ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

11(159)
2009

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Информация о журнале доступна по сети Internet по адресу http://www.**informika.ru/text/magaz/it/** или **http://novtex.ru/IT.**

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 004.42

Б. М. Михайлов, д-р техн. наук, проф., зав. каф.,
А. Е. Александров, д-р техн. наук, проф.,
Московский государственный университет приборостроения и информатики,
e-mail: femsystem@narod.ru

Решение граничных обратных задач теплопроводности на основе технологии порождающего программирования

Представлено описание алгоритмов для нахождения коэффициентов чувствительности нелинейных граничных обратных задач теплопроводности. Показано, что для нахождения коэффициентов чувствительности могут быть использованы аналогичные численные схемы, что и для параболического уравнения теплопроводности. Приведена технология разработки прикладных программных приложений на основе описанных алгоритмов и разработанного базового программного обеспечения.

Ключевые слова: *порождающее программирование, обратные задачи теплопроводности, система управления версиями, библиотека компонентов, проблемно-ориентированный высокоуровневый язык, метод конечных элементов.*

Введение

В работе [1] представлена разработка программного обеспечения на основе новой технологии, получившей впоследствии название "технология порождающего программирования". Данная технология позволяет стандартизовать многократно используемые компоненты и создавать из них, как из кубиков, прикладные программные системы. В настоящей работе технология порождающего программирования использована для решения граничных обратных задач теплопроводности. Реализованные едиными программными средствами методы решения как прямой задачи, так и обратной позволяют существенно повысить эффективность моделирования. Авторами работы [7] приводится большое число примеров, когда ценность математической модели существенно падает ввиду отсутствия достоверной информации о граничных условиях или выбранных значениях физических параметров. Предложенный авторами подход состоит в том, что отсутствующая информация в прямой задаче считается неизвестной, а для восстановления ее используется косвенная, получить которую значительно проще, чем первоначальную. Для восстановления отсутствующей информации используется аппарат для решения обратных задач.

В настоящей работе восстановление граничных условий в задаче теплопроводности рассматривается без существенных ограничений на форму исходной геометрической области и ее размерность, с возможностью включения в геометрическую область фрагментов из разнородных материалов, свойства которых зависят от температуры. Такая общность решения задачи восстановления основана на разработанной методике расчета коэффициентов чувствительности, реализованной в конечно-элементном подходе. Рассчитанные в соответствии с данной методикой коэффициенты чувствительности позволяют учесть все особенности физического процесса в рамках разработанной конечно-элементной модели, что, в конечном счете, приводит к повышению точности расчетов, максимально приближая разрабатываемую математическую модель к реальному физическому процессу.

При разработке и проектировании программных приложений, включающих решение обратных задач теплопроводности, использована базовая расширяемая библиотека компонентов, а также высокоуровневая расширяемая среда программирования, описанные в работах [1, 3].

Постановка граничной обратной задачи

Необходимость решения граничных обратных задач возникает довольно часто. В качестве примера приведем задачу расчета теплового и напряженно-деформированного состояния тормозных дисков в системе торможения авиационных колес [2].

К этому классу относятся задачи расчета теплового состояния технических конструкций в случае обтекания их поверхности двухфазным потоком. В этом случае математическая модель внешнего процесса может отсутствовать. Поэтому единственным выходом является восстановление внешних условий по показаниям датчика, установленного на некотором расстоянии от поверхности. Специфика экспериментальных исследований такова, что получаемая с датчика информа-

ция является косвенной, т. е. характеризует состояние внутренней области исследуемого объекта, а искомая величина — внешние условия определяются по их проявлениям.

Дадим постановку граничной обратной задачи применительно к параболическому уравнению на примере уравнения теплопроводности. Пусть задано уравнение процесса

$$\sum_{i=1}^{3} \frac{\partial}{\partial x_i}\left(\lambda_{x_i} \frac{\partial T}{\partial x_i}\right) + Q_V = c(T)\rho(T)\frac{\partial T}{\partial \tau} \qquad (1)$$

где τ — время, с; x_i — пространственная координата, м; $\lambda(T)$ — коэффициент теплопроводности материала, Вт · м$^{-1}$ °С$^{-1}$; $c(T)$ — теплоемкость материала, Дж · кг °С$^{-1}$; $\rho(T)$ — плотность материала, кг · м$^{-3}$; Q — плотность тепловыделения, Вт · м$^{-3}$; T — температура, °С.

Начальное распределение температуры $T(x_i, \tau_0) = T_0(x_i)$, геометрическая форма и размеры исходного тела $x_i \in V$, а также теплофизические параметры $\lambda(T)$ и $c(T)$, $\rho(T)$ являются известными величинами. Известно распределение температуры $T(d, \tau)$ в произвольно заданной точке (или в нескольких точках), находящейся внутри заданного тела.

Распределение температуры $T(d, \tau)$ определяется по результатам измерений датчика температуры и поэтому содержит погрешности. Необходимо определить тепловой поток на одной из поверхностей искомой области $q(x_i, \tau)|_s$ (как правило, на той поверхности, где с внешней стороны протекает сложный физический процесс, математическое описание для которого отсутствует).

Распределение температуры в любой момент времени τ и в произвольной точке P пространства определяется из решения уравнения (1). Для решения этого уравнения используем метод конечных элементов (МКЭ). Задавая различные функции на границе $q(x_i, \tau)|_s$, будем получать различные температурные кривые $T = T(x_i, \tau)$ в любой точке заданного тела, в том числе и в точке P $T_p(q, \tau)$.

Задача восстановления граничных условий будет состоять в том, чтобы при подобранном потоке на границе $q(x_i, \tau)|_s$ уклонение рассчитанной температуры $T_p(q, \tau)$, как можно меньше отстояло от температуры $T(d, \tau)$, измеренной с помощью датчика температур в той же точке. В точной экстремальной постановке определение функции $q(x_i, \tau)$ соответствует минимизации невязки, характеризующей уклонение температуры $T_p(q, \tau)$, рассчитанной для некоторой плотности теплового потока, от экспериментально найденной температуры $T(d, \tau)$ в метрике пространства входных данных:

$$F(q) = \int_0^s \int_0^\tau [T_p(q, \tau) - T(d, \tau)]^2 d\tau ds. \qquad (2)$$

Величина $F(q)$ представляет собой функционал в пространстве функций $q(x_i, \tau)$, его численное значение определяет расстояние в функциональном пространстве L_2, между экспериментально найденной $T(d, \tau)$ и рассчитанной $T_p(q, \tau)$ температурами.

