

## ОПЫТ ТЕСТИРОВАНИЯ GPSS WORLD

Ю. И. Рыжиков (Санкт-Петербург)

Ни один из языков моделирования не оказал на имитацию столь большого воздействия, как GPSS. Эта система относится к самым популярным системам программирования и (наряду с TeX'ом и Фортраном) являет редчайший пример долголетия. Ее крупным недостатком всегда была жесткость – практическая невозможность выйти за пределы хоть и богатого, но ограниченного спектра возможностей. Низкий (в особенности по нынешним понятиям) уровень входного языка, невозможность реализации нестандартных вычислений, скрытость от программиста внутренней логики поведения модели резко ограничивали реальную полезность системы, в том числе ее версий для персональных компьютеров. Поэтому большие надежды связывались с появлением Windows –версии, а конкретно GPSS World (далее GPSS/W).

### Нововведения в GPSS/W

Принципиальные особенности GPSS/W можно сгруппировать в несколько разделов.

1. Каждая разработка в духе современных технологий программирования рассматривается как *проект* (эта терминология в электронной документации к системе не используется). Проект состоит из объектов четырех типов: исходный модуль, Имитация, отчеты и текстовые файлы. Модель может иметь иерархическое строение и тогда включает в себя командой INCLUDE ранее разработанные фрагменты. Для каждого из шагов проекта есть свои *инструменты*: полноэкранный Текстовый редактор, пункт Create Simulation в меню Command, множество окон контроля и команд управлением моделированием, автоматический генератор отчетов, средства настройки отчетов, статистического анализа. В системе реализованы элементы объектного программирования в классическом его понимании: *наследование свойств* (установок) Имитации от исходной модели и *полиморфизм* – автоматическое преобразование операндов выражений к требуемому типу.

2. Интерфейс ориентирован на пользователя. Однострочный текстовый редактор для набора модели заменен на полноэкранный с реализацией стандартных функций Windows. Есть возможность замены традиционно используемого вместо знака умножения # на привычную \*. Компилятор формирует *список* синтаксических ошибок, обеспечивает автоматический подвод курсора к месту очередной ошибки и вывод диагностики в статусную строку. Каждый выбор элемента списка вызывает сообщение о типе ошибки в статусную строку и помещает точку вставки точно перед ошибочным элементом. Динамика модели может отслеживаться как в числовой, так и в графической форме с помощью окон около 20 разных типов в детальном и обзорном режимах. Степень загруженности ресурсов показывается цветом, можно наблюдать за перемещением *активного* транзакта. В окна можно выводить СЧА и накопленную статистику. Отображение внизу главного окна значения *таймера* облегчает диагностику зависания.

Открытие online-окон замедляет моделирование, поскольку система вынуждена ждать их обновления. Для иллюстрации степени замедления приведем данные по длительности прогонов одной и той же модели:

- все окна закрыты – 4 с;
- выводится таймер – 97 с;
- открыты окно блоков и таймер – 5271 с (почти полтора часа!).

К сожалению, иконки свернутых динамических окон неразличимы, и при их восстановлении трудно выбрать желаемое. Имеются средства интерактивного ввода

команд, контрольных остановов и трассировки. Результаты моделирования могут быть *автоматически* представлены в виде гистограмм соответствующих распределений.

3. Резко расширены вычислительные возможности. Отменены обязательная целочисленность операндов и принудительное округление результатов. Максимальное число измерений матрицы увеличено до 6. Используемый при инициализации матриц атрибут UNSPECIFIED предупреждает работу с «неприсвоенными» значениями. Средства GPSS дополнены языком программирования расчетов PLUS, близким к традиционным алгоритмическим. Это практически сняло ограничения на формируемые результаты и способы их обработки.

4. Система берет на себя стандартную статистическую обработку результатов моделирования. По результатам прогона программы автоматически формируется отчет. Содержание отчета при необходимости корректируется через Установки модели.

