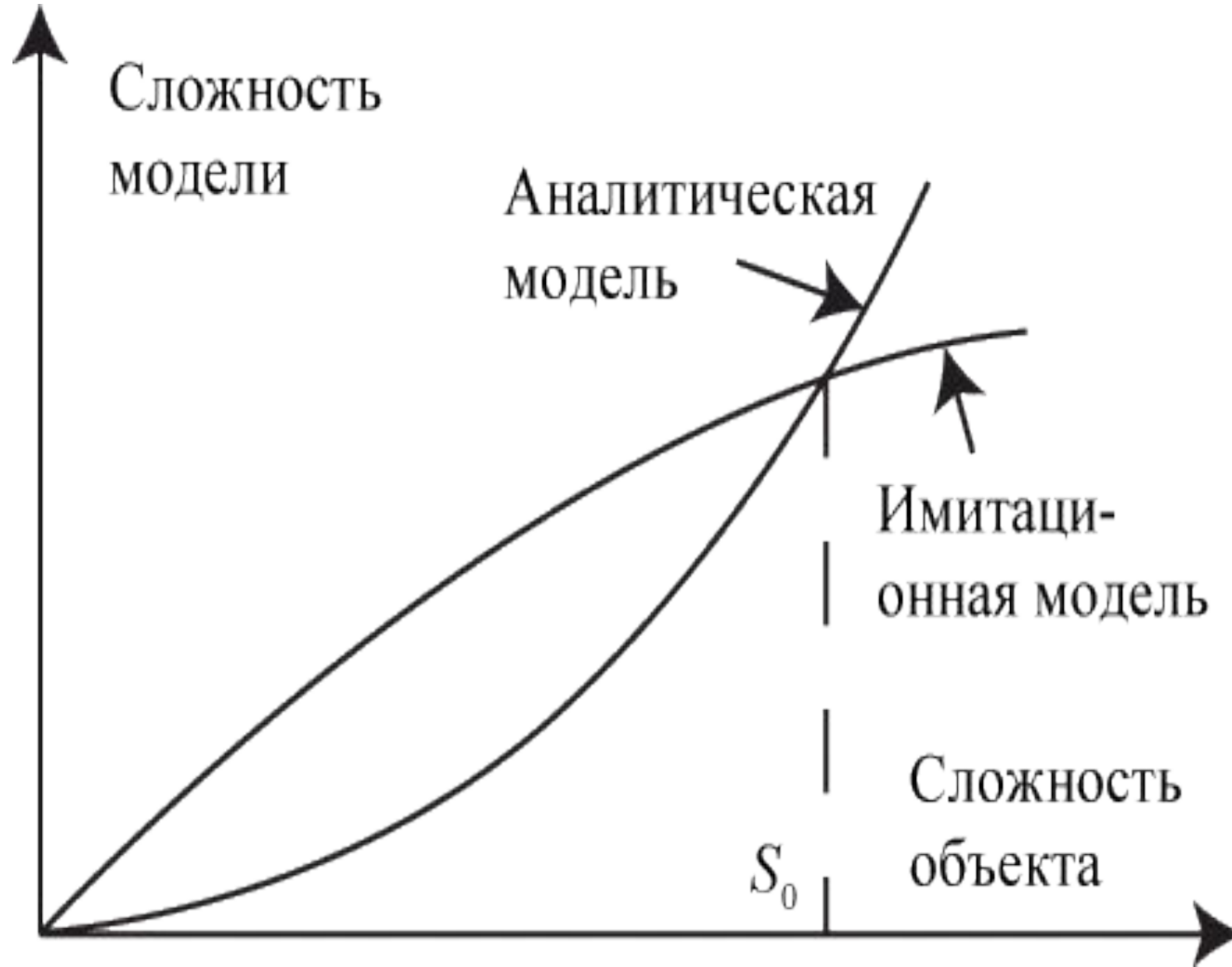


# моделирование



# МОДЕЛИ

Аналитическая модель описывается на языке математики.

Имитационная модель (ИМ) в подавляющем числе случаев – на языке программирования (ИМ – это программа, эмулирующая объект моделирования).