Аппроксимируем функцию $q(x_i, \tau)$ кусочно-постоянной функцией на поверхности S $q(\phi, \varphi, \tau) = q_{nmk}$, где $\phi_{n-1} \leqslant \phi \leqslant \phi_n$, $\varphi_{m-1} \leqslant \varphi \leqslant \varphi_m$, $\tau_{k-1} \leqslant \tau \leqslant \tau_k$, ϕ и φ — параметрические координаты на заданной поверхности S. При введенной аппроксимации для $q(x_i, \tau)$ функционал $F(q)$ становится функцией $(N \cdot M \cdot K)$-переменных: $F(q) = \psi(q_1, q_2, ..., q_{NMK})$. Таким образом, задача ставится так, чтобы подбором величин $(q_1, q_2, ..., q_{NMK})$ обеспечить минимум величины $\psi(q_1, q_2, ..., q_{NMK}) = \Sigma[T(d, \tau, q_1, q_2, ..., q_{NMK}) - T(d, \tau)]^2 \Delta\phi_n \Delta\varphi_m \Delta\tau_k$ на всей поверхности S и временном интервале $\tau \in (0, \tau_m)$, где $F(q)$ — уже целевая функция многих переменных.

Известно большое число работ, в которых дается решение граничной обратной задачи в данной постановке. Однако в большинстве из них для решения прямой задачи авторы использовали метод конечных разностей. Трудности использования этого метода при моделировании объемных тел общеизвестны. В данной работе при решении граничной обратной задачи в качестве прямого метода был использован метод конечных элементов. Это позволило существенно расширить класс решаемых задач, включая геометрически нелинейные задачи, имеющие широкое распространение при моделировании реальных технологических процессов. Для решения граничной обратной задачи было использовано понятие коэффициентов чувствительности. На основе данного понятия в работе [4] было дано систематическое описание решения задач данного класса.

Построение алгоритмов для нахождения коэффициентов чувствительности нелинейных граничных обратных задач

Для нахождения коэффициентов чувствительности продифференцируем уравнение теплопроводности по q_s. В результате получим

$$\sum_{i=1}^{3} \left\{ \frac{\partial}{\partial x_i}\left[\lambda(T)\frac{\partial W_s}{\partial x_i}\right] + \frac{\partial}{\partial x_i}\left[\frac{d\lambda(T)}{dT} W_s \frac{\partial T}{\partial x_i}\right] \right\} = \rho(T)c(T)\frac{\partial W_s}{\partial \tau} + \frac{d[\rho(T)c(T)]}{dT} W_s \frac{\partial T}{\partial \tau}, \qquad (3)$$

где $W_s = \dfrac{\partial T}{\partial q_s}$ — коэффициенты чувствительности, °С · м^2 · Вт$^{-1}$; q — тепловой поток, Вт · м$^{-2}$.

Величины $\frac{d\lambda(T)}{dT}$ и $\frac{d[\rho(T)c(T)]}{dT}$ могут быть определены из заданных зависимостей для $\lambda(T)$ и $c(T)$, а значения $\frac{\partial T}{\partial x_i}$ и $\frac{\partial T}{\partial \tau}$ из предварительно рассчитанной прямой краевой задачи теплопроводности.

Используя метод Галеркина, получим конечно-элементную модель для нахождения коэффициентов чувствительности при решении граничной обратной задачи:

$$-\int_V \sum_{i=1}^{3} \frac{\partial [N]^T}{\partial x_i} \cdot \frac{\partial [N]}{\partial x_i} \lambda(T) dV\{W_s\} + \int_S [N]^T \lambda(T) \frac{\partial W_s}{\partial n} dS - \int_V \sum_{i=1}^{3} \frac{\partial [N]^T}{\partial x_i} [N] \frac{d\lambda(T)}{dT} \frac{\partial T}{\partial x_i} dV\{W_s\} + \int_S \sum_{i=1}^{3} [N]^T [N] \frac{d\lambda(T)}{dT} \cdot \frac{\partial T}{\partial x_i} l_{x_i} dS\{W_s\} = \int_V [N]^T [N] c(T)\rho(T) dV \frac{\partial \{W_s\}}{\partial \tau} + \int_V [N]^T [N] \frac{\partial (c(T)\rho(T))}{dT} \cdot \frac{\partial T}{\partial \tau} dV\{W_s\}, \quad (4)$$

где $[N]$ — базисные функции формы в конечно-элементной модели.

Интегралы в выражении (4) вычисляются по объему и по поверхности исследуемого объекта. Введем обозначения в уравнение (4) для матриц, аналогичные обозначениям для нестационарного уравнения теплопроводности, записанного в виде конечно-элементной модели: $[\mathbf{C}] = \sum_{FE} \int_V [N]^T [N] c(T)\rho(T) dV$ — аналог матрицы теплоемкости;

$$[\mathbf{K}] = \sum_{FE} \int_V \sum_{i=1}^{3} \frac{\partial [N]^T}{\partial x_i} \cdot \frac{\partial [N]}{\partial x_i} \cdot \lambda(T) \cdot dV + \int_V [N]^T [N] \cdot \frac{\partial [c(T)\rho(T)]}{dT} \cdot \frac{\partial T}{\partial \tau} \cdot dV + \int_V \sum_{i=1}^{3} \frac{\partial [N]^T}{\partial x_i} [N] \cdot \frac{d\lambda(T)}{dT} \cdot \frac{\partial T}{\partial x_i} \cdot dV - \int_S \sum_{i=1}^{3} [N]^T [N] \cdot \frac{d\lambda(T)}{dT} \cdot \frac{\partial T}{\partial x_i} l_{x_i} dS \quad (5)$$

— аналог матрицы теплопроводности;

$$\{\boldsymbol{F}\} = \sum_{FE} \int_V [N]^T \lambda(T) \frac{\partial W_s}{\partial n} dS \quad (6)$$

— аналог вектора правой части.

В представленных зависимостях интегралы рассчитываются по объему конечного элемента, а результирующие матрицы — суммированием по всем конечным элементам.

