

УДК 004.021

**Классификация методов имитационного моделирования сложных дискретных систем с очередями**

*Сальманова Д.Э., студент  
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,  
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

*Научный руководитель: Рудаков И.В., к.т.н., доцент  
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана  
[irudakov@bmstu.ru](mailto:irudakov@bmstu.ru)*

В настоящее время имитационное моделирование является одним из наиболее эффективных средств исследования сложных дискретных систем. По сравнению с другими методами такое моделирование позволяет рассматривать большее количество альтернатив, улучшать качество принимаемых решений и прогнозировать их последствия. При исследовании функционирования сложных дискретных систем важно наиболее правильно выбрать метод моделирования.

В качестве первого критерия классификации выберем способ представления модельного времени[4]. Среди дискретных систем выделяют событийно-, процессно-ориентированные методы, а также метод сканирования активностей. Методы имитационного моделирования непосредственно связаны с классом системы имитационного моделирования.

При *событийном подходе* исследователь описывает события, которые могут изменять состояние системы, и определяет логические взаимосвязи между ними. Имитация осуществляется путем выбора из списка будущих событий ближайшего по времени и его выполнения. Выполнение события приводит к изменению состояния системы и генерации будущих событий, логически связанных с выполняемым. Эти события заносятся в список будущих событий и упорядочиваются в нем по времени наступления. В событийных системах модельное время фиксируется только в моменты изменения состояний. Данный подход реализуют языки SLAM II, GASP IV, SIMASCRIPТ II, СИМКОН.

При использовании *подхода сканирования активностей* разработчик описывает действия, в которых принимают участие элементы системы, и задает условия, определяющие начало и окончание этих действий. События, которые начинают или

завершают действие, не планируются разработчиком модели, а инициируются по условиям, определенным для данного действия. Условия начала или окончания действия проверяются после очередного продвижения имитационного времени. Если заданные условия удовлетворяются, происходит соответствующее действие. Подход сканирования активностей реализуют языки CSL, DRAFT, HOCUS, HEADLANDS.

При *процессно-ориентированном подходе* исследователь описывает последовательность компонентов модели, которые возникают по определенной схеме. Логика возникновения определенных компонентов задается одним оператором языка. Имитатор моделирует процесс продвижения активных элементов через систему, который сопровождается соответствующей последовательностью событий. Процессно-ориентированный подход сочетает в себе элементы событийного подхода и подхода сканирования активностей. Из данной категории наиболее часто употребляются GPSS, SLAM II, СИМУЛА, SOL, Q-GERT, SIMAN, PAWS, QNAR.

Так как этап программирования является неотъемлемой частью процесса моделирования, приведем классификацию по типу используемого языка. Данная классификация связана с историческими этапами развития языков имитационного моделирования[1,3]. Графически она представлена на рис. 1.