Языки программирования делятся на:

1. Языки общего назначения
2. Языки имитационного моделирования (языки, специализированные на описание ИМ)

# Два основных класса языков ИМ

Языки ИМ делятся на два основных класса:

1. Для имитации непрерывных процессов
2. Для имитации дискретно-событийных процессов

Т.к. в настоящее время Аналоговый вычислительные машины практически не применяются, для непрерывного моделирования используются численные методы (например, разностные методы). Для удобства описания непрерывных моделей существуют языки, описывающие модель в виде аналоговых элементов: усилителей, интеграторов, генераторов функций и т. п. Описанная с помощью таких языков модель представляет собой систему дифференциальных уравнений и моделирование сводится к решению этой системы. Например, в языке DYNAMO описываются значения функции в моменты времени  $t$  и  $t+dt$ , а также разности производных функции в моменты времени  $t$  и  $t+dt/2$ .

# Особенности языков имитационного моделирования

Языки ИМ специализированные, поэтому они не всегда могут подойти для решения определенной задачи. Следовательно, сначала необходимо сформулировать задачу, а затем выбирать определенный язык ИМ.

Общий язык программирования можно использовать в том случае, когда для реализации ИМ не рационален ни один из языков ИМ.

# Время в имитационной модели

При имитационном моделировании применяется **модельное время**. Это время может не иметь связи со временем реальным. Зачастую модельное время необходимо для того, чтобы сравнивать характеристики различных объектов (например, сравнить производительность двух архитектур вычислительной системы).

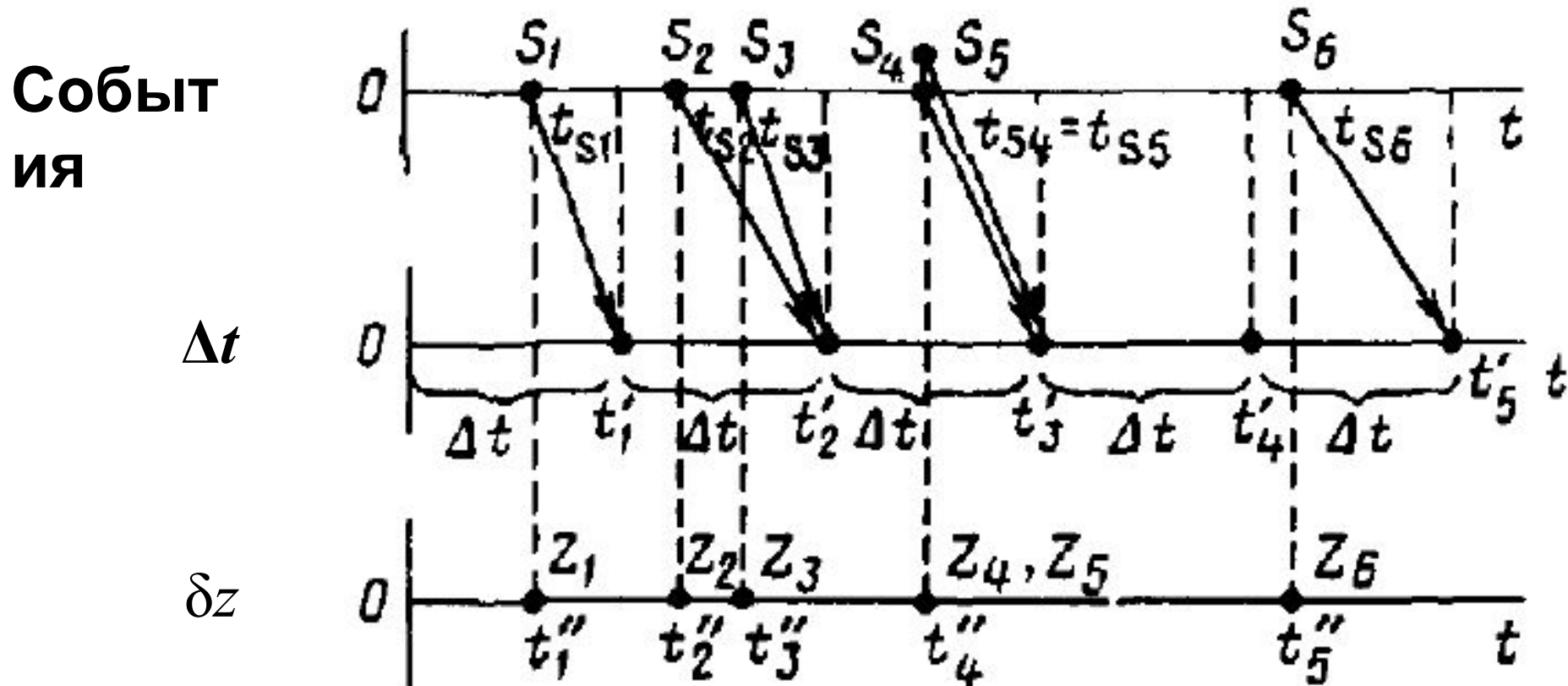
Наиболее важные задачи при создании имитационной модели и выборе языка программирования для нее:

