

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ И ЕГО ПРИЛОЖЕНИЯ

В.А. Балаш, О.С. Кузнецова, С.Н. Купцов

2008

Глава 1

МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН

1.1 Общее представление о методе Монте-Карло

Метод *Монте-Карло* — это численный метод решения математических задач при помощи моделирования случайных величин. Создателями этого метода считают английских математиков Дж. Неймана и С. Улама, которые анонсировали данный метод в статье *The Monte Carlo method*, J. Amer. statistical assoc., 1949. Название метода происходит от города в княжестве Монако, знаменитого своим игорным домом, так как одним из простейших механических приборов для получения случайных величин является рулетка.

До появления ЭВМ этот метод не мог найти широкого применения в силу трудоемкости моделирования случайных величин вручную. Возникновение метода Монте-Карло как универсального численного метода стало возможным благодаря появлению ЭВМ.

Пример. Пусть нужно вычислить площадь плоской произвольной фигуры S с границей, заданной графически или аналитически, или состоящей из нескольких кусков. Пусть фигура S расположена внутри квадрата единичной площади.

Выберем в единичном квадрате N случайных точек "равномерно разбросанных" по всему квадрату. Обозначим через N' число точек, попавших внутрь фигуры S приблизительно равна отношению N'/N . Чем больше N , тем точнее будет вычислена площадь S .

Для использования метода Монте-Карло составляется программа для осуществления одного случайного испытания, затем испытание повторяется требуемое количество раз, причем каждый опыт не зависит от остальных, а результаты всех опытов осредняются. Поэтому *метод Монте-Карло* называют методом *статистических испытаний*.

Ошибка вычисления при использовании этого метода пропорциональна числу

$\sqrt{D/N}$, где D — некоторая постоянная и N — число испытаний. Поэтому для уменьшения ошибки в 10 раз, то есть чтобы получить в результате еще один верный десятичный знак, нужно увеличить число испытаний в 100 раз.

Общая схема метода Монте-Карло

Пусть требуется вычислить некоторую неизвестную величину m . Попробуем придумать такую случайную величину ξ , чтобы $M(\xi) = m$. Пусть $D\xi = b^2$.

Рассмотрим N случайных величин ξ_2, \dots, ξ_N , распределения которых совпадают с распределением ξ . При достаточно большом N в силу *Центральной Предельной Теоремы (1)* распределение суммы $\rho_N = \xi_1 + \xi_2 + \dots + \xi_N$ будет приблизительно нормальным с параметрами $a = Nm$, $\sigma^2 = Nb^2$.

Применим *правило трех сигм*:

$$P\left(Nm - \frac{3b}{\sqrt{N}} < \rho_N < Nm + \frac{3b}{\sqrt{N}}\right) \approx 0,997$$

или

$$P\left(\left|\frac{1}{N} \sum_{j=1}^N \xi_j - m\right| < \frac{3b}{\sqrt{N}}\right) \approx 0,997.$$

Получено важное для метода Монте-Карло соотношение, дающее метод расчета m и оценку погрешности. В самом деле, найдем N значений случайной величины ξ . Среднее арифметическое этих значений будет приближенно равно m . С большой вероятностью ошибка такого приближения не превосходит величины $3b/\sqrt{N}$, и эта ошибка стремится к нулю при росте N .

1.2 Случайные величины

Метод вычисления площади, сформулированный в предыдущем пункте, справедлив только тогда, когда случайные точки не *просто случайны*, но и *равномерно разбросаны* по всему квадрату. Чтобы придать этим словам точный смысл, познакомимся с определением случайных величин и некоторыми их свойствами.

Для задания случайной величины надо указать, какие *значения* она может принимать и каковы *вероятности* этих значений.

1.2.1 Дискретные, непрерывные и нормальные случайные величины

Определение 1.2.1. Случайная величина ξ называется дискретной, если она может принимать дискретное множество значений (x_1, x_2, \dots, x_n) . Допускается и бесконечное число значений $(x_1, x_2, \dots, x_n, \dots)$.

Дискретная случайная величина ξ определяется таблицей (распределением случайной величины ξ):

$$\xi = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix},$$

где (x_1, x_2, \dots, x_n) — возможные значения величины ξ ; (p_1, p_2, \dots, p_n) — соответствующие им вероятности:

$$P(\xi = x_i) = p_i.$$

Числа (x_1, x_2, \dots, x_n) могут быть любыми, а вероятности (p_1, p_2, \dots, p_n) удовлетворяют условиям

1. $p_i > 0$,
2. $p_1 + p_2 + \dots + p_n = 1$, то есть ξ обязана в каждом случае принять одно из значений (x_1, x_2, \dots, x_n) .

Определение 1.2.2. Математическим ожиданием *случайной величины ξ* называется число

$$M\xi = \sum_{i=1}^n x_i p_i.$$

Если формулу для $M\xi$ записать в виде

$$M\xi = \sum_{i=1}^n x_i p_i = \frac{\sum_{i=1}^n x_i p_i}{\sum_{i=1}^n p_i},$$

то виден смысл математического ожидания — это *среднее значение* величины ξ , то есть это осреднение ξ с весами, равными вероятностям (p_1, p_2, \dots, p_n) .

Свойства математического ожидания:

1. $M(\xi + c) = M\xi + c$,
2. $M(c\xi) = cM\xi$,
3. $M(\xi + \eta) = M\xi + M\eta$,

где ξ, η — две любые случайные величины, а c — не случайная величина.

Определение 1.2.3. Дисперсией *случайной величины ξ* называется число

$$D\xi = M[\xi - M\xi]^2.$$

Таким образом, дисперсия $D\xi > 0$ — это математическое ожидание *квадрата* отклонения случайной величины ξ от ее среднего значения $M\xi$.

Получим еще одну формулу для дисперсии:

$$D\xi = M[\xi^2 - 2M\xi \cdot \xi + (M\xi)^2] = M(\xi^2) - 2M\xi \cdot M\xi + (M\xi)^2,$$

$$D\xi = M(\xi^2) - (M\xi)^2.$$

Свойства дисперсии:

1. $D(\xi + c) = D\xi$,
2. $D(c\xi) = c^2 D\xi$,

где ξ случайная величина, а c — не случайная величина.

Для *независимых* случайных величин справедливы соотношения:

1. $M(\xi\eta) = M\xi M\eta$,
2. $D(\xi + \eta) = D\xi + D(\eta)$.

Независимость случайных величин означает следующее: если кроме случайной величины ξ наблюдают за случайной величиной η , причем распределение ξ не меняется от того, что нам уже известно значение, которое приняла случайная величина η , то считают, что ξ от η *не зависит*.

Пример. Рассмотрим случайную величину θ с распределением

$$\xi = \begin{pmatrix} 3 & 4 \\ 1/2 & 1/2 \end{pmatrix}.$$

Реализация данной случайной величины — игра в орлянку с условием, что орел это 3 очка, а решка — 4 очка.

$$M\theta = 0,5 \cdot 3 + 0,5 \cdot 4 = 3,5;$$

$$D\theta = 0,5 \cdot (3^2 + 4^2) - (3,5)^2 = 0,25.$$

Определение 1.2.4. Случайная величина ξ называется непрерывной, если она может принимать любое значение из некоторого интервала $(a; b)$. Допускаются случаи $a = -\infty$ и $b = +\infty$.

Непрерывная случайная величина ξ определяется заданием интервала $(a; b)$, содержащего возможные значения этой величины и функции $p(x)$, которая называется *плотностью вероятностей* случайной величины ξ (или *плотностью распределения* ξ).

Смысл функции $p(x)$ следующий: пусть $(a'; b') \subset (a; b)$ — произвольный интервал, $(a \leq a', b' \leq b)$. Тогда вероятность того, что ξ окажется в интервале $(a'; b')$, равна интегралу

$$P(a' < \xi < b') = \int_{a'}^{b'} p(x) dx.$$

Свойства плотности $p(x)$

1. $p(x) > 0$;
2. $\int_a^b p(x) dx = 1$.

Определение 1.2.5. Математическим ожиданием *непрерывной случайной величины* ξ называется число

$$M\xi = \int_a^b x \cdot p(x) dx.$$

Если формулу для $M\xi$ записать в виде

$$M\xi = \frac{\int_a^b x \cdot p(x) dx}{\int_a^b p(x) dx},$$

то видно, что смысл математического ожидания — это *среднее значение* ξ , так как значением ξ может быть любое число из интервала (a, b) , которое входит в интеграл с весом $p(x)$.

Формулы, свойства и определения математического ожидания и дисперсии для непрерывных случайных величин аналогичны приведенным выше для дискретных случайных величин.

Приведем без доказательства формулу для математического ожидания случайной функции. Выберем произвольную непрерывную функцию $f(x)$ и рассмотрим случайную величину $\eta = f(\xi)$, которую назовем *случайной функцией*. Тогда

$$Mf(\xi) = \int_a^b f(x) \cdot p(x) dx, \quad Mf(\xi) \neq f(M\xi).$$

Определение 1.2.6. *Непрерывная случайная величина* γ называется *равномерно распределенной* на интервале $(0, 1)$, если ее плотность имеет вид

$$p(x) = \begin{cases} 1 & , \quad 0 \leq x \leq 1 \\ 0 & , \quad x \notin (0; 1) \end{cases}$$

и интегральная функция распределения

$$F(x) = \begin{cases} 0 & , \quad x < 0 \\ x & , \quad 0 \leq x \leq 1 \\ 1 & , \quad x > 1 \end{cases}$$

Для всякого интервала $(a'; b') \subset (0; 1)$ вероятность того, что γ попадет в $(a'; b')$ равна длине интервала $(a'; b')$

$$\int_{a'}^{b'} p(x) dx = b' - a'.$$

Если мы разделим весь интервал $(0; 1)$ на любое число интервалов равной длины, то вероятность попадания γ в любой из этих интервалов будет одинакова. Заметим также, что

$$M\gamma = \int_0^1 x \cdot p(x) dx = \int_0^1 x dx = \frac{1}{2},$$

$$D\gamma = \int_0^1 x^2 \cdot p(x) dx - (M\gamma)^2 = \frac{1}{3} - \frac{1}{4} = \frac{1}{12}.$$

Определение 1.2.7. *Непрерывная случайная величина ξ называется равномерно распределенной на интервале (a, b) , если ее плотность и интегральная функция распределения имеют вид*

$$p(x) = f_\xi(x) \begin{cases} \frac{1}{b-a} & , \quad a \leq x \leq b \\ 0 & , \quad x \notin (a; b) \end{cases}, \quad F(x) = F_\xi(x) = \begin{cases} 0 & , \quad x < a \\ \frac{x-a}{b-a} & , \quad a \leq x \leq b \\ 1 & , \quad x > b \end{cases}$$

Математическое ожидание и среднее квадратическое отклонение для равномерно распределенной на интервале (a, b) случайной величины:

$$M(X) = \int_{-\infty}^{+\infty} x \cdot f(x) dx = \dots = \frac{b+a}{2}, \quad \sigma(X) = M(X^2) - (M(X))^2 = \dots = \frac{b-a}{2\sqrt{3}}.$$

Упражнение: восстановить пропущенные вычисления.

Определение 1.2.8. *Нормальной (гауссовской) случайной величиной называется случайная величина $\zeta \in N_{a,\sigma}$, определенная на всей числовой оси и имеющая плотность*

$$p(x) = f_N(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-a)^2}{2\sigma^2}},$$

где a и σ — числовые параметры.

Изменение параметра a не влияет на форму кривой $p(x)$, его изменение приводит только к сдвигу всей кривой вдоль оси x . При изменении параметра σ форма кривой $p(x)$ меняется, так как

$$\max_x p(x) = p(a) = \frac{1}{\sqrt{2\pi}\sigma},$$

и если уменьшать σ , то $\max p(x)$ будет возрастать, а вся площадь под кривой $p(x)$ равна 1. Поэтому кривая "вытянется" вверх в окрестности точки $x = a$, но будет убывать при всех достаточно больших значениях x (см. рис. ??).

Функция распределения нормального распределения имеет вид

$$F(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(t-a)^2}{2\sigma^2}} dt.$$

При $a = 0$ и $\sigma = 1$ нормальное распределение называется стандартным: $X \in N_{0,1}$. Плотность стандартной СВ имеет вид

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad x \in R,$$

а ее интегральная функция распределения называется *функцией Лапласа*

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt.$$

$$\begin{aligned}\Phi(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^0 e^{-\frac{t^2}{2}} dt + \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{t^2}{2}} dt = \\ &= \frac{1}{\sqrt{2\pi}} \cdot \sqrt{\frac{\pi}{2}} + \Phi_0(x) = \frac{1}{2} + \Phi_0(x).\end{aligned}$$

Функция $\Phi_0(x)$ табулирована.

Найдем основные числовые характеристики показательного распределения и одновременно установим смысл параметров a и σ нормального распределения.

$$\begin{aligned}M(X) &= \int_{-\infty}^{\infty} x \cdot f(x) dx = \frac{1}{\sigma \cdot \sqrt{2\pi}} \int_{-\infty}^{\infty} x e^{-\frac{(x-a)^2}{2\sigma^2}} dx = \\ &= \left| t = \frac{x-a}{\sqrt{2}\sigma} \right| = \frac{1}{\sigma \cdot \sqrt{2\pi}} \int_{-\infty}^{\infty} (\sqrt{2}\sigma t + a) e^{-t^2} \sqrt{2}\sigma dt = \\ &= \frac{\sigma\sqrt{2}}{\sqrt{\pi}} \int_{-\infty}^{\infty} t e^{-t^2} dt + \frac{a}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-t^2} dt = 0 + \frac{a}{\sqrt{\pi}} \cdot \sqrt{\pi} = a,\end{aligned}$$

то есть математическое ожидание нормально распределенной случайной величины равно ее параметру a : $M(X) = a$. Первый интеграл равен нулю, так как подынтегральная функция нечетная, а пределы интегрирования симметричны относительно 0, а второй интеграл — Пуассона.

Теперь посчитаем дисперсию нормально распределенной случайной величины:

$$\begin{aligned}D(X) &= \int_{-\infty}^{\infty} (x-a)^2 f(x) dx = \int_{-\infty}^{\infty} (x-a)^2 \frac{1}{\sigma \cdot \sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}} dx = \\ &= \frac{1}{\sigma \cdot \sqrt{2\pi}} \int_{-\infty}^{\infty} 2\sigma^2 t^2 e^{-t^2} \sigma\sqrt{2} dt = \frac{2\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{\infty} t^2 e^{-t^2} dt = \\ &= \frac{2\sigma^2}{\sqrt{\pi}} \left(-\frac{1}{2} t e^{-t^2} \Big|_{-\infty}^{+\infty} + \frac{1}{2} \int_{-\infty}^{\infty} e^{-t^2} dt \right) = \frac{2\sigma^2}{\sqrt{\pi}} \cdot \frac{1}{2} \sqrt{\pi} = \sigma^2.\end{aligned}$$

Таким образом, $D(X) = \sigma^2$, то есть второй параметр нормального распределения σ является средним квадратическим отклонением нормально распределенной СВ.

