

Моделирование систем и языки моделирования.

Алгоритмические языки при моделировании систем служат вспомогательным аппаратом в разработке машинной реализации и анализа характеристик моделей. *Основная задача* – это выбор языка.

Каждый язык имеет свою систему абстракций, лежащую в основе формализации функционирования сложных систем.

Для программирования модели могут использоваться следующие языки:

1. Универсальные алгоритмические языки высокого уровня.
2. Специализированные языки моделирования: языки, реализующие событийный подход, подход сканирования активностей, языки, реализующие процессно-ориентированный подход.
3. Проблемно-ориентированные языки и системы моделирования.

Качество языков моделирования характеризуется:

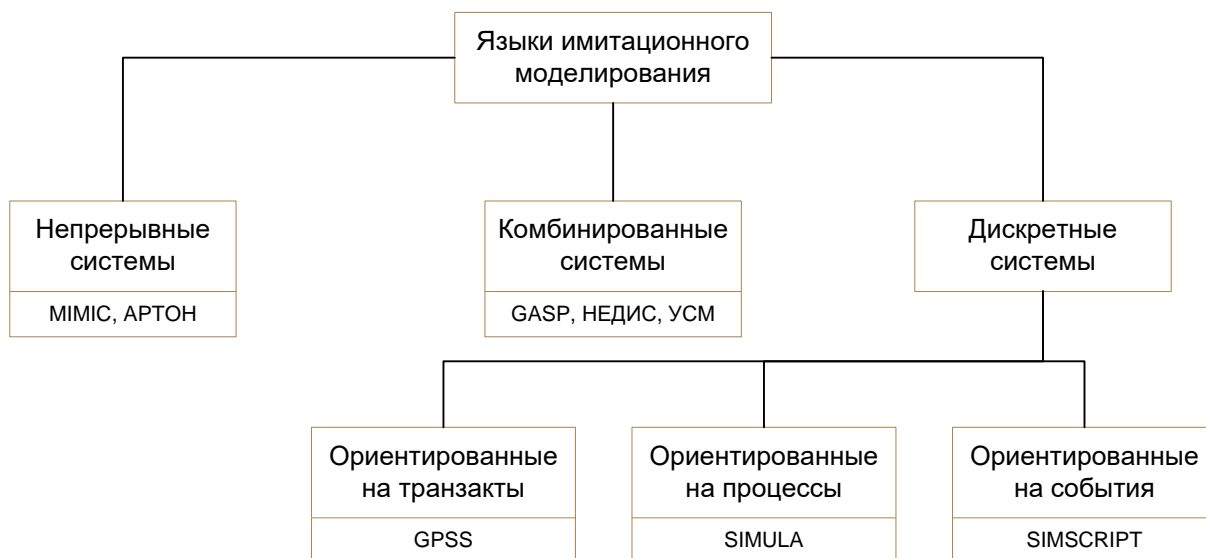
1. Удобство описания процесса функционирования;
2. Удобство ввода исходных данных, варьирования структуры, алгоритмов работы и параметров модели;
3. Эффективностью анализа и вывода результатов моделирования;
4. Простотой отладки и контроля работы моделирующей программы;
5. Доступностью восприятия и использования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

В большинстве своем языки моделирования определяют поведение систем во времени с помощью модифицированного событийного алгоритма. Как правило, он включает в себя список текущих и будущих событий.

//здесь будет схема из мобилы.

Классификация языков имитационного моделирования.

Основа классификации – принцип формирования системного времени.



Непрерывное представление систем сводится к представлению дифференциальных уравнений, с помощью которых устанавливают связь между входной и выходной

функциям. Если выходные переменные модели принимают дискретные значения, то уравнения являются разностными.

GASP – комбинированный, в основе лежит язык FORTRAN. Предполагается, что в системе могут наступать события двух типов:

- события, зависящие от состояния
- события, зависящие от времени.

Состояние системы описывается набором переменных, причем некоторые из них меняются непрерывно. При таком подходе пользователь должен составлять процедуры, описывающие условия наступления событий. Законы изменения непрерывных переменных, правила перехода от одного состояния к другому, т.е. реализуется классический принцип ДУ.

