

К МЕТОДИКЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

С. Н. Ковалев (Москва)

Наиболее широкое применение методика имитационного моделирования находит при исследовании систем массового обслуживания (СМО). В настоящее время для имитационного моделирования разработано большое количество специальных языков (GPSS, SLAM, GASP, SIMSCRIPT и др.). Не будем останавливаться на особенностях имитационного моделирования с помощью этих языков – достаточно хороший анализ этих и других языков имитационного моделирования приведен в работах [1, 4–8]. Эти языки (и их модификации), как правило, написаны или базируются на других языках типа: ассемблер, фортран, паскаль или других языках высокого уровня. При этом авторы разработок этих языков в качестве основных положительных качеств своих разработок зачастую отмечают простоту и компактность разрабатываемых моделей и отсутствие необходимости в изучении достаточно сложных языков программирования, которые послужили базовой основой для этих разработок. Конечно это не совсем так: чтобы грамотно и достаточно адекватно описать моделируемую систему необходимо знать все тонкости не только языка моделирования, на котором разрабатывается имитационная модель, но и среду моделирования и базовый язык программирования.

Рассмотрим особенности имитационного моделирования систем массового обслуживания тремя методами: **1)** с применением только языков высокого уровня (далее для простоты этот метод будем называть ЯВУ); **2)** с применением языка моделирования SIMPAS [5], событийная часть которого основана на языке моделирования GPSS [8]; **3)** с применением языка (и пакета) визуально-ориентированного программирования SIMULINK, входящего составной частью в систему MATLAB [2, 3].

Применение ЯВУ для имитационного моделирования систем массового обслуживания

Опыт многочисленных отечественных и зарубежных разработок в имитационном моделировании систем массового обслуживания показывает, что на языках высокого уровня типа: С, С++, Pascal и других можно с успехом решать эти задачи. В последний период наряду с развитием мощных средств объектно-ориентированного программирования в этих языках получили развитие и визуальные среды разработки типа: Borland С++ Builder, Visual С++, Delphi и др., которые позволяют достаточно быстро и качественно выполнить все необходимые операции по имитационному моделированию.

Приведем последовательность действий по разработке имитационной модели и реализации ее на компьютере, выполняемых с помощью ЯВУ: **а)** определение цели и словесная постановка задачи моделирования; **б)** разработка концептуальной модели, определяющая структуру моделируемой системы, свойства ее элементов и причинно-следственные связи, присущие этой системе; **в)** формализация модели, включающая разработку математической схемы рассматриваемой системы и соответственно модели, а также различные математические соотношения, логические условия, ограничения и т. п.; **г)** программная реализация модели на одном из языков программирования; **д)** планирование и реализация модельных экспериментов; **ж)** анализ вероятностно-временных характеристик процесса функционирования модели и синтез структуры системы.

Программная реализация модели выполняется на одном из языков высокого уровня С, С++, Pascal и др.

Для моделирования сравнительно простых СМО обычно используют Pascal, который по сравнению с языками типа C, C++ отличается более простыми конструкциями, меньшим количеством стандартных процедур и функций и т. п. До настоящего периода наиболее широкое применение получили следующие версии компиляторов Pascal: **а)** версии компилятора, работающие под управлением MS DOS в реальном режиме процессора (TURBO.EXE, TPC.EXE); **б)** версии компилятора, работающие под управлением MS DOS в защищенном режиме процессора (BP.EXE, BPC.EXE); **в)** версии компилятора, работающие под управлением WINDOWS (BPW.EXE).

В более простых случаях имитационного моделирования с помощью языка Pascal применяется структурная методология программирования. В более сложных случаях для имитационного моделирования применяют объектно-ориентированное программирование (ООП). Разработанная на базе языка Pascal объектно-ориентированная система визуального программирования DELPHI, позволяет получать приложения для WINDOWS, что в свою очередь позволяет за сравнительно короткое время решать задачи имитационного моделирования на современном уровне.