Найдем вероятность попадания СВ $X \in N_{a,\sigma}$ на интервал (α, β) . Для этого воспользуемся формулой

$$P\{a < X < b\} = \int_a^b f(x) dx.$$

Получим

$$P\{\alpha < X < \beta\} = \frac{1}{\sigma\sqrt{2\pi}} \int_{\alpha}^{\beta} e^{-\frac{(x-a)^2}{2\sigma^2}} dx = \left| \frac{x-a}{\sigma} = t \right| =$$

$$= \frac{1}{\sqrt{2\pi}} \int_{\frac{\alpha-a}{\sigma}}^{\frac{\beta-a}{\sigma}} e^{-\frac{t^2}{2}} dt = \frac{1}{\sqrt{2\pi}} \int_0^{\frac{\beta-a}{\sigma}} e^{-\frac{t^2}{2}} dt - \frac{1}{\sqrt{2\pi}} \int_0^{\frac{\alpha-a}{\sigma}} e^{-\frac{t^2}{2}} dt.$$

Правило трех сигм.

Каковы бы ни были a и σ , величина интеграла

$$\int_{a+3\sigma}^{a-3\sigma} p(x) dx = P(a - 3\sigma < \zeta < a + 3\sigma) \approx 0,997.$$

Другими словами, при одном испытании практически невозможно получить значение ζ , отличающееся от $M\zeta$ больше, чем на 3σ (см. рис. ??).

Центральная предельная теорема.

Рассмотрим N одинаковых независимых случайных величин $\xi_1, \xi_2, \dots, \xi_N$ с совпадающими плотностями вероятностей, математическими ожиданиями и дисперсиями. Обозначим

$$M\xi_1 = M\xi_2 = \dots = M\xi_N = m,$$

$$D\xi_1 = D\xi_2 = \dots = D\xi_N = b^2.$$

Обозначим через ρ_N сумму всех этих величин

$$\rho_N = \xi_1 + \xi_2 + \dots + \xi_N,$$

тогда

$$M(\rho_N) = M(\xi_1 + \xi_2 + \dots + \xi_N) = Nm,$$

$$D(\rho_N) = D(\xi_1 + \xi_2 + \dots + \xi_N) = Nb^2.$$

Рассмотрим также нормальную случайную величину ζ_N с такими же параметрами: $a = Nm, \sigma^2 = Nb^2$.

Теорема 1. Плотность суммы ρ_N приближается к плотности нормально распределенной величины ζ_N при $N \rightarrow \infty$:

$$\lim_{N \rightarrow \infty} |p_{\rho_N}(x) - p(\zeta_N)(x)| = 0.$$

Таким образом, сумма большого числа одинаково распределенных случайных величин приблизительно нормальна: $p_{\rho_N}(x) \approx p(\zeta_N)(x)$.

Замечание. В общем случае слагаемые ξ_2, \dots, ξ_N не обязаны быть одинаковыми и независимыми. Существенно только то, чтобы отдельные слагаемые не играли слишком большой роли в сумме. Именно ЦПТ поясняет, почему нормально распределенные случайные величины часто встречаются в природе. Каждый раз, когда сталкиваются с суммарным воздействием большого числа незначительных случайных факторов, результирующая случайная величина оказывается *нормальной*.

1.2.2 Генераторы случайных чисел

Различают 3 способа получения случайных чисел: таблицы случайных чисел, генераторы случайных чисел и метод псевдослучайных чисел.

- **Таблицы случайных чисел.** Пусть проводится опыт, состоящий в следующем: на одинаковых карточках напишем цифры от 0 до 9. Положим эти карточки в урну, перемешаем и по одной будем их доставать, записывать полученное на карточке число. Далее карточка возвращается назад и перемешивается с остальными. Полученные таким образом цифры запишем в виде таблицы - Таблицы случайных чисел (цифр). Таковую таблицу можно использовать для определения значения, которое принимает случайная величина с равномерным дискретным распределением

X	0	1	...	9
p	0,1	0,1	...	0,1

Таблицы случайных чисел можно также получить, используя рулетку — диск, разбитый на 10 одинаковых секторов. Вращая этот диск и затем резко его останавливая, выбирают ту цифру, на которую покажет неподвижная стрелка. Составленные таблицы проверяются на "случайность" с помощью статистических тестов: не противоречат ли свойства группы чисел гипотезе о том, что данные числа — значения случайной величины.

Таблица случайных цифр

86515	90795	66155	66434	56558	12332	94377	57802
69186	03393	42502	99224	88955	53758	91641	18867
41686	42163	85181	38967	33181	72664	53807	00607
86522	47171	88059	89342	67248	09082	12311	90316
72587	93000	89688	78416	27589	99528	14480	50961
52452	42499	33346	83935	79130	90410	45420	77757
76773	97526	27256	66447	25731	37525	16287	66181
04825	82134	80317	75120	45904	75601	70492	10274
87113	84778	45863	24520	19976	04925	07824	76044
84754	57616	38132	64294	15218	89571	49286	42903

- В качестве генераторов случайных чисел используют, например, шумы в электронных лампах: если за фиксированный промежуток времени Δt уровень шума превысил заданный порог четное число раз, то записывают 0, а если нечетное число раз, то записывают 1.

Пусть m таких генераторов работают параллельно и постоянно засылают случайные нули и единицы в двоичные разряды ячейки. Каждый такт означает

m -разрядное двоичное число, которое рассматривается как приближенное значение $0, \alpha_1, \alpha_2, \dots, \alpha_m$ случайной величины γ , равномерно распределенной на интервале $(0, 1)$. Каждая величина α_i имитирует случайную величину с распределением

X	0	1
p	0,5	0,5

- Числа, получаемые по некоторой формуле, и имитирующие значения случайной величины γ , называются *псевдослучайными числами*. Слово "имитирующие" означает, что эти числа удовлетворяют ряду тестов так, если бы они были значениями случайной величины.

Первый алгоритм получения псевдослучайных чисел был получен Дж. Нейманом. Он называется методом середины квадратов. Приведем пример его использования.

Пусть задано четырехзначное число $n_1 = 9876$. Возведем его в квадрат. Получим $n_1^2 = 97535376$. Выберем 4 средние цифры из этого числа и обозначим $n_2 = 5353$. Возведем его в квадрат и снова возьмем 4 средние цифры. Получим $n_3 = 6546$, и т.д. В качестве значений случайной величины γ предполагалось использовать значения $0,9876, 0,5353, \dots$. Однако такой метод оказался неудачным, так как давал больше, чем нужно, малых значений.

Одним же из современных способов получения псевдослучайных чисел является применение формул Бокса-Мюллера (1.11).

1.3 Преобразования случайных величин

Значение любой случайной величины можно получить с помощью преобразования значений некоторой ("стандартной") случайной величины γ (в частности, равномерно распределенной на $(0, 1)$).

Процесс нахождения значения любой случайной величины ξ путем преобразования одного или нескольких значений γ называть разыгрыванием случайной величины ξ .

1.3.1 Разыгрывание дискретной случайной величины

Пусть нужно получить значение случайной величины ξ с распределением

X	x_1	x_2	\dots	x_n
p	p_1	p_2	\dots	p_n

Рассмотрим интервал $0 < y < 1$ и разобьем его на n интервалов, длины которых равны p_1, p_2, \dots, p_n . Перенумеруем эти интервалы. Каждый раз, когда нужно получить разыграть значение, будем выбирать значение γ и строить точку $y = \gamma$. Если эта точка попадет в интервал с номером i , то будем считать, что $\xi = x_i$.

Случайная величина γ равномерно распределена на $(0, 1)$, поэтому вероятность того, что γ окажется в этом интервале, равна длине этого интервала. Поэтому

$$\begin{aligned} P(0 < \gamma < p_1) &= p_1, \\ P(p_1 < \gamma < p_1 + p_2) &= p_2, \\ &\dots\dots\dots \\ P(p_1 + p_2 + \dots + p_{n-1} < \gamma < 1) &= p_n. \end{aligned}$$

Тогда $\xi = x_i$ тогда, когда

$$p_1 + p_2 + \dots + p_{i-1} < \gamma < p_1 + p_2 + \dots + p_i, \quad P(\xi = x_i) = p_i.$$

1.3.2 Разыгрывание непрерывной случайной величины

Предположим, что нужно получить значение случайной величины ξ , распределенной в интервале (a, b) с плотностью $p(x)$.

Докажем, что значения ξ можно находить из уравнения

$$\int_a^\xi p(x) dx = \gamma, \tag{1.1}$$

то есть выбрав очередное значение γ , надо решить уравнение (1.1) и найти очередное значение ξ .

Рассмотрим функцию

$$y = \int_a^x p(x) dx.$$

В силу свойств плотности $y(a) = 0$, $y(b) = 1$, $y'(x) = p(x) > 0$, то есть функция $y(x)$ строго монотонно возрастает от 0 до 1. Поэтому уравнение (1.1) имеет только одно решение.

Выберем произвольный интервал $(a', b') \subset (a, b)$. Точкам этого интервала $a' < x < b'$ отвечают ординаты кривой $y = y(x)$, удовлетворяющие неравенству $y(a') < y(x) < y(b')$. Поэтому если $\xi \in (a', b')$, то $\gamma \in (y(a'), y(b'))$ и наоборот. Значит,

$$P(a' < \xi < b') = P(y(a') < \gamma < y(b')).$$

Так как γ равномерно распределена в $(0, 1)$, то

$$P(y(a') < \gamma < y(b')) = y(b') - y(a') = \int_{a'}^{b'} p(x) dx,$$

последнее равенство означает, что случайная величина ξ , являющаяся корнем уравнения (1.1), имеет плотность вероятностей $p(x)$.

Метод Неймана

для разыгрывания непрерывной случайной величины.

В случаях, когда интеграл в уравнении (1.1) не выражается через элементарные функции или когда плотность $p(x)$ задана графически, разрешить уравнение (1.1) оказывается затруднительно.

Пусть случайная величина ξ определена на конечном интервале (a, b) и плотность ее ограничена:

$$p(x) \leq M_0.$$

Разыгрывать значение ξ можно следующим образом:

- Выбираем два значения случайной величины γ и строим случайную точку $\Gamma(\eta', \eta'')$ с координатами $\eta' = a + \gamma'(b - a)$, $\eta'' = \gamma'' M_0$;
- если точка Γ лежит под кривой $y = p(x)$, то эту пару (γ', γ'') отбрасываем и выбираем новую пару (γ', γ'') .

1.4 Примеры применения метода Монте-Карло

1.4.1 Расчет системы массового обслуживания

Рассмотрим простейшую систему массового обслуживания. Она состоит из n линий (или каналов, или пунктов обслуживания), каждый из которых может "обслуживать потребителей". В систему поступают заявки, причем моменты их поступления случайные. Каждая заявка поступает на линию номер 1. Если во время T_k поступления k -й заявки эта линия свободна, то она приступает к обслуживанию заявки, что продолжается t минут (t — время занятости линии.) Если в момент T_k линия номер 1 занята, то заявка мгновенно передается на линию номер 2 и т.д.

Если все n линий в момент T_k заняты, то система выдает отказ.

Требуется определить, сколько в среднем заявок обслужит система за время T и сколько отказов она даст?

Задачи такого типа встречаются при исследовании организации работы любых предприятий, а не только предприятий бытового обслуживания. В некоторых очень частных случаях удается найти аналитические решения. Однако в сложных случаях метод Монте-Карло оказывается единственным методом расчета.

Определение 1.4.1. Простейшим потоком (потоком Пуассона) называется такой поток заявок, когда промежуток времени τ между двумя последовательными заявками есть случайная величина, распределенная на интервале $(0; +\infty)$ с плотностью

$$p(x) = ae^{-ax}. \quad (1.2)$$

Закон 1.2 называют экспоненциальным распределением.

Математическое ожидание τ :

$$M\tau = \int_0^{+\infty} xp(x) dx = \int_0^{+\infty} xae^{-ax} dx.$$

Интегрируя по частям ($u = x$, $dv = ae^{-ax} dx$), получаем

$$M\tau = -xe^{-ax} \Big|_0^{\infty} + \int_0^{+\infty} e^{-ax} dx = -\frac{e^{-ax}}{a} \Big|_0^{+\infty} = \frac{1}{a}.$$

Определение 1.4.2. Параметр a называется плотностью потока заявок.

Формула для розыгрыша τ можно получить из уравнения 1.1, которое для экспоненциального распределения запишется так:

$$\gamma = \int_0^{\tau} ae^{-ax} dx \iff 1 - e^{-a\tau} = \gamma \iff \tau = -\frac{1}{a} \ln(1 - \gamma).$$

Так как величина $1 - \gamma$ имеет с γ одинаковое распределение, то

$$\tau = \frac{1}{a} \ln \gamma. \quad (1.3)$$

Теперь каждой линии системы массового обслуживания поставим в соответствие ячейку внутреннего накопителя ЭВМ, где будем фиксировать момент освобождения этой линии. Обозначим момент освобождения i -й линии через t_i . Начальным моментом отсчета выберем момент поступления первой заявки $T_1 = 0$. Можно считать, что в этот момент все t_i равны $T_1 = 0$, то есть все линии свободны. Время окончания расчета $T_{end} = T_1 + T$.

Первая заявка поступает на линию номер 1. Значит в течение t эта линия будет занята. Поэтому мы заменяем t_1 на новое значение $(t_1)_{new} = T_1 + t_3$, добавляем 1 к счетчику выполненных заявок и переходим к выполнению этой заявки.

Пусть k заявок уже рассмотрены. Тогда надо разыграть момент поступления $(k+1)$ -й заявки. Для этого выбираем очередное значение γ и по формуле (1.3) вычисляем очередное значение $\tau = \tau_k$. А затем вычисляем момент поступления

$$T_{k+1} = T_k + \tau_k.$$

Для того, чтобы установить, свободна ли первая линия в этот момент, нужно проверить условие

$$t_1 \leq T_{k+1}. \quad (1.4)$$

Если это условие выполнено, то к моменту T_{k+1} линия уже освободилась и может обслужить эту заявку. Мы должны заменить t_1 на $T_{k+1} + t$, добавить 1 к счетчику выполненных заявок и перейти к следующей заявке.

Если условие (1.4) не выполнено, то к моменту T_{k+1} первая линия занята. Тогда нужно проверить, свободна ли вторая линия, то есть выполняется ли неравенство

$$t_2 \leq T_{k+1}. \quad (1.5)$$

Если условие (1.5) выполнено, то заменяем t_2 на $T_{k+1} + t$, добавляем 1 к счетчику выполненных заявок и переходим к следующей заявке.

Если условие (1.5) не выполнено, то переходим к проверке условия

$$t_3 \leq T_{k+1}. \quad (1.6)$$

Может оказаться, что при всех i от 1 до n $t_i > T_{k+1}$, то есть все линии в момент T_{k+1} заняты. Тогда надо добавить 1 к счетчику отказов и потом перейти к следующей заявке.

