты машин­ного времени при использовании того или иного принципа; не­обходимый объем машинной памяти для реализации модели, построенной по принципу Δt и δz; трудоемкость программирования и отладки.

1. **Механизм протяжки модельного времени.**

1 **Время реальной системы** – время в которой живет и функционирует реальная система.

2 **Модельное время** – искусственное время в котором живет модель, или время которое явл-ся имитационным временем реальной сист-ы.

3 **Реальное время** – время в кот-ом живет исследователь и компьютер; время необходимое для моделирования.

Обычно в моделирующих системах выделяют два основных списка событий: cписок текущих событий (СТС) и список будущих событий (СБС). Каждое событие ассоциируется с динамическим объектом. В список текущих событий входят все события, запланированные на текущий момент модельного времени. Программа управления моделированием просматривает в первую очередь этот список и пытается переместить по модели те динамические объекты, для которых выполнены условия. Если в этом списке таких динамических объектов нет, процесс управления моделированием обращается к другому списку - списку будущих событий. Она переносит все события, которые запланированы на ближайший момент модельного времени, из этого списка в список текущих событий и повторяет его просмотр.

При моделировании модель­ное время может меняться быстрее или медленнее, чем в реальной сис­теме. Это зависит от степени детализации модели и сложности описа­ния изучаемого процесса. В любом случае модельное время изменяется при выполнении некоторых событий, а события в системе моделирова­ния возникают в результате перемещения динамических объектов. Причиной изме­нения модельного времени может послужить явная задержка динамического объекта на некоторый отрезок модельного времени.

Упрощенная схема протяжки модельного времени.

1- Начало моделирования;

2- СТС пуст?

3- Продвижение активного объекта в модели;

4- Движение активного объекта прекращено;

5- Извлечение динамического объекта из СБС с ближайшим временем активизации;

6- Изменение текущего модельного времени;

7- Поместить все объекты со временем выхода, равным тек. модельному времени, из СБС в СТС;

8- Конец моделирования?

9- Нормальное завершение моделирования.



1. **Объектно-ориентированная моделирующая система.**

В большинстве случаев с помощью имитационных моделей исследуются характеристики и поведение системы на определенном отрезке времени, поэтому для моделирующей системы необходимым является реализация **двух функций**:

1. предоставление средств для формализованного описания системных компонентов, дисциплин выполнения различных работ, для задания структуры модели, привязки объектов модели к временной и пространственной координате;
2. осуществление координации событий, определение путей прохождения динамических объектов модели, изменение состояний узлов и передачу управления между блоками модели.

Существует несколько основных понятий, являющихся общими для большинства современных моделирующих систем:

1. **Граф модели.** Все процессы независимо от количества уровней структурного анализа, объединяются в виде направленного графа (многослойный иерархический).
2. **Динамические объекты.** Это объекты, моделирующие формальный запрос на какое-либо обслуживание.

Динамический объект может выполнять следующие действия:

* Порождать группы (семейства) других динамических объектов (GPSS, Split);
* Поглощать другие динамические объекты конкретного семейства;
* Захватывать ресурсы и использовать их некоторое время, а затем освобождать;
* Определять времена обслуживания, накапливать информацию о пройденном пути и иметь информацию о своем дальнейшем пути и о путях других динамических объектов.

Основные параметры динамических объектов:

* Уникальный идентификатор объекта;
* Идентификатор (номер) семейства, к которому принадлежит объект;
* Наборы различных ресурсов, которые динамический объект может захватывать и использовать какое-то время;
* Время жизни динамического объекта;
* Приоритет – неотрицательное число;
* Параметры обслуживания в каком-либо обслуживающем устройстве.

1. **Узлы графа модели** представляют собой центры обслуживания динамических объектов. В узлах динамические объекты могут задерживаться, обслуживаться, порождать семейства новых динамических объектов, уничтожать другие динамические объекты. Динамический объект всегда принадлежит одному из узлов графа модели.
2. **Событие** – факт входа или выходы из узла одного динамического объекта. Функция управления событиями отдана специально управляющей программе – координатору, автоматически внедряемому в состав модели.
3. **Ресурс** независимо от его природы в процессе моделирования может характеризоваться тремя общими параметрами: мощностью, остатком и дефицитом. **Мощность** – это макс. число ресурсных единиц. **Остаток ресурса** – число незанятых на данный момент единиц. **Дефицит ресурса** – количество единиц ресурса в суммарном запросе динамических объектов, стоящих в очереди к данному ресурсу.
4. **Пространство** – географическое, декартова плоскость. Узлы, динамические объекты и ресурсы могут быть привязаны к точкам пространства и мигрировать в нем.

Программная реализация моделирующих систем использует объектно-ориентированный способ представления процессов. Динамические объекты (транзакты), узлы, события и ресурсы являются основными объектами имитационной модели.