5. Имеются средства автоматической разработки, проведения и разработки статистических экспериментов (включая дробные факторные) по регрессионному анализу и оптимизации моделей. Программа Эксперимента может быть составлена как ручную, так и автоматически – с помощью диалога, реализуемого через меню Edit/Insert Experiment. Аргументы определяют, какая часть поверхности отклика подлежит исследованию в процессе Эксперимента. EXPERIMENT заполняет глобальную Матрицу Результаты, которая затем передается процедуре ANOVA.

Факторные и оптимизационные эксперименты по автоматически создаваемым Имитациям реализуются очень медленно (на ПЭВМ Pentium I/166 – *часами*).

### Тестирование GPSS/W

Общим правилом работы с новым программным обеспечением математического характера является его тестирование на задачах с известным решением, предпочтительно аналитическим. Напомним, что основным назначением GPSS всегда был анализ систем с очередями, т. е. *массового обслуживания*.

Для системы  $M/M/1$  при коэффициенте загрузки 0.9 теоретическая средняя длина очереди  $q = \rho^2 / (1 - \rho) = 8.1$ . Среднее время ожидания  $w$  в соответствии с формулой Литтла (закон сохранения стационарной очереди) должно быть равно  $q / \lambda$ , и в данном случае ( $\lambda = 1$ ), численно совпадать со средней длиной очереди. Теоретическая вероятность незанятости системы равна  $1 - \rho = 0.1$ . Следовательно, в среднем десятая часть заявок должна получать обслуживание без ожидания. Сопоставим с этими ожиданиями результаты моделирования – табл. 1.

Формула Литтла и доля заявок, принимаемых к обслуживанию без ожидания, подтверждаются с достаточно высокой точностью. Это дает основания доверять интерпретатору GPSS/W. Показатели, определяемые только *средними* значениями моделируемых первичных величин (среднее время обслуживания и коэффициент загрузки), также вполне приемлемы. Среднее время ожидания

$$w = \frac{\lambda b_2}{2(1 - \rho)}$$

зависит уже от двух моментов распределения времени обслуживания. Результаты, связанные с ожиданием, оставляют желать лучшего. Причина может быть только одна: недостаточно качественный генератор псевдослучайных чисел. Увеличение числа проведенных испытаний в общем приближает результаты к ожидаемым, но даже при 500 тыс. испытаний погрешность составляет около 17 %.

Таблица 1

## Тестирование GPSS/W на модели M/M/1

Показатель	Теория	Число испытаний		
		50 000	200 000	500 000
Коэффициент загрузки	0.900	0.896	0.897	0.899
Среднее время обслуживания	0.900	0.899	0.902	0.901
Число входов	–	49843	199042	499032
Из них с нулевым ожиданием	–	5340	21001	50691
Средняя длина очереди	8.100	5.132	5.757	6.690
Среднее время ожидания	8.100	5.148	5.785	6.703

В дополнение к приведенным результатам отметим, что для системы *M/M/1* распределение времени пребывания заявки в системе также подчинено показательному закону со средним  $1/(1/0.9-1/1)=9$ . При ограничителе таймера 200 000 была заказана гистограмма распределения времени пребывания, получены среднее значение 9.264 и среднеквадратическое отклонение 9.147. Требуемое для показательного распределения их равенство приближенно выполняется, однако погрешность самих значений составляет около 2 %.

Выше были приведены контрольные результаты для системы *M/M/1*. Сопоставим с этими ожиданиями результаты моделирования при различных способах задания показательных распределений (встроенная *Exponential*, табличная ДФР и теоретически точное обращение  $-\ln(1-U)$ ) и числе испытаний – табл. 2. Результаты, связанные с ожиданием, оставляют желать лучшего даже в случае с теоретически точным переходом от равномерного к экспоненциальному распределению. Причина та же: недостаточно качественный генератор равномерных псевдослучайных чисел. Эти ошибки усугубляются погрешностями перехода к показательному распределению в *Exponential* и отчасти нейтрализуются (случайный эффект) при использовании табличного преобразования. Увеличение числа проведенных испытаний в общем приближает результаты к ожидаемым.