Каждый раз, вычислив T_{k+1} , надо проверять условие окончания опыта

$$T_{k+1} > T_{end}.$$

Когда это условие окажется выполненным, опыт заканчивается. В счетчике выполненных заявок и в счетчике отказов будут стоять числа μ_{done} и μ_{rej} . Такой опыт повторяется N раз с использованием различных γ . Затем результаты всех опытов осредняются:

$$M\mu_{done} \approx \frac{1}{N} \sum_{j=1}^N \mu_{done}(j), \quad M\mu_{rej} \approx \frac{1}{N} \sum_{j=1}^N \mu_{rej}(j),$$

где $\mu_{done}(j)$ и $\mu_{rej}(j)$ — значения μ_{done} и μ_{rej} в j -м опыте.

Данный метод позволяет рассчитывать более сложные системы. Например, величина t может быть не постоянной, а случайной и различной для разных линий (что соответствует различному оборудованию или различной квалификации обслуживающего персонала). Схема расчета останется такая же, но значения t придется каждый раз разыгрывать и формула разыгрывания для каждой линии своя.

Можно рассматривать *системы с ожиданием*, в которых отказ выдается не сразу, заявка хранится в течение некоторого времени t_n (время пребывания заявки в системе), и если за это время какая-нибудь заявка освободится, то она обслужит эту заявку.

Можно рассматривать системы, в которых очередную заявку принимает та линия, которая раньше всех освободилась. Можно учесть случайный выход из строя отдельных линий и случайное время ремонта каждой из них. Можно учесть изменение плотности заявок во времени и т.д.

Чтобы получить результаты, имеющие практическую ценность, надо выбрать хорошую *модель*. Для этого необходимо тщательно изучать действительные потоки заявок, проводить хронометраж отдельных узлов, и т.д. Метод Монте-Карло позволяет вычислить вероятностные законы работы всей системы, насколько сложной она бы не была.

Такие методы расчета полезны при планировании предприятий, т.к. вместо дорогостоящего или невозможного эксперимента в реальности проводится численный эксперимент, моделирующий разные варианты организации работы или использования оборудования.

1.4.2 Расчет качества и надежности

Рассмотрим изделие S , состоящее из некоторого числа элементов. Например, если S — электрический прибор, то его элементами могут быть сопротивления $R_{(k)}$, емкости $C_{(k)}$ и т.д. Пусть качество изделия определяется значением одного выходного параметра U , которое можно вычислить, зная параметры всех элементов

$$U = f(R_{(1)}, R_{(2)}, \dots, C_{(1)}, C_{(2)}, \dots). \quad (1.7)$$

Если, например, U — это напряжение на рабочем участке электрической цепи, то по законам Ома можно составить уравнения для цепи и найти U .

На самом деле параметры элементов не равны в точности указанным значениям. Возникает вопрос: как повлияют отклонения параметров всех элементов от номинальных на значение U ?

Можно оценить пределы изменения U , выбирая для всех элементов "худшие" значения параметров. Но не всегда известно, какой набор будет действительно "худшим". Если число элементов велико, то такая оценка окажется завышенной, и маловероятно, чтобы все параметры одновременно оказались наихудшими.

Поэтому считают параметры всех элементов и саму величину U случайными величинами и оценивают математическое ожидание MU (среднее значение U всей партии изделий) и дисперсию DU , которая показывает, какие отклонения U от MU будут встречаться на практике, причем

$$MU \neq M(f(R_{(1)}, R_{(2)}, \dots, C_{(1)}, C_{(2)}, \dots)).$$

Вычислить аналитически распределение U практически невозможно, но можно сделать экспериментально, посмотрев большую партию готовых изделий. Последнее тоже возможно далеко не всегда, особенно только на стадии проектирования изделий.

Для применения метода Монте-Карло здесь необходимо:

- знать вероятностные характеристики всех элементов;
- знать функцию f (уметь вычислять значение U по любым фиксированным значениям $(R_{(1)}, R_{(2)}, \dots, C_{(1)}, C_{(2)}, \dots)$).

Вероятностное распределение параметров для каждого отдельного элемента можно получить экспериментально, и часто это распределение *нормально*. Таким образом, схема расчета состоит в следующем: для каждого элемента разыгрывается значение параметра, затем по формуле (1.7) вычисляется значение U . Повторив опыт N раз и получив значения U_1, U_2, \dots, U_N , приблизительно считаем, что

$$MU \approx \frac{1}{N} \sum_{j=1}^N U_j, \quad DU \approx \frac{1}{N-1} \left[\sum_{j=1}^N (U_j)^2 - \frac{1}{N} \left(\sum_{j=1}^N U_j \right)^2 \right].$$

При моделировании можно получить больше информации, чем только математическое ожидание дисперсию интересующей величины. Например, в результате N опытов получили N значений случайной величины. Тогда, по этим значениям, можно построить и приближенную плотность распределения на основании экспериментально построенной *гистограммы*, которая и служит приближением к неизвестной плотности случайной величины U .

1.4.3 Вычисление определенного интеграла

Рассмотрим функцию $g(x)$, $x \in (a, b)$. Требуется вычислить приближенно интеграл

$$I = \int_a^b g(x) dx. \quad (1.8)$$

Выберем произвольную плотность распределения $p_\xi(x)$, $x \in (a, b)$. Наряду со случайной величиной ξ , определенной на интервале (a, b) с плотностью $p_\xi(x)$, введем случайную величину

$$\eta = \frac{g(\xi)}{p_\xi(\xi)}.$$

Тогда

$$M\eta = \int_a^b \left(\frac{g(\xi)}{p_\xi(\xi)} \right) p_\xi(\xi) dx = I.$$

Теперь рассмотрим N одинаково распределенных $\eta_1, \eta_2, \dots, \eta_N$ и применим к их сумме ЦПТ и правило трех сигм:

$$P \left\{ \left| \frac{1}{N} \sum_{j=1}^N \eta_j - I \right| < 3\sqrt{\frac{D\eta}{N}} \right\} \approx 0,997. \quad (1.9)$$

Последнее означает, что если мы выберем N значений $\xi_1, \xi_2, \dots, \xi_N$, то при достаточно большом N

$$\frac{1}{N} \sum_{j=1}^N \frac{g(\xi_j)}{p_\xi(\xi_j)} \approx I. \quad (1.10)$$

Ошибка приближения (1.10) с большой вероятностью не превосходит $3\sqrt{D\eta/N}$.

Для расчета интеграла (1.8) можно использовать любую случайную величину ξ , определенную в (a, b) . В любом случае

$$M\eta = M(g(\xi)/p_\xi(\xi)) = I.$$

Но дисперсия D_{eta} и оценка погрешности формулы (1.10) зависят от выбора ξ . Примем без доказательства, что $D(\xi)$ будет минимальной, если $p_\xi(x)$ пропорциональна $|g(x)|$.

На практике интегралы вида (1.8) вычисляют редко, так как для этого есть более точные методы — квадратурные формулы. Однако при переходе к многократным

интегралам квадратурные формулы усложняются, а метод Монте-Карло остается почти без изменений.

1.5 Индивидуальные задания

1.5.1 Генерация нормальных случайных чисел с помощью преобразование Бокса-Мюллера

Задание 1.

1. Создать две исходные совокупности r и φ равномерно распределенных на интервале $(0; 1)$ случайных чисел. (Использовать, например, функцию СЛЧИС(), Excel).

2. Моделировать числа, имеющих стандартное нормальное распределение, применяя формулы *Бокса-Мюллера* для генерации чисел, имеющих стандартное нормальное распределение:

$$\boxed{z_0 = \cos(2\pi\varphi) \cdot \sqrt{-2 \cdot \ln r}}, \quad \boxed{z_1 = \sin(2\pi\varphi) \cdot \sqrt{-2 \cdot \ln r}}. \quad (1.11)$$

3. Моделировать числа, имеющих нормальное распределение с математическим ожиданием a , равным номеру студента в журнале, и дисперсией σ^2 , равной номеру группы, используя формулу $Y = a + \sigma \cdot X$.

1.5.2 Разыгрывание дискретных и непрерывных случайных величин

Общая постановка задачи: найти значение a изучаемой случайной величины.

Метод решения: Выбрать случайную величину X такую, что ее математическое ожидание совпадает со значением моделируемой случайной величины: $MX = a$.

Далее по выборке n значений случайной величины X вычислить выборочное среднее

$$a^* = \bar{X} = \frac{\sum_{i=1}^n x_i}{n}.$$

Принять полученную *точечную оценку* искомого числа (значение моделируемой случайной величины) $a = a^*$.

Пусть требуется разыграть дискретную случайную величину X , то есть получить последовательность ее возможных значений, зная ее закон распределения вероятностей

X	x_1	x_2	\dots	x_n	(*)
p	p_1	p_2	\dots	p_n	

Разыгрывание дискретной случайной величины

Рассмотрим равномерно распределенную случайную величину R на $(0; 1)$ и разобъем интервал $(0; 1)$ точками с координатами $p_1, p_1 + p_2, \dots, p_1 + p_2 + \dots + p_n$ на n частичных интервалов $\Delta_1, \Delta_2, \dots, \Delta_n$.

Теорема 2. Если каждому случайному числу $r_j, 0 \leq r_j < 1$, которое попало в интервал Δ_i , ставить в соответствие возможное значение x_i , то разыгрываемая дискретная случайная величина будет иметь заданный закон распределения (*).

Пример. Разыграем 10 значений дискретной случайной величины X , закон распределения которой известен:

X	2	3	6	8
p	0,1	0,3	0,5	0,1

1. Разобъем интервал $(0;1)$ на частичные интервалы $\Delta_i, i = 1, 2, 3, 4$:

Δ_1	Δ_2	Δ_3	Δ_4
$(0; 0, 1)$	$(0, 1; 0, 4)$	$(0, 4; 0, 9)$	$(0, 9; 1)$

2. Генерируем 10 равномерно распределенных на $(0;1)$ чисел

0, 09; 0, 73; 0, 25; 0, 33; 0, 76; 0, 52; 0, 01; 0, 35; 0, 86; 0, 34.

Посчитаем количество случайных чисел, попавших в каждый частичный интервал.

В интервал $\Delta_1 = (0; 0, 1)$ попали 1-е и 7-е числа (два числа), поэтому разыгрываемая дискретная случайная величина X принимает значение $x_1 = 2$.

В интервал $\Delta_2 = (0, 1; 0, 4)$ попали 3-е, 4, 8 и 10-е числа (четыре числа), поэтому разыгрываемая дискретная случайная величина X принимает значение $x_2 = 3$.

В интервал $\Delta_3 = (0, 4; 0, 9)$ попали 2-е, 5, 6 и 9-е числа (четыре числа), поэтому разыгрываемая дискретная случайная величина X принимает значение $x_3 = 6$.

В Δ_4 пусто.

Поэтому разыгранные возможные значения случайной величины X :

2, 6, 3, 3, 6, 6, 2, 3, 6, 3.

Задание 2. Разыграть $10 + N$ (N — порядковый номер фамилии студента в журнале) значений дискретной случайной величины X , если

- X распределена по биномиальному закону с параметром $1/3$, при четном номере группы студента;
- X распределена по геометрическому закону с параметром $4/5$ при нечетном номере группы.

Разыгрывание непрерывной случайной величины

Пусть требуется разыграть непрерывную случайную величину X , то есть получить последовательность ее возможных значений x_i , $i = 1, 2, \dots, n$ зная интегральную функцию распределения $F(x)$.

Теорема 3. (Метод обратных функций) Если r_j — случайное число, $0 \leq r_j < 1$, то возможное значение x_i разыгрываемой случайной величины X с заданной интегральной функцией распределения $F(x)$, является корнем уравнения $F(x_i) = r_i$.

Пример. Разыграем 3 значения непрерывной случайной величины X , распределенной равномерно на интервале $(5; 8)$.

Имеем

$$F(x) = \frac{x - 5}{3} \Leftrightarrow \frac{x_i - 5}{3} = r_i \Leftrightarrow x_i = 3r_i + 5.$$

Создаем 3 случайных числа 0,23; 0,09; 0,56. Решаем три уравнения относительно x_i .

Задание 3. Разыграть $K + N$ значений непрерывной случайной величины X , если

$$F(x) = \frac{x - K}{N},$$

K — номер группы, N — номер студента в журнале.

Глава 2

ОСНОВЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ В СРЕДЕ

Ithink

В основе идеологии построения и применения пакета *Ithink* лежат успехи в реализации современных Desk-Top—графических интерфейсов, достижения по разработке и реализации алгоритмов имитации динамического поведения, а также осознание того, что оперативное общение менеджера с компьютером более предпочтительнее на идеографическом уровне, чем даже на уровне развитого буквенного письма (тем более, уровня традиционных языков-запросов, имеющих, как правило, алгебраическую основу).

Действительно, процесс формирования первичных представлений предметной области можно свести к процессу „рисования“ простейших идеограмм, скрывающих подчас за внешней примитивностью весьма содержательный смысловой материал.

Дальнейшее уточнение, усовершенствование и усложнение моделей сводится практически к „рисованию“ обратных связей и циклов, а также к композициям таких идеограмм в более развернутые графические ансамбли.

Трудно не согласиться, что большинство экономических процессов, так или иначе, развернуты во времени и имеют дело с разного рода накапливаемыми фондами (резервы, запасы, кредиты, объемы, денежные средства и т.д.), управлению которыми наиболее соответствуют схемы потокового типа. Подобные схемы чрезвычайно привлекательны для исследования самых разнообразных сложных явлений деловой и финансовой активности, таких как организация безналичных расчетов и кассовая работа банков, планирование и регулирование производственных и финансовых ресурсов, определение и выбор вариантов капиталовложений, организация управлением производственными процессами и т.д. Потоковые схемы предполагают отображение их поведения динамикой функционирования всем нам хорошо известной „водопроводной“ сети. Тогда нетрудно представить и возможное управление входными и вы-

ходными потоками с помощью „вентилей“, регулирующих интенсивность, скорость, дискретную задержку по тактам модельного времени и другие самые разнообразные характеристики и режимы наполнения и исчерпывания соответствующих фондов. Далее остается лишь „офизичить“ задачу и попытаться описать ее в терминах „водопровода“ достаточно простыми идеографическими средствами пакета *Ithink*.

Потоковые сети, несмотря на свою кажущуюся простоту и наглядность, позволяют строить весьма нетривиальные механизмы управления, что делает возможным использовать их на уровне обобщенных, концептуальных представлений методологии реинжиниринга. Тогда, уже на самых ранних стадиях обдумывания и предварительного анализа проблем, можно ввести в модель прецизионные механизмы реакции на возможные промахи, локальные бедствия и риски. А это, в свою очередь, поможет выявлению и оценке потенциальных резервов по демпфированию нежелательных последствий в случаях формирования таких стратегических решений, как реинвестирование части банковского капитала в бизнес, выбор финансовой или инвестиционной компании на рынке поглощения и образования синергизма, сокращение или расширение сети торговых филиалов и т.д.