Окончательно уравнение для вычисления коэффициентов чувствительности в виде конечно-элементной модели будет иметь следующий вид:

$$[\mathbf{C}] \frac{\partial \{W_s\}}{\partial \tau} + [\mathbf{K}\{W_s\} + \{\boldsymbol{F}\} = 0. \quad (7)$$

Матрица [**C**] и вектор правой части в уравнении (5) формируются так же как и для нестационарного уравнения теплопроводности.

Матрица [**K**] при формировании имеет ряд особенностей. Рассмотрим формирование составляющих для этой матрицы.

Первая составляющая

$$[\mathbf{K}]_1 = \sum_{FE} \int_V \sum_{i=1}^{3} \frac{\partial [N]^T}{\partial x_i} \cdot \frac{\partial [N]}{\partial x_i} \lambda(T) dV \quad (8)$$

полностью аналогична составляющей в матрице теплопроводности.

Вторая составляющая

$$[\mathbf{K}]_2 = \sum_{FE} \int_V [N]^T [N] \cdot \frac{\partial (c(T)\rho(T))}{dT} \cdot \frac{\partial T}{\partial \tau} \cdot dV \quad (9)$$

может быть сформирована только после предварительного расчета прямой задачи и последующего вычисления величины $\frac{\partial T}{\partial \tau}$. Данная производная может быть определена с использованием конечно-разностной аппроксимации дифференциального оператора:

$$\frac{\partial T}{\partial \tau} = \frac{\{T\}_{k+1} - \{T\}_k}{\Delta \tau}$$

$$\text{или } \frac{\partial T}{\partial \tau} = \frac{\{T\}_{k+1} - \{T\}_{k-1}}{2 \cdot \Delta \tau}, \quad (10)$$

где $\{T\}_{k+1}$, $\{T\}_k$ и $\{T\}_{k-1}$ — вектор температур, полученный из предварительного расчета прямой задачи, соответственно для моментов времени $(k+1)$, (k) и $(k-1)$. Для вычисления интеграла была использована численная процедура метода Гаусса. Значения производных $\frac{\partial T}{\partial \tau}$ в точках интегрирования по объему конечного элемента рассчитывались с помощью базисных функций, которые принимались такими же, что и для коэффициентов чувствительности. Производная теплофизических параметров $\frac{\partial (c(T)\rho(T))}{dT}$ как функция температуры определялась из температурной зависимости $c(T)\rho(T) = f(T)$ с помощью операции численного дифференцирования.

Третья составляющая в матрице [**K**]:

$$[\mathbf{K}]_3 = \sum_{FE} \int_V \sum_{i=1}^{3} \frac{\partial [N]^T}{\partial x_i} [N] \cdot \frac{d\lambda(T)}{dT} \cdot \frac{\partial T}{\partial x_i} dV \quad (11)$$

может быть сформирована после предварительного вычисления прямой задачи.

Для вычисления интегралов была также использована численная процедура метода Гаусса. Компоненты $\frac{\partial T}{\partial x_i}$ находили из предварительно рассчитанного температурного поля решения прямой задачи. Последняя, четвертая составляющая в матрице теплопроводности представлена следующей зависимостью:

$$[\mathbf{K}]_4 = \sum_{FE} \left[- \int_S \sum_{i=1}^{3} [N]^T [N] \frac{d\lambda(T)}{dT} \cdot \frac{\partial T}{\partial x_i} l_{x_i} dS \right]. \quad (12)$$

Таким образом, представленные зависимости для формирования глобальных матриц и глобального вектора правой части дают возможность использовать аналогичные численные схемы для определения коэффициентов чувствительности, что и для параболического уравнения теплопроводности.

Пример проектирования и реализации обратной задачи на основе проблемно-ориентированного высокоуровневого языка

Для решения обратной задачи была использована базовая компонентная библиотека и высокоуровневая среда программирования, описанные в работе [1].

Решение обратной задачи может быть реализовано в виде итерационной схемы (рис. 1). При формировании метода решения обратной задачи необходимо создать методы для решения прямой задачи теплопроводности и нахождения коэффициентов чувствительности.

Реализация метода решения прямой задачи на основе разработанного высокоуровневого языка была описана в работе [1], и включает класс **CFEMTempTask** (класс описания задачи теплопроводности) и его метод **SolveNstUnlnQTask** (решение нестационарной, нелинейной задачи теплопроводности с заданием граничного условия в виде теплового потока).

Для решения задачи определения коэффициентов чувствительности был создан класс **CFEMSKTask**, являющийся наследником класса **CFEMTempTask** (рис. 2). Обратим внимание, что в список компонентов для формирования системы линейных алгебраических уравнений (СЛАУ) дополнительно включены компоненты **cmK2, cmK3, cmK4, cmF,** соответствующие реализациям согласно соотношениям (9), (11), (12) и (6).

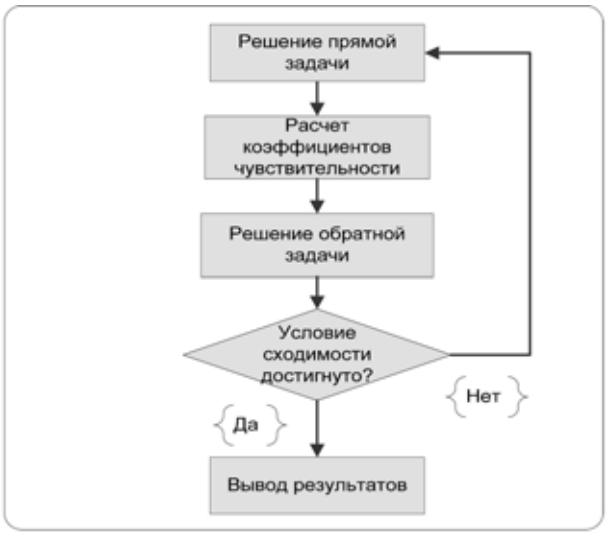


Рис. 1. Блок-схема решения обратной задачи

Описание класса расчета коэффициентов чувствительности

```
CLASS CFEMSKTask(CFEMTempTask)
      COMPONENT_LIST ={cmK2, cmK3, cmK4, cmF}
      METHOD SolveSKTask( )
END_CLASS
```

Реализация метода расчета коэффициентов чувствительности

```
CLASS CFEMSKTask METHOD SolveSKTask( )
      ODU(KRANK_NIKOLSON)
      BEGIN_CICLE
      GET_TIME()
      ADD_COMPONENT(cmK)
      ADD_COMPONENT(cmK2)
      ADD_COMPONENT(cmK3)
      ADD_COMPONENT(cmK4)
      ADD_COMPONENT(cmF)
      ADD_COMPONENT(cmC)
      GENERATE_SLAU()
      END_CICLE
END_METOD
```

Рис. 2. Описание решения задачи по расчету коэффициентов чувствительности

Реализация метода расчета коэффициентов чувствительности (рис. 2) включает организацию цикла по временным шагам, формирование и решение СЛАУ.