Группы языков моделирования, ориентированные на дискретное время, используются при построении именно имитационных моделей, но при этом используются разные способы описания динамического поведения исследуемого объекта.

Формальное описание динамики моделируемого объекта.

Будем считать, что любая работа в системе совершается путем выполнения **активностей**. Т.е. **активность** является наименьшей единицей работы и её рассматривают как единый дискретный шаг. Следовательно, активность является, единым динамическим объектом, указывающим на совершение единицы работ. **Процесс** – это логически связанный набор активностей.

Пример: активность установки головки жесткого диска, активность передачи информации с жесткого диска.

Активности проявляются в результате свершения событий. **События** – это мгновенное изменение состояния некоторого объекта системы. Рассмотренные объекты (активности, процессы, события) являются конструктивными элементами для динамического описания поведения системы. На их основе строятся языки моделирования системы. 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 системы, достаточно описать поведение каждого класса процессов и задать значение атрибутов для конкретных процессов.

Задачи построения модели.

Построение модели состоит из решения двух основных задач:

1. Первая задача сводится к тому, чтобы описать правила, описывающие виды процессов, происходящих в системе.
2. Вторая задача заключается в том, чтобы описать правила задания атрибутов или задать правила генерации этих значений. При этом система определяется на конкретном уровне детализации в терминах множества описаний процессов, каждый из которых в свою очередь включает множество правил и условий возбуждений активности. Такое описание системы может быть детализировано на более подробном или более иерархическом уровне представления с помощью декомпозиции процессов (в идеальном случае в активности). Это обеспечивает многоуровневое исследование системы.

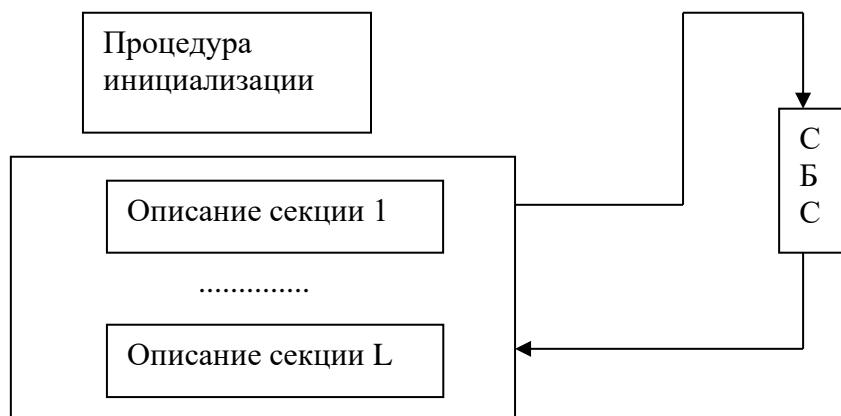
Т.к. система в общем случае служит для описания временного поведения, то язык моделирования дискретных систем должен обладать средствами, отображающими время.

В реальной системе совместно выполняются несколько активностей, принадлежащим как связанным, так и не связанным процессам. Имитация их действий должна быть строго последовательной. Таким образом, модель системы можно рассматривать как модель описаний, активностей, событий или процессов. Отсюда и деление языков моделирования.

Языки, ориентированные на события.

Моделирующая программа организована в совокупности в виде секций событий (процедуры отражающие события). Процедура состоит из набора операций, которые выполняются после выполнения какой-либо активности. Синхронизация происходит с помощью списка будущих событий.