1. корректировка временной координаты состояния системы «продвижение» времени, организация «часов»);
2. обеспечение согласованности различных блоков и событий в системе (синхронизация во времени, координация с другими блоками).

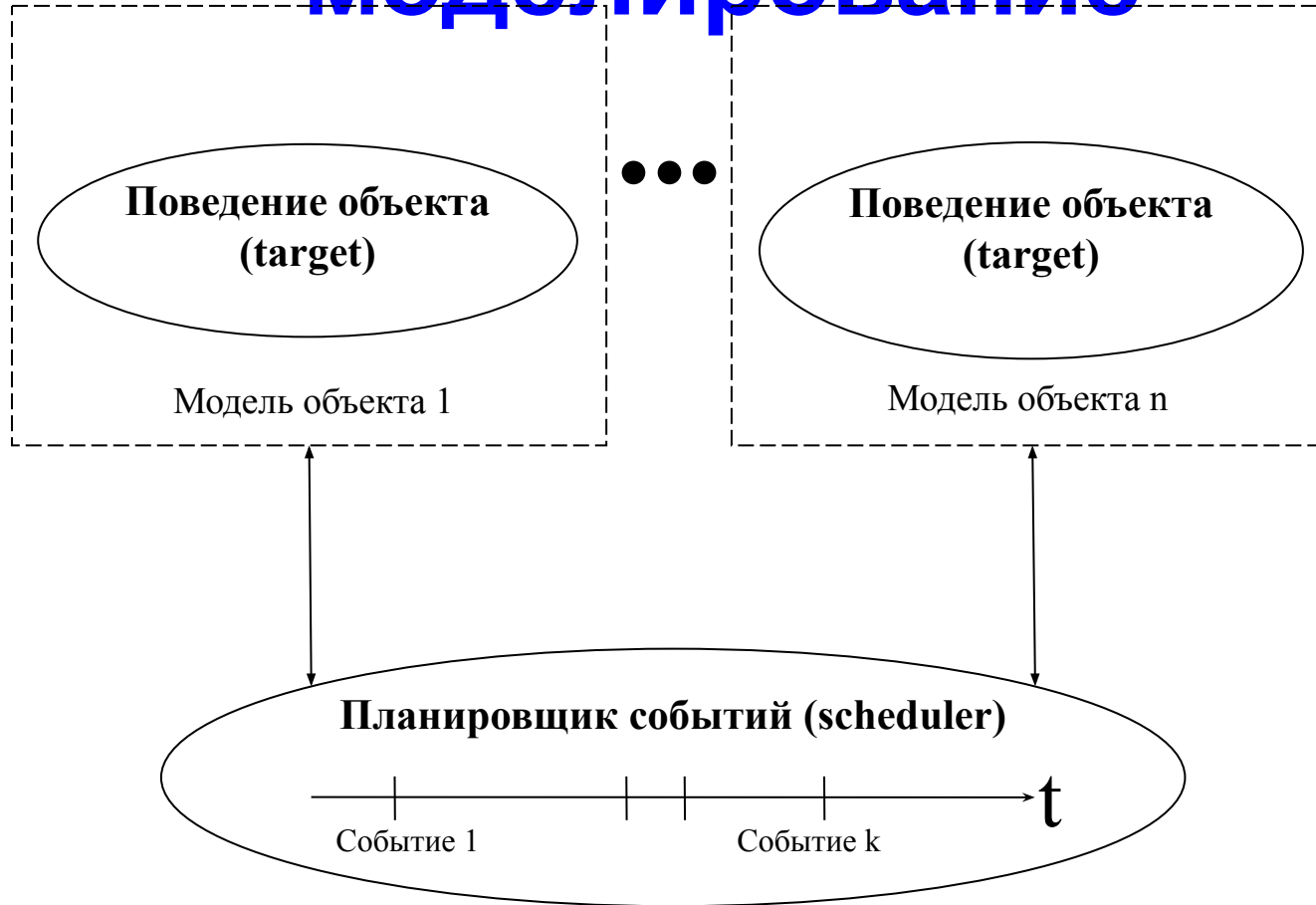
# МОДЕЛИ

Существует два подхода к организации модельного времени:

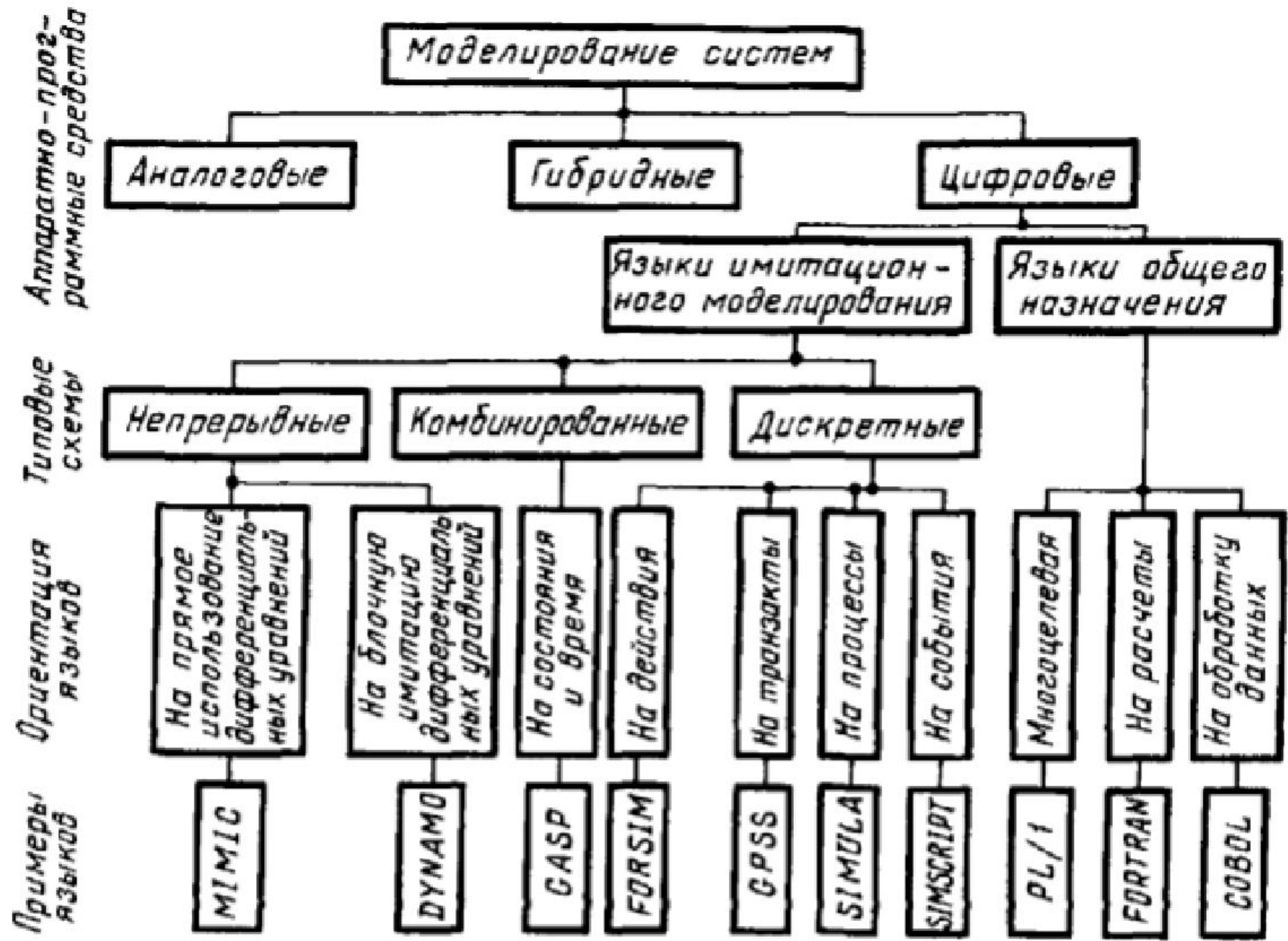
1. С постоянным шагом (стратегия  $\Delta t$ )
2. С переменным шагом – дискретно-событийное (стратегия  $\delta z$ )