Для моделирования достаточно сложных СМО используются языки программирования типа C, C++. Появившийся в начале 70-х годов язык C использовался в основном как средство системного программирования. В настоящее время этот язык и его более мощный последователь C++ стали теми языками, на которых пишут практически любые задачи прикладного характера, в том числе и программное обеспечение для операционных систем (UNIX, Linux и Microsoft Windows), компьютерных сетей и для прикладных программ модели клиент/сервер.

Приведем наиболее часто используемые компиляторы и системы программирования для языков C и C++: Turbo C, Borland C++3.1, Borland C++4.5, Borland C++5.02. Первые два компилятора предназначены для компиляции программ в операционной системе (ОС) MS DOS, вторые две системы программирования имеют мощную интегрированную среду разработки (IDE) и предназначены для работы как в ОС MS DOS, так и в WINDOWS.

В настоящее время при программировании достаточно сложных задач, какими являются в большинстве своем и задачи имитационного моделирования, предпочтение отдается языку C++ по сравнению с языком C по следующим причинам: в языке C++ реализован объектно-ориентированный подход к программированию, который позволяет решить проблему сложности программ; кроме того, C++ обеспечивает более тщательный контроль соответствия типов, поток ввода-вывода C++ более прост для применения, чем функции printf и scanf в языке C и т. п.

В настоящее время стандарт языка C++ определяется документом ANSI/ISO (Американский национальный институт стандартов / Международная организация стандартов). Наиболее популярной средой разработки для C++ является продукт, совместно разработанный компаниями Microsoft (Visual C++) и Borland (Borland C++ Builder) и предназначенный для работы под управлением операционных систем Microsoft Windows. С помощью этого продукта можно не только достаточно детально реализовывать (моделировать) на компьютере реальные системы и процессы, но и обеспечивать для этих моделей визуальный интерфейс и широкое распространение моделей среди пользователей с помощью компьютерных сетей.

В стандартных библиотеках, приведенных языков высокого уровня, содержится множество функций, в том числе и функций, возвращающих случайные числа, которые используются для имитационного моделирования СМО.

Применение SIMPAS для имитационного моделирования систем массового обслуживания

Система моделирования Simpas (разработанная в МГТУ им Н.Э.Баумана) написана на языке Pascal и может быть реализована в среде различных операционных систем: MS DOS, Windows 95/98, Windows NT, Linux.

Система Simpas предназначена для имитационного моделирования, по реализации представляет собой методологию построения дискретных и непрерывных событийных моделей и набор поддерживающих данную методологию процедур. Эту систему отличает простота и компактность записи моделей.

Построение моделей предусматривает раздельное определение модельной среды и выполняемых в ней действий. Фактически дискретная событийная модель состоит из трех частей: **1)** часть описания среды моделирования; **2)** часть инициализации модельной среды; **3)** событийная часть, включающая в себя цикл моделирования и блок выборки событий. Как ранее отмечалось, в рассматриваемой системе моделирования используется форма записи моделей, по своему виду напоминающая модели на языке имитационного моделирования GPSS (разработанного американскими учеными). Таким образом, в ней сочетаются два подхода: событийное описание моделей и процессно-ориентированная форма записи, что повышает наглядность представления моделей и сохраняет возможность описания сложных событий.

В системе Simpas, кроме обычно используемых в языке Pascal типов данных, предусмотрены специфические для Simpas типы данных, в частности: скалярные и множественные типы данных. К последнему типу (объектам) относятся: транзакты, приборы, очереди, накопители и др. Для обработки и выполнения соответствующих действий над этими типами используются специальные процедуры.

В системе Simpas определены генераторы, позволяющие производить последовательности случайных чисел, распределенных в соответствии с часто используемыми законами, в том числе: **1)** RAND01(V) – генератор случайных чисел, равномерно распределенных в интервале $[0,1]$, V – переменная – источник; **2)** RANDAB(A,B,V) – генератор случайных чисел, равномерно распределенных в интервале $[A,B]$; **3)** RANDEXP(LAMBDA,V) – генератор экспоненциально распределенных случайных чисел с интенсивностью LAMBDA, которая задает среднее число случайных событий, происходящих в единицу времени; **4)** RANDNORM(XMEAN,DISP,V) – генератор случайных чисел, распределенных по нормальному закону со средним XMEAN и дисперсией DISP.

