

# Моделирование (конспект лекций)

Лектор: Рудаков Игорь Владимирович

Редактор Архипов Дмитрий Романович

МГТУ им. Н. Э. Баумана, Москва, осенний семестр, 2019

*Данный файл был скомпилирован 22 декабря 2019 г., 19:25*

## Лекция №1

### 1 Философские аспекты моделирования

Почему моделирование?

- дешевле;
- моделировать можно любую фантастическую идею.

**Определение.** *Адекватность модели – "приближенность" к реальному миру. 20% факторов определяют 80% функционирования объекта*

**Определение.** *Объект – все то, на что направлена человеческая деятельность*

**Определение.** *Методика – упорядоченная совокупность методов*

**Определение.** *Гипотезы – предсказания основанные на небольшом количестве опытных данных, наблюдений, догадок*

Быстрая и полная проверка гипотез должна обеспечиваться моделированием

**Определение.** *Аналогия – суждения о частном сходстве и различии объектов*

Гипотезы и аналогии отображающие реальный объективно существующий мир, должны обладать наглядностью или сводиться к удобным для исследования логическим схемам.

Такие логические схемы, упрощающие рассуждения и логические построения, или позволяющие проводить эксперимент уточняющий природу явления называют моделью.

**Определение.** *Модель – объект-заместитель объекта-оригинала обеспечивающий изучение некоторых свойств оригинала.*

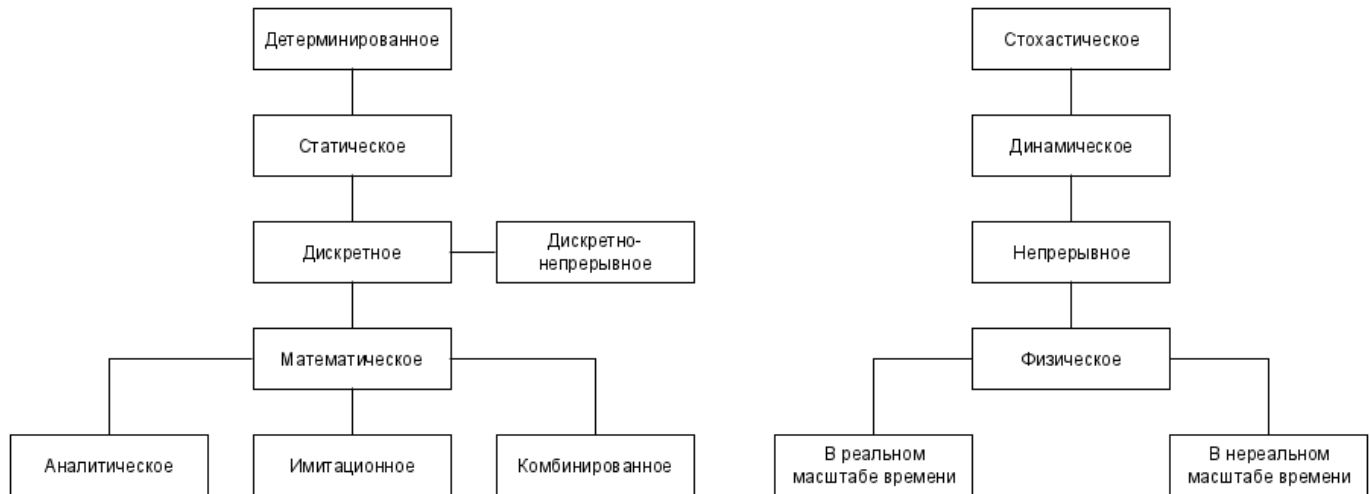
**Определение.** *Моделирование – замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-модели.*

Главный вопрос – можем ли мы познать объект с помощью методов моделирования?

<формализм сетей Петри ??? для моделирования ПО на наличие тупика>

## 2 Виды моделирования

В качестве признака классификации берем характер изучаемых процессов. Все виды моделирования будем делить на детерминированное и стохастическое.



Детерминированное отображает детерминированные процессы, т.е. такие, в которых предполагается отсутствие всяких случайных процессов или воздействий.

Стохастическое отображает вероятностные процессы и события.

Статическое служит для описания объекта в какой-либо момент времени.

Динамическое отображает поведение объекта во времени.

Дискретное описывает в дискретные моменты времени, непрерывное – в непрерывные.

Под математическим моделированием будем понимать процесс установления данному реальному объекту некоторого математического объекта, называемой математической моделью и исследовании этого объекта. Вид математической модели зависит как от природы рассматриваемого объекта так и от цели моделирования. Математическая модель описывает реальный объект с некоторой степенью приближения.

Кроме указанных в схеме так же существуют:

- информационное моделирование;
- ситуационное моделирование;
- структурное моделирование.

Кроме математического (мысленного), существует:

- наглядное моделирование
- символическое
  - языковое;
  - знаковое.

Физическое можно также делить на:

- натурное;
- комплексные испытания;
- производственный эксперимент.

Для аналитического моделирования характерно то, что в процессе функционирования системы процессы записываются в виде функциональных соотношений (алгебраических, интегро-дифференциальных, конечно-разностных и т.д.) или логических условий. Отобразена связь вход-выход.

Аналитическая модель может быть исследована 3-я способами:

- аналитическим – стремятся получить в общем виде зависимости от искомых характеристик;
- численным – если уравнение в общем виде решить нельзя, оно может получать решения уравнений для конкретных начальных данных;
- качественным – не имея решения, можно найти некоторые свойства решения, например оценить его устойчивость.

При имитационном моделировании реализующий модель алгоритм воспроизводит процесс функционирования системы во времени причем имитируются элементы явления, состава процесса, с сохранением их логических структур и последовательностей протекания во времени, что позволяет по исходным данным получить сведения о состоянии процесса в определенный момент времени, дающие возможность оценить характеристики системы.

Основное преимущество по сравнению с аналитическим методом – возможность решения более сложных задач.

Имитационная модель позволяет достаточно просто учитывать такие факторы как:

- наличие дискретных и непрерывных элементов;
- нелинейные характеристики;
- многочисленные случайные воздействия.

Имитационная модель часто подразумевается тем же самым что и программная.

Результаты выдачи имитационной модели должны быть статистически обработаны из-за большого числа вероятностных параметров лежащих в основной модели.

Целесообразно в качестве метода компьютерной имитационной модели использовать метод стохастического моделирования – метод Монте-Карло. Это численный метод применяемый для моделирования случайных величин и функций, вероятностные характеристики которых совпадают с решением аналогичных задач.

Комбинированный метод позволяет объединить достоинства разных видов. Самое важное – декомпозиция процесса функционирования на подпроцессы так, чтобы описать часть из них аналитически, а часть – имитационно.

Виды имитационного моделирования

1. агентное

Используется для исследования децентрализованных систем, динамика функционирования которых выделяется не глобальными правилами и их законами, а когда эти глобальные правила являются результатами индивидуальной активности членов группы

Цель – получить представление об общем поведении системы исходя из предположений об индивидуальном поведении отдельных активных агентов

Агент – некоторая сущность обладающая активностью, некоторой активностью поведения. Может принимать решение в соответствии с некоторым набором правил взаимодействия с окружением, а также самостоятельно изменяться.

Пример: транспортные потоки

2. дискретно-событийное подход, предлагающий абстрагироваться от непрерывной природы событий и рассматривать только основные события моделируемой системы. Наиболее развито

Пример: сеть: есть главные события пород

3. системная динамика парадигма где для исследования системы строятся графические диаграммы причинных связей и глобальных влияний одних событий на другие во времени, выявление причинно-следственных связей объектов друг на друга

Пример: бизнес-процессы, динамика популяций и тд

- Трудности имитационного моделирования связаны с обеспечением адекватности описания системы;
- Интерпретацией результатов стохастической сходимости моделирования
- Решением проблемы разверности
- трудоемкость

Имитационная модель – динамическая поэтому перед построением модели часто полезно осуществить предварительный статистический анализ системы.

При этом определяются и специфицированные функции, выполн элементами, потоки работ и тд

Самое главное – предварительно анализировать уменьш размерность имитационной модели, тем самым повышая эффективность.

На этом этапе составляется тз можно ограничить входные данные

1. создание имитационной модели на универсальном ЯП или на специальном языке, или на объектно ориентированном языке;
2. использование при разработке модели проблемно-ориентированных систем. Как правило не требует навыков программирования но могут моделировать только узкий класс систем, при этом модель строится самой системой чаще всего в ходе диалога с пользователем;
3. использование методов ИИ и знаний при принятии решений.

## 2.1 Управление имитационным экспериментом

Нерешенной проблемой неформального этапа остается перевод поставленной задачи на язык математики. Где одна проблема – наличие сложной системы управления.

**Замечание.** Составить графическую схему в виде системной динамики

- предмет
- основная математическая абстракция

Потом иерархически выстроить в соотв с уровнями абстракции

## 2.2 Подходы имитационного моделирования с точки зрения уровней абстракции

| дискретно-событийное моделирование | агентное моделирование                     | системная динамика       |
|------------------------------------|--|--------------------------|
| заявки                             | активные объекты                           | накопители               |
| потокосовые диаграммы              | индивидуальные правила поведения           | обслуживающие аппараты   |
| сети                               | прямое и не прямое взаимодействие объектов | потоки данных            |
| низкий уровень абстракции          | динамика среды                             | правила (обратные связи) |

### 2.2.1 Дискретно-событийное моделирование

Объекты: заявки, характеризующиеся набором присущих им индивидуальных параметров; обслуживающие устройства характеризующиеся методом работы с заявками (задержка, выделение ресурса, проверка условий и т.д.)

Методы: определение глобального процесса, представляемые в модели хронологической последовательностью отдельных процессов, реализуемых блоками программы (Совокупности процессов образуют потокосовые диаграммы)

### 2.2.2 Агентное моделирование

Объекты: агенты

Методы: "децентрализация модели разбиение на индивидуальные объекты и их окружение; поведение системы – результат совокупностей деятельности объектов.

### 2.2.3 Системная динамика

Объект: система функционирует как совокупность потоков, самое важное – их структура

Методы: потоковая статификация; диаграмма причинно-следственных связей; потоковые диаграммы моделей

## Лекция №3

В теории систем имеются специальные базовые понятия:

**Определение.** Система – это множество элементов находящихся в отношениях и связях, между собой

**Определение.** Элемент – часть системы, представление о котором нецелесообразно подвергать дальнейшему членению при моделировании.

**Определение.** Сложная система – систем, характеризуемая большим числом элементов и большим числом взаимодействий элементов

Сложные системы определяются также видом взаимодействий элементов, свойствами

- целенаправлен.;
- целостности;
- членимости;
- иерархичности;
- много??

**Определение.** Подсистема – часть системы (подмножества элементов и их взаимосвязи, которые имеют свойства системы) под системой

Система по отношению к которой рассматривается система является подсистемой

**Определение.** Структура – отображение совокупности элементов системы и их взаимосвязей. От понятия "система" отличается также тем, что при описании структуры принимают во внимание описание лишь типов элементов и связей без конкретной структуры и значений их параметров.

**Определение.** Параметр – величина выражающая свойство или системы, или ее части и влияющей на систему среды.

**Определение.** *Целенаправленность* – свойство искусственной системы выражающее назначение системы, необходимое для оценки эффективности вариантов системы.

**Определение.** *Целостность* – свойство системы, характеризующее взаимодействие элементов и наличия зависимости выходных параметров от параметров элемента. При этом, большинство выходных параметров не является повторением или суммой параметров элем.

**Определение.** *Иерархичность* – свойство сложной системы, выражающее возможность и целесообразность ее иерархических описаний, т.е. представления в виде уровней, между компонентами которыми имеется отношение "целое-часть"

Моделирование сложной дискретной системы имеет 2 крайне отличные задачи:

- modelling – собственно создание модели системы
- simulation – анализ свойств систем на основе использования их модели

### 3 Технические средства математического моделирования

Классы используемых устройств

1) Цифровая вычислительная техника



процессоры отличаются набором команд

**Определение.** *Процессор* – устройство предназначенное для выполнения арифметических операций (базово сложение и сдвиг) и управления над ними

Самая большая проблема ограничение быстродействия

2) Аналоговая

В отличие от дискретной в основе аналоговой заложен принцип моделирования а не счета. При использовании в качестве модели некоторой задачи электронных цепей каждой переменной величине ставится в соответствие определенная переменная величина электрической цепи.

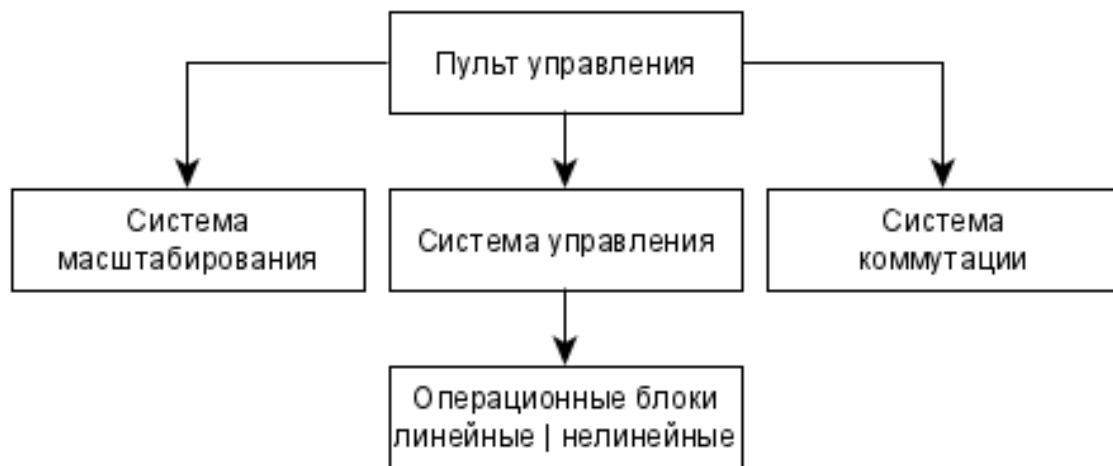
При этом основой построения такой модели является изоморфизм (подобие) исследуемой задачи и соответствия ей электронной модели.