# Дискретно-событийное моделирование



В дискретно-событийных системах моделирования контроль за событиями осуществляет программный блок «Планировщик событий»(scheduler). Поведение отдельных объектов задается с помощью программного блока target. (свой target на каждый объект).





# МОДЕЛИРОВАНИЯ

**Имитационное моделирование (данные этапы выделяли в своих работах Р. Нэнси и Ф. Кивиат):**

**Этап 1** (1955-1960) – языки программирования общего назначения.

**Этап 2** (1961-1965) – первые специализированные языки моделирования: GPSS, SIMSCRIPT, SIMULA, CSL, SOL.

**Этап 3** (1965-1970) – второе поколение языков моделирования: GPSS V, SIMSCRIPT II.5, SIMULA 67.

**Этап 4** (1971-1978) – повышение эффективности языков моделирования .

**Этап 5** (1979-1984) – интегрированные средства имитационного моделирования, поддерживающие полный жизненный цикл модели от создания до обработки результатов запусков модели. Появляется сбор и обработка статистики результатов моделирования

**Этап 6** (1985-1994) - перенос программного обеспечения для имитационного моделирования на персональные ЭВМ с использованием средств графического интерфейса. Развитие графического моделирования.

**Этап 7** (1995-1998) – появление средств технологической поддержки процессов распределенного имитационного моделирования на мультипроцессорных ЭВМ и сетях.

**Этап 8** (1999 - ...) – Развитие и усовершенствование разработанных средств моделирования. Виртуальная реальность (в последнее время данная технология значительно подешевела).

# System

Система GPSS была разработана сотрудником фирмы IBM Джеффри Гордоном в 1961 году.

Программный продукт осуществляет моделирование согласно Q-схеме

В 1980-х годах интерес к данному программному продукту значительно угас на фоне появления объектно-ориентированного программирования. К тому же GPSS не имел средств графической интерпретации, что значительно снижало удобство работы с системой.

# Основные объекты GPSS

Транзакт – обслуживаемое требование;

Блок – описание обслуживающего устройства;

Канал обслуживания;

Функция

Очередь

Накопитель

Генератор случайных чисел

# Основные блоки GPSS

GENERATE A, B, C, D, E – генерация транзакта (могут задаваться закон распределения случайного процесса, количество генерируемых транзактов, время начала генерации и т. д.);

TERMINATE A – уничтожение транзакта (A – на сколько уменьшить счетчик транзактов);

GENERATE-TERMINATE – сегмент (в модели может быть несколько сегментов).

Пример:

```
GENERATE70,20
```

```
TERMINATE 1
```

```
START1000
```

ADVANCE A,B – задержка (время) обслуживания транзакта. В блоке может находиться произвольное количество транзактов.

```
GENERATE70,20
```

```
ADVANCE50,30
```

```
TERMINATE
```

# Основные блоки GPSS

**SEIZE** (занять) и **RELEASE** (освободить) – блоки для моделирования ограниченных СМО. (A – номер или имя устройства). Если транзакт вошел в блок, то он становится занятым и не пускает в себя другие транзакты.