В результате работы модели собираются данные о работе очередей, приборов, накопителей. Чтобы сохранить эти данные в файле результатов, используются специальные процедуры. Статистические данные, собранные в результате моделирования, распечатываются процедурой PRINTALL.

Применение SIMULINK для имитационного моделирования систем массового обслуживания

Пакет Simulink позволяет осуществлять исследование (моделирование) сложных динамических систем. Ввод параметров систем производится в интерактивном режиме, путем графической сборки схемы соединений элементарных блоков, в результате чего получается модель исследуемой системы. Блоки, включаемые в создаваемую модель, могут быть связаны друг с другом как по информации, так и по управлению. Тип связи зависит от типа блока и логики работы модели.

С помощью Simulink автоматизируется наиболее трудоемкий этап имитационного моделирования – составление и решение сложных систем алгебраических и дифференциальных уравнений, описывающих заданную схему (модель)

Используя библиотеку блоков Simulink, представляющую собой набор визуальных объектов, можно собирать произвольные схемы. Из отдельных блоков можно получать требуемое число копий, которые в свою очередь можно индивидуально настраивать. При этом настройке подлежат как внутренние параметры блоков, так и внешние.

В Simulink реализован механизм подсистем. С его помощью любая сложная система представляется в виде совокупности подсистем (субмоделей). Внутри подсистем первого уровня, в свою очередь, могут располагаться подсистемы второго уровня и т. д.

В пакете Simulink имеется большой набор блоков, обеспечивающих при моделировании воздействия с разными функциональными и временными зависимостями, а также блоки получателей информации. Первый тип блоков входит в раздел Sources (источники) и включают свыше 15-ти блоков, в том числе: Constant – источники постоянной величины; Random Number – источник дискретного сигнала, амплитуда которого является случайной величиной, распределенной по нормальному закону и т. д. Второй тип блоков входит в раздел Sinks (получатели) и включает шесть блоков, в том числе: Scope – осциллограф; XY Graph – графопостроитель; Display – устройство вывода на экран дисплея; To File – устройство записи данных в файл; To Workspace – устройство записи в переменную рабочего пространства; Stop Simulation – блок остановки моделирования.

В Simulink большой набор генераторов случайных чисел. Если же и этих средств оказывается недостаточно, в модель могут быть включены и другие инструменты системы MatLab, в частности, набор инструментов Statistics Toolbox, в который входит более 20 генераторов случайных чисел.

Пример имитационного моделирования систем массового обслуживания, выполненного методами DELPHI, SIMPAS, SIMULINK

Приведем простой пример имитационного моделирования для системы массового обслуживания, выполненного методами: DELPHI, SIMPAS, SIMULINK.

Условие задания

На вычислительную систему, состоящую из двух процессоров, поступают задания через 11 ± 5 сек. Задания поступают последовательно сначала к первому, затем ко второму процессору. Обработка на первом процессоре занимает 11 ± 7 сек., а на втором – 11 ± 6 сек. Задания с суммарным временем обработки более 26 сек, покидают систему, с временем обработки от 20 до 26 сек. направляются ко второму процессору, а при времени обработки меньше 20 сек. должны пройти обработку в первом, затем во втором процессорах.

Смоделировать процесс обработки в системе 500 заданий. Определить функцию распределения времени обработки и вероятности повторения полной и частичной обработки. При выходе заданий без повторной обработки менее 92% обеспечить системе мероприятия, дающие гарантированный выход 90% заданий, с суммарным временем обработки более 20 сек.

В таблице приведены основные результаты расчетов.