Для формирования и решения обратной задачи был создан класс базовой библиотеки **CRevTask**. В реализации метода решения обратной задачи класса **CRevTask** использован метод регуляризации Тихонова с регуляризатором нулевого порядка.

Решение обратной задачи теплопроводности на основе методов, изложенных в работе [1],

предполагает создание комплексной модели. Класс описания комплексной модели обратной задачи **CRevCM** создан как наследник базового класса **CComplexModel** (рис. 3).

В список задач, входящих в класс описания комплексной модели **CRevCM,** включены объекты классов библиотеки исследователя **CFEMTempTask** и **CFEMSKTask,** имеющие идентификаторы DirectTask и SKTask, а также объект класса базовой библиотеки **CRevTask,** имеющий идентификатор RevTask.

Алгоритм решения комплексной задачи — самой обратной задачи — реализуется методом SolveProblem (рис. 3).

Для решения каждой из задач необходимо сформировать соответствующие проекты, включающие полное описание исходных данных. Описание проектов проводится в среде формирования конечно-элементной задачи.

Описание класса решения обратной задачи

```
CLASS CRevCM (CComplexModel)
      TASK_LIST ={CFEMTempTask DirectTask,
                  CFEMSKTask SKTask,
                  CRevTask RevTask}
      METHOD SolveProblem( )
END_CLASS
```

Реализация метода расчета обратной задачи

```
CLASS CRevCM METHOD SolveProblem( )
      DirectTask  LOAD_DATA(DirectTask.prj)
      SKTask  LOAD_DATA(SKTask.prj)
      RevTask  LOAD_DATA(RevTask.prj)
      BEGIN_CICLE
            DirectTask  SolveNstUnlnQTask ( )
            LINK_TEMP_WITH_SK(DirectTask,
      SensativeTask)
            SKTask  SolveSKTask()
            LINK_SK_WITH_REV(SKTask, RevTask)
            RevTask  SOLVE_TASK( )
            RevTask  CHECK_CONVERGENCE( )
            LINK_REV_WITH_TEMP(RevTask,
      DirectTask)
      END_CICLE
      RevTask  SAVE_RESULT( )
END_METHOD.
```

Рис. 3. Описание реализации обратной задачи

Инициализация расчета обратной задачи

```
PROGRAM
      CRevCM RCM
      RCM  SolveProblem( )
END_PROGRAM
```

Рис. 4. Инициализация расчета обратной задачи

Доступ к описанию исходных данных объекты DirectTask, SKTask и RevTask получают с помощью метода **LOAD_DATA** (рис. 3).

Первым шагом цикла обратной задачи является решение прямой задачи, которое проводится с помощью вызова метода **SolveNstUnlnQTask** объекта класса **CFEMTempTask.**

Затем результаты решения тепловой задачи с помощью метода **LINK_TEMP_WITH_SK** передаются как исходные данные для решения задачи нахождения коэффициентов чувствительности.

Далее с помощью метода **SolveSKTask** объекта класса **CFEMSKTask** решается задача нахождения коэффициентов чувствительности, результаты решения которой с помощью метода **LINK_SK_WITH_REV** передаются как исходные данные для формирования и решения обратной задачи.

Решение обратной задачи проводится с помощью метода **SOLVE_TASK** объекта класса **CRevTask.** В случае, если сходимость обратной задачи не достигнута, полученные результаты с помощью метода **LINK_REV_WITH_TEMP** передаются как исходные данные для прямой задачи.

Цикл повторяется до достижения сходимости обратной задачи, после чего с помощью метода **SAVE_RESULT** объекта класса **CRevTask** полученные результаты сохраняются на жестком диске и выполнение метода, реализующего комплексную модель решения обратной задачи, завершается.

Инициализация расчета, связанного с комплексной моделью обратной задачи, сводится к объявлению объекта класса, описывающего комплексную модель, и вызову метода, содержащего алгоритм решения комплексной задачи (рис. 4).

Предложенная технология существенно расширяет возможности математического моделирования (рис. 5).

Исследователь предметной области получает инструмент по созданию базы данных граничных условий, определяемых из решения обратной задачи теплопроводности, т. е. по сути можно говорить о настройке математической модели технологического процесса или технической системы на реальные условия эксплуатации.

Вместе с тем, можно рассматривать задание температуры внутри исследуемого объекта как некую образцовую характеристику, которую необходимо обеспечить для получения необходимых характеристик реального объекта. Тогда можно говорить об управлении процессом термического воздействия на свойства реального объекта путем формирования граничного условия, определяемого из решения обратной задачи. Сюда можно отнести различные процессы термической обработки.

Сопровождение и развитие базового и прикладного программного обеспечения на основе предложенной технологии

Описанная технология включает два уровня разработки и сопровождения программного обеспечения: первый уровень — разработка и сопровождение базовой компонентной библиотеки и высокоуровневой среды программирования; второй уровень — разработка и сопровождение прикладных программных приложений, организованных в виде предметно-ориентированной библиотеки моделей.