1. **Событийный и процессно-ориентированный подход к построению моделей.**

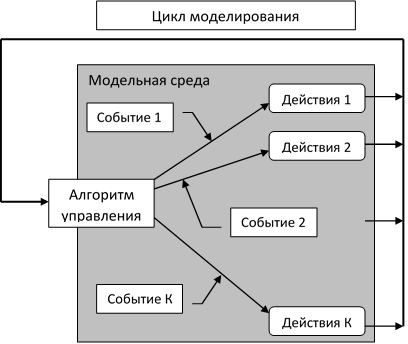
Существуют два различных подхода к построению имитационных моделей, они связаны с определенными элементами абстракций, важнейшими из которых являются понятия события и процесса.

**Событием** называется последовательность логически связанных действий, происходящих в некоторый фиксированный момент модельного времени. Появление события связано с изменением состояния модельной среды. Примером события может служить генерация сообщений, передача сообщения по каналу связи, обработка сообщения узлом и т.п.

**Событийный** подход основан на формировании потока событий. Такой поток образует сгруппиров. последоват-сти действий:



Группы событий последоват-но, происх-щих в один момент вр, условно заключены в скобки.



Несколько другая картина характерна для процессно-ориентированного подхода. Языки, реализующие процессно-ориентированный подход, имеют в своем составе блоки или операторы, позволяющие описать процесс продвижения компонентов через систему. В моделях, использующих подобную схему, описываются не события и условия их возникновения, а процесс, развивающийся в ней.

Для этого используются средства языка системы моделирования. Естественно, эти средства должны быть достаточно представительными для того, чтобы описывать широкий круг процессов. В частности, для процессного подхода необходимыми являются средства явного (или неявного) определения участков модели, на которых необходимо представление квазипараллельных процессов.

Оба подхода имеют как достоинства, так и недостатки. К достоинствам процессно-ориентированного представления моделей следует отнести компактность и наглядность. Здесь стрелками показано направление развития процессов. Событийные модели обладают большей гибкостью, но они уступают процессно-ориентированным системам в простоте и наглядности составления моделей.

1. **Инструментальные средства моделирования**

Универсальным инструментальным средством создания моделей яв-ся языки программир. общего пользования (Pascal, C/C++ и др.). Кроме того, сущ-ет множ-во специализир. средств моделир-я, позволяющих быстрее и с меньшими затратами создавать и исследовать модели. В развитии специализир. ср-в можно выделить **два направления**: **1)** Ср-ва моделир-я для анализа достаточно широкого класса систем: языки имитац. моделир-я (GPSS, SIMSCRIPT и др.), пакеты приклад. программ, использующих для моделир-я аналитич. методы, такие как MathCad, MathLab, SAS и др. Основной недостаток: их применение требует от исследователя спец. подготовки. **2)** Програм. комплексы, специализирующиеся на моделир-нии узкого круга систем одной конкретной предметной области. Недостаток: ограниченность применения таких программ одной предметной областью. Преимущ-ва: легкость их освоения специалистами в данной предмет. области, и эффективность применения вследствие узкой специализации.

В каждом цикле создания программной модели можно выделить **этапы**: **1) Формулирование проблемы**: описание исслед. проблемы, установление границ и ограничений моделируемой системы, определение целей исследования. **2) Разработка модели**: переход от реальной системы к некоторой логической схеме (абстрагирование). **3) Подготовка данных**: отбор данных, необх. для построения модели, и представление их в соответств. форме. **4) Трансляция модели**: описание модели на языке имитац. моделир-я. **5) Оценка адекватности**: повышение до приемлемого уровня степени уверенности, с кот. можно судить относительно корректности выводов о реальной системе, полученных на основании обращения к модели. **6) Планирование**: определение условий проведения машинного эксперимента с имитац. моделью. **7) Экспериментирование**: многократн. прогон имитационной модели на компьютере для получения требуемой инф-ии. **8) Анализ результатов**: изучение рез-тов для подготовки выводов и рекомендаций по решению проблемы. **9) Реализация и документирование**: реализация рекомендаций, получ. на основе имитации, и составление документации по модели и ее использованию.

1. **Система имитационного моделирования AnyLogic**

**AnyLogic**– уникальный инструмент имит. моделир-я, поддерживающ. на единой платформе абсолютно все существующ. подходы дискретно-событийного и непрерывного моделир-я. AnyLogicимеет развитый базовый язык дискретного и смешананный дискретно-непрерывного моделир-я. Библиотека AnyLogicEnterprise Library предоставляет высокоуровнев. интерфейс для быстрого создания дискретно-событийных моделей с пом. блок-схем. Графич. представление систем с пом. блок-схем широко исп-ся во многих важных сферах деятельности: производстве, логистике, системах обслуживания, бизнес-процессах, моделир-ии компьют. и телекоммуникацион. сетей.