В большинстве случаев при определении критериев подобия используются специальные приемы масштабирования (соответствие значения параметрической модели и переменных задачи)

Аналоговая техника относится к классу специальных машин выполняет узкий класс вычислений

Т.о. под АВМ будем понимать совокупность электрических элементов, организованных в систему, позволяющих изоморфно моделировать динамику исследования объекта

Структурная схема АВМ



+быстродействие

-низкая точность вычислений

-неуниверсальность

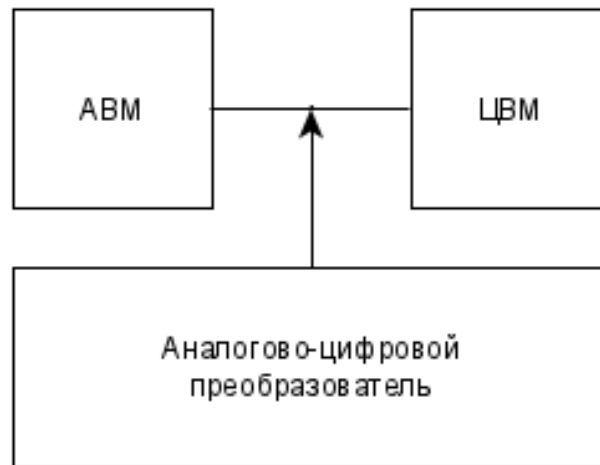
3) Гибридные вычислительные машины

Это широкий класс вычислительных систем, использующий как аналоговую так и дискретную форму представления и обработки информации

Подклассы ГВМ:

1. АВМ, использующие цифровые методы численного анализа;
2. АВМ, программируемая с помощью ЦВМ (Apache);
3. АВМ с цифровым управлением и логикой;
4. АВМ с цифровыми элементами (вольтметры и амперметры);
5. Цифровые машины с аналоговыми арифметическими устройствами
6. ЦВМ, допускающие программирование аналогового типа (например, дифференциальные анализаторы)





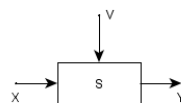
## Классификация по мощности решения интегро-дифференциальных уравнений

| тип информации                         | непрерывный   | дискретный                |
|--|---|---------------------------|
| изменение значений                     | величиной напряжения  | числовые значения         |
| базовые операции                       | арифметические операции и интегрирование                            | арифметические операции   |
| принцип вычисления                     | высокопараллель   | последовательно-параллель |
| режим реального времени                | без ограничений   | ограничен                 |
| динамическое изменение решаемой задачи | посредством системы коммутации                                      | в диалоговом режиме       |
| требования к пользователю              | профессиональные знания, методика моделирования                     | знание основ ПО ЭВМ       |
| уровень формализации задачи            | ограничен моделью решаемой задачи                                   | высокий                   |
| точность вычислений                    | ограничена  | ограничена разрядностью   |
| класс решения задач                    | динамика исследования объекта: алгебра и дифференциальные уравнения | любые                     |
| специальные функции                    | ограниченный набор  | неограниченный набор      |
| уровень миниатюрности                  | ограничен   | высокий                   |
| сфера применения                       | ограничена  | практически любая         |
| пользовательский интерфейс             | низкий уровень  | высочайший уровень        |

## Лекция №4

### 4 Основы теории моделирования

Объект  $S$ , множество входных воздействий



Модель объекта моделирования можно представить в виде множества величин, описанием процесса функционирования реальной системы и образованием в общем множестве следующего подмножества

- совокупность входных воздействий  $x_i \in X, i = \overline{1, n_x}$
- совокупность воздействий внешней среды  $v_l \in V, l = \overline{1, n_l}$
- совокупность внутренних собственных параметров системы  $h_k \in H, k = \overline{1, n_k}$
- совокупность выходных характеристик системы  $y_j \in Y, j = \overline{1, n_j}$

Эти вектора являются в общем случае элементами непересекающихся подмножеств и содержат как детерминированные так и статистические характеристики (составл)

При анализе функционирования сложной системы входные воздействия, воздействия внешней среды и внутренние параметры являются независимыми то есть экзогенными переменными, которые в векторной форме имеют вид:

$$\begin{aligned}\vec{x}(t) &= (x_1(t), x_2(t), \dots, x_{n_x}(t)) \\ \vec{v}(t) &= (v_1(t), v_2(t), \dots, v_{n_v}(t)) \\ \vec{h}(t) &= (h_1(t), h_2(t), \dots, h_{n_h}(t))\end{aligned}$$

А выходные характеристики являются зависимыми (эндогенными) переменными и имеют вид:

$$\vec{y}(t) = (y_1(t), y_2(t), \dots, y_{n_y}(t))$$

Процесс функционирования некоторой системы S описывается во времени некоторым оператором некоторого преобразования независимых переменных в зависимые

$$\vec{y}(t) = F_s(\vec{x}, \vec{v}, \vec{h}, t) \quad (1)$$

Параметр "время определяющий, мы выносим его как особо важный т.к. он порождает множество алгоритмов.

Зависимость 1 называют законом функционирования системы. В общем случае он может быть задан функцией, функционалом, логических условий, в алгоритмическом или табличном виде.

Важнейшим является понятие алгоритма функционирования под которым будем понимать метод получения выходных параметров  $y(t)$  с учетом входных воздействий, воздействий внешней среды и соотв. параметр. системы. Очевидно что один и тот же закон функционирования  $F_s$  может быть реализован различными способами, т.е. с помощью множества алгоритмов функционирования.

Соотношение 1 может быть получено через свойства системы в конкретные моменты времени которые называются самостоятельными и характеризуются вектором состояний  $Z(t) = (z_1(t), z_2(t), \dots, z_{n_m}(t))$

Если рассматривать функционирование системы как последовательную смену состояний, то они могут быть интерпретированы как координаты точки в m-мерном фазовом пространстве, причем каждой реализации соответствует некоторая фазовая траектория.

Совокупность  $Z(t)$  всех возможных состояний системы называют пространством состояний объекта моделирования. Состояние системы S в момент времени t от начала и до конца моделирования обозначается:  $Z^0 = (z_1^0, z_2^0, \dots, z_m^0)$ ,  $z_1^0 = z_1(t_0) \dots$  Состояние определяется начальным состоянием, входными воздействиями, внутренними переменными, ... которые имеют место за промежуток времени  $t = t_n - t_0$ , с помощью следующих уравнений

$$\begin{aligned} z(\vec{t}) &= \Phi(z^0, \vec{x}, \vec{v}, \vec{h}, \vec{t}) \\ y(\vec{t}) &= F(\vec{z}, t) \\ y(\vec{t}) &= F(\Phi(z^0, \vec{x}, \vec{v}, \vec{h}, t)) \end{aligned}$$

В общем случае время в модели м.б. непрерывно на интервале моделирования а может быть дискретным (число интервалов дискретизации)

Под математической моделью реальной системы понимают конечное множество элементов  $\vec{x}, \vec{v}, \vec{h}$  вместе с математическими связями между ними и характеристиками  $y(t)$

Главное – введение закона функционирования в виде 1, оно позволяет:

- выделить множество входных характеристик, а главное внутр. парам;
- показать как  $F_s$  переводит эти значения в выходные значения;
- даются ограничения на то, что делается

## 5 Типовые математические схемы

В практике моделирования на первоначальных этапах формализации объекта используют так называемые типовые математические схемы, к которым относят хорошо проработанные и проверенные математические объекты

| процесс функционирования системы    | типовая математическая схема                        | обозначения |
|-------------------------------------|---|-------------|
| непрерывно детерминированный подход | дифференциальные уравнения                          | D-схемы     |
| дискретно детерминированный подход  | конечные автоматы, мур                              | F-схемы     |
| дискретно-стохастический подход     | вероятностные автоматы                              | P-схемы     |
| непрерывно-стохастический подход    | системы массового обслуживания, системы с очередями | Q-схемы     |
| обобщенный универсальный характер   | агрегативные системы                                | A-схемы     |

В А-схеме какие угодно блоки, самое главное определить операции над ними. Переходя со слоя на слой пытаемся построить математическую схему для каждого слоя.

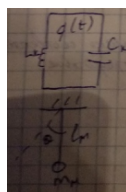
Самая важная операция для агрегативных систем композиция и декомпозиция.

Д-схемы

Если неизвестны функции многих аргументов частных производных, если одна неизвестная - обыкновенное дифференциальное уравнение.

Называется D-схема т.к. схемы такого вида отображают динамику изучения схемы.

Независимая переменная обычно время



$$\begin{aligned} \vec{h} &= (h_0, h_1, h_2) \\ h_0 &= L_k = m_M L_m^2 \end{aligned}$$

$$\begin{aligned}
 h_1 &= 0 \\
 h_2 &= \frac{1}{C_k} = m_M g l_M \\
 z(\vec{t}) &= (q(t)), z(\vec{t}) = \theta(t) \\
 L_k \frac{d^2 q(t)}{dt^2} + \frac{q(t)}{C_k} &= 0 \\
 \text{Период } \frac{2\pi}{\sqrt{LC}} \\
 m_M l_M^2 \frac{d^2 \theta(t)}{dt^2} + m_M g l_M \theta(t) &= 0 \\
 h_0 \frac{d^2 z(t)}{dt^2} + h_1 \frac{dz(t)}{dt} + h_0 z(t) &= 0
 \end{aligned}$$

Т.о. использование D-схем позволяет формализовать процесс функционирования непрерывной детерминированной системы и оценить их основные характеристики. Применяя аналитический или имитационный подход, реализованный на соответствующем языке моделирования.

## 6 Сущность компьютерного моделирования

Проведение эксперимента с моделью

**Определение.** Модель – некоторый программный комплекс описывающий формально или алгоритмически поведение системы.

Основные требования к модели:

1. полнота (возможность получения необходимого набора характеристик);
2. гибкость (модель должна предоставлять возможность воспроизводить различные ситуации при варьировании структуры, алгоритма и параметров модели)  
Блочная структура – возможность изменять блок модели без изменения структуры и т.д.
3. компьютерная реализация модели должна соответствовать имеющимся технологическим ресурсам (быстродействие, память и т.д.)

|        |
|--------|
| ТЗ     |
| Анализ |
| Синтез |

Итерационный процесс – определение постоянно уточняемого ТЗ.

Процесс моделирования включает в себя ... и является итерационным пока не будет получена модель которую можно считать адекватной в рамках решения поставленной задачи.

## 6.1 Основные этапы моделирования больших систем

### 6.1.1 1. Формируется модель и строится ее схема

- основное назначение этапа – переход от содержательного описания к его математической модели;
- исходные данные – содержательное описание

Этап – построение концептуальной модели. Последовательность действий:

1. проведение границы между системой и внешней средой;
2. исследование модели объекта с точки зрения выделения основных составляющих;
3. переход от содержательного описания к формальному описанию системы (к концептуальной модели) этот переход сводится к исключению и рассмотрению некоторых второстепенных признаков;
4. оставшиеся элементы группируются в блоки  
1 группа – блоки имитаторы воздействия внешней среды  
2 группа – блоки-модели функции системы  
3 группа – вспомогательные блоки
5. процесс функционирования системы т.к. разбивается на подпроцессы, чтобы построение модели отдельных подпроцессов было элементарно и не вызывало особых трудностей (элементарно = типовые математические модели)

### 6.1.2 2. Алгоритмизация модели и ее полная реализация математической модели

математическая модель → программный комплекс

- блочная логическая модель;
- БД
- язык

Последовательность действий:

1. разработка схемы модели алгоритма (блочная схема модели);
2. разработка диаграммы классов;
3. выбор технических средств для реализации модели;

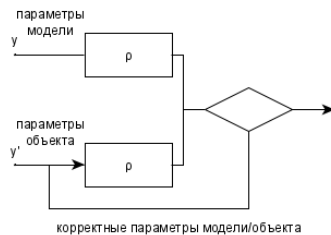
4. программирование модели;
5. отладка программы (отладка, тестирование);
6. проверка достоверности на примерах;
7. составление технической документации (спецификации, иллюстрации, технические требования)

### 6.1.3 3. Этап получения интерпретации результатов

- полное использование для проведения рабочих расчетов
- в результате расчетов подвести анализ и сделать выводы о том, что характеристики соответствуют поставленной задаче

Последовательность действий:

1. планирование компьютерного эксперимента (найти материал по)
  - машинный эксперимент
  - активный/пассивный эксперимент
  - стратегическое планирование и тактика
2. проведение рабочих расчетов



3. статистическая обработка результатов отчета
4. интерпретирование результатов модели
5. подведение итогов

3 класса ошибок:

1. ошибки формализации (неполная модель)
2. ошибки решения (некоторый/упрощенный метод)

### 3. ошибки задания параметров модели

Нарушение адекватности:

- проверка адекватности модели некоторой системы заключается в анализе ее со-размерности и равнозначности системы
- адекватность нарушается из-за идеализирования выполнения условий, особенностей режима функционирования, пренебрежение некоторыми случайными факторами
- считаем, что модель адекватна системе если вероятность того что отклонение выходного параметра не превышает предельной величины больше допустимой вероятности

Нужно проводить оценку по всем параметрам!

Проверка:

- модели элементов
- модели входных условий
- концептуальной модели
- как измерили/оценили значения выходных характеристик
- программируемые модели

Проверка приводит к изенениям модели

1. глобальные (методические) ошибки
2. локальные
3. параметрические

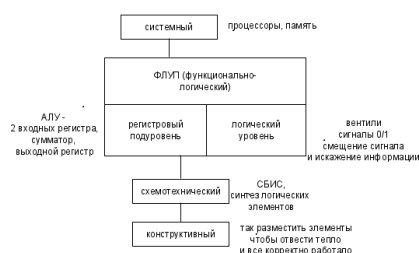
Зафиксировать область применения модели – множество условий при соблюдении которых точность результатов моделирования находится в допустимых пределах.

## Схема взаимодействия технологических этапов моделирования



## 7 Вычислительная система как объект моделирования

Уровни выделяемые среди вычислит техники При моделировании в результате



выделения уровней на каждом из них свои типовые математические модели

Структурный подход возможен или если 230 тыс знаков (сверху вниз, модульный)

Интегральные схемы моделирования снизу-вверх

При моделировании новых и модернизации существ вычислительных систем необходимо оценивать эффективность их функционирования с учетом различных вариантов структурной реализации. Эти варианты могут отличаться составом и характеристиками модулей (устройств), структурой межмодульных связей, режимами работы, а также алгоритмами управления.



**Определение.** Под вычислительной системой будем понимать комплект аппаратных и программных средств который в совокупности выполняет определенные рабочие функции

**Определение.** ОС - набор ручных и автоматизированных процедур которые позволяют группе людей (коллектив пользователей) эффективно использовать вычислительную установку.

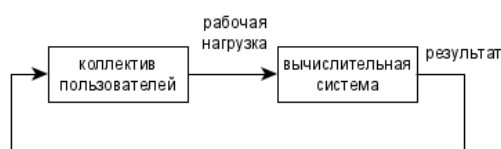
**Определение.** Коллектив пользователей - сообщество таких людей которые используют вычислительную систему для удовлетворения своих нужд

Входные данные

- программы
- данные
- команды

Создаются коллективом пользователей и называются рабочей нагрузкой

Схема вычислительной установки



Введем понятие индса производительности - некоторый общий ?описатель? который в общем случае используется для определения производительности вычислительных систем.

Различают количественные и качественные:

| количественные   |   |
|--|---|
| пропускная способность, объем информации обрабатываемый в единицу времени  |   |
| время реакции (ответа)   | время между предъявляемые системе входным данным запросам и появление соответствующей выходной информации |
| коэффициент использования оборудования, отношение времени использования указ части оборудования к длительности интервала времени |   |

| качественные           |
|------------------------|
| нравится/не нравится   |
| мощность набора команд |

Концептуальная модель включает в себя сведения о входных и конструктивных параметрах системы, ее структура, особенности работы каждого элемента характера взаимодействия между ресурсами системы. Сюда же включен характер реализуемых прикладных задач самое главное – это цель моделирования системы.

