

# Алгоритмы моделирования в сложных системах

Н.С. Румянцев

Самарский национальный исследовательский университет  
им. академика С.П. Королева  
Самара, Россия  
rumancevn07@gmail.com

С.В. Востокин

Самарский национальный исследовательский университет  
им. академика С.П. Королева  
Самара, Россия  
eastst@mail.ru

**Аннотация**—Моделирование является стремительно развивающимся методом познания, но даже оно не справляется со сложными системами. Для них нужен особый подход, который развивает саму концепцию моделирования. Такое развитие повлекло за собой создание специального блока в системе, который использует для работы специальные алгоритмы моделирования. Данная статья занимается их классификацией, описанием и сравнением по характеристикам.

**Ключевые слова**— алгоритмы моделирования, сложные системы, моделирование, параллельные вычисления, имитационное моделирование

## I. ВВЕДЕНИЕ

Моделирование – процесс создания моделей и работы с ними. Для создания модели применяются многие подходы, но чаще всего используется системный подход. Сложные системы же, требуют особого подхода. Сами сложные системы представляют собой крупные комплексы, состоящие из множества составных частей. Типичным примером сложной системы может стать обычный ПК. У него иерархическая структура, которая работает за счет функционирования всех его частей. Также уровни иерархии представляют собой различные уровни абстракции, причем каждый уровень может быть рассмотрен отдельно со своим набором функций и устройств.

Проведение моделирования для таких сложных систем является важным элементом, ведь с увеличением сложности систем увеличивается и цена ошибки в таких системах, однако просто так провести моделирование данной системы, не всегда представляется возможным. Тогда в такую систему можно добавить блок, который выполняет свою работу, следуя определенным алгоритмам, которые подходят под любой код в данной системе, подстраиваясь под неё и проводя моделирование. Именно о таких алгоритмах, позволяющих строить моделирование любых сложных систем, и пойдет речь. Будут рассмотрены алгоритмы, их достоинства и недостатки, а также их реализации.

## II. АЛГОРИТМЫ МОДЕЛИРОВАНИЯ

В качестве меры времени в имитационных моделях используют модельное время – переменная  $t_0$ , отвечающая за синхронизацию всех событий, в том числе параллельных, в модели. Именно эта  $t_0$  и является главным отличием алгоритмов моделирования между собой, поэтому данный критерий может послужить обоснованием для выделения конкретных алгоритмов данного обзора. Оптимального менеджмента модельного времени  $t_0$  можно добиться меняя сами способы (формулу) измерения или же меняя организацию

алгоритма, в результате чего алгоритм за модельное время будет обрабатывать разные пакеты данных.

В настоящее время известны 4 основных алгоритма для моделирования сложных систем. Алгоритм  $\Delta t$ , особых состояний, в которых меняется способ измерения  $t$ . А также алгоритмы последовательной обработки и параллельной работы объектов, которые затрагивают организацию алгоритма.

### A. Алгоритм $\Delta t$

Алгоритм  $\Delta t$  состоит в том, что происходит имитация работы реальной системы в фиксированные моменты времени, называемые  $\Delta t$ . То есть отсчитывается шаг времени:  $t, t+\Delta t, t+2\Delta t, t+3\Delta t$  и так далее по формуле

$$t_i = t_{i-1} + \Delta t, \text{ где } i = 1, 2, \dots, N. \quad (1)$$

Для такого отслеживания вводится счётчик времени, который отсчитывает время, начиная от нуля до  $\Delta t$  на каждом цикле моделирования, и фиксирует изменения системы. Перед началом работы необходимо задать начальные значения: начало  $Z_i(0)$ , шаг  $\Delta t$ , конец  $K$ . Для моделирования такой системы подходит описание объектов системы в виде:

$$Z_i(t) = F(Z_i(t-1), x_j(t)), \text{ где } Z_i, x_j - \text{ вых. и вх. сигнал.} \quad (2)$$

Стоит отметить, что система работает всё время моделирования, а не только во время этого шага времени, такой счётчик позволяет только фиксировать изменения системы. Примерный алгоритм  $\Delta t$  изображен на рисунке 1 ниже.

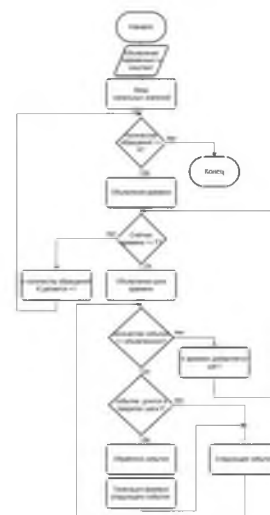


Рис. 1. Схема алгоритма  $\Delta t$ .

### B. Алгоритм особых состояний

Алгоритм особых состояний представляет собой скачкообразное изменение характеристик системы от обычного до особого (состояния изменения) системы.

Алгоритм отслеживает именно такие моменты, когда в системе происходят изменения, то есть переход от одного особого состояния в другое, что для моделирования является важнейшей задачей.