Рис. 5. Математическая модель процесса, настроенная на реальные условия эксплуатации

Сопровождение программного обеспечения первого уровня требует создания необходимого организационного и методического обеспечения. В этом направлении на кафедре "Персональные ЭВМ и сети" Московского государственного университета приборостроения и информатики на основе инновационных технологий разработки открытого программного обеспечения [5] ведется работа по созданию базовой библиотеки компонентов, регламента ее использования и расширения и созданию документации по описанию ее функционального состава. В настоящий момент уровень развития базовой библиотеки дает возможность решать с использованием конечно-элементного метода задачи трех типов: тепловые стационарные и нестационарные задачи теплопроводности; упругие и термоупругие прочностные задачи; обратные задачи теплопроводности. На кафедре установлена система управления версиями, использующая централизованную модель с единым хранилищем документов, управляемая специальным сервером. Это дает возможность взаимодействовать с пользователями, использующими базовую библиотеку и создающими прикладное программное обеспечение через удаленный доступ (рис. 6, см. вторую сторону обложки). Второй уровень организационно поддерживается пользователями, разрабатывающими программное приложение. Это могут быть кафедры или научно-производственные предприятия, которые используют в своей работе описанные выше модели или их модификации. В случае отсутствия необходимого инструмента для создания программного приложения в базовой версии пользователь формирует требования для их реализации. В результате совместных консультаций создается документ по расширению функционального наполнения базового программного обеспечения, определяются сроки и ресурсы для его создания.

Предложенная организационная и методическая форма разработки и сопровождения программного обеспечения на основе представленной технологии дает возможность существенно ускорить процесс реализации моделей различного назначения.

Список литературы

1. **Александров А. Е.** Разработка и проектирование прикладных программных приложений на основе порождающего программирования. // Информационные технологии. 2008. № 9. С. 2—9.

2. **Александров А. Е.** Диагностика теплового состояния системы торможения авиационных колес // Контроль. Диагностика. 2002. № 10.

3. **Александров А. Е.** Методология проектирования прикладных адаптивных программных систем с использованием многоуровневой инструментальной среды: дис. ... д-ра техн. наук. 05.13.11, 05.13.18. М., 2006.

4. **Бек Дж., Блакуэлл Б., Сент-Клер И., мл.** Некорректные обратные задачи теплопроводности: Пер. с англ. М.: Мир, 1989. 312 с.

5. **Михайлов Б. М., Зеленко Г. В., Рощин А. В.** Проблемы информационного обеспечения открытого образования // Межвузовский сборник научных трудов. М.: МГАПИ. 2003. Вып. 6. С. 10—15.

6. **Михайлов Б. М., Ульянов М. В.** Инструментальные средства для исследования эффективности алгоритмов: концепция и основные принципы построения // Информационные технологии. 2005. № 2. С. 54—57.

7. **Тихонов А. Н., Кальнер В. Д., Гласко В. Б.** Математическое моделирование технологических процессов и метод обратных задач в машиностроении. М.: Машиностроение, 1990. 264 с.

УДК 004.031

Э. А. Мухачева, д-р техн. наук, проф.,
Э. И. Хасанова, аспирант,
Уфимский государственный авиационный технический университет,
e-mail: xasel@mail.ru

Гильотинное размещение контейнеров в полосе: комбинирование эвристических технологий

Рассматривается NP-трудная задача ортогонального размещения контейнеров сквозными рядами. Последние означают гильотинное разделение размещения в прямоугольной полосе. При этом имеются дополнительные ограничения — условия комфортности загрузки и разгрузки предметов. К приведенным моделям сводятся многие проблемы загрузки крупногабаритных грузов на палубы судов и отсеки самолетов. Для решения задач используются уровневая и мультиметодная технологии. Приведены результаты численных экспериментов.

Ключевые слова: *гильотинное размещение, уровневые алгоритмы, эволюционные метаэвристики, генетические алгоритмы, комбинирование эвристик.*

Введение

Задачи размещения крупногабаритных грузов на палубах судов, в отсеках самолетов или железнодорожных вагонах в простейшем случае описываются классическими моделями задач раскроя-упаковки с гильотинным или негильотинным размещением. Под гильотинным понимается размещение сквозными рядами с обеспечением подходов к каждому из них. Аналогом такого размещения является гильотинный раскрой, реализуемый только сквозными резами, параллельными кромкам материала. В этой статье нас будет интересовать задача размещения, классическая постановка которой описывается моделью гильотинного раскроя. Впервые эта задача была поставлена Л. В. Канторовичем еще в 1939 г. [1]. Она рассматривалась им как задача получения максимального числа комплектов деталей в заданном ассортиментном отношении. Для ее решения был предложен метод *разрешающих* множителей, который предвосхитил появившееся позднее линейное программирование. Тогда, Л. В. Канторович применил непрерывную релаксацию к решению целочисленной задачи раскроя. Метод был описан Л. В. Канторовичем и В. А. Залгаллером в 1951 г. (1971 г.) [2]. Различные вычислительные схемы (*сеточный метод* и *метод склейки*) были разработаны и исследованы Э. А. Мухачевой [3] и И. В. Романовским [4]. Независимо появились аналогичные работы и за рубежом: Gilmory P., Gomory R. [5] и Terno J., Scheithauer G. [6]. По сути, классическая проблема массового раскроя оказалась закрытой. Производственные модификации были развиты на основе линейного программирования Э. А. Мухачевой [7]. М. Гэри и Д. Джонсон доказали, что целочисленная задача одномерного раскроя является *NP*-трудной [8]. А это, в свою очередь, означает, что целочисленная проблема гильотинного размещения является NP-трудной в сильном смысле. Для решения таких задач используется полный перебор, а точных алгоритмов полиномиальной сложности пока неизвестно. Поэтому стали бурно развиваться приближенные и эвристические методы, в том числе и метаэвристики, обобщение которых привело к появлению технологий конструирования следующих рациональных раскроев и размещений: *нижний-левый* [9]; *комбинирования эвристик* (мультиметодная технология [10]); *блочная* [11]; *уровневая* [12]. Технологии применяются на каждой итерации метаэвристики в качестве декодера построения дочерней карты размещения. В этой статье в качестве базовой используется уровневая технология. Она приведена в разделе 1, в разделе 2 описаны ее модификации с использованием мультиметодных технологий. Уровневые алгоритмы основаны на *послойных методах*, которые были ранее независимо предложены A. Adamovich, A. Albano [13] и Э. А. Мухачевой [14].