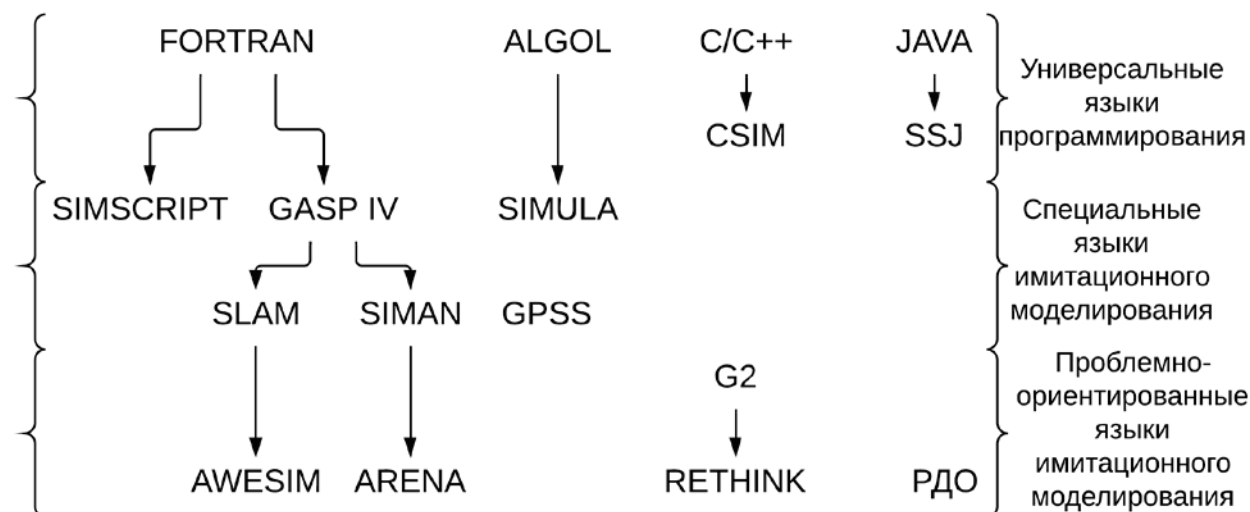


Рис. 1. Классификация методов имитационного моделирования по типу используемого языка

1. Использование универсальных алгоритмических языков программирования Паскаль, C/C++, FORTRAN, ALGOL и др. В этом случае программист имеет практически неограниченные возможности по созданию эффективной имитационной

модели, наилучшим образом использующей ресурсы ЭВМ и особенности операционной системы. Однако такой подход обладает и рядом недостатков:

- Большие трудозатраты на создание модели;
- Привлечение специалистов различного профиля;
- Требуется высокая квалификация программистов.
- Невозможность использования имитационной модели для других (аналогичных) задач без изменения.

2. Создание и использование специализированных языков имитационного моделирования. К данной группе относятся языки SLAM II, GASP IV, GPSS и др.

Языки ИМ за счет снижения гибкости и универсальности позволяют создавать имитационную модель на несколько порядков быстрее и не требуют работы системных программистов. Они обладают двумя важными достоинствами: удобством программирования и концептуальной выразительностью. При использовании специализированных языков ИМ для подготовки имитационной модели требуется описывать структуру и физические процессы объекта абстрактными понятиями в терминах используемого языка, как правило, далекими от понятий предметной области.

3. Создание и использование проблемно-ориентированных систем моделирования, например ARENA, ReThink и ряда других.

Эти системы, как правило, не требуют от пользователя знания программирования и позволяют моделировать лишь относительно узкие классы сложных систем. При этом имитационная модель генерируется самой системой в процессе диалога с пользователем, что позволяет сделать ее написание быстрым, эффективным и избежать многих ошибок программирования. Таких систем насчитывается несколько десятков.

Из рассмотренных выше инструментальных средств имитационного моделирования следует, что существует противоречие между универсальностью (широтой класса систем, для моделирования которых может быть использовано данное средство) и гибкостью (легкостью описания конкретной системы). Это означает, что если средство может быть использовано для моделирования широкого класса систем, то язык описания, как правило, сложен и неудобен для описания конкретной системы, и наоборот.

Еще одной проблемой имитационного моделирования является то, что неотъемлемой частью сложной системы является система управления (представляющая собой сложную систему), которую также необходимо моделировать. Ситуация усугубляется еще и тем, что система управления сложным объектом, как правило, включает в себя человека, принимающего решения. Существующие средства имитационного

моделирования обладают ограниченными возможностями моделирования системы управления. В ряде средств имитационного моделирования логика системы управления закладывается непосредственно в алгоритм имитационной модели, в других имеются средства для ее формализации, такие как выбор приоритетов, заполнение таблиц решений, запросы к оператору.

Таким образом, в процессе моделирования принципиальным является вопрос получения параметров модели. По данному признаку методы имитационного моделирования можно разделить на следующие группы:

- использование средств графического ввода информации об объекте моделирования;
- использование диалоговых систем ввода информации об объекте моделирования;
- гибридные системы, в которых объединены преимущества различных методов и подходов.

Указанные проблемы и ряд других обусловили новый этап развития имитационного моделирования – интеллектуальный[2]. Известно, что интеллектуальные системы обладают большой гибкостью, универсальностью и могут имитировать деятельность человека при принятии решений. Интеллектуализация процесса имитационного моделирования призвана решить указанные выше проблемы и сделать метод имитационного моделирования более эффективным и привлекательным для системных аналитиков, проектировщиков и многих других специалистов, имеющих дело со сложными системами и явлениями. Однако в этой области сделаны лишь первые шаги. Среди систем, которые претендуют на то, чтобы называться интеллектуальными, можно назвать ReThink + G2 и ARENA.