Типовая математическая схема – система массового обслуживания

Задачи:

1. определения принципов организации вычислительной системы, выбор архитектуры
2. уточнение функций и их декомпозиция
3. разработка структурной схемы

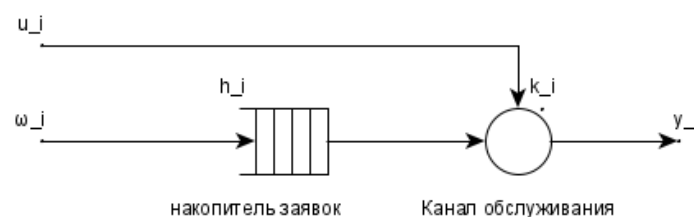
## 8 Непрерывно-стохастические системы (системы с очередями, Q-схемы)

При моделировании систем такого класса с успехом используется теория массового обслуживания

В любом элементарном акте обслуживания можно выделить 2 составляющие:

- ожидание обслуживания
- обслуживание заявки

Это можно изобразить в виде  $i$ -го прибора обслуживания



$i$ -ый прибор обслуживания состоит из

- накопителя заявок емкостью  $l_i = 0, L_i^H$
- канал обслуживания

$\omega_i$  - поток заявок в накопитель

$u_i$  - поток заявок в канал

**Определение.** Поток событий – последовательность событий проходящих одно за другим в случайные моменты времени.

Поток событий называется однородным если он характеризует только моментами наступления этих событий. Время от 0 до окончания моделирования

Поток называют неоднородным если он задается 2-мя компонентами: вызывающим моментом и набором признаков  $\{t_n, f_n\}$

Если интервалы времени независимы между собой и являются случайной величиной то поток будет с ограниченным действием

Поток называется ординарным если вероятность того что на малый интервал времени  $\Delta t$ , примыкающий к моменту времени  $t$  пренебрежительно мала по сравнению с вероятностью того что попадет ровно одно событие.

Поток называется стационарным если вероятность события

Для ординарного потока среднее число сообщений поступающих за  $\Delta t \rightarrow 0$  равно  $p_1(t, \Delta t)$

Среднее число сообщений поступивших за интервал времени  $\Delta t$  в единицу времени равно  $\lim_{\Delta t \rightarrow 0} \frac{p_1(t, \Delta t)}{\Delta t} = \lambda(t)$  – интенсивность ординарного потока

Для стационарного потока  $\lambda(t) = \lambda = const$

В обслуживающем устройстве 2 потока:

- появление заявок на входе канала
- поток обслуживания

$y_i$  включает и обработанные заявки и те которые обработать нельзя/не вышло

Процесс функционирования  $i$ -го прибора можно представить как процесс изменения состояния его элементов

$$\vec{Z}_i = (Z^H, Z_i^k)$$

В практике моделирования элементарные Q-схемы обычно объединены, при этом если каналы различных устройств соединены параллельно то имеет место многоканальное обслуж, если параллельно многофазное

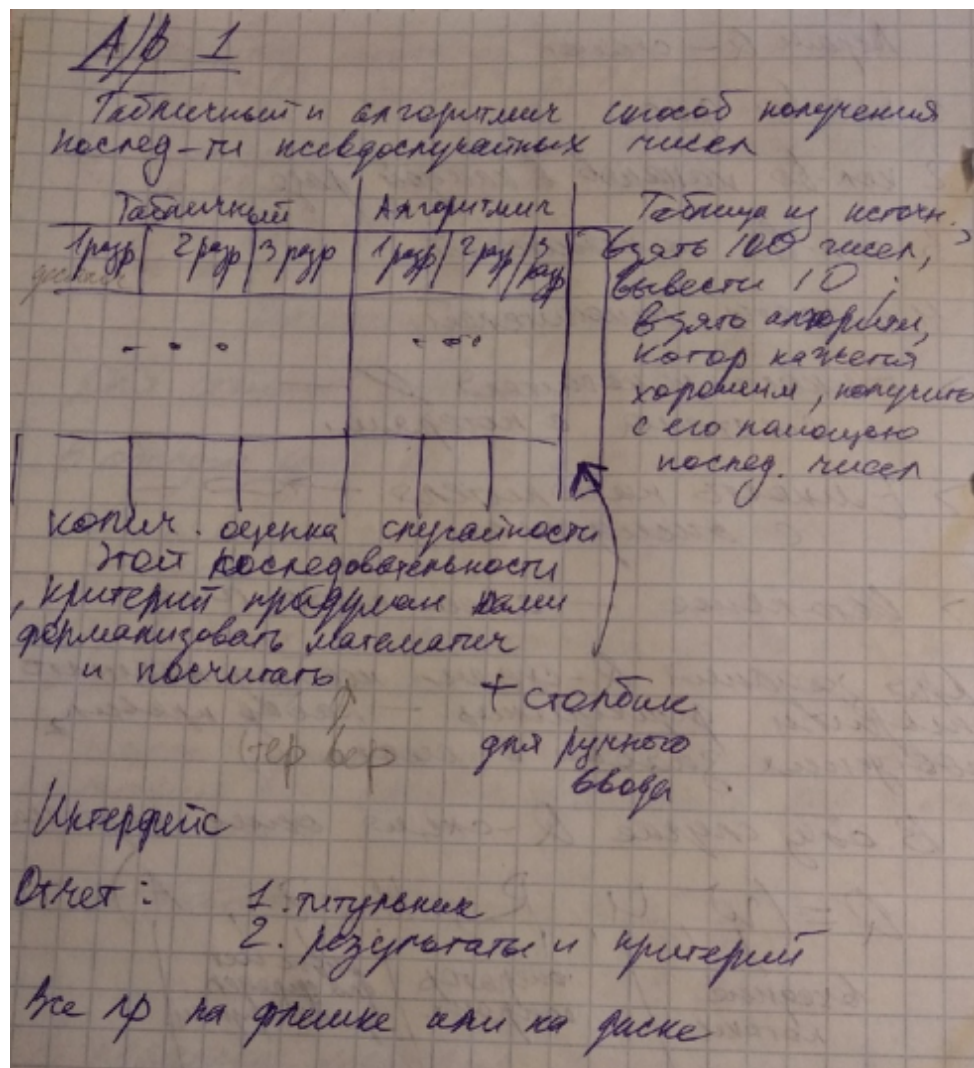
Для соединения необходимо использовать некоторый оператор сопряжения

Параметры Q-схемы

1. количество фаз
2. количество каналов в каждой фазе
3. количество накопителей
4. емкости накопителей
  - емкость накопителя 0 – система с потерями
  - емкость накопителя  $\rightarrow \inf$  – с ожиданием
  - остальное – смешанного типа

Для задания Q-схемы необходимо описать алгоритм функционирования – набор правил поведения заявок в системе

В общем случае Q-схема описывается кортежем  $Q=(w,u,R,H,Z,A)$ , где  $w$  – входные данные,  $u$  – поток обслуживания,  $R$  – оператор сопряжения,  $H$  – множество внутренних параметров,  $Z$  – множество состояний для определений закона функционирования,  $A$  – алгоритм функционирования.



Для получения соотношений связ. характеристики, описывается функционирование Q-схем введем некоторые допущения

- входных пот.
- функций распределения
- длит. обслуж. запросов
- дисциплин. обслуживание

Для математической формализации функционирования устройств процесс в котором развивается в форме случайного процесса может быть с успехом применен аппарат из теории вероятностей (теории массового обслуживания) для так называемых марковских случайных процессов.

Случайный процесс протекающий в некоторой системе S, называют марковским если для каждого момента времени вероятность любого состояния в будущем зависит только от его состояния в настоящем.

Примера нет, т.е. всегда нужна предыстория

Решив уравнение Колмогорова, определить вероятность нахождения системы в этом состоянии.

В марковских случайных процессах будущее развитие зависит только от настоящего и не зависит от предыстории.

Уравнение Колмогорова в общем виде  $F = (P'(t), P(t), \Lambda) = 0$

$\Lambda$  - коэффициенты

Для стационарного состояния  $p = (P(t), \Lambda) = 0$

$p = P(\Lambda)$

Тогда выходная характеристика (базисная/базовая модель)

$Y = P(\Lambda)$

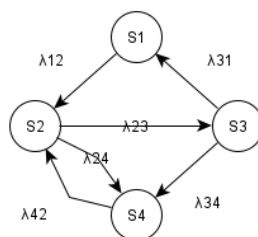
В общем виде  $Y = Y(X, V, H)$

Тогда необходимо  $= (X, V, H)$  – интерфейсная модель

Математическая модель строится как совокупность базисной и интерфейсной модели что позволяет использовать одни и те же базисные модели для решения задач проектирования осуществляет настройку ее на соотв. конкретную задачу, меняя интерфейсную модель.

Математическая модель должна обеспечивать вычисление времени реакции на запрос и производительность системы (в Q-схемах)

Пример:



1. выделить состояния

2. "взвесить" граф (граф формализует состояние системы)

$\lambda_{ij}$  – плотность вероятности

Найдем  $P_1$ , т.е. вероятность того что в момент времени  $t$  система будет в состоянии  $S_1$

Придадим  $t$  некоторое малое приращение  $\Delta$ , и найдем вероятность что в момент времени ...

$t + \Delta t$  система будет в состоянии  $S_1$

Тогда или

- система не перешла в  $S_2$
- система перешла из  $S_3, S_1$

Вероятность того что система была в состоянии  $S_1$  и за время  $\Delta t$  не вышла из этого состояния найдем как  $P_1(t) * \text{на усл. вероятность того что будучи в состоянии } S_1$

система за  $\delta t$  не перейдет в состояние  $S_2$ , эта вероятность с точностью до бесконечно малых высших порядков  $-1 - \lambda_{31} >$

$p_1(t)(1 - \lambda_{12}\Delta t) + p_3(t)\lambda_{31}\Delta t = p_1(t + \Delta t)$  – вероятность нахождения системы в состоянии  $S_1$

$$\lim_{\Delta t \rightarrow 0} \frac{p_1(t+\Delta t) - p_1(t)}{\Delta t} = -\lambda_{12}p_1(t) + \lambda_{31}p_3(t)$$

$$p_1'(t) = -\lambda_{12}p_1(t) + \lambda_{31}p_3(t)$$

S2

- система перешла из  $S_1$  или  $S_4$
- система осталась в  $S_2$
- система

$$p_2(t + \Delta t) = p_2(t)(1 - (\lambda_{23} + \lambda_{12})\Delta t) + p_1(t)\lambda_{23}\Delta t$$

$$\lim_{\Delta t \rightarrow 0} \frac{p_2(t+\Delta t) - p_2(t)}{\Delta t} = p_1(t)\lambda_{12} - (\lambda_{23} + \lambda_{12})p_2(t) + \lambda_{24}p_2(t) + \lambda_{42}p_4(t)$$

S3

- не перейти в  $S_1$
- не перейти в  $S_4$
- из  $S_2$  в  $S_3$

$$p_3(t + \Delta t) = p_3(t)(1 - (\lambda_{34} + \lambda_{31})\Delta t) + p_2(t)\lambda_{23}\Delta t$$

$$\lim_{\Delta t \rightarrow 0} \frac{p_3(t+\Delta t) - p_3(t)}{\Delta t} = -(\lambda_{34} + \lambda_{31})p_3(t) + p_2(t)\lambda_{23}$$

S4

- перейти из  $S_3$  в  $S_4$
- перейти из  $S_2$  в  $S_4$
- не перейти из  $S_4$  в  $S_2$

$$p_4(t + \Delta t) = p_4(t)(1 - \lambda_{42}\Delta t) + p_3(t)\lambda_{34}\Delta t + p_2(t)\lambda_{24}\Delta t$$

$$\lim_{\Delta t \rightarrow 0} \frac{p_4(t+\Delta t) - p_4(t)}{\Delta t} = -p_4(t)\lambda_{42} + \lambda_{34}p_3(t) + p_2(t)\lambda_{24}$$

Интегрирование этой системы дает исходные вероятности состояния как функции времени.

Начальные условия берутся в зависимости от того, какого было начальное состояние системы.

Если в момент времени  $t=0$  система находится в состоянии

$$S_1 : X = (1000)$$

$$S_2 : X = (0100)$$

Кроме того должно выдерживаться уравнение нормировки

## 8.1 Правило построения уравнений Колмогорова

В левой части стоит производная вероятности состояния, правая часть содержит столько членов сколько стрелок связано с этим состоянием. Стрелка направлена из состояния - знак „-“ в состояние - „+“

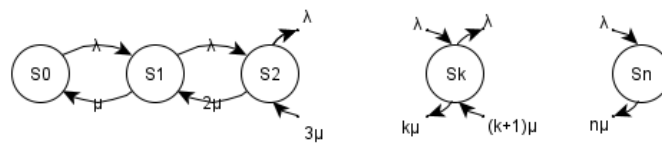
Каждый член равен произведению плотности вероятности перехода (интенсив) соответствует данной стрелке, умнож на вероятность того состояния из которого исходит стрелка

## 9 Многоканальная СМО (система массового обслуживания) с отказами

Будем нумеровать состояния системы по числу занятых каналов т.е. по числу заявок в системе

Вводим состояния

- $S_0$  – все каналы свободны
- $S_1$  – занят 1 канал, остальные свободны
- $S_k$  – занято  $k$  каналов, остальные свободны
- $S_n$  – заняты все  $n$  каналов



Разметим граф, т.е. проставим стрелки соответствующие интенсивности потока заявок  $\lambda$

Поток обслуж заявок с интенсивностью  $\mu$

Пусть система находится в состоянии  $S_1$ , тогда как только закончится обработка заявки в этот момент интенсивность будет  $\mu$

Для двух каналов –  $2\mu$  и тд

### 9.1 Уравнение Колмогорова

- $S_0 : p'_0(t) = -p_0(t)\lambda + p_1(t)\mu$
- $S_n : p'_n(t) = p_n(t)\lambda - p_n(t)n\mu$
- $S_1 : p'_1(t) = -p_1(t)\lambda + p_2(t) \cdot 2\mu - \mu p_1(t) + p_0(t)\lambda$

$$\bullet S_k : p'_k(t) = -p_k(t)\lambda + p_{k+1}(t)(k+1)\mu - k\mu p_k(t) + p_{k-1}(t)\lambda$$

Предельные вероятности состояний для данной системы характер. установившийся режим работы  $\Rightarrow$  производная от константы  $= 0$

$$p_0 = \frac{1}{1 + \frac{\lambda/\mu}{1!} + \frac{(\lambda/\mu)^2}{2!} + \dots + \frac{(\lambda/\mu)^n}{n!}} = \frac{1}{1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!}}$$

$$p_k = \frac{(\lambda/\mu)^k}{k!} p_0 = \frac{\rho^k}{k!} p_0$$

Обозначим  $\frac{\lambda}{\mu} = \rho$  – среднее число заявок в системе

$$\text{Вероятность отказа } p_n = \frac{\rho^n}{n!} p_0$$

$$\text{Относительная пропускная способность } q = 1 - p_n$$

Среднее число заявок обслуж в единицу времени (абсолют. проп. способность.)

$$\lambda q = \lambda(1 - p_n) = A$$

Полученное соотношение можно рассматривать как базисную модель оценки производительности системы, где  $\lambda = \frac{1}{t}$  – усредненная характеристика пользователей ( $t$  – время обработки)

$\mu$  характеризует технические средства реализации системы

$$\text{Среднее число зан каналов } \bar{k} = \frac{A}{\mu} = \rho(1 - \frac{p_n}{n} p_0)$$

Если время I/O инф по каждой задаче мало по сравнению с временем реш, то можно принять след:  $t = \frac{1}{\mu}$  – отношению среднего числа операций процессора на 1 задаче к среднему времени быстроедействия.  $t$  – среднее время решения задачи.