1. Классическая задача гильотинного размещения

В качестве основной будем рассматривать задачу гильотинного размещения предметов в полубесконечную полосу (*2-Dimensional Guillotine Strip Placing*, ***2DGSP***). Имеется прямоугольная полоса заданной ширины W и неограниченной длины, а также набор прямоугольных деталей заданных размеров w_i, l_i, $i = \overline{1, m}$, где m — число прямоугольников; w_i, l_i — длина и ширина i-й детали. Таким образом, исходная информация задачи может быть представлена следующим набором данных:

$$< W;\ m;\ w = (w_1, w_2, ..., w_m);\ l = (l_1, l_2, ..., l_m) >.$$

Введем прямоугольную систему координат: оси Ox и Oy совпадают соответственно с нижней и боковой сторонами полосы. Положение каждого прямоугольника P_i зададим координатами (x_i, y_i) его левого нижнего угла. Решение рассматриваемой задачи может быть представлено в виде набора

$$GP = < x = (x_1, ..., x_i, ..., x_m),\ y = (y_1, ..., y_i, ..., y_m) >,$$

где x_i, y_i — координаты левого нижнего угла прямоугольника, $i = \overline{1, m}$.

Набор $GP = <x, y>$ называется *допустимым гильотинным размещением*, если выполняются следующие условия:

1. Ребра предметов параллельны ребрам полосы:

$$(\rho_x^i = l_i) \wedge (\rho_y^i = w_i), \quad i = \overline{1, m},$$

где ρ_x^i, ρ_y^i — проекции заготовки i на оси координат Ox и Oy.

2. Взаимное непересечение деталей: $\forall i \neq j : i, j = 1, ..., m,$

$$((x_i \geqslant x_j + l_j) \vee (x_j \geqslant x_i + l_i)) \vee ((y_i \geqslant y_j + w_j) \vee \vee (y_j \geqslant y_i + w_i)).$$

3. Непересечение деталей с гранями полосы: $\forall i = 1, ..., m,$

$$(x_i \geqslant 0) \wedge (y_i \geqslant 0) \wedge (y_j + w_i \leqslant W).$$

Набор $GP = <x, y>$ называется *гильотинным прямоугольным размещением*, если кроме условий 1—3 выполняется еще условие гильотинности.

4. Условие гильотинности:

для любого прямоугольника P с размерами (w, l), $w \neq w_i \vee l \neq l_i$, $i = \overline{1, m}$, выполняется условие разделения на два прямоугольника $P'(w', l')$ и $P''(w'', l'')$ такие, что $((w' = w'' = w) \wedge (l' + l'' = l)) \vee \vee ((w' + w'' = w) \wedge (l' = l'' = l))$ и $\forall i = \overline{1, m}$, если $P_i \in P'$, то $P_i \notin P''$ и если $P_i \in P''$, то $P_i \notin P'$.

Задача 2DGSP. При исходных данных $<W; m; w; l>$ требуется минимизировать

$$L(x; y) = \max_{i = \overline{1, m}} (x_i + l_i) \qquad (1)$$

на множестве размещений $\{GP = <x, y>\}$, удовлетворяющих условиям 1—4.

Как указывалось выше, рассматриваемая задача является *NP*-трудной, т. е. для нее неизвестен точный алгоритм, гарантирующий получение оптимума за полиномиальное от m время. Поэтому наряду с точными методами разрабатываются эвристические алгоритмы.

Что касается дополнительных ограничений на проход (проезд) между рядами размещенных предметов, то они учитываются при корректировке данных: к длине и ширине каждого предмета добавляются припуски, либо используется методика корректировки конечного решения. В этом случае задача решается для $\tilde{W} = W - \tilde{\Delta}_W$, где $\tilde{\Delta}_W$ — общий припуск на ширину области размещения. На втором этапе ряды с предметами раздвигаются, чтобы освободить проходы. Это действие описательно-организационное, на эффективность алгоритма оно никак не влияет и в дальнейшем подробнее не описывается.

В уровневых алгоритмах используется прием, когда каждый новый элемент упаковывается с выравниванием по левому и нижнему краю. Через правую сторону прямоугольника максимальной длины текущего уровня проводят сквозную вертикальную линию. Исходный прямоугольный объект оказывается разделенным на уровни прямоугольной формы, содержащие целиком входящие прямоугольники. Что касается стратегии выбора прямоугольников, то она может исходить из известных правил. При этом различаются конструктивные и эволюционные уровневые алгоритмы.

2. Конструктивные и эволюционные уровневые алгоритмы

Конструктивные эвристики. Конструктивные эвристики могут использоваться самостоятельно для решения задачи за один проход алгоритма или в качестве декодеров в рамках эволюционных алгоритмов. Обзор конструктивных эвристик подробно представлен в работе [15]. Здесь мы воспользуемся стратегией *первый подходящий* (*First Fit,* **FF**), которая состоит в следующем:

1. Первый предмет помещаем в первый уровень.

2. На k-м шаге находим уровень с наименьшим номером, в котором размещается k-й предмет. Если такого уровня нет, то открываем новый уровень и помещаем предмет в него.

Цикл продолжается до полной упаковки. При этом каждый новый элемент размещается с выравниванием по левому нижнему краю.

Алгоритм FF — конструктивный, позволяет получить конечное допустимое решение за одну итерацию.

Вход: $<W; m; w = (w_1, w_2, ..., w_m); l = (l_1, l_2, ..., l_m)>$

Выход: L — длина занятой части полосы; CC — коэффициент раскроя; $GP = <x, y>$ — гильотинное прямоугольное размещение.

Пусть задан список $\pi = (\pi_1, \pi_2, ..., \pi_\nu, ..., \pi_m)$, где π_ν — номер детали в ν-й позиции списка π.

Шаг 1. *Выделение первого уровня в полосе:* в качестве левого нижнего угла первого уровня принимается начало координат $x' = 0$, $y' = 0$.

Шаг 2. *Размещение первой детали в текущий уровень:* первая неиспользованная деталь из списка π укладывается в левый нижний угол текущего уровня.

Шаг 3. *Проверка возможности размещения неиспользованных деталей из списка* π *в уровень:* **если** ширина очередной детали не превышает остаточной ширины, **то** она размещается сверху уже уложенных деталей с выравниванием по левому нижнему краю.