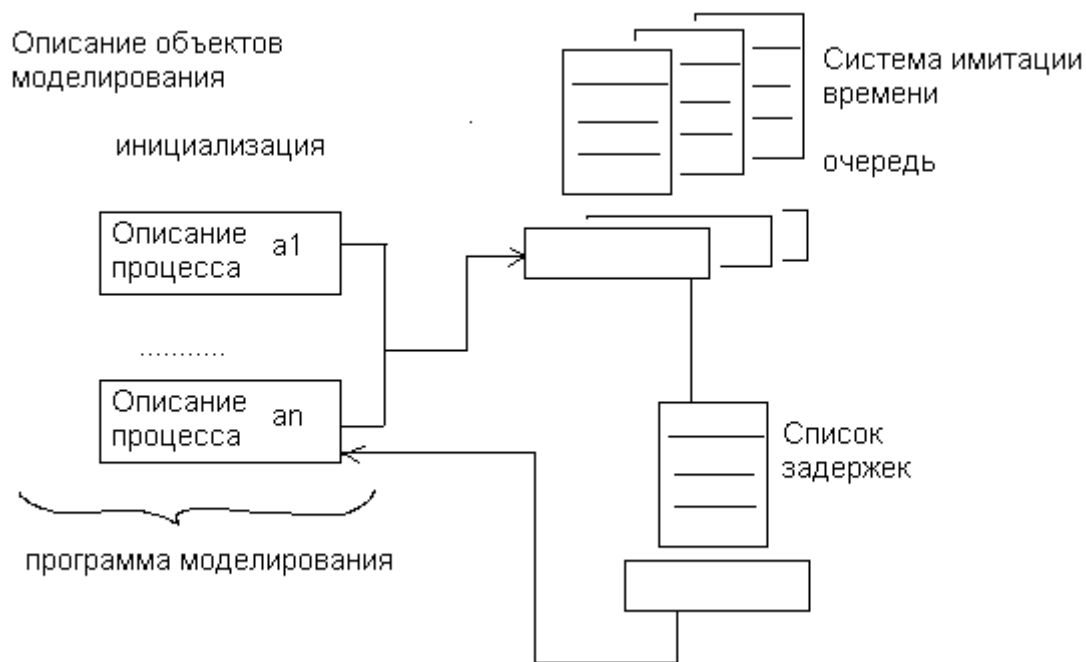
Структуру рассмотрим на примере языка SIMSCRIPT.



Языки, ориентированные на процессы.

Моделирующая программа организуется в виде набора описаний процесса. Каждый из которых описывает один класс. Описание процесса функционирования устанавливает атрибуты и активности всех процессов. Синхронизация операций во времени реализуются так же с помощью списка будущих событий, который содержит точку возобновления конкретного процесса (точка прерывания).

На примере языка SIMULA:



Результаты экспертных оценок сравнения различных языков при моделировании большого класса систем.

Критерии:

1. Возможность языка. Выше всех находится SIMULA -> SIMSCRIPT -> GPSS -> C -> PASCAL
2. Простота применения: GPSS -> SIMSCRIPT -> SIMULA -> C -> PASCAL
3. Предпочтение пользователей: GPSS -> SIMSCRIPT -> SIMULA -> PASCAL -> C

При использовании универсальных алгоритмических языков программист имеет практически неограниченные возможности по созданию имитационной модели, наилучшим образом используя ресурсы системы, особенности ОС и т.д. Но в то же самое время требует больших трудозатрат, работы программистов высокой квалификации, взаимодействие специалистов разных профилей (программисты, эксперты предметной области и т.д.). В результате модель получается узко направленной и, как правило, использование этой модели в других разработках невозможно.

При использовании языков имитационного программирования снижается гибкость и универсальность. Модель создается в несколько раз быстрее и не требует присутствия системы от программистов. Модели обладают свойством концептуальной выразительности. Используются специальные термины языка в области, исследуемой задачи.

Алгоритм сканирования активности включает в себя цикл по всем действиям активностей, дальше срабатывает проверка выполнения действия. И если условие выполняется, то выполняется действие, изменяющее состояние модели. Иначе цикл по всем действиям заканчивается.

Сравнение универсальных и специализированных языков программирования при моделировании:

Преимущества	Недостатки
Универсальные	
Минимум ограничений на выходной формат	Значительное время, затрачиваемое на программирование
Широкое распространение	Значительное время, затрачиваемое на отладку
Специализированные	
Меньше затраты времени на программирование	Необходимость точно придерживаться ограничений на форматы данных
Более эффективные методы выявления ошибок	Меньшая гибкость модели
Краткость, точность понятий, характеризующих имитируемые конструкции	
Возможность заранее строить стандартные блоки, которые могут использоваться в любой имитационной модели	
Автоматическое формирование определенных типов данных, необходимых именно в процессе имитационного моделирования	
Удобство накопления и представления выходной информации	
Эффективное использование ресурсов	

Основные концепции языка РДО (Ресурсы, действия, операции).