Потоковые диаграммы *Ithink* - это идеографическое изображение моделей на среднем, базовом уровне представления. Ниже располагается уровень соответствующих исполняемых спецификаций, который доступен и прост для прочтения программистом, но для Вас не представляет особого интереса (хотя, при желании, в нем достаточно просто и быстро можно разобраться). Отдельные фрагменты потоковых диаграмм можно инкапсулировать во фреймы процессов с соответствующими информационными и управляющими соединениями и получать адекватные графические изображения на самом внешнем уровне в виде наборов квадратов, соединенных стрелками.

2.1 Основные инструменты и меню *Ithink*

Ithink предоставляет в Ваше распоряжение следующий набор инструментов, с помощью которых, формируются поведенческие потоковые модели:

- „РУКА“, „КИСТЬ“, „ДИНАМИТ“— на высокоуровневом представлении моделей и самом низком уровне выполняемых спецификаций;
- „РУКА“, „КИСТЬ“, „ДИНАМИТ“ и „ФАНТОМ“— на базовом уровне потоковых схем.

РУКА - многоцелевой инструмент редактирования. В основном, он используется для выбора, передвижения, открытия, а также редактирования строительных блоков и объектов. Практически в любой ситуации при использовании *Ithink* Вы имеете дело с инструментом РУКА. Так как РУКА воспринимается как особый, привилегированный инструмент, после использования любого другого инструмента управление по умолчанию передается инструменту РУКА.

КИСТЬ служит для дополнительной раскраски элементов модели. Подержав некоторое время мышью указатель на кончике кисти пиктограммы, можно вызвать палитру цветов. Цвет контура выбирается щелчком правой кнопки мыши на соответствующем цвете, а цвет заливки - щелчком левой кнопки мыши. Тяните мышью - и у Вас окажется „в руке“ вместо стрелки-указателя изображение кисти. Протаскивайте мышью кисть до объекта, который нуждается в раскраске и один раз щелкните мышью. Объект (или его контур) окрасится выбранным цветом.

ДИНАМИТ предназначен для мгновенного стирания объекта и не нуждается на первых шагах знакомства с *Ithink* в особых разъяснениях.

ФАНТОМ является чрезвычайно эффективным инструментом при конструировании топологически сложных потоковых схем. В таких схемах коннекторы настолько перепутываются, что невозможно отследить и наглядно представить взаимодействия, а это в свою очередь сводит на нет все преимущества и достоинства идеографического подхода. С помощью ФАНТОМ-ов можно создавать копии, „призраки“ объектов, а затем - размещать их рядом с изначально удаленными объектами и тем самым „распутывать“ и упрощать исходные потоковые схемы, делая их наглядными и структурированными.

Набор инструментов *Ithink* можно увидеть практически на любом предыдущем рисунке, где он располагается поверх операционного окна правее набора строительных блоков. Динамический характер, который присущ любому факту применения инструментов, делает практически невозможным на статическом рисунке продемонстрировать их функционирование на сколь-нибудь содержательном примере, поэтому Вам рекомендуется оценить достоинства инструментария *Ithink* непосредственно на компьютере.

Объекты

Ithink предоставляет в Ваше распоряжение следующий основной набор объектов, с помощью которых комментируются, преобразуются и отображаются входные данные и результаты имитационных экспериментов:

- „ТЕКСТ“, „ГРАФИК“, „ТАБЛИЦА“ и „ЧИСЛОВОЙ ИНДИКАТОР“ — на высокоуровневом представлении моделей;

- „ТЕКСТ“, „ГРАФИК“, „ТАБЛИЦА“, „ЧИСЛОВОЙ ИНДИКАТОР“, „ФРЕЙМ“ и „КОМПРЕССОР“ — на базовом уровне потоковых схем.

Свойства объектов, как и инструментов, доопределяются соответствующими диалоговыми окнами.

Здесь можно увидеть, что на одной странице можно расположить несколько графиков поведения элементов потоковых диаграмм, собрать страницы в стопку, „закрыть на замок“ любую страницу стопки и т.д.

При отладке правильности функционирования модели полезно параллельно графикам формировать соответствующие таблицы, которые аналогичным образом определяются своими диалоговыми окнами.

Следует упомянуть о таких специфических объектах *Ithink*, как ФРЕЙМ и КОМПРЕССОР, подразумевающих средства композиции функционально близких элементов потоковых идеограмм. Композиция через ФРЕЙМ действует на базовом уровне представления и автоматически отражается на самом верхнем уровне представления в конструкциях соответствующих подмоделей. КОМПРЕССОР действует в рамках лишь базового уровня, формируя новый тип объекта и способствуя существенному структурному упрощению потоковых схем повышенной сложности. Техника работы с подмоделями требует более тщательного и подробного рассмотрения и не укладывается в рамки первоначального знакомства с пакетом. Тем не менее Вам предстоит в дальнейшем столкнуться с необходимостью постепенного освоения этого механизма по мере усложнения и расширения круга Ваших моделей и проблем.

Фонды

Фонд (Stock) можно понимать как резервуар или емкость, содержащую в данный момент времени определенное количество чего-либо. Фонды в *Ithink* изображаются прямоугольниками. Для того чтобы поместить фонд на рабочее поле, необходимо поставить РУКУ на кнопку Stock панели инструментов и щелкнуть левой клавишей мышки, затем переместить РУКУ в нужное место и еще раз щелкнуть левой клавишей мышки. Выбранному фонду можно присвоить имя, набрав его на клавиатуре. Фонды пополняются через входящие потоки и растрачиваются через выходящие потоки.

Фонд в *Ithink* более всего похож на:

- долговременную память;
- буфер;
- ресурс;
- резервуар.

Как буфер, фонд может использоваться для обеспечения балансировки скорости накапливания и исчерпывания. Такая роль фонда - основа концепции динамического поведения потоковых моделей *Ithink*.

Идеограммы фондов в качестве буферов можно активно использовать в потоковых моделях самых разнообразных сфер деловой активности, где в центре внимания стоит задача организации баланса процессов.

Займы, доходы от продаж, выпуск акций или продажа активов - по своей природе ресурсы пополняемые. Текущие расходы, текущие активы, фиксированные активы, сокращения пассивов и выплаты собственникам - ресурсы исчерпываемые. И

пополняемые, и исчерпываемые ресурсы в *Ithink* целесообразно интерпретировать идеограммами фондов.

И все же наиболее похожи фонды *Ithink* на резервуары. В то время, как предыдущие три аналога (долговременная память, буфер и ресурс) представляют достаточно ограниченные модельные ситуации, фонд как резервуар - многоцелевой модельный механизм, поддерживающий практически все возможные варианты интерпретации динамики потоковых процессов.

Потоки

Поток - это процесс, протекающий непрерывно во времени, оценить который можно в физических или денежных единицах, соотнесенных с каким-либо временным интервалом (рубли в месяц, литры в час, стоимость акции на время закрытия биржи в данный день и т.д.). В *Ithink* поток изображается фигурой состоящей из путепровода, вентиля, регулятора потока и указателя направления.

Поток „течет“ по путепроводу в направлении указателя. С помощью вентиля возможно управлять интенсивностью потока через значение, задаваемое некоторым алгебраическим выражением. Значение выражения увеличивается - вентиль приоткрывается шире, значение уменьшается - вентиль зауживает путепровод. Можно также сказать, что потоки через свои регуляторы воздействуют на вентили, а те, в свою очередь, управляют активностями.

По характеру использования, потоки подразделяются на ограниченные и неограниченные, однонаправленные и двунаправленные, конвертируемые и неконвертируемые.

Поток, как правило, ограничивается фондом. Однако нередки случаи, когда модельная ситуация требует использования неограниченных потоков, и тогда *Ithink* прикрепляет к соответствующему источнику или приемнику „облако“ неограниченности. Действительно, как можно ограничить поток заказов при стихийно складывающихся ценах на рынке потребительских товаров? Какими рамками можно ограничить степень доверия к той или иной финансовой компании? Наконец - каким образом количественно выразить границы существования таких идеализируемых концептов, как поток возможных инвестиций, перманентное и неконтролируемое исчерпывание природных ресурсов и т.д.

Кроме того, встречаются модельные ситуации, требующие использования двунаправленных потоков. Например, цены на рынке колеблются, а при этом, желательно использовать лишь один поток, характеризующий эти колебания. Или - динамика изменения заработной платы, которая не только будет расти в связи с инфляцией, но и - падать в зависимости от конъюнктуры рынка труда и складывающихся цен на потребительские товары и услуги.

Наконец, потоки иногда требуют конвертации своего содержимого для того, чтобы поставить в соответствие объемы соединяемых потоками фондов. Действительно,

они соотносятся, как правило, в отношении 1 : 1 (наличные деньги налогоплательщиков и Государственный бюджет, материальные запасы оптовика и наличие товаров у розничных торговцев, объем хранилища топлива на бензозаправочной станции и объем бензобака при заправке и т.д.). Однако, как без конвертации соотнести количество комплектующих к числу собранных агрегатов при монтаже; или - количество бревен к числу досок при распилке (соотношение 1 : n)? Подобные проблемы разрешаются соответствующими средствами *Ithink*, которые будут продемонстрированы по мере необходимости на конкретных примерах.

Конверторы

Конверторы в *Ithink* - преобразователи модельных единиц, которые можно использовать для детализации и уточнения поведения потоковых схем. Изображаются они окружностями, соединенными с другими элементами стрелками-коннекторами (см. **коннекторы**). Конверторы часто используются в качестве таких переменных (или заменяющих их алгебраических выражений), как доход, цена, численность, рейтинг и т.д. В противоположность фондам, конверторы - это не память и они не умеют ничего аккумулировать; их значения пересчитываются в каждый такт модельного времени.

Конверторы могут выполнять роль механизмов уточнения и эквивалентной замены фондов. Однако чаще всего конверторы используются в роли обыкновенных промежуточных переменных и служат для управления взаимодействиями, регулируя интенсивность потоков. Значения конверторов могут задаваться полноценными алгебраическими выражениями с использованием представительного набора встроенных функций.

Так как конверторы - не аккумуляторы, у них нет времени на задержку и последний конвертор в цепочке конверторов срабатывает синхронно с первым. Это обстоятельство, при несоблюдении осторожности, может привести к фатальным закликиваниям. Однако это сигнализирует и о том, что в модели недостает одного или нескольких аккумуляторов. Коллизия, как правило, разрешается добавлением структур фонд/поток.

Наконец, конверторы используются в роли внешних генераторов, как например - генераторов серий значений модельных единиц в привязке к интервалам имитационного времени, а также в качестве графических функций.

Графические функции - часто используемое и эффективное средство *Ithink* для параметризации и управления активностью потоковых моделей. Работа с графическими функциями организуется *Ithink*-ом через специальные диалоговые механизмы, подробно описанные в сопроводительной документации и кратко представленные в соответствующих help-ах.

В *Ithink* определены следующие три способа использования графических функций:

- для графического отображения конвертором или регулятором потока произвольно задаваемых дискретных значений (в частности - значения для регулятора потока могут задаваться по оси имитационного времени через вызов встроенной функции TIME);
- для графического отображения конвертором или регулятором потока дискретных значений содержимого фонда;
- для графического отображения конвертором или регулятором потока эффектов воздействия идеализированных понятий (эффекта от стимулирования — на количество единиц отгружаемой продукции, предрасположенности к финансовому риску — на колебание курса акций, коммерческого преувеличения при рекламе — на рост объема продаж и т.д.).

Коннекторы

Коннекторы в *Ithink* служат для логической связи всех других элементов потоковых диаграмм между собой. В предыдущих примерах мы уже их использовали.

Коннектор может соединять в логические пары конвертор с потоком, фонд с потоком, фонд с конвертором, поток с потоком, поток с конвертором и конвертор с конвертором. Подробности, связанные с особенностями соединений, раскрываются в дальнейшем по мере обсуждения конкретных схем и подтверждаются текстами соответствующих help-ов и сопроводительной документацией пакета.

Обратите внимание, что стрелка коннектора никогда не указывает на фонд, так как все управление фондами в *Ithink* всегда осуществляется через входные и выходные потоки.

Коннекторы, как и потоки, могут участвовать в конструировании блок-схем, которые *Ithink* интерпретирует как самый внешний уровень представления моделей.

Меню

Меню в *Ithink* располагаются под строкой заголовка пакета, внешне организованы подобно меню в Windows и содержат следующие пункты для каждого из трех вышеперечисленных уровней представления модели: - File, Edit, Map & I/O, Run, Help - File, Edit, Diagram, Run, Help - File, Edit, Equations, Run, Help

В левом нижнем углу программного окна *Ithink* расположена кнопка „бегущий человек“, дублирующая функции пункта меню Run. Эта кнопка (как и пункт меню Run) определяет весь диапазон разнообразных услуг, которые предоставляются *Ithink* для реализации имитационных экспериментов над потоковыми моделями. Подробности об этом можно почерпнуть из сопроводительной документации, или же обратившись за справкой к пункту меню Help. Главенствующая форма общения, которая определяется самой сутью *Ithink* - это общение с помощью мыши. Однако позволительно также, при желании, активно использовать клавиатуру, или же -

комбинировать то и другое. Зачастую при работе с меню предпочтительнее все же клавиатура.

Пользуясь меню, Вы выбираете для себя то, что Вам в данный момент необходимо или желательно. Чтобы не запутаться в большом количестве предоставляемых возможностей, они разделены на специальные группы, так называемые подменю. Каждое такое подменю имеет свое собственное название, представленное в основном меню. Так, подменю Diagram входит в основное меню базового представления модели.

При выходе на любой уровень представления модели, Вы попадаете в основное меню, что можно также сделать нажатием <Alt> , или передвижением мышью стрелки указателя (далее будем считать, что при работе с меню для каждого действия с клавиатурой предусмотрен соответствующий „мышиный“ эквивалент). То, что какой-то пункт меню активен, определяется по инверсному представлению надписи. Выбор между пунктами меню осуществляется клавишами перемещения курсора. Альтернативный вариант состоит в выборе подчеркнутой буквы в соответствующем названии. Например, в меню RUN подчеркнута буква R . Если Вы нажмете <R>, выделенным окажется пункт RUN. Тем самым можно избавиться от целого ряда ненужных нажатий клавиш. Нажмите, например, <Alt>+<R>, и на экране сразу появятся все команды меню RUN.

Внутри подменю отдельные команды выбираются при помощи клавиш „стрелка вверх“ и „стрелка вниз“, а подтверждение выбора осуществляется нажатием клавиши <Enter>. Более быстрый выбор соответствующих команд осуществляется нажатием клавиши с подчеркнутой буквой соответствующей команды. Команды, за которыми следуют точки, не подлежат моментальному выполнению. Их можно назвать расширенными командами. Выбор соответствующей расширенной команды повлечет за собой появление диалогового окна.

Некоторые команды включают в себя соответствующие подменю. Они помечены треугольником. Функциональные меню соответствующего подменю открываются сбоку и содержат упорядоченные по группам команды.

Если команда выведена не темными, а бледно-серыми буквами, то она не может в данный момент выполниться, т.е. является недоступной. Причина состоит в отсутствии необходимых условий для выполнения данной команды. Так, нельзя исполнить команды Pause или Stop, если модель не запущена на имитирование командой Run.