## 10 Лабораторная работа 2

Таблица

n от 1 до 10

Переводя курсор по клеткам заполнить  $\lambda_{ab}$ , найти время нахождения в выбранном состоянии

Мы говорим о поиске времени в установившемся состоянии

## 11 Немарковские случайные процессы, сходящиеся к марковским

Реальные процессы не являются марковскими т.к. часто обладают ... действием. Иногда при исследовании таких процессов удается воспользоваться методами разработать для марковских цепей.

Наиболее распространены:

- метод разложения случайного процесса на фазы (метод псевдосостояний)
- метод вложенных цепей Маркова



## 11.1 Метод псевдосостояний

Состояния системы, потоки переходов из которых является немарковскими, заменяются эквивалентной группкой фиктивных состояний, поток переходов из которых уже является Марковским.

Возникает необходимость в создании условий статистич. эквив. реальных состояний и фиктив. В каждом конкретном случае они выбирают по-разному.

$$\begin{cases} p_{13}(t) = p_{12}(t) \\ p_{12}(t) = p_{11}(t) \\ \mu p_1(t) = \lambda p_0(t) \end{cases} \rightarrow \frac{p_0(t)}{p_{11}(t)} = \frac{\mu}{\lambda}$$

$$p_1 = p_{11}(t) + p_{12}(t) + p_{13}(t)$$

$$p_1(t) + p_0(t) = 1$$

$$p_0 = \frac{\mu}{\mu + 3\lambda}$$

$$p_1 = \frac{3\lambda}{\mu + 3\lambda}$$

## 11.2 Метод вхождения цепей Маркова

Образуется следующим образом: в исходном случайном процессе выбирается случайный процесс таким образом в котором значения образуют марк. цепь. Моменты времени обычно выбираются случайно и зависят от свойств процесса. Решаются только эти времена.

Частный случай – полумарковские процессы

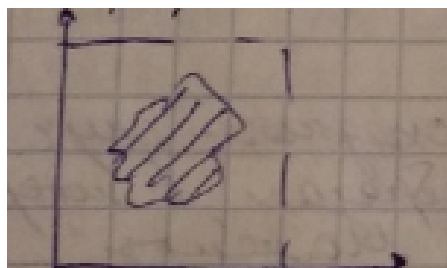
Случ. процесс с конечным или счетным числом состояний называется полумарковским или с заданной вероятностью перехода системы из одного состояния в другое  $(p_i, p_j)$  и распредел времени пребывания процесса в каждом из состояний. Например в виде функции распределения или функции плотности распределения.

Реально на практике почти все процессы - не марковские, поток заявок не всегда пуассоновский, показат распределение – еще реже.

Для произв потоков событий переводящих систему из состояний в сост аналит решения получены только для отдельных частных случаев.

Будем использовать метод статистических испытаний (Метод Монте-Карло)

Идея - вместо аналитического описания проводится ?розыгрыш? т.е. моделирования некоторой величины с помощью некоторой процедуры дающей "случайный" результат. Производя розыгрыш  $n$  раз ( $n \rightarrow \infty$ ) получаем статистический набор – множество реализаций случайного явления (предложено Фон Кейманом)



Суть метода

1. Любым способом получаем  $x_i y_i$ , подчиняется равномерному закону распределения на отрезке  $[0,1)$
2. Считаем что одно число определяется координат первое по  $x$ , второе - по  $y$
3. Анализируем, принадлежит ли т. поверхности
4. Считаем число попавших и не попавших

Процесс генерации случайных чисел с заданным законом распределения и проверка принадлежности проводится  $n$  раз

Например в качестве критерия можно взять

$$\min \int_{t_1}^{t_2} [\lambda_i(\tau) - \lambda_i(\tau)] dt$$

За счет расширенного числа систем некотор процессы удается точно свести к марковским. Создается т.о. система статистич эквивалентна или близка к реальной системе (Далее исследуем с помощью уравн Колмогорова)

В случае потока Эрланга  $k$ -го порядка интервал времени между соседними событиями представл сумму  $k$  независимых случайных распределений имеющих показательный закон. Поэтому сведение потока Эрланга  $k$ -го порядка к пуассон осуществляется введением  $k$  псевдосостояний.

Интенсивность перехода между псевдосост равны кот в состоянии потока Эрланга. Получ таким образом случайн проце является марковским т.к. интервалы времени его нахождения в различных состояниях подчинены показат закону распред

Пример:

Есть некоторое устройство  $S$ , которое выходит из строя с интенсивностью  $\lambda$ . Поток отказов пуассоновский. После отказа устройство восстанавливается твост распределенное по закону Эрланга 3-го порядка.

$$f(t) = 0.5\mu(\mu t)^2 e^{-\mu t}$$

Найти предельные вероятности состояний системы

Решение:

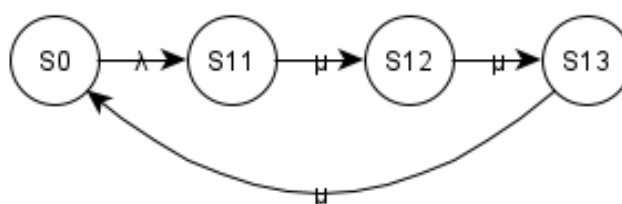
$S_0$  – устройство работает

$S_1$  – устройство отказало и восстанавливается

$S_0 \rightarrow S_1 = \lambda$  – закон Пуассона

$S_1 \rightarrow S_0 = \mu$  – закон Эрланга 3-го порядка (3 временных интервала)

$$T = t_1 + t_2 + t_3$$



$$\begin{cases} p'_0(t) = -\lambda p_0(t) + \mu p_{13}(t) \\ p'_{11}(t) = \lambda p_0(t) - \mu p_{11}(t) \\ p'_{12}(t) = -\mu p_{12}(t) + \mu p_{11}(t) \\ p'_{13}(t) = -\mu p_{13}(t) + \mu p_{12}(t) \end{cases}$$

Определим площадь как

Погрешность –  $\epsilon < \sqrt{\frac{1}{n}}$

+ универсальность, обусловленная возможностью статистического исследования объекта

- для реализации возможно нужны точные статистические сведения о параметрах элемента

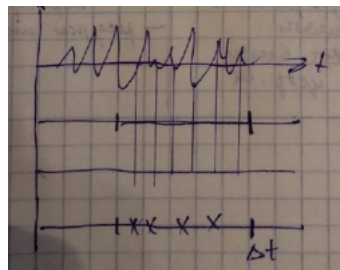
- большой объем вычислений => важность выбора n

## 12 Способы получения последовательности случайных чисел

На практике применимы 3 способа

### 12.1 Аппаратный

Использование шума устройства



Случайная величина вырабатывается специальной электрической приставкой (генератор случайных чисел) как правило внешнее устройство компьютера не требует других устройств и операций кроме обращения к устройству

### 12.2 Табличный

Записанные в таблицы

## 12.3 Алгоритмический

| Способ          | Достоинства  | Недостатки  |
|-----------------|--|---|
| Аппаратный      | запас чисел неограничен<br>мало вычислительных операций<br>не занимает место в памяти          | требуется периодическая проверка на случайность<br>нельзя воспроизвести<br>использование специальных устройств<br>меры для обеспечения стабильности |
| Табличный       | однократная проверка на случайность<br>воспроизводимость                                       | запас чисел ограничен<br>занимает место в памяти<br>время на обращение к памяти   |
| Алгоритмический | однократная проверка<br>воспроизводимость<br>мало памяти<br>не используются внешние устройства | запас чисел ограничен периодом<br>ресурсы системы   |

Вопросы на самостоятельную проработку

1. Свойства функций и плотностей распределения случайного числа, взаимосвязь (интеграл / производная)
2. Графическое отображение функции плотности
  - равномерно распределение на  $[a, b]$
  - нормальное распределение
  - Пуассон
  - Экспоненциальное
  - k-распределение Эрланга

Функция плотности, мат ожидание, дисперсия

## 13 Простейшие алгоритмы генерации псевдослучайных чисел

Один из первых способов получения последовательности псевдослучайных чисел – выделение дробной части многочлена первой степени.

$$y_n = Ent(a \cdot n + b)$$

Если  $n$  пробегает ряд натуральных чисел то поведение  $y_n$  хаотично

Якоби доказал что при рациональном коэффициенте  $a$ , множество  $y_n$  конечно, при иррациональном – бесконечно и всюду плотно в интервале  $(0, 1)$

Бэйль решил соотв задачу для коэффициентов больших степеней. Предложен критерий равномерности от натурального ряда чисел: среднее по реализациям псевдослучайных чисел = среднее по всему их множеству с вероятностью 1.

Проблемы из-за иррациональности – разрядная сетка конечна

### 13.0.1 Алгоритмы

1) Фон Нейман: каждое последующее случайное число образуется возведением предыдущего в квадрат и отбрасыванием цифр с обоих концов

2) Конгуэртный метод

$y_{n+1} = (ky_n + l) \bmod M$  при  $l = 0, M = 2^n$  наибольший период достигается при  $k=3+8i$  и  $k = 5+8i$

Главное чтобы начальное число было нечетным

Но! Было доказано что все такие числа лежат на 15 параллельных плоскостях.

"От отчаяния" используют 2 или 3 разных генератора, смешивая их значения.

Если разные генераторы независимы, то сумма их дисперсий = сумма всех дисперсий, поэтому вырастает случайность рядов

3) На основе ряда Фибоначи

Бит переноса (обычно последняя цифра)

Старые программы генерируют случайные числа от 0 до 1 на Fortran включен подбор констант

## 13.1 Генерация случайных чисел

(n=0 – настройка, n=1 – равномерное распределение, n=2 = нормальное распределение)

```
const m34:double
      m35
      m36
      m37

function Rand(n:integer):double;
var S,W:double;
    i = integer;
begin
    if n = 0 then
    begin
        x:=m34; Rand:=0; exit;
    end;
    S:=-2.5
    for i:=1 to S do
    begin
        x:=5.0*x;
        if x>m37 then x:=x-m37;
        if x>m36 then x:=x-m36;
        if x>m35 then x:=x-m35;
        w:=x/m35;

        if n = 1 then
        begin
            Rand:=w; exit;
        end;
        S:=S+W;
    end;
end;
```

$S := S * 1.154919;$

$\text{Rand} := (\text{sqr}(5) - 3.0) * S * 0.01 + S;$

end;

Для имитации равномерного распределения на интервале  $[a, b]$  используется обратное преобразование функции плотностей

$$\frac{x-a}{b-a} = R \Rightarrow x = a + (b-a)R$$

$R$ -случайная величина в интервале  $[0, 1]$

В основе генерации случайных чисел с законом отличным от равномерного лежит метод преобразования последовательности случайных чисел равномерного закона распределения в заданный закон распределения.

В основе – теорема, которая утверждает, что  $F(t) = \int_{-\infty}^t f(x)dx = R$ , где  $x$  – Закон распределения,  $R$  – Случайная величина от 0 до 1

Для распределения по показательному закону

...

$$x = -1/\lambda \cdot \ln(1 - R)$$

Распределение Пуассона относится к числу дискретных т.е. переменная может принимать только целочисленные значения в том числе 0, с математическим ожиданием и дисперсией  $\lambda > 0$

Можно использовать метод точек: генерировать числа от 0 до 1 пока не будет справедливо  $\prod_{i=0}^x R_i \geq e^{-s} > \prod_{i=0}^{x+1} R_i$

При получении случайной величины функция распределения, которая не позволяет найти решения уравнения  $F(t)$  в явной форме, можно произвести кусочно-линейную аппроксимацию и вычислить приближенное значение корня

Кроме того для получения случайной величины часто используют те или иные свойства распределения

Например распределение Эрланга с параметром  $\lambda$  и  $k$

Известно, что поток Эрланга может быть получен прореживанием потока Пуассона  $k$  раз, поэтому достаточно получить  $k$  значений случайной величины, распределенной по показательному закону и усреднить их  $x = \frac{1}{k} \sum_{i=1}^k (-\frac{1}{\lambda} \ln(1 - R_i)) = -\frac{1}{k\lambda} \sum_{i=1}^k \ln(1 - R_i)$

Нормально распределенная случайная величина может быть получена как сумма большого числа случайных величин, распределенных по одному и тому же закону с одними и теми же параметрами.

Случайная величина  $x$ , имеющая нормальное распределение  $Z$

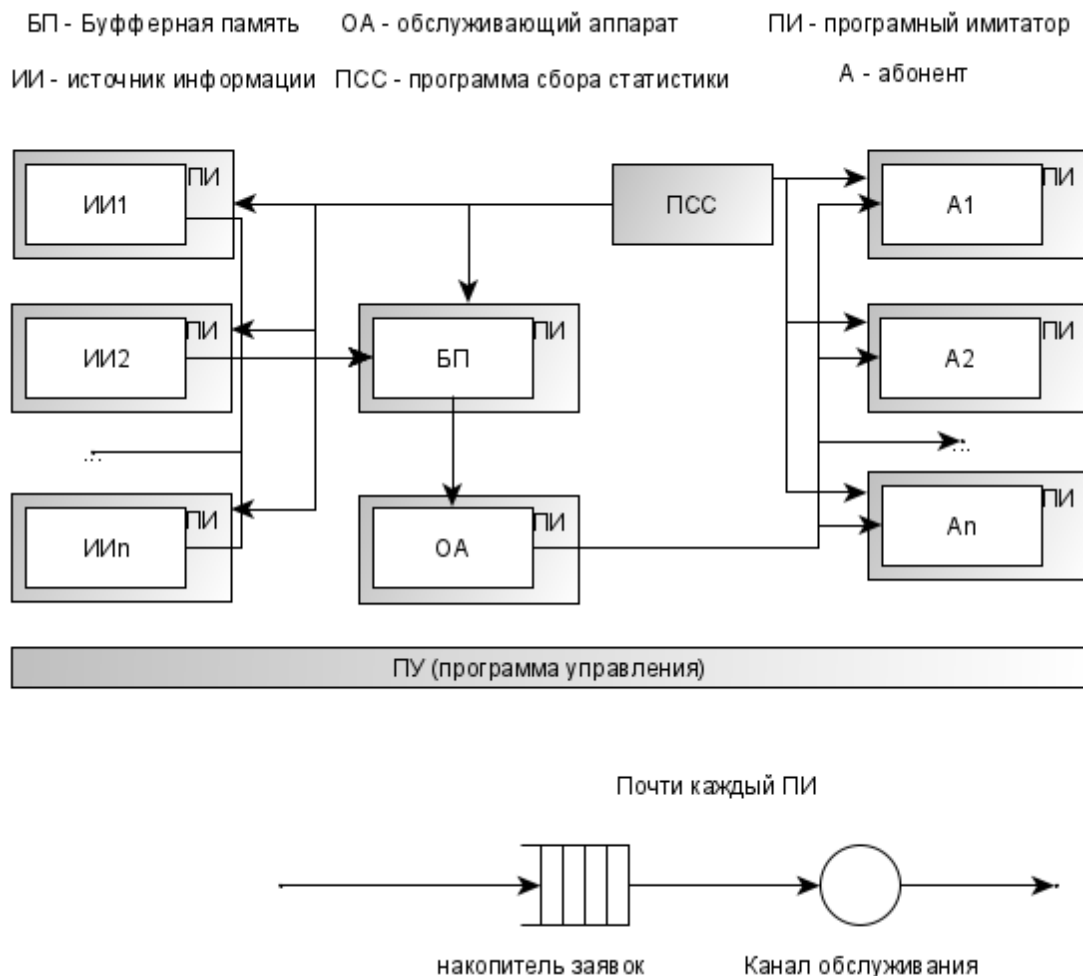
$$N = (m_x, \sigma_x^2)$$

$$x = \sigma_x \sqrt{\frac{12}{n}} \left( \sum_{i=1}^n R_i - \frac{n}{2} \right) + m_x$$

## 14 Методика построения программной модели

Для разработки программной модели исходная система должна быть представлена как стохастическая система массового обслуживания.