Обсудим, наконец, временные показатели сравниваемых версий ДСЧ. Для достижения отметки таймера 500000 при табличной реализации логарифма и при работе с *Exponential* потребовалось практически одинаковое время – около 135 с (процессор *Pentium 1/166*). По-видимому, *Exponential* реализована таблично. При точном обращении показательного распределения было затрачено 187 с.

Прогон аналогичной программы на *Фортране 77* при тех же исходных данных дал среднюю длину очереди 8.683. Это заметно ближе к теоретической оценке, чем лучший из приведенных в таблице результатов. Прогон занял всего 6.26 с. Столь большая разница является платой за универсальность интерпретатора GPSS.

Таблица 2

## Влияние способов генерации показательно распределенных чисел

Показатель	Exponential		Табличная ДФР		-ln(1-U)	
	50 000	500 000	50 000	500 000	50 000	500 000
Коэффициент загрузки	0.896	0.899	0.900	0.900	0.909	0.902
Среднее время обслуживания	0.899	0.901	0.899	0.901	0.903	0.896
Число входов	49843	499032	50048	499390	50320	503176
Из них с нулевым ожиданием	5340	50691	5022	49823	4764	50498
Средняя длина очереди	5.132	6.690	7.770	7.911	7.163	6.831
Среднее время ожидания	5.148	6.703	7.763	7.920	7.118	6.788

## Общая оценка GPSS World

GPSS/W является весьма ценным инструментом имитационного моделирования, свободным от ограничений аналитических и численных методов, достаточно «прозрачным», допускающим нестандартную обработку данных и снимающим с программиста множество нетривиальных проблем программирования и отладки моделей. Тем не менее, приходится отметить наличие у нее ряда серьезных недостатков:

1. Оставляет желать лучшего электронная документация к системе. При вызове Help'a по ключевому слову пользователю представляется «слепое» множество альтернатив, так что приходится выполнять их перебор. Часто встречаются пережитки предыдущих версий (GPSS/PC). Заявление о числовом значении «his value has a magnitude limited to 306 decimal digits» можно понять как длину мантииссы, хотя на самом деле речь идет о модуле десятичного порядка. Ничего (кроме факта существования) не сказано о библиотеке. В разделе определения табличных функций обратная функции заменена *кумулянтной*. В заголовке определения команды MATRIX неверно указано число операндов, не выделен строкой и буквой неиспользуемый первый операнд. Этот перечень можно было бы продолжить.

2. Громоздкость системы и явная перегруженность встроенными возможностями (многообразие примитивов), множество пережитков ассемблерного стиля программирования. В разных блоках одно и то же по смыслу значение закреплено за разными полями операндов: было бы гораздо лучше работать с ключевыми операндами, имеющими значимые имена, или автоматически выводить подсказку при наборе типового блока. Совершенной дикостью является то, что использование кириллических символов даже в комментариях исключают правильную работу Имитации (во *всех* системах программирования компиляторы игнорируют примечания). Отметим *важность* этого дефекта для русскоязычных пользователей: им приходится отказываться от комментариев, что резко усложняет сопровождение программ.

3. Медленная работа интерпретатора, связанная с его крайней громоздкостью. Вероятно, проблему можно было бы снять введением перекомпиляций интерпретатора применительно к структуре заказанной модели.

4. Отсутствие концептуального единства. Достаточно указать различие в обращении к элементам матрицы при простой ссылке и при изменении значения; круглые индексные скобки в основном тексте и квадратные – в PLUS-выражениях; обязательность приставки MX\$ при ссылке на глобальную матрицу в тексте модели и столь же обязательное ее отсутствие внутри процедур; контекстно зависимый вид ссылок на параметры активного транзакта; потерявшее смысл в GPSS/W различие переменных VARIABLE и FVARIABLE; выражение коэффициента использования устройства в тысячных долях.