Ниже приведен пример раскрытия типичных для *Ithink* пунктов меню.

File		Run	
New	Ctrl+N	Run	Ctrl+R
Open	Ctrl+O	Pause	
Close Model		Stop	
Close Window	Ctrl+W	Sector Specs..	
Save	Ctrl+S	Sensi Specs..	Ctrl+Y
Save As		Time Specs..	
Save As PICT		Range Specs..	
Revert			
Page Setup..			
Print Map	Ctrl+P		
Author Prefs			
Exit	Ctrl+Q		

В тех случаях, когда какая-то функция не может быть вызвана при помощи простой команды, *Ithink* предоставит в Ваше распоряжение диалоговое окно, подобно тому, как это происходит в Windows, когда необходимо перед выполнением данной команды сообщить некоторую, более подробную информацию. Если, например, модель надо сохранить на жестком диске, пакету необходимо знать, на какой логический диск, в какой директорий и под каким именем следует записать информацию.

2.2 Базовые потоковые процессы *Ithink*

Динамика процессов и систем, как бы они сложны ни были, выражаются в *Ithink* через функционирование всего пяти типов базовых конструкций: - приращение фондов; - исчерпывание фондов; - рабочий процесс; - совмещение потоков; - адаптация фондов. Каждая из этих конструкций входит практически в любые инфраструктуры современного менеджмента; осознанное и грамотное их использование - ключ к существенному сокращению времени проектирования и более полному пониманию поведения моделей.

Приращение фондов

Процесс приращения фондов используется в ситуациях, когда необходимо организовать регулирование роста фонда интенсивностью его приращения.

Такой тип базового процесса находит свое применение в самых разнообразных сферах деловой активности и управляется соответствующим входным потоком через вычисление значения выражения

$$\text{входной поток} = \text{фонд} * \text{интенсивность приращения.}$$

Пример на Рис. демонстрирует простейшую модель годового роста сальдо банковского счета, скорость которого зависит от колеблющейся ежемесячной процентной

ставки. Хорошо известно, что если после очередного интервала начисления доход (т. е. начисленные за данный интервал проценты) не выплачиваются, а присоединяются к денежной сумме, имеющейся на начало этого интервала, для определения наращенной суммы применяют формулы сложных процентов. Такие сложные ссудные проценты в настоящее время являются весьма распространенным видом процентных ставок.

Рассматриваемая модельная ситуация и результаты имитационного эксперимента в *Ithink* компактно и лаконично изображаются потоковой диаграммой, таблицей и графиком.

Предполагается, что процентная ставка задается через конвертор interest rate соответствующей графической функцией, текущие значения которой также можно видеть в таблице.

Исчерпывание фондов

Процесс исчерпывания фондов используется в ситуациях, когда необходимо организовать модельное регулирование феноменов упадка, старения и износа.

На Рис. демонстрируются примеры применения процесса исчерпывания, управляемого вычислением значения выражения

$$\text{выходной поток} = \text{фонд} * \text{интенсивность исчерпывания.}$$

Следующий пример взят из области исследования процессов и явлений в монетарной политике, опирающейся на способность денежно-кредитной системы влиять на денежное предложение и, соответственно, на ставку процента, а через нее и на инвестиции и реальный валовой национальный продукт.

На Рис. демонстрируется кривая спроса на денежные средства, которая отражает взаимосвязь количества денег, которое фирмы и население предпочитает держать у себя в настоящий момент, с процентной ставкой, выплачиваемой по обычным неденежным активам, которая тесно связана с альтернативными издержками владения денежными средствами. Кривая спроса, представленная исчерпыванием фонда „денежная наличность“, имеет негативный наклон, что означает - по мере роста ставки процента по неденежным активам люди будут меньше хотеть держать деньги у себя в наличности и больше желать неденежных активов, таких как, например, Государственные краткосрочные обязательства, краткосрочные казначейские векселя или акции промышленных компаний. И наоборот, по мере падения ставки процента, владение неденежными активами будет сокращаться и возрастет владение денежными средствами. Регулирование оттоком наличных денежных средств осуществляется выходным потоком „спрос“. Процентная ставка в примере задается соответствующей графической функцией.

Рабочий процесс

Рабочий процесс можно считать наиболее часто используемой потоковой структурой, отражающей семантику производственной и деловой активности. Применяется в ситуациях изменения содержимого разнообразных фондов, когда дисциплина такого изменения определяется наперед заданным базовым ресурсом и интенсивностью (производительностью) его исчерпывания.

На Рис. демонстрируются примеры рабочих процессов, управляемых вычислением значения выражения

$$\text{рабочий процесс} = \text{ресурс} * \text{производительность}.$$

В условиях развивающейся рыночной экономики в центре внимания — разнообразные процессы, связанные с вложениями денежного капитала в различного вида ценные бумаги (долевое участие в предприятиях, займы другим предприятиям под векселя или иные долговые обязательства). Естественно желание сохранения капитала от обесценения в условиях инфляции. Простейшая модель выплаты процентов по долговым обязательствам при колеблющейся ставке процента (задаваемой графической функцией конвертора ставка процента) продемонстрирована на Рис. Долговые обязательства определяются депозитным сертификатом номиналом 300000\$, выданным на 1 год с ежемесячной выплатой процента, колеблющейся от 0,01 до 0,03.

Совмещение потоков

Процесс совмещения потоков используется при необходимости выразить факт взаимодействия процесса с некоторым другим, управляющим и протекающим параллельно процессом. Интенсивность и характер воздействия управляющего процесса определяется значениями конверсионного коэффициента. На Рис. приведены примеры, демонстрирующие применение базового процесса совмещения потоков в разнообразных контекстах и сферах деловой активности, где "управление" определяется вычислением значения выражения

$$\text{управление} = \text{управляющий поток} * \text{конверсионный коэффициент}.$$

Предполагается, что взаимодействующие процессы развиваются параллельно (практический опыт и обучение в высшем учебном заведении, успешная генерация объектного кода программы и возникновение ошибок в текстах программ, наем рабочих и проблемы повышения производительности на производстве и т.д.), однако один из них зачастую понимается как управляющий, от которого в той или иной мере зависит характер поведения процесса управляемого (практический опыт способствует успеху учебного процесса, поспешная и необдуманная кодогенерация неотлаженных программ является причиной вала ошибок, дополнительный наем рабочих как-то корреспондируется с изменением производительности труда и т.д.). Именно подобные несимметричные взаимодействия определяют семантику совмещения потоков в

рамках специальной конструкции базового процесса и могут оказать помощь в исследовании нетривиальных модельных ситуаций.

Пример на Рис. показывает простейшую модель возможной организации взаимодействия между типичными бизнес-процессами маркетинга через совмещение потоков „продажи“ и „выручка“. Здесь в качестве конверсионного коэффициента используется колеблющаяся цена на единицу продукции, а в качестве управляющего воздействия - синхронизированные по модельному времени с ценой факты продаж.

Адаптация фондов

Эта потоковая конструкция применяется при необходимости отобразить в *Ithink* бизнес-процессы, в которых содержимое фонда адаптируется к меняющимся условиям и приводится в некоторое состояние равновесия. Естественно, что такая интерпретация возможна лишь при использовании в конструкции двунаправленного потока.

Конструкция, в силу своей нетривиальной семантики, представляет несомненный интерес в проектировании сложных поведенческих моделей и требует к себе повышенного внимания. Грубые аналогии подобных модельных ситуаций можно встретить в самых разнообразных сферах проявления активностей. Так, процесс адаптации фонда можно усмотреть при наблюдении физических процессов нивелирования температуры по замкнутому объему газового потока, при исследовании эффекта выравнивания объема популяции в условиях установления экологического равновесия, при попытках определения усредненных значений рейтинга общественного деятеля в периоды политического затишья и т.д.

2.3 Задачи для самостоятельного решения

Глава 3

ДИСКРЕТНО—СОБЫТИЙНОЕ МОДЕЛИРОВАНИЕ. ОСНОВЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СРЕДСТВАМИ GPSS World

3.1 Детерминированные и стохастические модели

При моделировании сложных реальных систем исследователь часто сталкивается с ситуациями, в которых случайные воздействия играют существенную роль. Стохастические модели, в отличие от детерминированных, учитывают вероятностный характер параметров моделируемого объекта. Например, в модели склада не могут быть определены точно моменты поступления заказов. Данные моменты являются случайными величинами, поэтому модель является стохастической: значения переменных величин модели, которые зависят от реализаций случайных величин, сами становятся случайными величинами. Анализ подобных моделей выполняется на компьютере на основе статистики, набираемой в ходе имитационных экспериментов при многократном прогоне модели для различных значений исходных случайных величин, выбранных в соответствии с их статистическими характеристиками.

Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания. Система массового обслуживания - это совокупность последовательно связанных между собой входящих потоков требований на обслуживание (машин, самолетов, пользователей и т.д.), накопителей, очередей, каналов обслуживания (станций техобслуживания, аэродромов, ЭВМ и т.д.) и выходящих

потоков требований после обслуживания.

В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д. При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования.

В любом элементарном акте обслуживания можно выделить две основные составляющие: ожидание обслуживания заявкой и собственно обслуживание заявки. Это можно изобразить в виде некоторого прибора обслуживания состоящего из накопителя, и канала обслуживания заявок. Заявки, обслуженные каналом и заявки, покинувшие прибор по различным причинам необслуженными (например, из-за переполнения накопителя), образуют выходной поток. Процесс функционирования прибора обслуживания можно представить как процесс изменения состояний его элементов во времени. Переход в новое состояние означает изменение количества заявок, которые в нем находятся (в канале и в накопителе).

Входящий поток требований - это последовательность входящих требований, нуждающихся в обслуживании в системе и подчиняющихся определенному закону.

Выходящий поток требований - это последовательность выходящих требований, обслуженных в системе и подчиняющихся определенному закону.

Требование. (транзакт) - это объект, поступающий в систему и нуждающийся в определенном обслуживании в данной системе. Требование является активным элементом в моделируемой системе. Под требованием можно понимать сообщение, машину, изделие, информацию, пользователя и т.д.

Канал обслуживания - устройство, в котором выполняется обслуживание требования. Основным параметром канала обслуживания является время обслуживания, которое, как правило, является случайной величиной.

Накопитель (буфер) - это место временного расположения требований, нуждающихся в обслуживании. Накопители характеризуются не временем обслуживания требований, а емкостью - максимально возможным количеством одновременно находящихся в накопителе требований. Неоднородность заявок в системе может учитываться с помощью введения классов приоритетов. Приоритеты могут быть статическими и динамическими. Статические приоритеты назначаются заранее и не меняются в процессе моделирования. Динамические используют при моделировании конкретных ситуаций.

В практике моделирования систем, имеющих более сложные структурные связи и алгоритмы поведения, для формализации используются не отдельные приборы

обслуживания, а схемы, образуемые композицией многих элементарных приборов обслуживания (сети массового обслуживания). Если каналы различных приборов обслуживания соединены параллельно, то имеет место многоканальное обслуживание, а если приборы и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание. В разомкнутой системе входной поток обслуженных заявок не может снова поступить на какой-либо элемент, т. е. обратная связь отсутствует, а в замкнутых схемах имеются обратные связи, по которым заявки двигаются в направлении, обратном движению вход-выход.

3.2 Основные подходы к моделированию дискретно-событийных систем

Разработано значительное количество программных средств, предназначенных для имитационного моделирования дискретно-событийных систем. По реализуемой парадигме моделирования можно выделить событийный подход, процессный подход, подход на основе диаграмм состояний.

Рис. 3.1. Классификация подходов к дискретно-событийному моделированию

Событийный	Моделируется время наступления событий и их последствия. При наступлении очередного события меняются значения переменных, описывающих состояние системы.
Диаграммы состояний	Для каждого элемента системы (ресурса, продукта и т.д.) изображается цикл активности в виде диаграммы состояний. Диаграммы соединены между собой в единую модель.
Процессный	Описывается процесс прохождения заявок (требований) через стандартные блоки обслуживания. Модель описывает логику прохождения заявок через систему.

Приведенная классификация не является исчерпывающей или взаимоисключающей. Многие программные системы позволяют комбинировать описанные подходы в рамках одной модели.

Продемонстрируем различия в парадигмах дискретно-событийного моделирования на простом примере. Рассмотрим модель офиса, в котором прием клиентов ведет один специалист. Специалист не может обслуживать одновременно нескольких клиентов. Поэтому, если специалист занят, клиенты ждут в очереди.

Событийный подход

Разработка модели начинается составлением списка событий, происходящих в системе и меняющих ее состояние. В нашем случае таких событий всего два:

- поступление нового клиента;
- завершение обслуживания клиента.

Следующий шаг описание реакции системы на событие. Это производится путем изменения значений переменных состояния с использованием условных операторов IF - THEN - ELSE. Введем переменные состояния Queue - число клиентов в очереди и Clerk = („занят “ / „свободен “) - занят или свободен специалист. В начале моделирования очередь пуста, Queue=0, специалист - свободен.

Поступление клиента :

$$\begin{aligned} & \text{IF (Clerk = „свободен “) THEN Clerk = „занят “} \\ & \qquad \qquad \text{ELSE Queue = Queue + 1} \end{aligned}$$

Завершение обслуживания :

$$\begin{aligned} & \text{IF (Queue > 0) THEN Queue = Queue - 1} \\ & \qquad \qquad \text{ELSE Clerk = „свободен “} \end{aligned}$$

Значения переменных состояния учитываются при определении действия, происходящего при наступлении очередного события. Для завершения модели необходимо уметь формировать „список событий “- время их наступления. Также необходимо уметь вычислять необходимую информацию о функционировании системы. Например, о числе наступивших событий, длине очереди, времени ожидания и т.д.

Диаграммы состояний

Выделим два ключевых элемента рассматриваемой системы: специалист и клиент. Специалист может находиться в одном из двух состояний „свободен “ / „обслуживание клиента “.

Диаграмма состояний имеет вид:

Выделим следующие состояния клиента: вне офиса, поступление на обслуживание, ожидание обслуживания, обслуживание. Диаграмма состояний имеет вид:

Для завершения модели необходимо задать связи между изменениями состояний специалиста и клиента. В итоге получим принципиальную схему модели.

Для того чтобы получить исполняемую модель необходимо задать время перехода элементов между состояниями.

Процессный подход

Процесс обслуживания описывается как последовательность действий:

Реализацию процессного подхода к дискретно-событийному моделированию рассмотрим более подробно на примере системы GPSS World.

3.3 Система имитационного моделирования GPSS World

GPSS World - это общецелевая система моделирования, значительно облегчающая процесс создания и выполнения программ имитационного моделирования.

Студенческую версию системы GPSS World можно бесплатно скачать с сайта фирмы Minuteman Software по адресу: www.minutemansoftware/download.

В процессе инсталляции система GPSSW по умолчанию устанавливается в каталог C:

Program Files

Minuteman Software

GPSS World Student Version.

Для системы GPSSW требуется IBM-совместимый компьютер с ОС Windows 95, 98 и выше.