Это объясняется тем что информация от внешней среды поступает в случайные моменты времени, длительность обработки различных типов информации может быть в общем случае различна, поэтому внешняя среда как бы является генератором сообщений, комплекс вычислительных устройств = обслуживающие устройства.



Источник информации выдает на вход буфера памяти независимо друг от друга сообщения. Закон появления сообщений произволен, но как правило формализован и задан.

В памяти сообщения записываются и выбираются в обслуживающий аппарат по принципу FIFO. Длительность обработки одного сообщения также случайна, но закон обработки также должен быть известен.

Быстродействие обслуживающего аппарата ограничено, поэтому на входе в память возможно сложение данных которые в общем случае образуют очередь заявок на те или иные ресурсы.

Главное – придумать программу управления (ПУ), которая определяет когда обратиться к какому из устройств.

Поток сообщений обычно моделируется моментами появления очередного сообщения в потоке (происх до этого  $+t_{i-(i+1)}$  между сообщениями), на которое мы продвигаем модельное время.

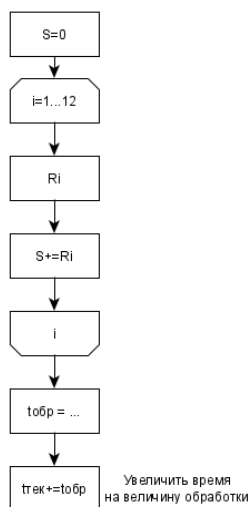
Таблица генерации времен

|                  |   |
|------------------|---|
| Нормальное       | $t = \sigma_t \sqrt{\frac{12}{n}} (\sum_{i=1}^n R_i - \frac{n}{2}) + m_x$ |
| Равномерное      | $t = a + (b - a)R$  |
| Экспоненциальное | $t = -\frac{1}{\lambda} \ln(1 - R)$                                       |
| Эрланга          | $t = -\frac{1}{k\lambda} \sum_{i=1}^k \ln(1 - R_i)$                       |

## 15 Моделирование работы обслуживающего аппарата

Программа моделирующая работу обслуживающего аппарата – это набор функций, вырабатывающих случайные отрезки времени соответств времени обслуживания

$N(m_x, \delta_x)$  – нормальное распределение Тобр =  $m_x + (\sum_{i=1}^{12} R_i - 6)\delta_x$



## 16 Моделирование абонентов

Абонент -

Необходимо составить функции требований

Абонент и сам может быть источником заявок т.о. абонент либо



## 17 Моделирование работы обслуживающего аппарата

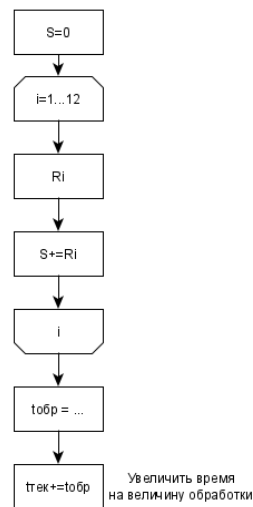
Программа имитирует работу обслуживающего аппарата – набор функций вырабатывающих случайные отрезки времени соответствующего длительного обслуживания

Требования от источников обработки по нормальному закону  $\mu(x)$  и  $\sigma(x)$

Написать время обработки обслуживания аппаратом события

$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$  тобр =  $MX + (\sum_{i=1}^{12} R_0 - 6)\sigma$ , где  $R_0$  – псевдослучайное число на (0,1)

Алгоритм работы формулы:



### 17.1 Работа моделирования абонентов

Абонент может рассматриваться как обслуживающий автомат, поток информации на котором поступает от процессора. Для моделирования работы абонентов необходимо составить функцию выработки времени обслуживания требований. Кроме того абонент сам может быть исполнителем заявок (генерации с помощью генератора сообщений) т.о. абонент либо как обслуживающий аппарат либо как генератор.

### 17.2 Моделирование буферной памяти

Память представляет собой сложный объект.

Разрабатываем концептуальную модель (вербально полностью определить и суть и свойства)

Дать определение, что такое память, указав свойства:

**Определение.** *Память – электромеханическое устройство предназначенное для записи выделения и хранения информации.*

Надо промоделировать саму запоминающую среду:

- запись осуществляется в ячейку по адресу
- чтение из ячейки по адресу, сразу генерирование следующего адреса

Самый простой способ – массив.

Если считываем информацию из пустой ячейки то с точки зрения модели оно должно быть зарезервировано, что был факт обращения к памяти.

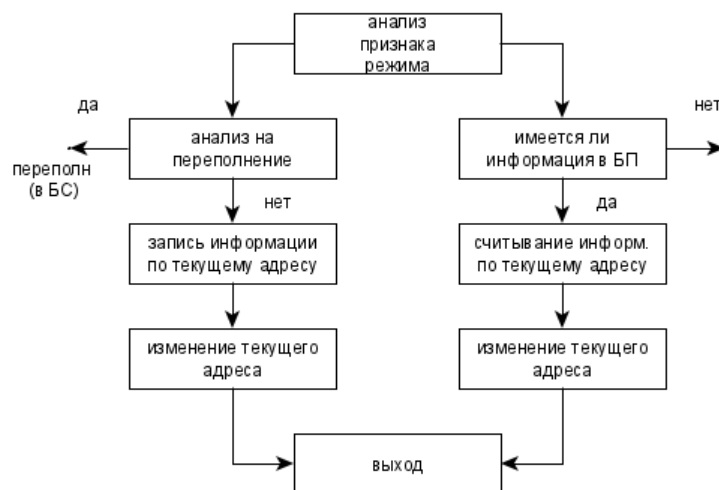
Блок памяти должен производить запись и считывание числа, выдавать сигналы переполнения и отсутствия данных.

В любой момент модельного времени располагать сведения о находящ в нем количестве требований.

Сама запоминающая среда в простейшем случае имитируется некоторым одномерным массивом размер которого ограничивает емкость памяти.

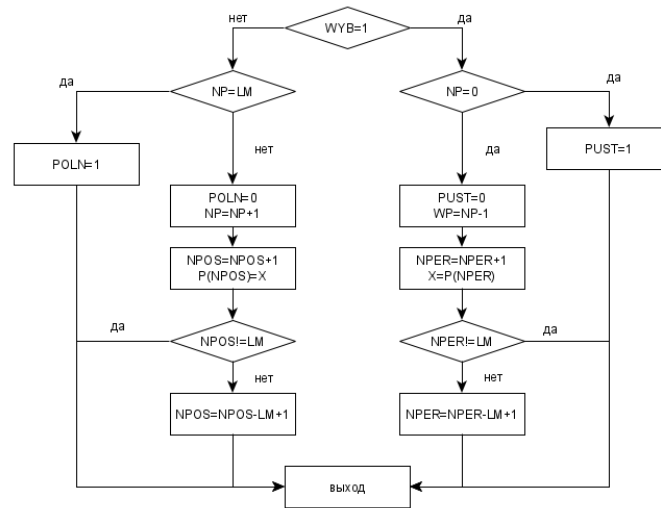
Каждый элемент этого массива может быть либо свободен либо занят. Если занят, ему может быть присвоено время появления этого объекта

Структурная схема модели памяти



Проблемы при моделировании памяти: время запроса (ускорение – введение базирования и индексирования)

Алгоритм реализации работы буферной памяти



|      |   |
|------|---|
| P    | массив обращения  |
| WYB  | признак обращения к буферной памяти<br>(0 – режим записи, 1 – режим выбора сообщения) |
| NP   | число сообщения в памяти  |
| LM   | объем буферной памяти   |
| NPOS | номер последнего сообщения в памяти   |
| NPER | номер первого сообщения в памяти  |
| POLN | признак переполнения (1-нет свободных ячеек)  |
| NPOS | =NPOS+1, если NPOS<LM<br>NPOS-LM+1 иначе  |
| X    | ячейка для сообщения  |
| PUST | признак отсутствия сообщения (1-в памяти нет сообщения)                               |
| NPER | =NPER-1, если NPER<1,<br>=NPER-LM+1 иначе   |

### 17.3 Разработка программы сбора статистики

Задача блока статистики заключается в накоплении численных значений, необходимых для выполнения статистических оценок заданных параметров работы моделируемой вычислительной системы

При моделировании простейшей СМО интерес представляет среднее время ожидания в очереди. Для каждого сообщения время ожидания в очереди. Для каждого сообщения время ожидания в очереди = разности между моментом времени когда оно было выбрано на обработку и и когда оно пришло в систему от источника информации. Коэффициент загрузки обслуживающего аппарата определяется как отношение времени работы обслуживающего аппарата к общему времени моделирования.

Подсчитав число потерянных сообщений можно определить вероятность отказа или потери сообщения.

$$\frac{N_{poteri}}{N_{poter}+N_{obrabot}}$$

## 17.4 Управляющая программа .. модели

Основная функция – реализация алгоритма функционирования

Самое простое – протягивание времени через  $\delta t$  и проверка попадания события ( $t + t_{обр} = \delta t$ ) Шаг за шагом просматривать до окончания  $t$  моделирования.

Если  $\delta t$  очень маленький – долго считать, если  $\delta t$  большое – может попасть больше 1-го значимого события т.е. результат моделирования не будет адекватно отображать процесс.

Второй способ – создание списка событий

Недостаток – список, который нужно постоянно анализировать

Третий – алгоритм Делфи (комбинирует два предыдущих метода)

Если программные имитаторы от источника обслуживающего аппарата и памяти имитируют работу отдельных устройств то управляющая программа имитирует взаимодействие отдельных устройств системы

Управляющая программа реализуется в основном по двум принципам:

1) Принцип  $\delta t$

Заключается в последовательном анализе состояний всех блоков системы в момент  $t + \delta t$  по заданному состоянию блоков в момент времени  $t$ , при этом новое состояние определяется в соответствии с их алгоритмическим описанием с учетом действующих случайных факторов (задается распределение вероятности)

В результате этого алгоритм принимает решение о том какое событие должно имитироваться на данный момент времени

Основной недостаток – значительные затраты модельного времени на его реализацию. При недостаточно малой  $\delta t$  – опасность пропуска отдельных событий в системе что приводит к нарушению адекватности всей модели.

Характерное свойство: состояние отдельных устройств изменяется в дискретный момент времени, совпадающий с моментом поступления сообщения в систему => моделирование с продвижением времени удобно проводить с использованием событийного принципа. При его использовании состояние всех блоков системы лишь в момент появления или наступления какого либо события

Момент наступления следующего события определяет ..  $t$  из списка будущих событий, представляющих собой совокупность моментов ближайшего изменения состояния каждого из блоков системы

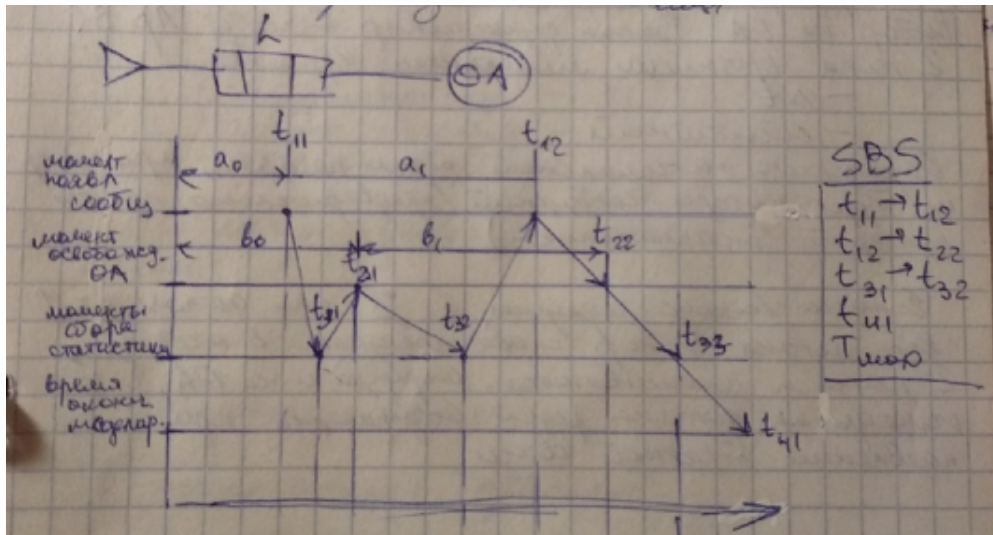
Пусть необходимо промоделировать систему состоящую из:

- генератора
- памяти
- обслуживающего аппарата

$t_{1i}$  – моменты появления сообщений

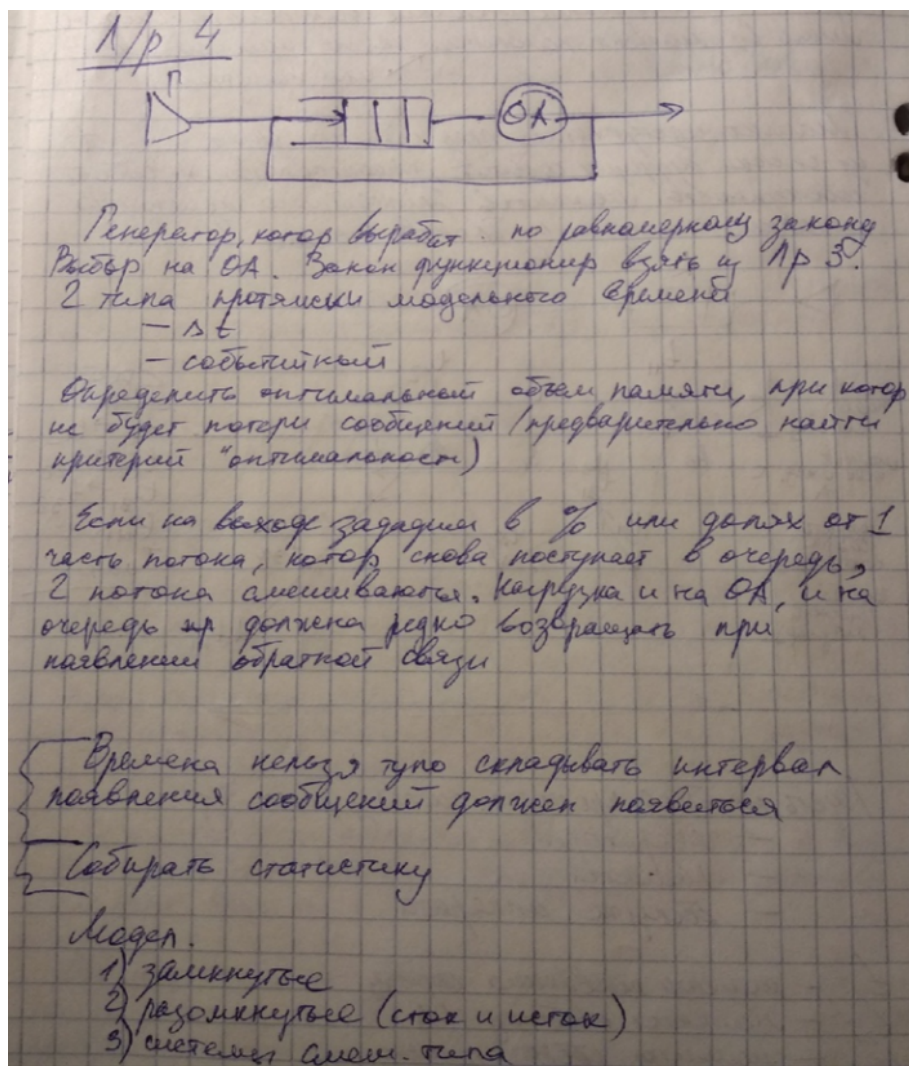
$t_{2i}$  – моменты освобождения ОА

$t_{3i}$  – моменты сбора статистики  $t_{4i}$  – моменты окончания моделирования



**Замечание.** Самостоятельно написать по пунктам методику написания моделирования блоков

Комбинированный метод – циклические списки, где каждый элемент – свой список, а самый нижний уровень – набор времен, продвигаемых по  $\delta t$



## 18 Классификация СМО

СМО классифицируется по следующим признакам -закону распределения входного потока -число обслуживающих приборов -закону распределения времени (в обслуживающих приборах) -числу мест в очереди -дисциплине обслуживания

Для краткости в литературе часто записывается СМО которое характеризуется следующими характеристиками  $A/B/C/D/E$ , где  $A$  — закон распределения интервалов времени между поступлением заявок, наиболее часто используемые (экспоненциальная, эрланговская, гиперэкспоненциальная, гамма, детерминированное распределение) для обозначения произвольного характера распределения используется  $G$ .  $B$  — закон распределения времени обслуживания в приборах СМО  $C$  — число обслуживающих приборов (для одноканальных записывается 1, для многоканальных в общем случае  $l$ )  $D$  — число мест в очереди (если неограничено, то может отсутствовать) для конечного числа мест в очереди принято обозначение либо  $r$  либо  $m$ .  $E$  — Дисциплина обслуживания наиболее часто используются следующие варианты (fifo, lifo, random) fifo если не указано

СМО с одним обслуживающим прибором, бесконечной очередью, экспоненциальными законами распределения, интервалами времени между поступлениями заявок и времени обслуживания. Дисциплина FIFO.

$m/m/1$

СМО с несколькими обслуживающими приборами, конечной очередью, эрланговским законом распределения заявок, гиперэкспоненциальным временем обслуживания lifo.

$E/h/l/r/lifo$

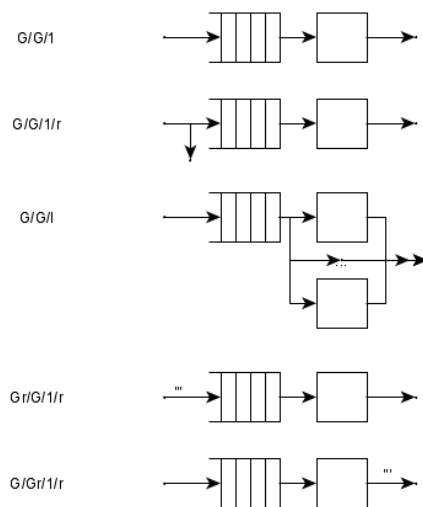
СМО с несколькими обслуживающими приборами, бесконечной очередью, произвольными законами распределения времени, между поступлениями заявок. FIFO

$G/G/l$

Для моделирования вычислительных систем и сетей чаще всего используются следующие типы СМО

- 1) Одноканальная СМО с ожиданием Представляет собой один обслуживающий аппарат с бесконечной очередью. Является наиболее распространенным формализмом при моделировании практически любых узлов сетей и вычислительных систем
- 2) Одноканальные СМО с потерями Представляют собой один обслуживающий аппарат с конечным числом мест в очереди Используются при моделировании каналов передачи.
- 3) Многоканальное СМО с ожиданием Представляют собой несколько параллельно работающих обслуживающих приборов с общей бесконечной очередью. Используются при моделировании групп абонентских терминалов работающих в диалоговом режиме.
- 4) Многоканальное СМО с потерями. Представляют собой несколько параллельно работающих обслуживающих приборов с общей очередью число мест в которой ограничено. Используются для моделирования каналов связи.
- 5) Одноканальные СМО с групповым поступлением заявок Представляют собой один обслуживающий прибор с бесконечной очередью. Перед обслуживанием заявки группируются в пакеты по определенному правилу.
- 6) Одноканальные СМО с групповым обслуживанием заявок. Представляют собой один обслуживающих прибор

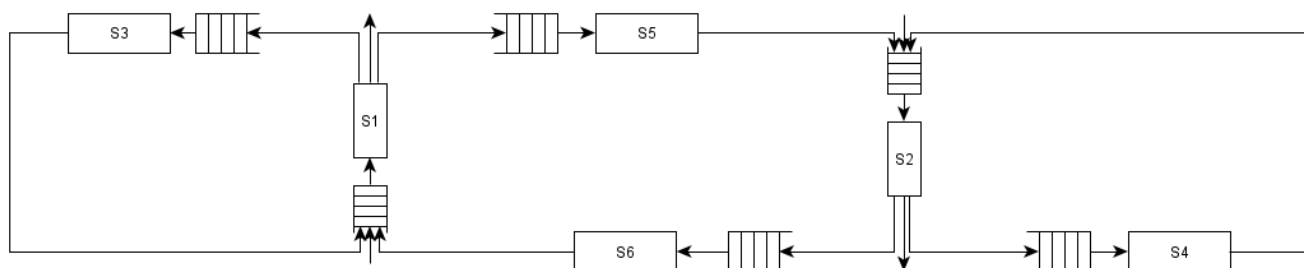
с бесконечной очередью. Заявки обслуживаются пакетами. Два последних типа СМО могут использоваться для моделирования центров коммутации.



Вычислительные сети могут быть в общем виде представлены в виде сети массового обслуживания. Различают открытые, замкнутые и смешанные сети.

Открытой называется сеть массового обслуживания состоящая из  $m$  узлов причем хотя бы в один из узлов сети поступает извне входящий поток ... (заявок?) и имеется сток заявок. Для открытых сетей характерным является то что интенсивность поступления заявок в сеть не зависит от состояния сети. Такие сети как правило используются для вычислительных сетей или систем работающих в не оперативном режиме.

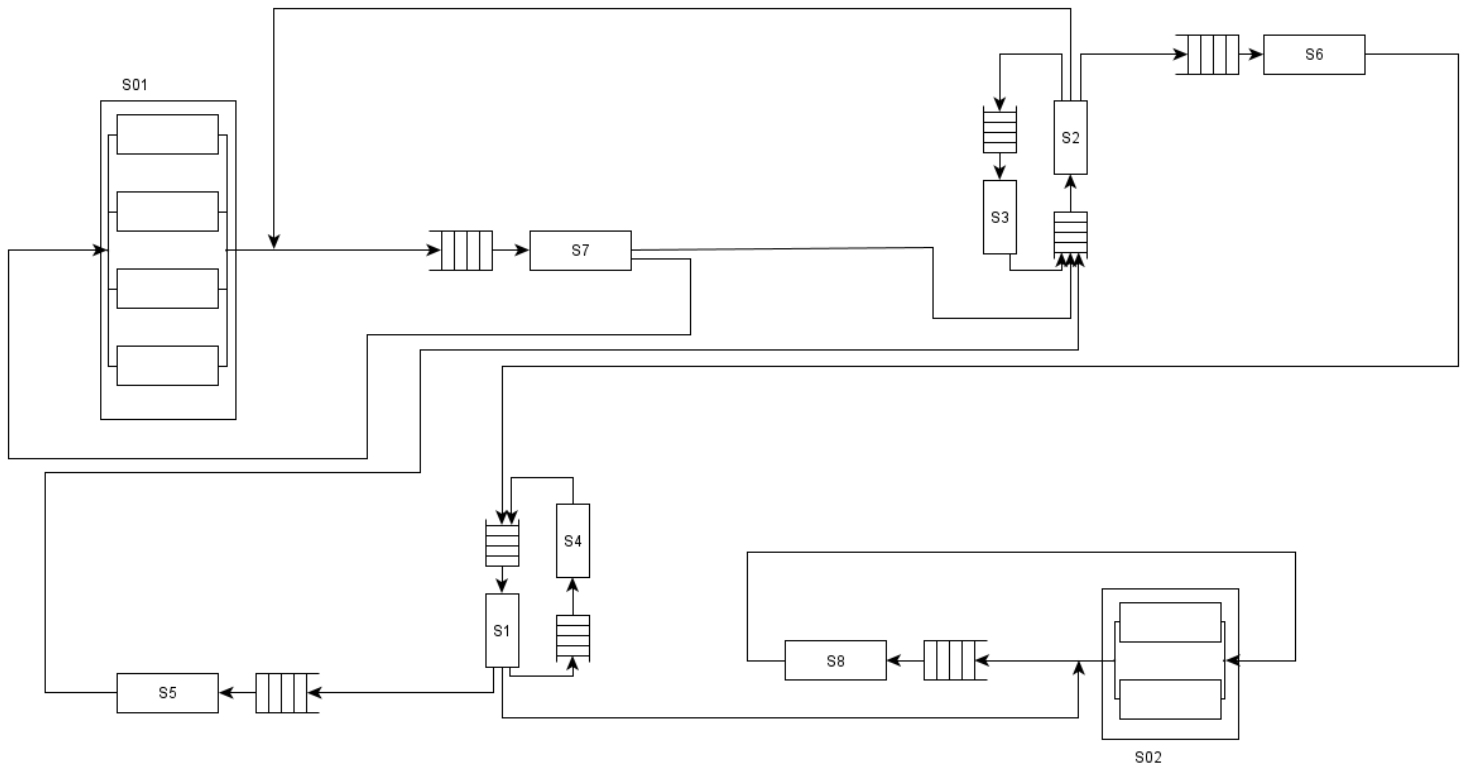
В сети циркулирует 2 потока заявок, каждая заявка поступает на вход соответствующего узла коммутации, где определяется место ее обработки. Затем заявка передается на «свой» сервер или по каналу связи на соседний, где и обслуживается. После чего возвращается к источнику и покидает сеть. Обязательно должно быть 2 истока заявок и 2 прибора в который входят извне потоки заявок.



S1,S2 – моделирует работу узлов, у них есть исток и исток S3,S4 – работу серверов S5,S6 – работу межузловых коммутаторов

Замкнутой называется сеть массового обслуживания с множеством узлов без источника и стока, в которой циркулирует постоянное число заявок. Замкнутые сети массового обслуживания используются для моделирования таких вычислительных

систем источниками информации для которых служат абонентские терминалы, работающие в диалоговом режиме. В этом случае каждая группа абонентских терминалов представляется в виде многоканальной сети массового обслуживания (системы массового обслуживания) с ожиданием. И она включается в состав устройств сети. Различают простой и сложный режим диалога работы абонента. Простой – абоненты не производят никакие действия, кроме отправки заданий в вычислительную систему и обдумывание ответа от системы. Абоненты из терминала посылают запросы которые по каналам связи поступают на узлы коммутации оттуда на обработку, на свой или соседний сервер.



S01,S02 – группы абонентских терминалов S1,S2 – узлы коммутации S3,S4 – сервер S5,S6 – каналы межузловой связи S7, S8 – каналы связи с абонентом

При сложном режиме работы представляется в виде совокупности операций некоторого процесса называемого технологическим. Каждая операция технологического процесса моделируется соответствующей сетью массового обслуживания. Причем часть операций предусматривает обращение к вычислительной сети а часть нет.

(та же схема что и сверху, группы абонентов замещаются технологическими операциями)

Смешанной называется такая сеть массового обслуживания в которой циркулирует несколько различных типов заявок, причем относительно одних типов сеть замкнута, относительно других открыта. С помощью смешанных сетей массового обслуживания моделируется части абонентов которые работают в диалоговом а часть в непрерывном режиме (**неоперативность**).

**Замечание.** Лабораторная работа 5 В какой то информационный центр приходят клиенты через интервалы времени  $10 \pm 2$  минуты. Если все три имеющихся опе-



ратора заняты клиенту отказывают в обслуживании. Операторы имеют разную производительность и могут обеспечивать обслуживание среднего запроса пользователей за  $20 \pm 5$ ,  $40 \pm 10$ ,  $40 \pm 20$ . Клиенты стремятся занять свободного оператора с максимальной производительностью. Полученные запросы сдаются в приемный накопитель откуда и выбираются на обработку на первый компьютер запрос от первого и второго операторов, на второй от третьего оператора. Время обработки запросов на первом и втором компьютере равны 15 и 30 минут соответственно. Смоделировать процесс прохождения трехсот запросов. Определить вероятность отказа.