Язык придуман в МГТУ.

Причина создания РДО:

- Универсальность имитационного моделирования относительно класса исследуемых систем и процессов.
- Легкости модификации моделей
- Моделирование сложных систем управления совместно с управляемым объектом (включая использования имитационного моделирования в режиме реального времени)

Язык РДО является реализацией т.н. интеллектуального подхода к имитационному моделированию. При этом подходе стараемся отойти от жесткого алгоритмического подхода в процессе принятия решения при моделировании для удобства автоматизирования той части процесса, где используются знания человека, и сделать процесс моделирования максимально гибким по способам представления информации об объекте. В основе языка РДО лежит продукционная система, которая состоит из трех элементов: классов и отношений, правил, управляющей структуры.

Классы и отношения трактуются как база данных, содержащая декларативные знания. Процедуры представляют собой набор модифицированных продукционных правил: ЕСЛИ – ТО ДЕЙСТВИЕ.

Управляющая структура – это интерпретатор правил, управляющий выборкой правил.

Условие – это проверка состояние базы данных.

Действие – изменяет некоторым образом базу данных.

Достоинства системы основанной на продукциях:

- Простота создания и понимания, отдельных правил
- Легкость модификации

Недостатки:

- Неясность взаимных отношений правил
- Сложность оценки целостного образа знаний
- Низкая эффективность обработки

Для имитационного моделирования **основным недостатком системы** продукции является отсутствие времени, поэтому такие продукционные правила применимы для анализа статических объектов, т.е. правила описывают причинно следственные связи и не описывают динамику процесса. Поэтому при интеллектуально моделировании, т.е. для моделирования динамики системы, на основе знаний о протекающих процессах в РДО используют модифицированное продукционное правило.

Как оно выглядит:

ЕЛСИ <УСЛОВИЕ>, ТО <СОБЫТИЕ 1> ... ЖДАТЬ (отведенный интервал) ... ТО <СОБЫТИЕ 2>

СОБЫТИЕ 2 наступает через некоторый интервал времени.

Если в течение указанного времени происходит нерегулярное событие, затрагивающее релевантное данному событию ресурсы, то событие 2 может не наступить или наступить в другой момент времени.

В РДО сложная дискретная система представляется в виде множества взаимодействующих между собой ресурсов.

Ресурс – это элемент сложной системы, внутренней структурой которого можно пренебречь, в то время как наличие и свойства его, как целого, важны и существенны для описания.

Каждый ресурс модели должен быть описан и должен иметь свое уникальное имя.

Ресурсы могут быть двух типов:

1. Постоянные. Они всегда присутствуют в системе.
2. Временные. Поступают в систему и покидают её в процессе функционирования. Причем они могут быть и результатом работы, т.е. здесь наличие обратной связи.

Все ресурсы системы образуют некоторое множество: $R(t) = \{r_i: i = 1, \dots, N(t)\}$

Где R_i – i -ый ресурс сложной дискретной системы

$N(t)$ – число ресурсов в текущий момент времени.

Каждый ресурс описывается множеством параметров, которые могут быть следующих типов:

1. **Описательные**, представляющие факты внутренне присущие каждому ресурсу.
2. **Указывающие**, используемые для дачи имени или обозначения ресурса, проще говоря, идентификаторы.

3. **Вспомогательные**, используемые для связи различных ресурсов, накопления статистики, графического вывода при имитации и т.д.

Состояние ресурса в момента времени t : $C_i(t) = \{C_{ij}(t): j = 1 \dots M_i\}$

Где C_{ij} – значение j -ого параметра i -ого ресурса

M_i – число параметров i -ого ресурса

Состояние всей системы является совокупностью состояний всех ресурсов.