Использование методов искусственного интеллекта может использоваться при принятии решений в процессе имитации, управлении имитационным экспериментом, реализации интерфейса пользователя, создании информационных банков имитационного моделирования, использовании нечетких данных и др.

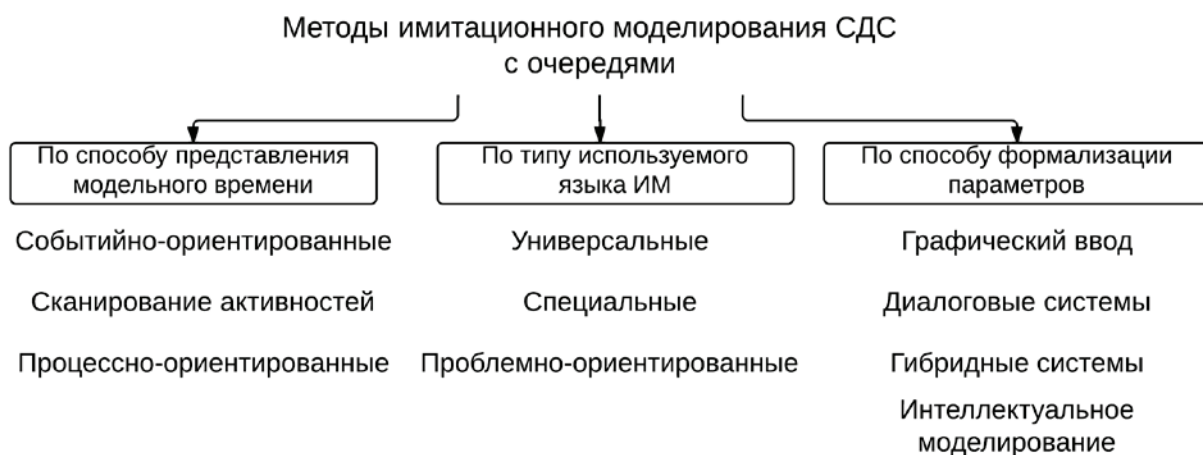


Рис. 2. Классификация методов имитационного моделирования

Из рассмотренных методов имитационного моделирования (полная классификация приведена на Рисунке 2), вытекает необходимость создания системы имитационного интеллектуального моделирования на базе одного из специальных языков имитационного моделирования. Создаваемая система должна обладать следующими качествами:

- наличие визуальной среды для работы с языком;
- наличие экспертной системы для подбора параметров модели;
- наличие команд для контроля и анализа результатов;
- наличие удобного дружественного интерфейса, с помощью которого легко вводить параметры моделирования, а также просматривать результаты;
- разработанная система должна содержать блок визуализации, который позволял бы отлаживать программу, а также демонстрировать результаты моделирования;
- по результатам моделирования необходима выдача отчета, позволяющего оценить значения исследуемых параметров модели; отчет может быть представлен в виде графика.

В качестве языка выбран язык GPSS, т.к. он отвечает ряду выше обозначенных требований, а также обладает следующими преимуществами:

- достаточная мощность языка для решения широкого класса задач;
- простота языка;
- наличие большого количества моделей на базе языка, в силу его распространенности.

Для моделирования сложных дискретных систем с очередями необходимо разработать препроцессор, который удовлетворял бы требованиям разрабатываемой системы, а именно включал команды для контроля и анализа результатов, а также

позволял бы осуществлять подбор параметров модели. Данный подход позволяет сохранить простоту разработки модели.

### Список литературы

1. Емельянов В.В., Ясиновский С.И. Имитационное моделирование систем: учеб. пособие для вузов.- М.: Изд-во МГТУ им. Н.Э.Баумана, 2009.- 584 с.
2. Кельтон В., Лоу А. Имитационное моделирование. Классика CS. СПб.: Питер; Киев: Издательская группа BHV, 2004.
3. Технология системного моделирования / С.В. Емельянов [и др.] М.: Изд-во «Машиностроение», 1988. – 520 с.
4. Nance, Richard E. (1993) A History of Discrete Event Simulation Programming Languages. Technical Report TR-93-21, Computer Science, Virginia Polytechnic Institute and State University.