GENERATE70,20

ADVANCE0,0

SEIZEPRIBOR

ADVANC50,30

RELEASEPRIBOR

TERMINATE

# Основные блоки GPSS

**STORAGE A** – многоканальное устройство обслуживания (A – количество каналов);  
**ENTER A,B – LEAVE A,B** – операторы «войти» и «выйти» для многоканального у-ва;

```
MKS3 STORAGE    3  
GENERATE 70,20  
ADVANCE  0,0  
ENTER MKS3  
ADVANCE 50,30  
LEAVE MKS3  
TERMINATE
```

**PREEMPTA – RETURN** – блок захватить устройства обслуживания (захват устройства означает, что приостанавливается обслуживание транзакта и начинается обслуживание другого, который имеет больший приоритет).

# Основные блоки GPSS

**QUEUE A,B - DEPART A,B** – очередь ожидания транзактов. («встать в очередь», «выйти из очереди»)

GENERATE 70,20

QUEUE OCH

SEIZE PRIBOR

DEPART OCH

ADVANCE 50,30

RELEASE PRIBOR

TERMINATE

# Основные блоки GPSS

**TRANSFER (передать)** – изменение маршрута движения транзакта.

Обычно транзакт движется от одного блока программы к другому. Оператор TRANSFER изменяет этот порядок в зависимости от определенных условий.

TRANSFER имеет 9 режимов работы:

- Безусловная передача.
- Случайная передача.
- Логическая передача – указывается два альтернативных блока (если в один из них транзакту не удастся войти, он передается во второй).



# Основные блоки GPSS

**FUNCTION A,B.** – задание распределения случайной величины с помощью точек, через которые она проходит.

Функция, заданная с помощью 24 точек:

```
EXPDIS FUNCTIONRN1,C24
```

```
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38/.8,1.6/.84,1.85  
.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/.97,3.5/.98,3.9/.99,4.6  
.995,5.3/.998,6.2/.999,7/.9998,8
```