5. Неудачны обозначения операторов отношения L,G,E (было бы лучше согласовать с фортрановскими LT,GE,EQ); арифметическое SQR используется для квадратного

корня (в Паскале так обозначается *квадрат*); в связи с «числовым» представлением логических значений и объединением понятий числового равенства и логической эквивалентности нарушено общепринятое старшинство операций; состояние логических ключей описывается как SET и RESET (буквальный перевод «установлен» и «установлен заново») вместо ON,OFF; операнд RE (традиционный смысл этой приставки – повторение действия) означает удаление – лучше было бы DEL. В системе невозможно непосредственно определить вектор. Нижняя граница индексов матрицы по любому измерению равна 1, что может нарушить естественность индексации (например, при расчете вероятностей состояний системы). Крайне неудобен просмотр матриц через динамическое окно: одновременно можно видеть только два столбца. Ненулевая инициализация матрицы требует отдельного оператора INITIAL на каждый элемент. Здесь напрашивается аналог фортрановского DATA.

6. Отсутствуют формулы (или алгоритмы расчета) моментов логарифмического распределения Лапласа, логарифмически логистического, распределений крайних значений и обратного гауссова и, что самое главное, – средства подбора параметров любых теоретических распределений по заданным моментам. Без таких средств предлагаемые генераторы имеют нулевую ценность. Нет непосредственных возможностей генерации коррелированных случайных величин.

7. Нельзя менять тип шкал графиков (исключаются логарифмические) и их разметку, цвет и структуру линий, что может сделать последние неотличимыми друг от друга и/или от фона при черно-белом выводе. Аргументом графиков может быть только *время*, так что распределение вероятностей состояний системы автоматически построить нельзя. Кстати, для такого графика потребовалась бы логарифмическая шкала.

8. «Универсальность» интерпретатора GPSS явно недостаточна – главным образом из-за ограничений на доступ к «цепям». Допустимы *только однократные* прерывания устройств. Вообще исключены *прерывания памяти*, которые могут быть использованы для моделирования многоканальных устройств обслуживания и, следовательно, окажутся подвержены прерываниям. Невозможно моделировать обслуживание с *динамическим приоритетом* (линейный рост приоритетов по времени ожидания с угловыми коэффициентами, зависящими от типа заявки): цепь может быть переупорядочена произвольно только в момент входа в нее новой заявки, тогда как для реализации динамического приоритета необходимо переупорядочение *в момент выборки*. Синтаксис GPSS не позволяет динамически формировать имя или метку блока, что исключает моделирование обслуживания *циклической системы очередей*. Невозможно точное моделирование «замкнутых» систем с интенсивностью потока, зависящей от числа заявок в сети.

9. Ценность оптимизационных экспериментов в GPSS сомнительна. Они невероятно трудоемки и выбирают только значения непрерывных переменных, тогда как оптимизацию систем обслуживания чаще приходится проводить по дискретным аргументам (выбор осуществляется из дискретных рядов устройств с фиксированными характеристиками – быстродействием, емкостью памяти, набором алгоритмов функционирования и т. п. и состоит в назначении *целого* числа устройств или расширяющих блоков). Никакая «универсальная» система подобную специфику учесть не может. Кстати отметим, что об *алгоритме* оптимизации модели нигде не упоминается, а возможности вмешательства в его работу (разумеется, под ответственность пользователя) отсутствуют.

### Выводы

Вышеизложенное вкуче с опытом тестирования вынуждает заявить, что при всей полезности GPSS/W эта модификация «заслуженного ветерана» является запоздалой и недостаточной. Давно назрела необходимость *радикального* обновления с учетом сделанных выше замечаний – даже ценой отказа от совместимости с ранее разработанными моделями.