Шаг 4. *Проверка выполнения условия останова:* **если** все детали уложены в полубесконечную полосу, **то** работа алгоритма заканчивается,

— коэффициент размещения

$$PC = \frac{\sum_{i=1}^{m} w_i l_i}{WL}; \qquad (2)$$

— длина занятой части полосы увеличивается на длину l уровня, т. е. $L = L + l$,

иначе открывается новый уровень; переход к **шагу 2**.

Одной из результативных модификаций конструктивной эвристики *FF* является первый по убыванию подходящей длины (*First Fit Decreasing Length*, FFDL). Согласно этому методу сначала все детали сортируются в порядке невозрастания длины l_i и в соответствии с полученным списком размещаются в полосу. При этом для обеспечения комфортности разгрузки предметы сортируются по принадлежности к пунктам доставки и размещаются группами в порядке, обратном маршруту следования [16].

Результат работы алгоритма всегда одинаков и зависит только от исходных данных и порядка их размещения в списке π.

Эволюционные алгоритмы. В методах эволюционных вычислений используются конструктивные эвристики в процессе построения решений. На каждой итерации в соответствии с выбранной стратегией конструируется допустимое решение, которое сопоставляется с полученным на предыдущих итерациях рекордом. Процесс продолжается до выполнения критерия останова: затраченное время, количество итераций, либо достижение целевой функцией заданного значения. Полученные при повторных запусках эволюционных алгоритмов размещения могут различаться при одних и тех же исходных данных.

Рассмотрим одноточечный эволюционный алгоритм (1 + 1)-EA с двумя способами перехода от одной особи к следующей: (1 + 1)-EA(m) — наивный эволюционный алгоритм и (1 + 1)-EA(2) — эволюционный алгоритм поиска решения в окрестности (m и 2 — число переставляемых в списке деталей на одной итерации).

Алгоритм (1 + 1)-EA(m) — наивный эволюционный алгоритм.

Вход: $<W$; m; $w = (w_1, w_2, ..., w_m)$;
$l = (l_1, l_2, ..., l_m)>$; число итераций.

Выход: L — длина занятой части полосы; PC — коэффициент размещения; $GP = <x, y>$ — прямоугольное размещение.

Шаг 1. *Применение алгоритма FF к исходному списку* π: найденное решение принимается в качестве рекорда.

Шаг 2. *Преобразование списка* π: список π генерируется случайным образом.

Шаг 3. *Выполнение алгоритма FF.*

Шаг 4. *Сопоставление с рекордным значением:* полученный на шаге 3 PC сравнивается с достигнутым на предыдущих итерациях рекордом R. Если $PC > R$, то в качестве рекордного значения R принимается PC.

Шаг 5. *Проверка выполнения условия останова:* **если** число итераций исчерпано, **то** работа алгоритма заканчивается, в качестве решения принимается достигнутый рекорд, **иначе** переход к **шагу 2**.

Алгоритм (1 + 1)-EA(2) — эволюционный алгоритм поиска решения в окрестности, отличается от алгоритма (1 + 1)-EA(m) способом генерации нового списка π. На шаге 2 случайно выбираются номера элементов i и j такие, что $i, j \in [l, m]$, $i \neq j$. Формируется новый список π, в котором детали с номерами i и j меняются местами. Таким образом, исследуются размещения не из всей области допустимых решений, а только из окрестности начального списка π.

Другой особенностью алгоритма (1 + 1)-EA(2) является то, что в отличие от наивного алгоритма в нем выполняются два вида итераций: простые и обобщенные. На каждой простой итерации текущий список π преобразуется путем перестановки двух случайно выбранных элементов. Таким образом, формируется окрестность начального решения. На обобщенной итерации происходит возврат к исходному списку π.

Численный эксперимент 1. Для сравнения описанных эвристик был использован набор примеров Евы Хоппер [17]. Он включает в себя по три примера из семи различных классов размещаемых элементов.

В качестве исходного списка π для эволюционных алгоритмов рассматривались случайный список деталей и список, полученный упорядочением деталей по невозрастанию длины.

Число итераций для наивного эволюционного алгоритма было принято 50 000. Для алгоритма поиска решения в окрестности экспериментальным путем были найдены лучшие значения для количества простых и обобщенных итераций при общем число итераций 50 000. Для заданного списка деталей это 225 простых и 222 обобщенных итераций, для упорядоченного списка — 5 простых и 10 000 обобщенных итераций.

Работа алгоритмов оценивалась с помощью коэффициента размещения.

Результаты сравнения приведены в табл. 1.

Из табл. 1 видно, что алгоритмы, в которых в качестве исходного принят упорядоченный список, позволяют получить лучшее решение по сравнению с алгоритмами, использующими случайный список деталей. При этом эволюционный алгоритм (1 + 1)-EA(2) только в двух случаях из 21 позволил незначительно улучшить результат

Таблица 1

Результаты численного эксперимента 1

№	Конструктивная эвристика	Эволюционные эвристики			
	FEDL	Случайный список		Упорядоченный список	
		(1 + 1)-EA(m)	(1 + 1)-EA(2)	(1 + 1)-EA(m)	(1 + 1)-EA(2)
1	**0,741**	**0,741**	**0,741**	**0,741**	**0,741**
2	**0,690**	0,667	0,645	**0,690**	**0,690**
3	**0,870**	0,833	0,833	**0,870**	**0,870**
4	**0,750**	0,682	0,682	**0,750**	**0,750**
5	**0,789**	0,750	0,789	**0,789**	**0,789**
6	**0,652**	**0,652**	**0,652**	**0,652**	**0,652**
7	**0,750**	0,667	0,682	**0,750**	**0,750**
8	**0,714**	0,625	0,600	**0,714**	**0,714**
9	**0,698**	0,638	0,638	**0,698**	**0,698**
10	**0,811**	0,594	0,612	**0,811**	**0,811**
11	**0,811**	0,571	0,556	**0,811**	**0,811**
12	**0,750**	0,600	0,594	**0,750**	**0,750**
13	**0,891**	0,672	0,677	**0,891**	**0,891**
14	**0,849**	0,608	0,625	**0,849**	**0,849**
15	**0,849**	0,604	0,604	**0,849**	**0,849**
16	**0,882**	0,591	0,588	**0,882**	**0,882**
17	0,822	0,524	0,536	**0,822**	**0,828**
18	**0,863**	0,566	0,556	**0,863**	**0,863**
19	**0,920**	0,524	0,527	**0,920**	**0,920**
20	0,848	0,457	0,449	**0,848**	**0,851**
21	**0,879**	0,429	0,442	**0,879**	**0,879**

FFDL. Таким образом, использование эволюционной стратегии в этом случае нецелесообразно.