**START A, B, C, D** – начать моделирование (A – количество генерируемых транзактов)

**RESET** – сбросить накопленную статистику прогона модели

**CLEAR** – сброс модели

# System

## Пример программы на GPSS

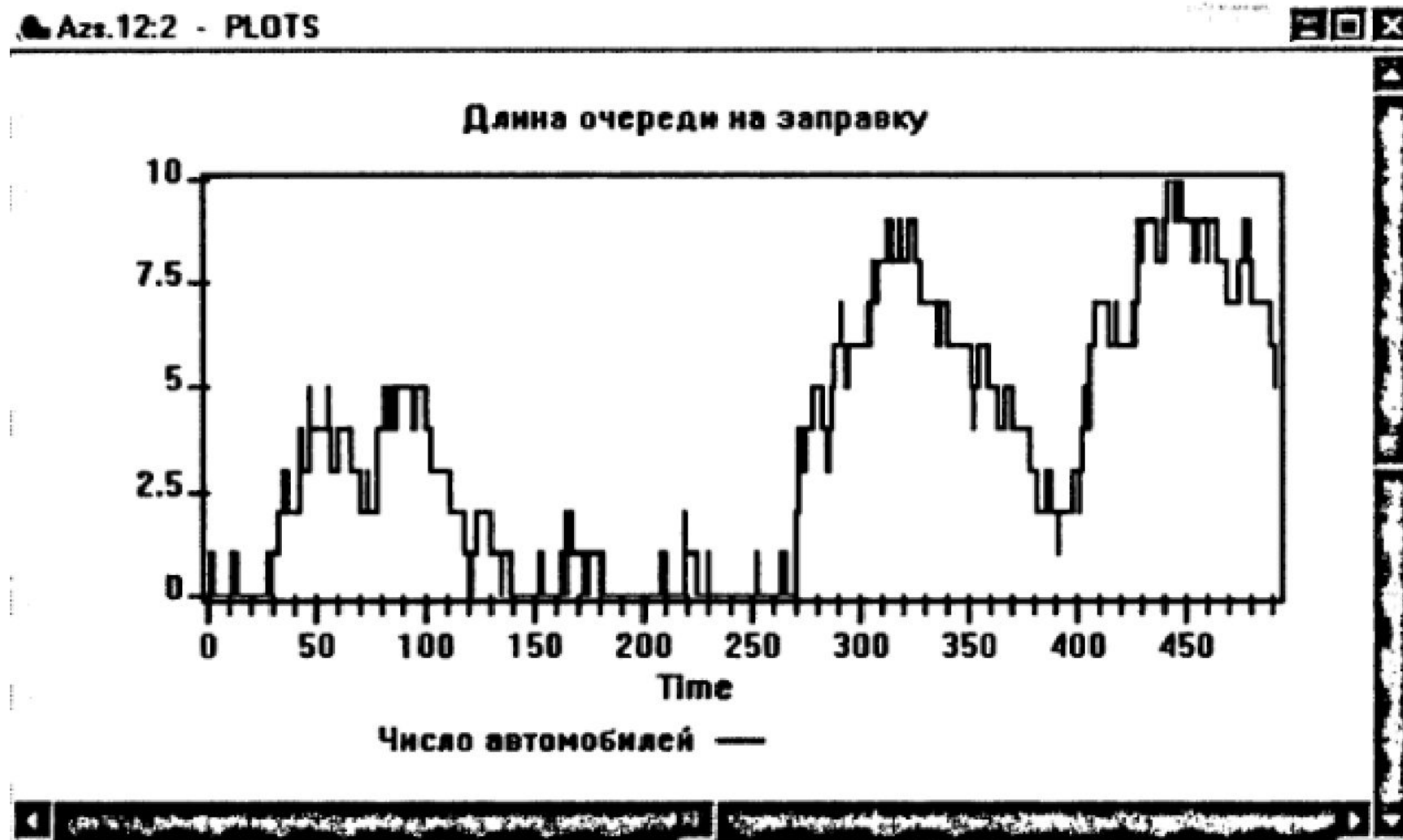
```
GENERATE (POISSON(1,40)) ; генерация потока транзактов  
                ; Поток транзактов пуассоновский со средним интервалом 40 единиц  
QUEUE mainQ ; Войти в регистратор очереди SEIZE F007  
                ; Попытка занять устройство DEPART mainQ ; Покинуть регистратор очереди  
ADVANCE (Normal(1,35,4)) ; Моделирование процесса обслуживания. Время обслуживания  
                ; распределено по нормальному (гауссову) закону  
                ; с математическим ожиданием 35 и среднеквадратическим отклонением 4  
RELEASE F007 ; Освободить устройство  
TERMINATE ; удаление транзакта
```

# заправку

```
Azs : GPSSU AZS.GPS|
*****
* Моделирование автомобильной *
* заправочной станции (АЗС) *
*****
      GENERATE (Exponential(1,0,6.5)) ; Поступление автомобилей на заправку.
      QUEUE Zapravka ; Вход автомобиля в очередь на заправку.
      TRANSFER Both,Kol_1,Kol_2 ; Переход автомобиля к свободной колонке.
Kol_1 SEIZE Kolonka_1 ; Определение занятости 1 колонки.
      DEPART Zapravka ; Выход автомобиля из очереди на заправку.
      ADVANCE 10,2.5 ; Заправка автомобиля на 1 колонке.
      RELEASE Kolonka_1 ; Освобождение 1 колонки.
      TRANSFER ,Next ; Переход к оператору с меткой Next.
Kol_2 SEIZE Kolonka_2 ; Определение занятости 2 колонки.
      DEPART Zapravka ; Выход автомобиля из очереди на заправку.
      ADVANCE 13,4 ; Заправка автомобиля на 2 колонке.
      RELEASE Kolonka_2 ; Освобождение 2 колонки.
Next SAVEVALUE Ave_Queue,QT$Zapravka ; Сохранение времени пребывания в очереди.
      TERMINATE ; Выход автомобиля из системы.
      GENERATE 480 ; Моделирование работы АЗС в течение смены.
      TERMINATE 1
      START 1
```

# System

График длины очереди на заправку



# Литература

1. Советов Б. Я., Яковлев С. А. С 56 Моделирование систем: Учеб. для вузов — 3-е изд., перераб. и доп. — М.: Высш. шк., 2001. — 343 с.
2. Поляков Н.А. История имитационного моделирования.  
URL:  
<http://simulation.su/uploads/files/default/obzor-polyakov-1.pdf>