При таком алгоритме у нас всё ещё присутствует  $\Delta t$ , но сначала фиксируется изменение, а потом уже делается шаг времени до этого события и изменяется значение счётчика на величину до этого события. Вкратце алгоритм состоит из следующих шагов:

1. Начало,  $t = 0$ , установка начального состояния
2. Вычисление очередного особого момента времени
3. Если  $t \geq K$ , то обработка и конец, иначе 4
4. Расчет состояния системы во время  $t$ . Идти к 2

#### C. Алгоритм последовательной обработки

Алгоритм последовательной обработки состоит в том, что каждая заявка отслеживается с самого начала, от поступления в систему, до самого конца, до момента выхода из системы. Только после этого происходит переход к следующей заявке.

Казалось бы, такой алгоритм довольно долгий, потому что за один раз обслуживается только одна заявка, однако в таком алгоритме есть понятие очередей и каналов, которые могут быть настраиваемыми. Система массового обслуживания является типичным примером систем с алгоритмом последовательной обработки

#### D. Алгоритм параллельной обработки объектов

Алгоритм параллельной работы объектов или принцип объектного моделирования разделяет целый алгоритм на отдельные объекты, которые независимы друг от друга, но при этом слаженно работают друг с другом. Такой подход к созданию моделей позволяет собирать сколько угодно сложные системы без изменения всей системы, а только меняя некоторые части. Это существенно удлинит жизненный цикл систем.

Примерный алгоритм параллельной работы объектов изображен на рисунке 2 ниже. Такой алгоритм состоит из отдельных блоков: генератора заявок, очереди, каналов и сбора статистики.

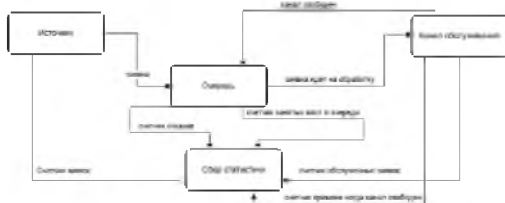


Рис. 2. Схема алгоритма параллельной работы объектов.

Блок генерации создает заявки и подсчитывает их количество, отправляя статистику в блок сбора статистики. Блок очередей отправляет заявку на обработку, если канал обслуживания свободен. В отличие от данного, первые три алгоритма плохо модернизируются.

### III. ЗАКЛЮЧЕНИЕ

В данном обзоре описаны алгоритмы моделирования, используемые при проведении моделирования сложных систем. Не существует идеального решения, объединяющего достоинства всех алгоритмов, поскольку оптимизация одних характеристик может приводить к ухудшению других. В таблице ниже приведены краткие результаты сравнения алгоритмов. Выбор компромиссного решения зависит от требований по времени адаптации под конкретную задачу, вычислительным ресурсам, гибкости, масштабируемости и отказоустойчивости.

Алгоритм  $\Delta t$  является самым универсальным и простым из всех, однако он самый неэкономичный, погрешность измерений равняется шагу времени –  $\Delta t$ . Алгоритм особых состояний экономит время моделирования и затраты производительности. Также его работа уменьшает погрешность, однако это довольно сложный алгоритм. Такой алгоритм может быть с дополнительными особенностями, которые усложняют разработку, но ещё больше уменьшат затраты и время моделирования. Алгоритм последовательной обработки подходит далеко не для всех систем и не является экономичным. А алгоритм параллельной работы объектов это масштабируемый алгоритм, но он имеет не слишком выдающиеся другие показатели.

Тем не менее, каждый из этих алгоритмов находит своё применение в распространяющейся компьютерной среде.

Таблица I. СРАВНЕНИЕ АЛГОРИТМОВ МОДЕЛИРОВАНИЯ МЕЖДУ СОБОЙ ПО ХАРАКТЕРИСТИКАМ

Характеристика	Алгоритм			
	$\Delta t$	особые состояния	последовательная обработка	параллельная работа объектов
Универсальность	+	-	-	+
Простота	+	-	+	-
Экономность	-	+	-	-
Точность	-	+	+	+
Время	-	+	+	-
Модернизация	-	-	-	+

#### ЛИТЕРАТУРА

- [1] Бирюков, Б.В. Кибернетика и методология наук / Б.В. Бирюков. – М.: Наука, 1974. – 416 с.
- [2] Филиповский В.М. Системы управления в пространстве состояний: учебное пособие / В.М. Филиповский. – СПб., 2022. – 75 с.
- [3] Гафаров, Ф.М. Параллельные вычисления: учебное пособие / Ф.М. Гафаров, А.Ф. Галиминов. – Казань: Издательство Казанского технического университета, 2018. – 149 с.
- [4] Асланов, Т.Г. Моделирование систем: учебное пособие / Т.Г. Асланов, У.Р. Тетакаев. – Махачкала: ДГТУ, 2020 – 167 с.
- [5] Теплов, А. М. Об одном подходе к сравнению масштабируемости параллельных программ / А.М. Теплов // Вычислительные методы и программирование. – 2014. – Т. 15, № 4. – С. 697-711.
- [6] Blom, S. Correct program parallelisations / S. Blom, S. Darabi, M. Huisman, M. Safari // International Journal on Software Tools for Technology Transfer. – 2021. – V. 23. – P. 741–763. DOI: 10.1007/s10009-020-00601-z
- [7] Сюзев, В.В. Алгоритмы многомерного имитационного моделирования случайных процессов / В.В. Сюзев, Е. В. Смирнова, А.В. Пролетарский // Компьютерная оптика. – 2021. – Т. 45, № 4. – С. 627–637. DOI: 10.18287/2412-6179-CO-770