1. нарисовать концептуальную модель
2. перевести в термины СМО
3. накопители, пять каналов при 2 накопителях
4. эндогенные переменные (зависимые) – время обработки  $i$ -ым оператором, и время решения на  $j$ -ом компьютере
5. экзогенные переменные.  $N0$  – число обслуженных клиентов,  $n1$  – число клиентов получивших отказываю

Уравнение модели  $N1/(N1+N0)$

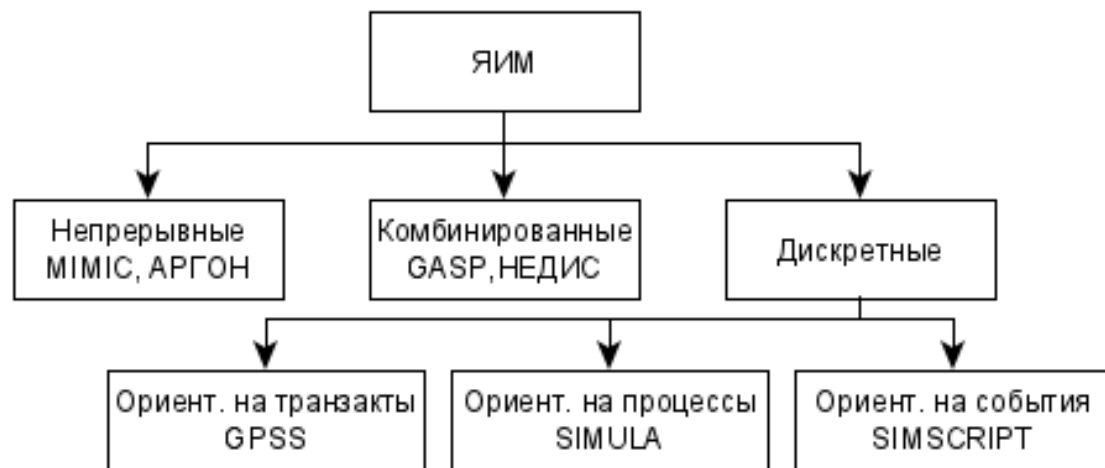
За единицу системного времени возьмем 1/100 минуты + в отчете все схемы придумать реальную систему (анализ работы которой может быть формализован системой массового обслуживания)

## 19 Основные принципы построения языков моделирования

GPSS почитать

Качество языков моделирования характеризует

1. Удобством описания процесса функционирования
2. Удобством ввода исходных данных, варьированием структуры алгоритмов работы и параметров модели
3. Эффективностью анализа и вывода результатов моделирования
4. Простотой отладки и контроля работы моделирующей программы
5. Доступность восприятия и использования языка



Все специальные языки моделирования определяют поведение систем во времени с помощью событийного или комбинированного метода протяжки модели.

Классическая классификация языков имитационного моделирования В качестве классификации принцип формирования системного времени

Непрерывное представление систем сводится как правило к составлению дифф. ур. С помощью которых устанавливается связь вход-выход. Если выходные переменные модели принимают дискретные значения, то получаются разностные уравнения.

Комбинированный подход предполагает что в системе могут происходить два типа событий:

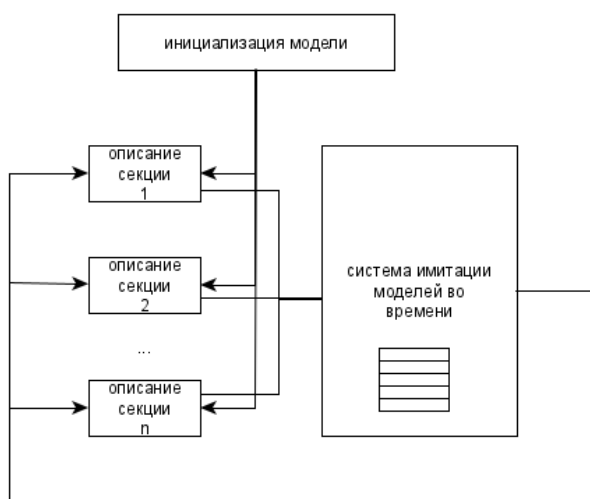
- Зависящие от состояния. Состояния системы в таком случае описывается набором переменных, некоторые из которых меняются непрерывно. Пользователь должен запрограммировать условие наступления событий, законы изменения непрерывных переменных, правила перехода от одного состояния к другому.
- Формальное описание динамики моделируемого объекта. Будем считать что любая работа в системе совершается путем выполнения активности. Активность является наименьшей единицей работы. Ее можно рассматривать как единый дискретный шаг и она имеет свое время выполнения. Следовательно активность это единый динамический объект указывающий на совершение единицы работы. Процесс это логически связанный набор активностей. Активность проявляется в результате совершения событий. Событие это мгновенное изменение состояния некоторого объекта системы. Состояние - характеристика в определенный момент времени.

Рассмотренные объекты активности процесса и события являются конструктивными элементами для динамического описания системы на их основе строятся языки моделирования. В то время когда динамическое поведение системы формируется в результате выполнения большого числа взаимодействующих процессов, сами эти процессы образуют относительно небольшое число классов, чтобы описать поведение систем достаточно описать поведение каждого класса процессов и задать значение атрибутов для конкретных процессов. Построение модели состоит из решения двух задач:

- Описание правил задающие виды процессов
- Указание значений атрибутов таких процессов или задать правила генерации этих значений

При данном подходе система описывается на определенном уровне детализации в терминах множества описаний процесса, каждый из которых включает в себя множество правил ?возбуждения? Активностей. Модель служит для отображения временного поведения системы, поэтому язык моделирования должен отображать тем или иным способом модельное время. В реальной системе совместно выполняется несколько активностей, причем как связанных с данным процессом так и независимых. Имитация этих действий на модели проходит в строгой последовательности в соответствие с алгоритмом.

Языки ориентированные на события Моделирующая программа организованная в виде совокупности секций. Секция — множество событий. Секция событий состоит из набора операций которые в общем случае выполняются после завершения какой либо активности. Выполнение этих процедур или секций синхронизируется во времени списком будущих событий.

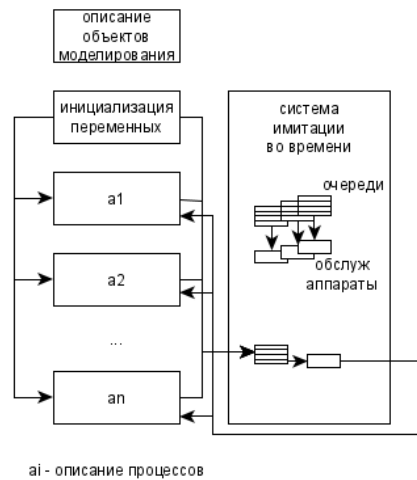


Языки ориентированные на процессы Моделирующая программа организуется в виде набора описания процессов каждый из которых описывает один класс процессов. Собственно описание процесса функционирования устанавливает атрибуты активности. Синхронизация во времени происходит также с помощью списка будущих событий, который содержит время возбуждения конкретного процесса.

#### Структура программы на языке SIMULA

Язык GPSS можно отнести к группе языков также ориентированных на процессы, его используют для описания движения объектов такие процессы называют транзактными и представляют собой элементы потоков. В процессе имитации транзакты создаются и уничтожаются. Если в программе появился хоть один блок Generate то должен быть блок Terminate, иначе модель заикнется.

По возможностям языка ...



Простота применения GPSS SIMSCRIPT SIMULA

Предпочтение пользователей . . .

Сравнение универсальных и специализированных языков Универсальный Пре-

имущества: минимум ограничений на выходной формат, широкая распространимость

Недостатки: значительно время затраченное на программирование, значительное время затраченное на отладку

Специализированные языки Преимущества: меньшие затраты на программирование; более эффективные методы выявления ошибок; краткость, точность выражений, понятий характеризующих процесс; возможность заранее встроить для пользователей стандартные функции, они могут применяться в любой имитационной модели; автоматическое формирование в определенных типах данных; удобство накопления и представления выходных данных; управление и контроль ресурсами (вычислительными). Недостатки: необходимо твердо придерживаться ограничений на выходные форматы; меньшая гибкость модели и большие затраты машинного времени

Возможности программного обеспечения и имитационного моделирования

- Анимация и динамическая графика
  - С целью отладки моделируемой программы
  - Показать что модель неправильна
  - Обучение обслуживающего персонала
- Статистические возможности
  - Необходимость поддерживать генератор псевдослучайных чисел

Существуют два типа анимаций:

- Совместная — осуществляется во время ?прогона модели?
- Раздельная анимация

Объектно ориентированное моделирование Система формируется из объектов которые взаимодействуют друг с другом. Объекты имеют данные и содержат методы. Данные описывают состояние объекта, методы взаимодействуют с объектом.

Преимущества: обеспечивает возможность повторного использования объектов и удобный способ их изменения; реализация иерархического подхода; упрощение изменения модели; возможность декомпозиции задачи на подзадачи.

Недостатки: пакеты очень трудны для изучения

Назовем некоторые предметно ориентированные пакеты имитационного моделирования: Для производственных систем – promodel Для сетей связи – itdecision, opnetmodelling

Универсальные пакеты имитационного моделирования arena, extend, asim, somul8, etc...

## 20 Методологии и системы компьютерного моделирования

1) Matcad – многофункциональная интерактивная вычислительная система позволяющая благодаря встроенным алгоритмам решать аналитические задачи которые решаются численными методами, большинство задач не прибегая к программированию

Рабочая книга – электронная книга с формулами вычисления автоматически в порядке записи формул

Простой и понятный интерфейс, хорошая 2D и 3D графика, возможность записи стандартным математическими символами, возможность подключения к офисным и конструкторским программам/сетям

2) Matlab – система технического программирования сверхвысокого уровня. Помимо обычных языковых конструкций позволяет выполняя процедурное объектно ориентированное и визуальное программирование большое число алгоритмов для расчетов и визуализации

Предназначен для создания професс технически-сложных высокопроизводительных приложений оперируя большими массивами данных. Характеризуется высокой точностью

3) Model Visual Study

Предназначена для визуального объектно-ориентированного моделирования сложных динамических систем.

## 21 Системы моделирования бизнес-процессов

Существует множество моделей описывающих различную производственную и коммерческую деятельность

Решаются задачи:

- определение средней загрузки оборудования
- потребности в мощностях
- логические задачи

Для автоматизации используются типы моделей, наиболее полная классификация которых содержится в стандартах IDEF

- IDEF0 – моделирование функций вход-выход
- IDEF1 – информационное моделирование "BibData" (моделирование данных)
- IDEF2 – динамическое моделирование
- IDEF3 – описание процессов
- IDEF4 – объектно-ориентированные методы проектирования
- IDEF8 – интерфейс пользователя
- IDEF10 – техническое проектирование
- проектирование вычислительных сетей

IDEF0 разработан на основе технологии моделирования SAT (Structured analyse technique)

Также используется методология структур. анализа и моделирования в виде иерархии потока данных

## 22 Объектно-ориентированные системы моделирования

Наиболее известная методология реализуется с помощью языка UML. Он является универсальным позволяющим одновременно с анализом создавать документацию для проектирования сложных систем (часто иерархических). В дальнейшем возможен переход на работоспособный код практически на любом ЯП.

Используя UML, можно содержательно описывать классы, объекты и компоненты, принадлежащие различным областям.

Язык объектно-ориентированный но не привязан ни к какому ЯП высокого уровня.

Для создания динамических моделей большое распространение получили методы, основанные на использовании сетей Петри и конечных автоматов + агентные модели

Для этих целей использование моделирования системы в которых используется динамические элементы различных формализмов (цепи Маркова, теории систем массового обслуживания)

| Имя                | Разработчик                   | Область применения   | Класс реализ модели      | граф коммен для виз. моделей                       | анимация | интегр с кейс и эксперт сист                 |
|--------------------|-------------------------------|--|--------------------------|--|----------|--|
| ARENA              | Systems Modelling Corporation | производств и бизнес-процессы  | дискретные               | палитра стандарт блоков и процессов                | да       | импорт функц моделей из BPWin                |
| EXTEND             | Imagine That Corporation      | экономич проц стратегич планир   | дискретные               | блоки функц преобразов                             | да       | нет  |
| GPSS WORD          |                               | моделир систем массового обслуж произ. и транспорт                     | дискретные               | нет (стыдно)                                       | нет      | нет  |
| IThink Analyst     | High Performance Inc          | моделир финансовой деятельн реинженеринг                               | непрерывные и дискретные | поточковые диаграммы                               | да       | интегр с эксперт сист                        |
| RETHINK            |                               | Моделирование процессов в производстве и в организация                 | дискретные и непрерывные | блоки функц преобраз и процессов                   | да       | совмещена с эксперт сист G2                  |
| Matlab++, Simulink | Mathworks                     | многофункц среда моделир информац и управляющих сист (компонентный ур) | дискретные и непрерыв    | блоки функц сост + спец карты состояний            | да       | напрямую нотация UML при описании сист динам |
| РДО                | МГТУ им Баумана               | моделир процесс управл и произв  | дискрет и непрерыв       | развитый графич интерф + возможны генерации графов | да       | совместимо с эксп системами и интеллект САПР |

## 23 Сети Петри

### 23.1 Формальное описание

**Определение.** *Сеть Петри – это математическая модель дискретных динамических систем (параллельных программ, ОС, компьютеров и их устройств, сетей компьютеров), ориентированных на качественный анализ и синтез таких систем (обнаружение блокировок, тупиковых ситуаций, "узких" мест автоматический синтез таких программ и компонентов комп)*

$$PN = \{\theta, P, T, F, M_0\}$$

$\theta = \{\theta = 0, 1, 2, \dots\}$  – дискретные моменты времени  $t$  начиная с 0

$P = \{p_1, p_2, \dots, p_n\}$  – непустое; позиции, места

$T = \{t_1, t_2, \dots, t_m\}$  – непустое; переходы

$$P \cap T = \emptyset$$

$F$  – функция инцидентности, связыв множества вершин

$$F : (P \times T) \cup (T \times P) \rightarrow \{0, 1, 2, \dots, k\}$$

2 отображения

- $F^p(p, t) = P \times T \rightarrow 0, 1, 2, \dots$  – для каждой позиции указываются переходы с учетом кратности
- $F^t(t, p) = T \times P \rightarrow [0, 1, 2, \dots]$  – для каждого перехода указываются связанные с ним переходы с учетом кратности

Из вершины позиции  $p_i$  дуга в вершину перехода  $t_j$  тогда и только тогда  $f_{ij}^p > 0$

В этом случае говорят, что  $t_j$  – выходной переход позиции  $p_i$

Множество позиций  $p_k$  для которых  $t_j$  – выходной переход, будем обозначать  $p_j$

Т.е.  $p_j = \{p_k : f_{kj}^p > 0\}$

Рассматривая переход из  $t_j$  дуга ведет в позицию  $p_i \in P$  тогда и только тогда когда  $f_{ji}^t > 0$

Говорят что  $p_i$  – выходная позиция перехода  $t_j$

Аналогично множество выходных позиций  $T^j = \{t_l : f_{lj}^t > 0\}$

Каждая позиция  $p_i$  содержит некоторый целочисленный ресурс  $\mu(p) \geq 0$ . Часто он отображается числом фишек/точек внутри позиции

$M = (\mu_0, \mu_1, \mu_2, \dots)$  называется маркировкой/разметкой сети Петри

$M : P \rightarrow (0, 1, 2, \dots)$

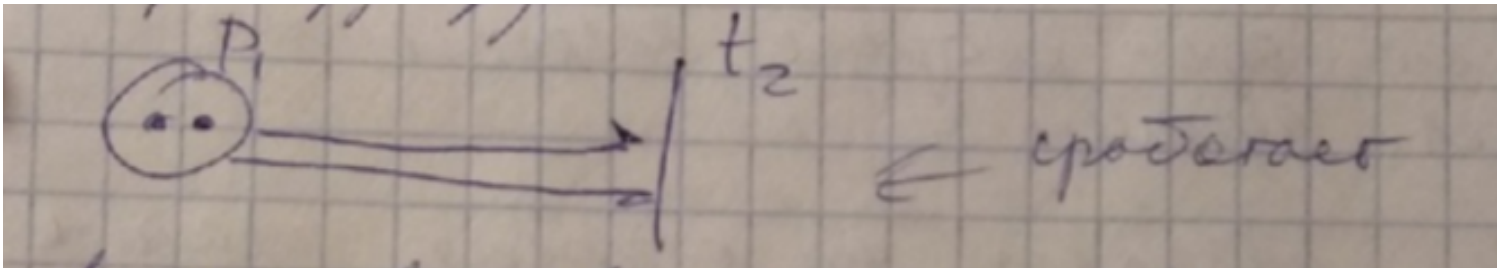
$\mu_0$  – начальная маркировка определяющая стартовые состояния сети Петри

Смена маркировок начиная с  $\mu_0$  происходит в результате срабатывания сети Петри

Переход  $t_j \in T$  может сработать при маркировке  $\mu$  для всех  $p_i \in P$

$$\mu_i(\theta) - f_{ij}^p(\theta) = 0$$

Т.е. если каждая входная позиция для данного перехода содержит как минимум столько фишек, какова кратность ведущей к  $t_j$ -му переходу дуги



Маркировка в след. момент времени  $\theta + 1$  сменяется на  $\mu(\theta + 1) = \mu_i(\theta) - f_{ij}^p(\theta) + f_{ij}^t(\theta)$

Т.е. переход  $t$  изымает из каждой входной позиции число фишек, равное кратности входных дуг и посылает в каждую выходную позицию число фишек, равное кратности выходных дуг

Если может сработать несколько, то срабатывает 1 любой

Функционирование сети останавливается если при некоторой маркировке ни один из переходов сработать не может (тупиковая маркировка)

В силу недетерминированного функционирования сеть Петри может порождать различные последовательности срабатывания переходов. Последовательности образуют слова, множество слов – язык сети Петри.