Установите на компьютер и запустите GPSS World Student Version. Откроется главное окно системы GPSSW.

В первой строке (строке заголовка) главного окна указано название - GPSS World. Во второй строке располагаются пункты главного меню, в третьей - стандартная панель инструментов, включающая ряд кнопок, за которыми закреплены наиболее часто используемые команды. Нижняя строка главного окна - строка состояния системы, в которой дается краткое описание выделенной команды.

Система GPSSW имеет иерархическую систему меню, состоящую из главного меню, систем выпадающих и всплывающих меню (подменю).

Главное меню служит для вызова выпадающих меню. Набор пунктов главного меню располагается во второй строке главного окна системы GPSS.

Выпадающее меню предназначено для вызова всплывающего меню, диалогового окна или соответствующей команды. Выпадающее меню располагается под соответствующим пунктом главного меню. Всплывающее меню позволяет вызывать другое всплывающее меню, диалоговое окно или соответствующую команду. Всплывающее меню располагается справа от выбранного пункта выпадающего меню. Для выбора пункта всплывающего меню достаточно щелкнуть по нему левой кнопкой мыши.

Главное меню обеспечивает доступ ко всем средствам системы GPSSW.

Пункт File (Файл) главного меню служит для работы с файлами документов.

Файлы имитационных моделей в системе GPSSW записываются в окне Model (Модель) и сохраняются с расширением .gps, которое указывается сразу после имени файла. Текстовые файлы системы GPSSW записываются в окне Text File (Текстовый файл) и сохраняются с расширением .txt. Они имеют текстовый формат, и могут модифицироваться при помощи любого текстового редактора.

Результаты проведенного моделирования создаются после сохранения содержимого окна REPORT (Отчет). При этом файл будет иметь расширение .gpr. Кроме

того, можно сохранить сообщения, появляющиеся в процессе моделирования систем. Эти сообщения, выводимые в окне JOURNAL (Журнал главного меню), можно сохранить в файле с расширением .sim.

Создание нового файла

При выборе пункта New или нажатии комбинации клавиш Ctrl+N обеспечивается вызов диалогового окна Новый документ.

Выберите New для того чтобы создать новый файл для моделирования с помощью пункта Model с расширением .gps.

Введите вручную следующие команды (или откройте с помощью пункта Open меню файл)

```
GENERATE 300,100
QUEUE Barber
SEIZE Barber
DEPART Barber
ADVANCE 400,200
RELEASE Barber
TERMINATE 1
```

Эта программа в системе GPSSW будет выглядеть как показано на рис. 3.3.

Данная программа предназначена для моделирования процесса обслуживания в парикмахерской. Новые клиенты появляются через случайные промежутки времени, равномерно распределенные на отрезке 300 ± 100 ед. Клиентов обслуживает один парикмахер (Barber). Время стрижки 400 ± 200 ед. Если парикмахер занят, клиенты ожидают в очереди. После стрижки клиенты покидают парикмахерскую.

Охарактеризуем каждое событие, возникшее в СМО:

Появление требования в системе (GENERATE - Генерировать).

Вход требования в очередь (QUEUE - Очередь).

Определение занятости канала обслуживания (SEIZE - Занять). Если канал занят, то требование остается в очереди, если свободен - то входит в канал обслуживания.

Выход требования из очереди (DEPART - Выйти).

Обслуживание требования в канале обслуживания (ADVANCE - Задержать).

Освобождение канала обслуживания (RELEASE - Освободить).

Выход требования из системы (TERMINATE - Завершить).

Использованные в программе операторы соответствуют содержанию и логике моделируемой системы. Справа от операторов записаны параметры (признаки, переменные), которые характеризуют данное событие.

В операторе GENERATE первое число - 300 - определяет средний интервал времени между поступлениями требований в систему на обслуживание, а второе - 100 - максимально допустимое отклонение от среднего времени.

В операторах QUEUE и DEPART символы Varber определяет имя очереди, в которую вошло и из которой собирается выйти требование. В операторах SEIZE и RELEASE символы Varber определяют символическое имя канала обслуживания, в который собирается войти требование, если он освободился, и выйти - если требование уже в нем обслужилось.

В операторе ADVANCE первое число - 400 - определяет среднее время обслуживания требования, а второе - 200 - максимально допустимое отклонение от этого времени.

Оператор TERMINATE выполняет удаление одного требования из системы.

Для запуска моделирования и просмотра стандартного отчета, в котором выводятся результаты моделирования, используются подпункты меню Command:

Меню Command

Выбор пункта Command главного меню или нажатие комбинации клавиш Alt+C вызывает выпадающее меню команд, изображенное на рис. 3.4.

Меню предлагает следующие пункты:

Create simulation - создать выполняемую модель,

Retranslate - перетранслировать,

Repeat last command - повторить последнюю команду,

Conduct - управление (возможность проведения эксперимента),

Start - запуск оттранслированной программы,

Step 1 - пошаговое выполнение оттранслированной программы,

Halt - прерывание процесса моделирования,

Continue - продолжение процесса моделирования,

Clear - возвращение моделирования в первоначальное состояние,

Reset - сброс статистики в первоначальное состояние,

Show - вызывает диалоговое окно для просмотра искомых параметров в окне Journal,

Custom - вызов диалогового окна для ввода команд управления пользователем во время моделирования.

Для создания выполняемой модели:

— щелкните по кнопке Command (Команда) главного меню. Появится выпадающее меню;

— щелкните по кнопке Create Simulation (Создать выполняемую модель).

Затем:

— щелкните по кнопке Command (Команда) главного меню. Появится выпадающее меню;

— щелкните по кнопке Start (начать моделирование);

— введите число требований, которое необходимо пропустить через систему. Введите START 100 и нажмите кнопку ОК.

Результаты моделирования будут выведены в окно отчета.

Стандартный отчет включает в себя результаты моделирования системы, в том числе:

- коэффициент использования канала обслуживания;
- средняя длина очереди;
- среднее время пребывания требования в очереди и ряд других.

Рис. 3.5. Стандартный вывод результатов моделирования

В стандартный отчет включаются следующие основные показатели моделирования системы:

- время моделирования системы - END TIME (Время окончания);
- число обслуженных требований в канале обслуживания - ENTRIES (Числе входов);
- коэффициент использования канала обслуживания- UTIL. (Использование);
- среднее время обслуживания требования в канале - AVE. TIME (Среднее время);
- максимальная длина очереди - MAX (Максимальная);
- средняя длина (содержимое) очереди - AVE.CONT. (Среднее содержимое);

Процессный подход среднее время пребывания требования в очереди - AVE. TIME - и ряд других показателей.

Система GPSSW позволяет эффективно работать с несколькими моделями. Под каждую модель отводится отдельное окно. Одно из них обычно является активным - то, с которым пользователь работает в настоящий момент. Состояние других окон „замораживается“, но их можно активизировать в любое время без новой загрузки и без исполнения связанной с окном программы. Система GPSSW позволяет организовать работу с несколькими окнами одновременно. Выбор пункта Window (Окно) главного меню или нажатие комбинации

Окно исходной модели

Окно исходной модели системы GPSSW предназначено для эффективной разработки, проверки и отладки программ в GPSSW. Это окно вызывается автоматически при открытии файла с программой на языке GPSS в диалоговом окне Open или создания новой модели с помощью подпункта меню New.

Создайте новую модель и введите следующую программу или откройте файл Elevator.gps.

* Моделирование работы карьера

* 4 автомобиля, 1 погрузчик

```
Q_Load  QTABLE  Loading,10,10,3
        GENERATE,4           ; Число машин в карьере
Cycle   QUEUE   Loading      ; Вход в очередь на погрузку
        SEIZE   Elevator     ; Ожидание освобождения погрузчика
```

DEPART Loading	; Выход из очереди на погрузку
ADVANCE 10,4	; Время погрузки 8 - 12 мин
RELEASE Elevator	; Освобождение погрузчика
ADVANCE 15,5	; Путь до пункта разгрузки 10 - 20 мин
TRANSFERCycle	; Возврат на погрузку
GENERATE440	; Моделирование работы
TERMINATE	; в течении 1440 мин

Программу, находящуюся в окне модели, можно оттранслировать и получить искомый результат. Для этого важно, чтобы окно модели было активным. Если вы не уверены в этом, щелкните мышью в любом месте окна и сделайте его активным. Чтобы загрузить программу на транслирование и выполнение:

— щелкните мышью по пункту `Command` главного меню.

Появится выпадающее меню; щелкните мышью по пункту `Create Simulation ...` (Создать выполняемую модель) выпадающего меню. Если программа без ошибок и в ней присутствует управляющая команда (оператор) `START`, которую не сопровождают символы `NP` (`Not Print` - Не печатать), то результаты моделирования появятся в окне `REPORT`.

Можно также нажать комбинацию клавиш `Ctrl+Alt+S`, которая дает команду на вызов транслятора и выполнение процесса трансляции исходной модели.

Если программа содержит синтаксическую ошибку, то сообщение об этом появится в окне `JOURNAL`.

Оттранслированная программа может быть запущена на выполнение. Для этого:

- щелкните по пункту `Command` главного меню. Появится выпадающее меню;
- щелкните по пункту `Start` выпадающего меню. Появится диалоговое окно `Start Command`

После щелчка по кнопке `ОК` появятся результаты моделирования.

Пункты (команды) выпадающего меню пункта `Command` главного меню становятся активными после транслирования исходной модели.

Команда `START` используется для запуска процесса моделирования. Эта команда может быть введена в модели или в интерактивном режиме.

Многokrатное моделирование может быть выполнено с использованием последовательности управляющих операторов `RESET`, `CLEAR` и `START`.

Остановить процесс моделирования можно несколькими способами:

- щелкните по пункту `Command` главного меню, а затем - по пункту `Halt` (Остановить) выпадающего меню;

- нажмите комбинацию клавиш Ctrl+Alt+H;
- нажмите на функциональную клавишу F4;
- щелкните по пункту Command главного меню, а затем - по пункту Custom выпадающего меню. Появится диалоговое окно Simulation Custom. Введите в поле диалогового окна команду HALT и щелкните по кнопке ОК;
- используйте командный файл.

Вы можете возобновить моделирование, используя команду CONTINUE или интерактивные команды HALT и SHOW выполняются в момент их ввода, другие команды ставятся в очередь. Они помещаются в конце списка команд, которые еще не были закончены к моменту ввода.

Визуализация процесса функционирования моделируемой системы

После трансляции модели система GPSSW обеспечивает возможность визуального наблюдения активных требований в процессе моделирования.

Для этого:

- выберите пункт Window главного меню системы
- выберите подпункт Simulation Window
- щелкните по пункту Block Windows. Появится диалоговое окно Block Entities

Для визуализации процесса движения требований:

- щелкните по пункту Command главного меню. Появится выпадающее меню;
- щелкните по пункту Start выпадающего меню. Появится диалоговое окно Start Command. Нажмите ОК. Кнопки Halt, Continue, Step позволяют останавливать, возобновлять или производить пошаговое выполнение программы.

Следует помнить, что визуализация процесса сильно замедляет скорость моделирования. Поэтому для быстрого получения результатов, окно Block Entities целесообразно закрыть.

Просмотр значений выражений в динамике

Для представления значений выражений в динамике следует использовать окно Expressions Window, которое можно открыть в процессе моделирования. Рассмотрим возможность представления значений выражений для созданной нами модели.

Будем отображать:

- текущую длину очереди Q\$Input;

- максимальную длину очереди QM\$ Loading;
- среднюю длину очереди QA\$ Loading;
- среднее ожидания в очереди Q\$ Loading;
- среднее время ожидания в очереди QT\$ Loading;
- коэффициент использования погрузчика FR\$Elevator.

Для этого:

- выберите пункт Window главного меню системы
- выберите подпункт Simulation Window
- щелкните по пункту Expression Windows.

Появится диалоговое окно Edit Expression Windows. Поочередно введите в поле Label названия выражений, в поле Expression - выражения, и нажимайте кнопку Memorize - запомнить. Затем с помощью кнопки View добавьте выражения в окно Window Contents. Нажмите ОК.

Запустите программу на выполнение:

- щелкните по пункту Command главного меню. Появится выпадающее меню;
- щелкните по пункту Start выпадающего меню. Появится диалоговое окно Start Command. Нажмите ОК.

Следует иметь в виду, что значение коэффициента использования, выдаваемое в окне EXPRESSIONS, нужно разделить на 1000. Таким образом, в нашей задаче коэффициент использования погрузчика равен 1, то есть погрузчик занят все время.

Замечание. Если сократить время погрузки до 4 - 8 мин, то погрузчик будет какое-то время простаивать.

ADVANCE 6,2 ; Время погрузки 4 - 8 мин

По окончании имитационного эксперимента коэффициент использования погрузчика составит 776 ед., т.е. погрузчик будет занят 77,6% времени.

Графическое представление результатов моделирования

Результаты моделирования могут представляться в графическом виде. Для этого:

- щелкните по пункту Window главного меню;
- щелкните по пункту Simulation Window выпадающего меню;
- щелкните по пункту Plot Window. Появится диалоговое окно Edit Plot Window.

Заполним поля для представления длины очереди к погрузчику. В поле Label - введем название графика, в поле Expression - Q\$Loading. Нажмем кнопки Plot, Memorize, ОК.

После этого:

- щелкните по пункту Command главного меню. Появится выпадающее меню;
- щелкните по пункту Start выпадающего меню. Появится диалоговое окно Start Command. Нажмите ОК.

Используя горизонтальную и вертикальную полосу прокрутки, можно просмотреть построенный график.

Если в программе предусмотрены операторы табулирования статистики об очередях, то существует возможность вывода гистограмм для параметров.

Модифицируем программу, добавив оператор табулирования Qtable. Метка Q_Load - задает имя оператора табулирования, Loading - имя очереди. Остальные параметры - определяют верхний предел первого интервала, ширину частотного интервала, число частотных интервалов.

* Моделирование работы карьера

* 4 автомобиля, 1 погрузчик

Q_Load	QTABLE	Loading,10,10,3	
	GENERATE	,4	; Число машин в карьере
Cycle	QUEUE	Loading	; Вход в очередь на погрузку
	SEIZE	Elevator	; Ожидание освобождения погрузчика
	DEPART	Loading	; Выход из очереди на погрузку
	ADVANCE	10,4	; Время погрузки 8 - 12 мин
	RELEASE	Elevator	; Освобождение погрузчика
	ADVANCE	15,5	; Путь до пункта разгрузки 10 - 20 мин
	TRANSFER	Cycle	; Возврат на погрузку
	GENERATE	440	; Моделирование работы
	TERMINATE		; в течении 1440 мин

После этого можно вывести соответствующую гистограмму.

Установка содержания отчета

Флажок In Windows (В окнах) обеспечивает текстовое представление каждого объекта сообщения вместо сохранения их в файле.