Ресурсы, принадлежащие к одному типу, наследуют общие свойства этого типа. Отношения наследования может использоваться как для отображения общности ресурсов, так и для идентификации ассоциативных связей.

Ресурсы взаимодействуют друг с другом в соответствии с определенным алгоритмом. Каждое действие связано с изменением состояния системы, которое связано с конкретным событием. Все события должны быть определены и зафиксированы в модели. Они могут быть *внешними* и *внутренними* по отношению к системе.

В РДО событие происходит в счетные моменты времени, которые фиксируются в модели с помощью независимой переменной. Эта переменная изменяется дискретно и служит базой для определения различий в наблюдении одного и того же свойства. Все события делятся на *регулярные* и *нерегулярные*.

Регулярные – это события, вызываемые штатным функционированием ресурсов. Они выражают логику взаимодействия ресурсов между собой.

Нерегулярные события происходят либо при нештатной работе (поломка или отказ) ресурса, либо из-за внешних по отношению к системе причин, т.е. в систему пришел новый временный ресурс.

В отличие от регулярных событий, нерегулярные носят стохастический характер.

Ресурсы в процессе функционирования выполняют определенные действия. С каждым действием связано два события:

1. Время начала
2. Время окончания

Действие представляет собой целенаправленное мероприятие, выполняемое под управлением некоторой подсистемы и направленное на достижение определенной цели. Поэтому действие планируется и может находиться в следующих состояниях:

- Запланировано
- Начато
- Окончено
- Прервано по какой-либо причине

Действие с нулевой длительностью представляет собой *событие*.

Процесс функционирования сложной дискретной системы строится как временная последовательность действий и нерегулярных событий. Действие может начинаться только в том случае, если значение параметров его релевантных ресурсов удовлетворяет необходимому условию.

В языке используется понятие **виртуального действия**, которое не привязано ни к началу, ни к окончанию, а привязано только к необходимому условию начала действия.

Множество виртуальных действий может быть разбито на небольшое число подмножеств, имеющих одинаковую природу, т.е. одинаковую логику взаимодействия ресурсов и

различаются лишь конкретно участвующими в них ресурсами. Для формального описания логики однотипных виртуальных действий используются понятия *операций*.

Операция – это фактически процедура, у которой формальные параметры условия выполнения и алгоритмы, а в качестве фактических параметров выступают подмножества ресурсов. При задании фактических ресурсов элементами соответствующих подмножеств однотипных ресурсов из операции получается виртуальное действие. Т.е. операция отражает логику взаимодействия ресурсов. Всякий раз, когда состояние соответствует условию срабатывания виртуального действия, происходит действие описывающее соответствующей операции с различными временами начала и окончания.

Следовательно, *операция* описывает, как происходит действие или виртуальное действие и с какими множествами релевантных ресурсов. Т.е. что может произойти в сложной системе при определенных условиях. А *действие* – что произошло, происходит, произойдет и в какое время.

Представление сложной дискретной системы в РДО методе.



СДС – сложная дискретная система

Основные моменты этого метода:

Все элементы сложной дискретной системы представлены как ресурсы.

Основные положения РДО-метода формируются следующим образом:

1. Каждый ресурс определенного типа описывается одними и теми же параметрами.
2. Состояние ресурса определяется вектором значений всех его параметров. Состояние системы в целом – совокупностью всех его параметров.
3. Процесс протекающий в сложной системе описывается как последовательность целенаправленных действий и нерегулярных событий, изменяющих состояние ресурсов. Действия ограничены во времени событиями начала и конца.
4. Нерегулярные события описывают изменение состояния сложной системы не предсказуемые в рамках продукционной модели системы. Моменты наступления нерегулярных событий случайны.

5. Действия описываются операциями, которые представляют собой модифицированные правила, учитывающие временные параметры. Операция описывает предусловия, которым может удовлетворять состояние, участвующих в операциях ресурсов. И правила изменения состояния ресурсов в начале и конце соответствующего действия.
 6. Множество ресурсов и множество операций образуют модель сложной дискретной системы.
-