2 сети Петри эквивалентны, если порождают один и тот же язык.

## 23.2 Теоретическо-графовое представление сети Петри

Явл. двудольный ориентированный мультиграф

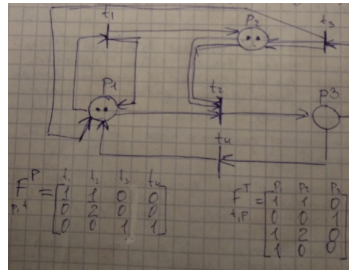
Граф содержит позиции (места) обозначенные  $O$ ; переходы, обозначены планками; ориентир. дуги, соединяющие позиции с переходами и переходы с позициями

Кратные дуги обозначаются несколькими параллельными дугами

Наличие параллельных дуг относит граф к мультиграфовой структуре

Т.к. 2 типа вершин граф называется двудольным





## 24 Пространство состояний сети Петри

Состояние сети определяется ее маркировкой, а подпространство состояний обладает  $n$  позициями, понимаем множество всех маркировок

Изменение в состоянии вызванное запуском перехода, определяется функцией перехода или функцией следующего состояния

**Определение.** *Переход* – изъятие фишек из позиции  $p_i$  и помещение в позицию  $p_k$

Процесс создания новых маркировок продолжается пока в выбранной сети Петри при данной маркировке существует хотя бы 1 разрешенный переход

Если же ни один переход не разрешен, то маркировка называется тупиковой. Получ 2 последовательности:

- последовательность маркировок
- последовательность запущенных переходов

$$M(\theta + 1) = \delta(M(\theta), t_{j\tau})$$

Множество достижимости  $R(PN, M)$  – наименьшее множество маркировок, такое что:

$$\begin{array}{l} M' \in R(PM, M) \\ M'' \in \delta(M', t_{j\tau}) \\ M'' \in R(PN, M) \end{array}$$

Пример (сх. выше)



Исходя из нашего примера при нач маркировке  $M_0 = [2, 2, 0]$  могут работать переходы

1.  $t_1$  получаем  $M'=[2,3,0]$

2.  $t_2$  получаем  $M'=[1,0,1]$

Каждая из полученных маркировок порождает новую в результате чего получаем дерево маркировок.

В дереве маркировок могут встречаться повторяющиеся маркировки, в этом случае дальнейшее построения дерева ведется только для 1 из них

Если выделить путь по дугам графа от  $\mu_0$  до  $\mu''$  и выписать подряд все символы переходов, то полученное слово образует последовательность срабатывания.

А их совокупность – язык сети Петри

Он начинается со специального символа  $\lambda$ (пустое)

Для примера  $\lambda, t_1, t_1 t_2, \dots$

## 25 Свойства сетей Петри

1. свойство ограниченности

Если для любой достижимой сети маркировки  $\mu$  существует такое  $k(\mu_i \leq k)$  то сеть называется ограниченной, если все позиции ограничены

Пр – сеть неограничена, т.к. возможен неограниченный рост  $\mu_2$

Выделяется маркировка  $\mu' = [2, \omega, 0]$

2. свойство безопасности

сеть называется безопасной, если при любой достижимой маркировке  $\mu_i \leq 1$  для всех  $i = \overline{1, n}$

Т.е. вектор маркировок состоит из 0 и 1

3. свойство консервативности

сеть называется консервативной если сумма фишек во всех позициях постоянна при работе сети

4. свойство живости

переход  $t_j$  называется живым, если существует достижимая из  $\mu_0$  такая то маркировка  $\mu'$ , при которой переход  $t_j$  может сработать

Если  $t_j$  – потенциально живое при любой достижимой в сети маркировке, то он называется живым

Переход  $t_j$  не является потенциально живым при начальной маркировке  $\mu_0$ , называется мертвым. Маркировка  $\mu_0$  в этом случае называется тупиковой для перехода  $t_j$

Если маркировка  $\mu_0$  – тупиковая для всех  $t_j$  то маркировка  $\mu_0$  – тупиковая

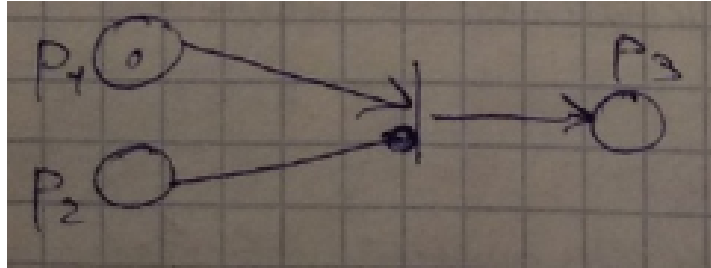
Переход является устойчивым если никакой другой переход не может лишить его возможности сработать при наличии для этого необходимых условий

## 26 Ингибиторные сети Петри

Это сети Петри, для которых функция инцидентности имеет вид  $F = F^p \sup F^t \sup F^t$  т.е. она дополнена специальной функцией инцидентности

$$F' = P \times T \rightarrow \{0, 1\}$$

Она вводит ингибиторные дуги для тех пар  $(p_i, t_j)$  где кратность таких дуг – всегда 1 т.е. позиция соедин. дугой, не должна содержать фишек

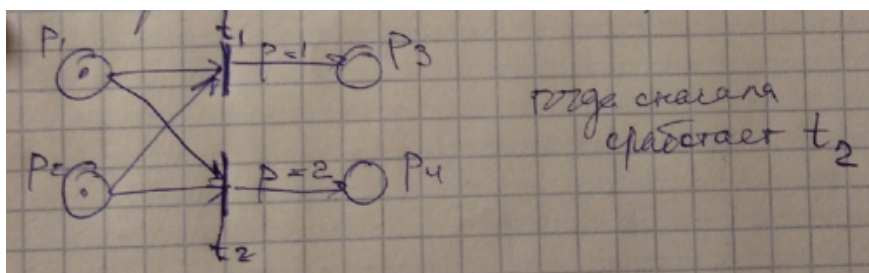


## 27 Сети с приоритетами

Характер работы сети Петри недетерминированный т.е. если имеется возможность срабатывания нескольких переходов, то может сработать любой

Необходимо регламентировать последовательность срабатывания для этого введя понятие множества приоритетов и приписав каждому переходу соответствующее целочисленное значение

В этом случае правила срабатывания модифицируются: если на такте сети Петри могут сработать несколько переходов, то срабатывает тот из них, который имеет наивысший приоритет

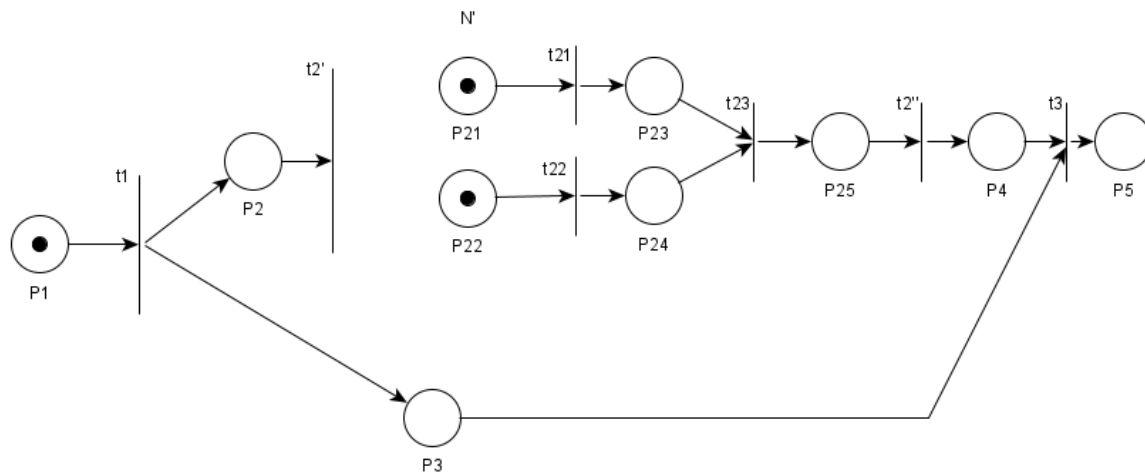


## 28 Иерархические сети Петри

Представляют собой многоуровневые структуры внутри которых присутствуют подсети различных уровней. Отсюда это замечательный инструмент формализации многоуровневых иерархических систем. В отличие от обыкновенных сетей Петри в иерархических имеется 2 типа перехода: простые и составные.

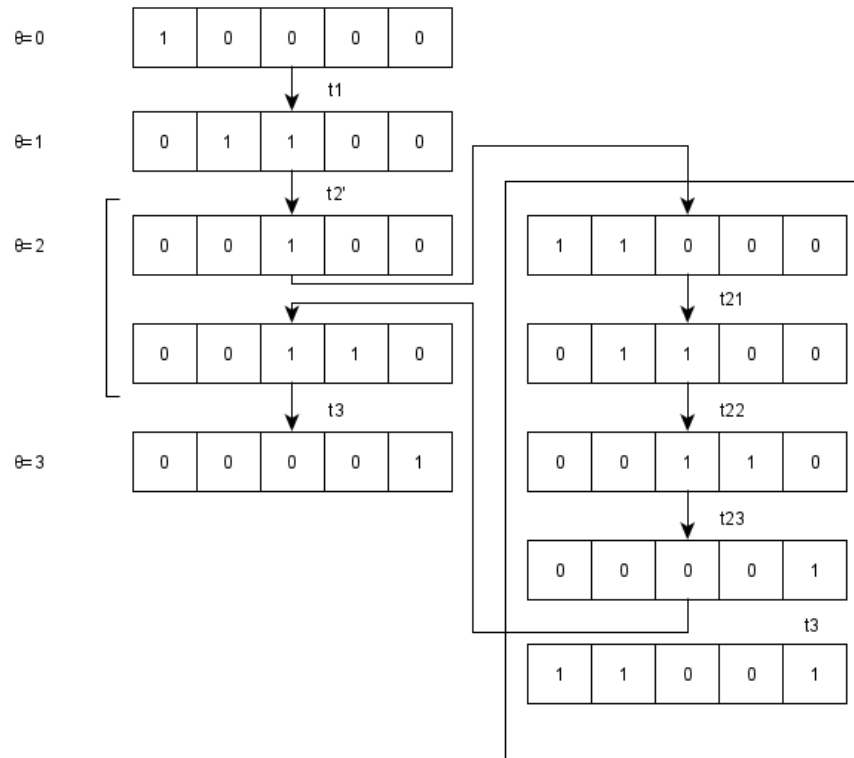
Простой переход ничем не отличается от того что мы рассматривали ранее. Составной переход содержит внутри себя сеть Петри более низкого иерархического уровня. Формально они состоят из входного (головного) и выходного (хвостового) перехода. Именно между ними находится сеть. Причем уровень вложенности иерархии может быть до  $n$  вложенности.

Пример иерархической сети



В данной сети имеется составной переход  $t_2$  содержащий сеть  $N'$ . В этой сети  $t_2'$  голова  $t_2''$  хвост. То что между это сеть состоящая из  $p_{21}$ - $p_{25}$ .

Срабатывание составных переходов является не мгновенным событием а составным действием. Поэтому мы говорим не о срабатывании перехода а его работе. На каждом шаге дискретного времени  $tetta$  составной переход может находиться в одном из двух состояний. В пассивном и активном. Начальное состояние всех переходов пассивное. Составной переход может стать активным в некоторые моменты времени  $tetta$  если до этого он был пассивен и имеются условия для срабатывания головного перехода. При этом производится изменение маркировки в сети верхнего уровня по обычным правилам. И одновременно запускается работа сети находящаяся внутри перехода. Как только запущена подсеть работа всей сети блокируется. Сеть нижнего уровня работает с учетом своей начальной маркировки до тех пор пока все ее переходы не станут пассивными после этого работает хвостовой переход и происходит изменение маркировки сети верхнего уровня. Составной переход принимает свое начальное пассивное состояние, а в сети нижнего уровня восстанавливается начальная маркировка.



На шаге  $tetta = 2$  происходит выполнение составного перехода в следующем порядке: срабатывает  $t_2'$  ... после чего окончательно срабатывает  $t_2$  и продолжение работы всей сети.

## 29 Цветные сети Петри CPN

CPN – функционирует через мультимножество.  $CPN = (tetta, sum, P, T, A, N, C, G, E, I)$

Цветная сеть Петри отличается от обычной только тем что мы можем накидать больше свойств чем в обыкновенной сети петри. Tetta – множество дискретных моментов времени, в которых происходит функционирование сети. Sum – конечное множество называемое «цветами»,  $P$  – конечное множество позиций,  $T$  – конечное множество переходов,  $A$  – конечное множество дуг связывающих между собой позиции и переходы. В отличие от обыкновенных сетей петри, где дуги задаются матрицами инцидентности, здесь дуги указываются с помощью множественного выражения. Такая нотация введена здесь для того чтобы можно было пару элементов соединить несколькими дугами, с различными свойствами.  $NN(A)$  узловая функция которая для каждой дуги указывает ее исходный и конечный узел. (дуга может быть указана !записью!)  $C$  – цветовая функция определяющая множество типа цветов разрешенных для каждой позиции.  $G$  – блокировочная или спусковая функция, описывающая дополнительные условия которые должны быть выполнены для срабатывания перехода  $t$ . Эта функция представляет собой предикат составленный из переменных принадлежащих к типу цветов.  $E$  – функция задающая выражения на дугах, причем для каждой дуги она форма-

лизована в виде мультимножества, состоящих из элементов описанных в множестве цветов.  $I$  – функция инициализации сети CPN.