Ниже приводится группа флажков, обеспечивающих вывод результатов функционирования того или иного элемента моделируемой системы:

- Blocks (Блоки);

- Queues (Очереди);
- Tables (Таблицы);
- Names (Имена);
- XN Groups (XN-группы);
- Num Groups (Числовые группы);
- Facilities (Каналы обслуживания);
- Storages (Накопители);
- CEC (Цепь текущих событий);
- FEC (Цепь будущих событий);
- Six Places (Шесть знаков);
- Procedures (Процедуры);
- Savevalues (Сохраняемые величины);
- Logicswitches (Логические переключатели);
- Userchain (Цепь пользователя);
- Matrices (Матрицы);
- Scientific (Научный);
- Experiments (Эксперименты).

Наличие галочки в том или ином окошке обеспечивает вывод в стандартном отчете результатов моделирования выделенных элементов.

Флажок Six Places устанавливает точность действительных значений с шестью десятичными знаками вместо трех.

Флажок Scientific используется для представления мантиссы чисел в виде десятичной экспоненты в сообщениях и потоках данных.

Поле Saved Plot Points (Сохраненное значение точек) сообщает системе GPSSW, сколько пространства выделить круговому буферу для точек.

Флажок Suppress Page Numbers (Не вставлять номера страниц) служит для удаления номеров страниц при печати текстовых объектов.

Флажок PLUS Trace (Трассировка PLUS) обеспечивает трассировку языка программирования PLUS.

Флажок Silence (Молчание) обеспечивает выдачу ошибок без звукового сопровождения.

Вставка блоков GPSS в модель

Редактирование модели удобно осуществлять в диалоговом режиме. Для вставки нового оператора:

- щелкните по пункту Edit главного меню;
- щелкните по пункту Insert GPSS Block into Model Object выпадающего меню.

В диалоговое окно Insert GPSS Block into Model Object (Вставить блок GPSS в модель) входит пятьдесят три блока.

Если щелкнуть по любому из них, появится соответствующее диалоговое окно. Допустим, что вы щелкнули по блоку GENERATE. Шаблон блока GENERATE появится в диалоговом окне Enter Block Information (Информация для ввода в блок), показанном на рис. 3.12, для ввода необходимой информации.

Поле Start Delay Time предназначено для задания времени появления первого транзакта (по умолчанию 0), Creation Limit - задает максимальное число сгенерированных требований, Priority - приоритета.

По умолчанию распределение между временем заявок предполагается равномерным. В этом случае, параметр A задает среднее время между появлениями заявок, B - максимальные отклонения от среднего. В приведенном примере предполагается равномерное распределение на интервале (200;400). Система GPSSW включает встроенные функции многочисленных типовых распределений вероятностей:

- Beta - бета-распределение;
- Binomial - биномиальное распределение;
- Discrete Uniform - дискретное равномерное распределение;
- Exponential - экспоненциальное распределение;
- Extreme Value A;
- Extreme Value B;
- Gamma - гамма-распределение;
- Geometric - геометрическое распределение;
- Inverse Gaussian - обратное гауссово распределение;
- Inverse Weibull - обратное распределение Вейбула;
- Laplace - распределение Лапласа;
- Logistic - логистическое распределение;

- LogLaplace - логлапласово распределение;
- LogLogistic - логлогистическое распределение;
- LogNormal - логнормальное распределение;
- Negative Binomial - отрицательное биномиальное распределение;
- Normal - нормальное распределение;
- Pareto - распределение Парето;
- Pearson Type V - распределение Пирсона типа V;
- Pearson Type VI - распределение Пирсона типа VI;
- Poisson - пуассоновское распределение;
- Triangular - треугольное распределение;
- Uniform - равномерное распределение;
- Weibull - распределение Вейбула.

Например, (EXPONENTIAL(20,0,50)) - задает распределение времени между появлением требований по экспоненциальному закону. Первый параметр соответствует номеру потока случайных чисел, два других - параметрам экспоненциального закона. Оператор Generate имеет вид:

GENERATE (EXPONENTIAL(20,0,50))

Основные операторы языка GPSS

Оператор		Описание
GENERATE	A,B,C,D,E,F,G,H	Сгенерировать требование
QUEUE	A,B	Увеличить содержимое очереди
DEPART	A,B	Уменьшить содержимое очереди
SEIZE	A	Занять канал обслуживания
RELEASE	A	Освободить канал обслуживания
ENTER	A,B	Увеличить вместимость накопителя
LEAVE	A,B	Уменьшить вместимость накопителя
ADVANCE	A,B	Задержать перемещение требования
ASSIGN	A,B	Модифицировать параметр требования
PREEMPT	A,B,C,D,E	Отстранить требование от обслуживания
RETURN	A	Вернуть канал обслуживания А
BUFFER		Идти в хвост цепи текущих событий
GATE	A,B	Переместить в зависимости от состояния
GATHER	A	Подождать родственные требования
LINK	A,B,C	Ввести в цепь пользователя
UNLINK	A,B,C,D,E,F,G	Вывести из цепи пользователя
<Имя> STORAGE	A	Определить вместимость накопителя
MARK	A	Создать временную метку
MATCH	A	Подождать другое требование
SELECT	A,B,C,D,E,F,G	Выбрать элемент
SAVEVALUE	A,B,C	Модифицировать сохраняемое значение
MSAVEVALUE	A,B,C,D	Модифицировать значение матрицы
PRIORITY	A,B	Изменить приоритет

SPLIT	A,B,C,D	Разделить требование
TABULATE	A,B	Собрать табличные данные элемента
TERMINATE	A	Уничтожить требование
TEST	A,B,C	Переместить согласно сравнению
TRANSFER	A,B,C,D	Передать в новое местоположение
ASSEMBLE	A	Собрать родственные требования
EQU	A,B	Определить эквивалентность
MATRIX	A,B,C	Задать матрицу
INITIAL	A,B	Инициализировать
RMULT	A,B,C,D,E,F,G	Установить значение генератора
<Имя> VARIABLE	A	Определить целую переменную
<Имя> FVARIABLE	A	Определить вещественную переменную
<Имя> BVARIABLE	A	Определить булеву переменную
<Имя> FUNCTION	A,B	Определить функцию
<Имя> TABLE	A,B,C,D,E	Определить таблицу
CITABLE	A,B,C,D	Определить таблицу для очереди
LOGIC	A	Изменить логический переключатель
LOOP	A,B	Изменить параметр требования и зациклить
REPORT		Сообщить
COUNT	A,B,C,D,E	Поместить индекс объекта внутрь параметра
ITRACE		Начать трассировку
UNTRACE		Завершить трассировку
INDEX	A,B	Увеличить параметр требования
EXECUTE	A	Выполнить блочную операцию элемента

WRITE		Разместить текстовую строку в потоке данных
READ		Считать текстовую строку из потока данных
RESET		Сбросить
ADOPT		Изменить номер ансамбля
CLOSE		Закрыть поток данных
JOIN		Увеличить группу членов
ALTER		Изменить атрибуты требования
DISPLACE		Переместить требование к блоку
EXAMINE		Исследовать член группы
FAVAL		Сделать канал обслуживания доступным
FUNAVAL		Сделать канал обслуживания недоступным
INTEGRATION		Включить или выключить интегрирование
PLUS		Оценить PLUS-выражение
SAVAL		Сделать накопитель доступным
SUNAVAL		Сделать накопитель недоступным
REMOVE		Уменьшить группу членов
SEEK		Установить позицию в потоке данных
SCAN		Восстановить информацию группы членов

Каждый оператор языка GPSS должен находиться на одной строке текста и включать в совокупности с другими элементами не более 250 символов. Он структурно состоит из нескольких полей. Выделяют следующие поля оператора:

- номер строки (необязательный);
- метка (необязательный);
- собственно оператор;
- операнды (зависят от введенного оператора);
- комментарий (необязательный).

Поле - переменное число печатаемых символов, за которыми следует незаполненное пространство или разделитель.

Номера строк, если они используются, должны вводиться с первой позиции (столбца). Однако номера строк в системе GPSS World игнорируются транслятором. Номер же строки, данный в сообщении об ошибках, - это число, обозначающее место строки в программе, а не указанный пользователем номер строки.

Поле метки - это поле, в котором может быть введено имя данного оператора - числовое или символьное - для обращения к нему из другого места программы.

Поле оператора - это поле, в котором записывается один из операторов языка имитационного моделирования GPSS.

Поле операндов - это поле ввода требуемых для данного процесса моделирования операндов (аргументов оператора). Большинство операторов имеют один или несколько операндов. Поля операндов часто называют полями переменных. Традиционно операнды разделяются запятыми. Допускается пропускать в операторах некоторые операнды - так называемые необязательные операнды. В поле переменных выделяют подполя, разделяемые при записи запятыми и служащие для указания числовых значений переменных, стандартных числовых атрибутов, числовых или символьных меток и т.п. Подполя могут быть пустыми. Стандартные числовые атрибуты служат для сокращенного указания различных величин, фигурирующих в модели, например:

- K15 - константа, равная 15;
- V12 - переменная под номером 12;
- Q\$kanal - средняя длина очереди под символьным именем kanal;
- XI7 - сохраняемая величина под номером 17;
- FN\$ECSP - функция с именем ECSP;
- P2 - значение параметра требования под номером 2;
- *4 - содержимое параметра требования под номером 4;

Поле комментария - это поле размещения комментария, относящегося к используемому оператору. Существует два основных вида комментария: входные и промежуточные. Первый вид комментария начинается с символов ; (точка с запятой) или * (звездочка), а далее идет любой текст. Модель в системе GPSSW - это последовательность операторов моделирования. Оператор моделирования может быть или оператором GPSS, или процедурой языка PLUS (Programming Language Under Simulation - Язык программирования для моделирования).

Рассмотрим более подробно некоторые из стандартных блоков GPSSW, применяемых для построения имитационных моделей реальных систем.

Блоки Split, Assemble, Transfer

Перечисленные блоки относятся к блокам создания и удаления транзактов. Блок Split создает заданное количество копий от входящего в него транзакта. Входящий транзакт называют транзактом - родителем, а копии транзактами - потоками.

SPLIT A,B,C

В поле А задается число создаваемых копий. Поле В - номер блока (метка) в который передается копии транзакта родителя. В поле С может быть задан номер параметра, используемого для присвоения копиям последовательных номеров.

Блок Assemble объединяет несколько транзактов, после чего из него выходит один транзакт, который переходит в следующий блок, а все остальные уничтожаются.

ASSEMBLE A

Блок Transfer позволяет изменять маршруты транзактов, в том числе осуществлять условные и безусловные переходы.

TRANSFER A,B,C,D

Блок TRANSFER имеет следующие режимы работы: безусловный, вероятностный, BOTH, ALL, PICK; функция; подпрограмма; параметр; SIM.

В приведенной далее программе блок SPLIT порождает одну копию требования, которое передается в блок SIEZE, по метке Chanel2. Входящий пакет рассортировывается на две части, каждый из которых обрабатывается со своей временной задержкой. Оператор TRANSFER осуществляет безусловный переход первой копии по метке OUT_SYS. Оператор ASSEMBLE формирует из двух частей один пакет.

	GENERATE	10,3,0,200
	SPLIT	1,Chanel2
	SEIZE	mash1
	ADVANCE	10,2
	RELEASE	mash1
	TRANSFER	,OUT_SYS
Chanel2	SEIZE	mash2
	ADVANCE	15,3
	RELEASE	mash2
OUT_SYS	ASSEMBLE	2
	TERMINATE	1

Оператор TRANSFER может использоваться для моделирования вероятностного выбора. Например,

TRANSFER 0.75, Ch_1, Ch_2

означает, что с вероятностью 75% входящий транзакт будет направлен в блок с меткой Ch_1, а с вероятностью 25% - в блок с меткой Ch_2.

	GENERATE	10,3
	TRANSFER	BOTH,Ch_1,Ch_2
Ch_1	SEIZE	mash1
	ADVANCE	10,1
	RELEASE	mash1
	TRANSFER	,OUT_SYS
Ch_2	SEIZE	mash2
	ADVANCE	14,5
	RELEASE	mash2
OUT_SYS	TERMINATE	1

Приведенный пример реализует модель следующей системы. Заявки могут обрабатываться на одном из двух приборов. Продолжительность обработки на первом 10 ± 1 мин., на втором - 14 ± 5 мин. Предпочтительнее обработка на первом приборе.

Блоки ASSIGN, LOOP, PREEMPT, RETURN

Блок ASSIGN предназначен для задания значений параметров транзактов. Параметры применяются для задания характеристик конкретного транзакта, их различения в потоке. Интерпретация параметров определяется разработчиком конкретной модели.

Рассмотрим пример. К кассе обменного пункта через случайные промежутки времени, распределенные равномерно на интервале от 60 до 180 секунд, подходят клиенты. Сумма, необходимая клиенту, распределена случайно на интервале от 50 до 150. Обслуживание происходит в порядке живой очереди. Время обслуживания пропорционально стоимости.

Блок Assign 10, SUM1 присваивает параметру под номером 10 значение переменной SUM1. Значение параметра 10 используется в операторе ADVANCE для определения времени задержки.

```

SUM1 Variable      (uniform(20,50,150))
GENERATE          120,60
ASSIGN            10,SUM1
QUEUE            CashBox
SEIZE             cashier
ADVANCE           P10
RELEASE           cashier
DEPART           CashBox
TERMINATE        1

```

Блок PREEMPT позволяет прерывать поток транзактов и передавать его для обработки другого потока, блок RETURN возвращает устройство для обработки первоначального потока. Пример. На обработку по равномерному закону поступают детали через интервалы 5 ± 2 мин. Обработка детали занимает 5 ± 1 мин. Каждые 8 часов (480 мин.) станок необходимо обслуживать. Обслуживание занимает 30 ± 15 мин.

* Обработка деталей на станке

```

GENERATE          5,2
SEIZE             Machine
ADVANCE           5,1
RELEASE           Machine
TERMINATE        1

```

* Обслуживание станка

```

GENERATE          480
PREEMPT           Machine
ADVANCE           30,15
RETURN            Machine
TERMINATE        1

```

Форма записи блока TEST

TEST X A,B,C

Блок TEST определяет номер следующего блока для вошедшего в него транзакта в зависимости от выполнения заданного условия. Во вспомогательном поле X записывается один из условных операторов:

LE - значение аргумента А меньше или равно значению аргумента В;

GE - значение аргумента А больше или равно значению аргумента В;

E - значение аргумента А равно значению аргумента В;

L - значение аргумента А меньше значения аргумента В;

G - значение аргумента А больше значения аргумента В;

NE - значение аргумента А не равно значению аргумента В. При истинности условия транзакт переходит к следующему по номеру блоку. Если отношение ложно, транзакт переходит к блоку номер или имя которого задано в поле С.

В следующем примере блок TEST использован в программе, моделирующей движение на перекрестке, оборудованном светофором. В направлении Север - Юг машины прибывают с интервалом 25 ± 10 сек., Восток - Запад с интервалом 15 ± 8 сек. Светофор имеет следующий цикл: красный свет в направлении Север - Юг в течении 90 секунд, затем красный свет в направлении Восток - Запад в течении 120 секунд. Требуется определить основные параметры движения через перекресток, после проезда через него 1000 автомобилей. Также в программе использованы операторы EQU - присвоения числовых значений переменным и SAVEVALUE - присвоение значений сохраняемым величинам.