Интер-	DELPHI	SIMPAS	SIMULINK
--------	--------	--------	----------

вал по- ступле- ния за- явки в сек.	В Е Р О Я Т Н О С Т Ь В %								
	повто- рения полной обра- ботки	повто- рения частич- ной об- работки	выхода без по- вторной обра- ботки	повто- рения полной обра- ботки	повто- рения частич- ной об- работки	выхода без по- вторной обра- ботки	повто- рения полной обра- ботки	повто- рения частич- ной об- работки	выхода без по- вторной обра- ботки
Меняем интервал поступления заявки от 14 до 16									
14	33,6	39,4	27,0	36,2	38,6	25,2	35,8	43,0	21,2
15	35,0	42,4	22,6	36,4	37,2	26,4	34,8	42,6	22,6
16	34,8	39,4	25,8	35,0	38,4	26,6	36,0	43,0	21,0
Меняем время обработки заданий в 1-м процессоре от 12 до 14									
12	23,6	53,6	22,8	27,8	48,2	24,0	21,4	53,5	25,1
13	14,8	51,4	33,8	18,4	44,4	37,2	15,9	54,2	29,9
14	7,8	43,4	48,8	8,2	43,4	48,4	6,3	48,4	45,3
Меняем время обработки заданий во 2-м процессоре от 13 до 15									
13	21,6	41,6	36,8	19,4	43,0	37,6	19,6	40,3	40,1
14	17,6	40,4	42,0	13,6	40,0	46,4	15,6	39,5	44,9
15	6,2	41,6	52,2	7,6	41,6	50,8	1,8	38,6	59,6

Вывод: а) результаты по трем методам сопоставимы между собой и различаются, в среднем, в пределах 5–10%; б) отличия объясняются, в основном, как различиями в реализациях этих методов, так и недостаточным объемом разыгрываемых рядов случайного процесса.

Оценка методов и систем имитационного моделирования

Приведем характеристику методов и систем имитационного моделирования, рассмотренных выше.

1. Метод ЯВУ (языки программирования Pascal, C, C++ и их модификации): а) в изучении эти языки достаточно сложны, причем этим отличается в большей степени язык C++; б) язык Pascal по сравнению с языками типа C, C++ отличается простотой программирования, хотя версия этого языка с указателями, объектами и т. п. по сложности программирования приближается к языку C++, а система DELPHI по своим возможностям близка к системе Borland C++ Builder; в) по сравнению с системами моделирования SIMPAS и SIMULINK метод ЯВУ отличается хорошей гибкостью в использовании различных подходов моделирования.

2. Система имитационного моделирования SIMPAS: а) сравнительно проста в изучении; б) однако при программировании требуется знать кроме самой системы моделирования SIMPAS также и систему программирования Pascal; в) очевидно, что выигрывая в простоте изучения и программировании, эта система проигрывает в гибкости использования различных подходов моделирования.

3. Система визуально-ориентированного программирования SIMULINK, примененная для имитационного моделирования: а) отличается сравнительной сложностью (по сравнению с системой имитационного моделирования SIMPAS) при изучении и простотой при программировании; б) по удобству графического пользовательского интерфейса, по обилию компонентов в множестве библиотек, разнообразию виртуальных средств регистрации и визуализации результатов моделирования эта система выгодно отличается от других систем моделирования.

Литература

1. **Бусленко Н.П.** Моделирование сложных систем. Москва. 1961 г.
2. **Гульятев А.** Визуальное моделирование в среде MATLAB: учебный курс – СПб: Питер, 2000 г.
3. **Дьяконов В. Simulink 4.** Специальный справочник. – СПб: Питер, 2000 г.
4. **Ковалев С.Н.** К применению информационно-компьютерных технологий в математическом моделировании систем. Сб. науч. тр. МАДИ (ГТУ). Москва: Изд-во МАДИ, 2001 г.
5. **Марков А.А.** Моделирование информационно-вычислительных процессов: учебное пособие для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 1999 г.
6. **Прицкер А.** Введение в имитационное моделирование и язык СЛАМ II: Пер. с англ. – М.: Мир, 1987 г.
7. **Советов Б.Я., Яковлев С.А.** Моделирование систем (учебник для вузов). – М.: Высшая школа, 1998 г.
8. **Шрейбер Т. Дж.** Моделирование на GPSS: Пер. с англ. – М.: Машиностроение, 1980 г.