Класс Entity яв-ся базовым классом для всех сообщений, кот. посылаются между активными объектами библиотеки Enterprise Library.

Под **заявкой** в библиотеке Enterprise Library может пониматься:

• заявка в ее обычном понимании (продукт, потребитель, пакет данных, документ),

• ресурс (оператор, машина, критическая секция),

• транспортер (поезд, автобус, корабль, автопогрузчик).

Заявки в их традиц. понимании генерир-ся объектами Source, проходят через смоделированную систему, где они обрабатыв-ся, обслужив-ся, транспортир-ся, конкурируют за право обладания ресурсами и эту систему покидают.

Ресурсы, созданные объектами Resource, могут быть заняты заявками для выполнения каких-то задач, после чего они освобожд-ся и возвращ-ся в объект Resource.

Транспортеры, также как и обычные заявки, создаются объектами Source, затем передаются в объекты Node и исп-ся для транспортировки других заявок между узлами и вдоль сегментов сети.

Объект класса Entity может исп-ся в любой из этих ролей, и при необходимости даже менять свою роль во время работы модели.

**Source** - Источник заявок. Обычно исп-ся в качестве начальной точки потока заявок, или как генератор ресурсов, транспортеров.

**Sink -** Уничтожает поступивш. заявки. Обычно исп-ся в качестве конечной точки потока заявок.

**Enter -** Пересылает заявки, переданные этому объекту либо “явно” через входной порт inputExternal, либо с помощью функции объекта take(), дальше по блок-схеме.

**SelectOutput -** Принимает заявку, и затем, в зависимости от заданного условия, передает ее на один из двух выходных портов.

**Queue -**  моделирует очередь, он хранит поступающ. заявки в опред. порядке: FIFO (заявки помещ-ся в очередь в порядке поступления), LIFO (заявки помещ-ся в порядке, обратном поступлению), RANDOM (заявки помещ-ся в произвольные места очереди) или PRIORITY (заявки помещ-ся в очередь в соответствии со значением своих полей priority).

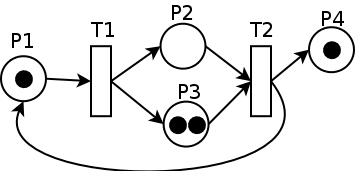
1. **Общие понятия сетей Петри**

**Сетевые модели (сети Петри)** исп-ся для анализа причинно-следственных связей в сложных системах. Аппарат теории сетей Петри позволяет описывать структуруру и взаимод-е параллельных систем и процессов. Сеть Петри (N-схема) задается 4-мя элементами: N = <B,D,I,O>,где B - конечное множество позиций; D - конечное множество переходов; I - входная функция (прямая функция инцидентности), I:BXD→{0,1}; O - выходная функция (обратная функция инцидентности), O:DXB→{0,1}.

I отображает переход dj в множество входных позиций biϵI(dj), а выходная функция О отображает переход dj в множество выходных позиций biϵD(dj). Для каждого djϵD можно определить множество входных позиций перехода I(dj) и выходных позиций перехода O(dj) какI(dj)={biϵB|I(bi,dj)=1}, O(dj)={biϵB|I(dj,bi)=1}.

Аналогично, для каждого biϵB можно определить множество входных переходов позиции I(bi) и выходных переходов позицииO(bi):I(bi)={djϵD|I(dj,bi)=1}, O(bi)={djϵD|O(bi,dj)=1}.

Графически сеть Петри изображается в виде двудольного ориентированного мультиграфа, представляющего собой совокупность позиций и переходов.

 **Пример сети Петри**. Белыми кружками обозначены позиции, полосками — переходы, чёрными кружками — метки.

Для представления динамических св-в объекта **вводится ф-я маркировки** (разметки) M:B🡪{0,1,2,…}. **Маркировка** - присвоение абстрактных объектов, называемых метками (фишками), позициям: NM = <B,D,I,O,M>.

Функционирование сети Петри отражается путем перехода от разметки к разметке. M0 - начальная разметка. **Смена разметок** - срабатывание одного из переходов сети. Необходимое условие срабатывания перехода dj: biϵI(dj) {M(bi)≥1}, где M(bi) – разметка позиции bi.

Срабатывание перехода dj изменяет разметку сети M(b) на M’(b) по следующему правилу: M’(b)=M(b)-I(dj)+O(dj), т.е. переход dj изымает по одной метке из каждой своей входной позиции и добавляет по одной метке в каждую из выходных позиций.

Сети Петри представляют удобный математич. аппарат для моделир-я параллел. технологич. процессов с разделяемыми ресурсами. **Преимуществом** сетей Петри явл-ся легкость построения иерархич. конструкций, что позволяет сначала исследовать отдельные подсистемы, а затем всю систему в целом. Модели, построенные на основе сетей П., предназначены для анализа с помощью имитации на компьютере. Такие модели довольно легко реализуются программно даже с помощью универс. языков программир-я.