* Моделирование движения автомобилей через перекресток * Движение в направлении Север-Юг

```
GENERATE      25,10
QUEUE         N_S
TEST E        X$Light_N_S,F$Cross ; Запрет движения на красный свет красный
SEIZE         Cross
DEPART        N_S
ADVANCE       8,3
RELEASE       Cross
TERMINATE     1
```

* Моделирование движения автомобилей через перекресток * Движение в направлении Восток - Запад

```
GENERATE      15,8
QUEUE         O_W
TEST E        X$Light_O_W,F$Cross ; Запрет движения, на красный свет
SEIZE         Cross
```

DEPART	O_W
ADVANCE	8,3
RELEASE	Cross
TERMINATE	1

* Моделирование работы светофора

GENERATE „,1 ;Генерирование состояния светофора

```

L_cycle  SAVEVALUE  Light_N_S,Red    ;Красный свет в направлении Север - Юг
        SAVEVALUE  Light_O_W,Green ;Зеленый свет в направлении Восток-Запад
        ADVANCE    Red_N_S        ;Время красного сигнала для Север - Юг
        SAVEVALUE  Light_N_S,Green ;Зеленый свет в направлении Север - Юг
        SAVEVALUE  Light_O_W,Red   ;Красный свет в направлении Восток-Запад
        ADVANCE    Red_O_W        ;Время красного сигнала для Восток-Запад
        TRANSFER   ,L_cycle
Red_N_S  EQU 120
Red_O_W  EQU 90
Red      EQU 100
Green    EQU 0
Start 1000

```

Фрагмент стандартного отчета имеет вид:

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
O_W	9	6	631	68	2.722	40.786	45.712	0
N_S	6	0	375	66	1.901	47.931	58.168	0

Средняя длина очереди в направлении Север — Юг составила 1.9, Восток — Запад — 2.7. Максимальная длина очередей 6 и 9 автомобилей соответственно. Изменяя в программе время переключения сигналов светофора, можно подобрать значения, обеспечивающие оптимальное движение через перекресток.

Для моделирования систем с ограниченной очередью или ограниченной емкостью накопителя используются блоки ENTER, LEAVE, GATE, TEST и операторы STORAGE, VARIABLE, INITIAL.

Для моделирования систем с ограниченной очередью или емкостью накопителя используют блоки ENTER, LEAVE и оператор STORAGE.

Оператор STORAGE позволяет задать емкость накопителя.

Блок GATE проверяет состояние устройств, памятей, логических ключей. Блоки ENTER и LEAVE уменьшают или увеличивают содержание накопителя. Оператор

INITIAL позволяет задавать начальные значения ячеек, элементов матриц, логических переключателей.

Приведем пример следующей программы. Требования прибывают через случайные промежутки с интервалом 10 ± 2 мин. Если прибор занят, то требования ожидают в накопителе. Емкость накопителя ограничена 4 единицами. Если в накопителе нет места, то требование теряется. Время обслуживания 11 ± 1 мин.

```
STOR_1 STORAGE 4
    GENERATE 10,2
    GATE SNF  STOR_1,Exit_1
    ENTER     STOR_1,1
    SEIZE     Cashier
    ADVANCE  11,3
    RELEASE  Cashier
    LEAVE     STOR_1,1
    TERMINATE1
Exit_1  TERMINATE1
```

Оператор

```
STOR_1 STORAGE 4
```

определяет накопитель с именем STOR_1, емкостью 4 единицы. Оператор GATE с помощью логического оператора SNF проверяет условие, что накопитель STOR_1 полностью заполнен. В этом случае происходит переход на блок с меткой Exit_1 и транзакт покидает систему.

Результаты моделирования обработки 1000 транзактов представлены в стандартном отчете.

LABEL	LOC	BLOCK	TYPE	ENTRY	COUNT	CURRENT	COUNT	RETRY
1		GENERATE		1003		0		0
2		GATE		1003		0		0
3		ENTER		917		2		0
4		SEIZE		915		1		0
5		ADVANCE		914		0		0
6		RELEASE		914		0		0
7		LEAVE		914		0		0
8		TERMINATE		914		0		0
EXIT_19		TERMINATE		86		0		0

FACILITY	ENTRIES	UTIL.	AVE.	TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY	
CASHIER	915		0.999	10.953		1	1001	0	0	0	2

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
STOR_1	4	1	0	4	917	1	3.366	0.842	0	0

В оператор Terminate с с меткой Exit_1 поступило 86 транзактов, то есть не поступили на обработку из-за переполнения накопителя потеряно 8,6% требований. Средняя наполненность накопителя составила 3,36 заявок.

3.4 ВАРИАНТЫ ЗАДАНИЙ ДЛЯ МОДЕЛИРОВАНИЯ

Задание 1. Моделирование работы станка с поломками

Задания поступают на станок в среднем один раз в час. Распределение величины интервала между ними экспоненциально. При нормальном режиме работы задания выполняются в порядке их поступления.

Перед выполнением задания производится наладка станка, время осуществления которой распределено равномерно на интервале 0.2-0.5 ч. Время выполнения задания нормально распределено с математическим ожиданием 0.5 ч и среднеквадратичным отклонением 0.1 ч.

Станок подвергается поломкам, при которых он не может продолжать выполнения задания. Интервалы между поломками распределены нормально с математическим ожиданием 20 ч и среднеквадратичным отклонением 2 ч. При поломке выполняемое задание удаляется со станка и помещается в начало очереди заданий к станку.

Общая продолжительность устранения поломки распределено равномерно на интервале от 0.1 до 0.5 часа.

Проанализируйте работу станка в течение обработки 500 деталей для получения информации по загрузке станка и времени выполнения задания.

Задание 2. Движение автомобилей через туннель

Светофоры, размещенные на обоих концах туннеля, управляют движением автомобилей.

Автомобили подъезжают к светофорам через интервалы времени, распределенные экспоненциально с математическим ожиданием, равным 12 с для 1-го направления и 9 с для 2-го направления.

Когда загорается зеленый свет, машины следуют по туннелю с интервалом 2 с.

Подъезжающий к туннелю автомобиль едет по нему без задержки, если горит зеленый свет и перед светофором нет машин.

Светофор имеет следующий цикл: зеленый в 1-м направлении, красный в обоих направлениях, зеленый во 2-м направлении, красный в обоих направлениях. Красный свет горит в обоих направлениях в течение 60 с для того, чтобы автомобили, следующие через туннель, смогли покинуть его до переключения зеленого света на другое направление.

Проанализируйте движение через перекресток 1000 автомобилей.

Цель имитационного эксперимента: определите такие значения „зеленых“ интервалов для обоих направлений, при которых среднее время ожидания всех автомобилей будет минимальным.

Задание 3. Работа бензоколонки

Из предыдущих наблюдений известно, что интервалы времени между прибытием клиентов в час пик распределены экспоненциально с математическим ожиданием, равным 0.1 единицы времени.

На бензоколонке имеется два заправочных автомата. Клиенты выбирают более короткую очередь. При равной длине очередей, а также при отсутствии очередей, клиенты отдадут предпочтение первому автомату. Длина каждой очереди ограничена и составляет не более 5 автомобилей. Если подъезд заполнен до отказа, прибывший клиент считается потерянным, так как он сразу же уезжает.

Продолжительность обслуживания всех колонок одинакова и распределена экспоненциально с математическим ожиданием, равным 0.5 единицы времени.

Проанализируйте процесс обслуживания 400 автомобилей.

Цель имитации определить: 1. Среднее число клиентов в каждой очереди. 2. Процент клиентов, которые отказались от обслуживания. 3. Интервалы времени между отъездами клиентов. 4. Среднее время пребывания клиента на заправке.

Задание 4. Имитация работы мелкооптового магазина

В магазине используется следующая процедура обслуживания клиентов. Клиенты, попадая в магазин, определяют по каталогу наименования товаров, которые они хотели бы приобрести. Предполагается, что клиент может выбрать от 1 до 5 наименований товаров с равной вероятностью. Интервалы между моментами поступления заявок на товары от клиентов экспоненциально распределены с математическим ожиданием, равным 15 мин.

После этого клиент обслуживается клерком, который идет на расположенный рядом склад и приносит необходимый товар. Время, которое затрачивает клерк на путь

к складу, равномерно распределено на интервале от 0,5 до 1,5 мин. Время поиска товара нужного наименования зависит от числа наименований, которое клерк должен найти на складе.

Это время нормально распределено с математическим ожиданием, равным утреннему числу искомых наименований, и среднеквадратичным отклонением, равным 0,2 математического ожидания.

Следовательно, если, например, со склада надо взять товар одного наименования, время на его поиск будет нормально распределено с математическим ожиданием, равным 3 мин, и среднеквадратичным отклонением, равным 0,6 мин. Время возвращения со склада равномерно распределено на интервале от 0,5 до 1,5 мин.

По возвращении со склада клерк рассчитывается с клиентом, которого он обслуживает. Время обслуживания равномерно распределено на интервале от 1 до 3 мин.

Цель имитации - определить следующее: загрузку клерка; среднее время, необходимое на обслуживание одного клиента с момента подачи заявки на товар до оплаты счета за покупку. Продолжительность имитационного прогона: обслуживание 1000 клиентов.

Задание 5. Система управления запасами с неудовлетворенным спросом

На оптовом складе планируется ввести систему управления запасами. Время между поступлением заказов распределено экспоненциально с математическим ожиданием 0.5 нед. Размер заказа является случайной величиной, равномерно распределенной на интервале от 20 до 100 ед. Если остаток на складе меньше размера заявки, то покупатель отправляется к другому поставщику.

Склад использует периодическую систему просмотра состояния запасов, в которой запас просматривается каждые 4 нед. и принимается решение о необходимости осуществления (размещения) заказа. Если текущее состояние запасов меньше или равно 200, то осуществляется размещение заказа, доводящего запас до контрольного уровня, составляющего 500 ед. Время доставки (время между размещением заказа и его получением) постоянно и составляет 2 нед.

Начальное состояние запаса - 500 ед.

Проведите имитацию системы управления запасами за 5 лет (520 заявок) для получения статистических данных о следующих величинах:

1. среднем размере запаса;
2. о числе удовлетворенных и неудовлетворенных заказов;
3. об общем объеме удовлетворенных и неудовлетворенных заказов;
4. о времени между несостоявшимися продажами.

Задание 6. Моделирование работы карьера

В карьере самосвалы доставляют руду от двух экскаваторов, причем после выгрузки руды у измельчителя самосвалы всегда возвращаются к одним и тем же экскаваторам. Время погрузки самосвала распределено экспоненциально с математическим ожиданием 5, время поездки постоянно и равно 3, время разгрузки распределено экспоненциально с математическим ожиданием 2, время обратной поездки постоянно и равно 1. К каждому экскаватору приписаны три самосвала. Очереди к каждому экскаватору организованы по правилу "первым пришел - первым обслужен".

Требуется проанализировать функционирование всей системы в течение 480 единиц времени для определения загрузки экскаватора и измельчителя и длин очередей к ним.

Задание 7. Производственная линия с пунктами технического контроля и настройки

Собранные изделия на заключительной стадии их производства проходят технический контроль. Время между поступлениями изделий в пункт контроля распределено равномерно на интервале 5-15 мин. Изделие проверяется на годность к эксплуатации. Время, необходимое на проверку одного изделия контролером, распределено равномерно на интервале 6-12 мин. В среднем 85% изделий проходят проверку успешно. Остальные 15% возвращаются в пункт настройки, обслуживаемый одним рабочим. Время настройки распределено равномерно на интервале 10-30 мин. После настройки изделие вновь возвращается на проверку.

Необходимо проимитировать работу пунктов контроля и настройки в течение 600 мин для оценки времени, затрачиваемого на обслуживание каждого изделия, а также загрузку контролера и настройщика.

Задание 8. Производственная поточная линия

На поточной линии предприятия выполняются две операции (соответственно имеется два рабочих места), осуществляемые в строгой последовательности, т.е. вторая операция всегда следует за первой. Обрабатываемые изделия громоздки, поэтому на одной поточной линии могут одновременно находиться только восемь изделий, включая уже обрабатываемые. В соответствии с предлагаемым планом между рабочими местами поточной линии выделяется пространство, достаточное для размещения двух изделий, а перед первым рабочим местом - для четырех изделий. На рис. 1 приведена предлагаемая схема поточной линии. Стратегия текущего управления производством на линии заключается в том, что обработка изделия, которое

не может разместиться в пределах линии из-за недостатка свободного пространства, откладывается.

Обследования поточной линии показали, что интервалы времени между запросами на обработку изделий распределены экспоненциально с математическим ожиданием, равным 0.4 единицы времени. Времена обработки изделий также распределены экспоненциально, причем на первом рабочем месте обработка занимает в среднем 0.25 единицы времени, а на втором - 0.5 единицы. Предполагается, что изделия автоматически транспортируются от первого рабочего места ко второму за очень малый промежуток времени. Если очередь ко второму рабочему месту заполнена до конца, т.е. в ней ожидают обработки два изделия, то происходит блокировка первого рабочего места, так как изделие с него не может быть убрано. На заблокированное рабочее место не может поступать для обработки другое изделие.

Для оценки предлагаемой схемы необходимо собрать за период, равный 300 единицам времени, статистику по следующим величинам:

1. загрузка рабочих мест;
2. время обработки одного изделия на поточной линии;
3. число изделий, обработка которых отложена (в единицу времени);
4. число изделий, находящихся в очереди к каждому рабочему месту;
5. доля времени, в течение которого первое рабочее место заблокировано.

Задание 9. Работа порта

В порту танкеры загружаются сырой нефтью, которую морским путем доставляют затем по назначению. Мощности порта позволяют загружать не более 2 танкеров одновременно. Танкеры, прибывают в порт через каждые 16 ± 12 ч. Время на погрузку - 12 ± 6 ч.

В порту имеется один буксир, услугами которого пользуются все танкеры при причаливании и отчаливании. Причаливание и отчаливание занимает около 1 ч, причем, если в услугах буксира нуждаются сразу несколько танкеров, приоритет отдается операции отчаливания.

По результатам имитационного моделирования в течении года (500 танкеров) определите:

- Среднее время пребывания танкеров в порту
- Среднюю длину очереди на погрузку

Задание 10. Работа инструментальной кладовой

К инструментальной кладовой приходят рабочие и с вероятностью 0,7 становятся в первую, 0,3 - во вторую очередь. Время поступления заявок распределено по равномерному закону с интервалом 7 ± 2 мин. Заявки обслуживают два кладовщика. Время обслуживания первым кладовщиком распределено по равномерному закону с интервалом 9 ± 2 мин. Время обслуживания вторым кладовщиком распределено по равномерному закону с интервалом 11 ± 2 мин.

По результатам имитационного моделирования обслуживания 1000 заявок определите:

1. Среднее время пребывания в каждой из очередей
2. Среднюю длину каждой из очередей