Сравнение эволюционных эвристик (1 + 1)-EA(m) и (1 + 1)-EA(2) показывает, что как для случайного, так и для упорядоченного списка поиск в локальной области позволяет получить решения лучше, чем наивный алгоритм.

В целом, полученные значения далеки от оптимума. Это видно и из рис. 1 (см. вторую сторону обложки), на котором приведена одна из карт размещения.

В целях улучшения решения включим в алгоритм процедуру заполнения боковых пустот.

3. Заполнение боковых пустот

Назовем незаполненные области каждого уровня справа от размещенных прямоугольников *боковыми пустотами*.

Пусть имеется некоторое частичное размещение. Будем считать, что боковые пустоты текущего уровня k еще не заполнены (рис. 1, см. вторую сторону обложки).

Обозначим: π — исходный список элементов; π^+ — множество элементов исходного списка, размещенных в полосе; π^- — множество не размещенных элементов исходного списка, т. е. $\pi = \pi^- \cup \pi^+$, $\pi^- \cap \pi^+ = \phi$.

Сначала необходимо выделить пустые прямоугольные области, которые можно гильотинно заполнить оставшимися прямоугольниками.

Способы выделения и заполнения боковых пустот. Будем различать горизонтальный и вертикальный способы выделения боковых пустот. Что касается их заполнения, то для выполнения этой процедуры воспользуемся полным перебором неразмещенных деталей. Для выбора одной из них определяем критерии размещения. Обе эти процедуры принадлежат мультиметодной технологии [10].

Горизонтальный способ выделения пустот является наиболее простым. Он заключается в том, что в качестве незанятой области рассматривается свободное пространство справа от каждой детали в частичном размещении. При этом ширина пустой области равна ширине детали, а длина — разности длины уровня и длины детали (рис. 2, *а*). Обозначим k — номер заполняемого уровня.

Алгоритм выделения и заполнения боковых пустот горизонтальным способом

Вход: $< W$; m; l_k; $w = (w_1, w_2, ..., w_m)$;
$l = (l_1, l_2, ..., l_m)>$

π^- — список неразмещенных элементов;

(x_i, y_i), $i \in \pi^+$ — частичное размещение;

x_k — координата x левого нижнего угла k-го уровня;

l_k — длина k-го уровня;

Выход: π_k^+ — список элементов, размещенных в k-м уровне;

(x_i, y_i), $i \in \pi_k^+ \cup \pi^+$ — частичное размещение;

Шаг 1. *Нахождение размеров и координат i-й пустой области:* для каждого элемента k-го уровня из частичного размещения находятся размеры и координаты расположенной справа пустой области.

Шаг 2. *Поиск пустой прямоугольной области с наибольшим значением y.*

Шаг 3. *Пересчет высоты верхней боковой пустоты:* если над самой верхней деталью в уровне

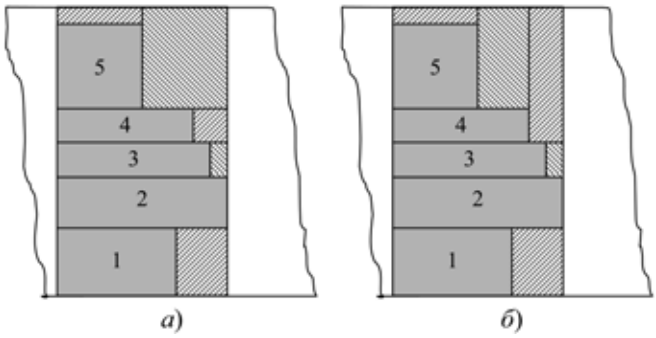


Рис. 2. Способы выделения боковых пустот:
а — горизонтальный; *б* — вертикальный

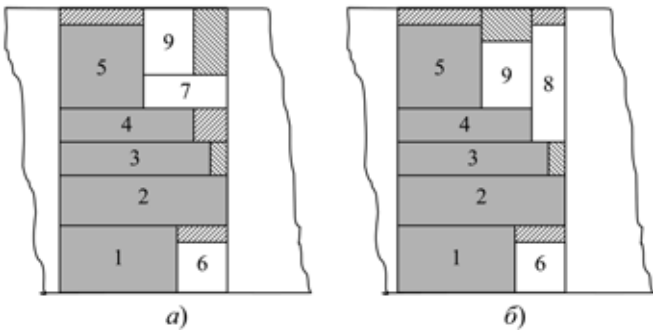


Рис. 3. Пример заполнения выделенных боковых пустот:
а — пример заполнения горизонтальных пустот; *б* — пример заполнения вертикальных пустот

имеется незанятая область, то эта область объединяется с пустотой, расположенной справа от самой верхней детали в уровне.

Шаг 4. *Заполнение выделенных боковых пустот:* перебор неразмещенных элементов для заполнения боковой пустоты по критерию коэффициента заполнения, при этом координаты элементов рассчитываются с учетом координат пустой области в полубесконечной полосе (рис. 3, *а*).

Вертикальный способ выделения пустот предполагает, что ранее уже были выделены горизонтальные области, способ основан на их преобразовании. Горизонтальные пустоты рассматриваются в порядке невозрастания длин расположенных слева от них деталей. Для каждой пустой горизонтальной области ищутся смежные пустоты и исследуется целесообразность их объединения в одну вертикальную область (рис. 2, *б*).

Алгоритм выделения и заполнения боковых пустот вертикальным способом

Вход: $<W; m; w = (w_1, w_2, ..., w_m);$
$l = (l_1, l_2, ..., l_m)>;$
π^- — список неразмещенных элементов;
(x_i, y_i), $i \in \pi^+$ — частичное размещение;
x_k — координата x левого нижнего угла k-го уровня;

Выход: π_k^+ — список элементов, размещенных в k-й уровень;
(x_i, y_i), $i \in \pi_k^+ \cup \pi^+$ — частичное размещение.

Шаг 1. Выделение боковых пустот горизонтальным способом.

Шаг 2. Определение первоначальных параметров пустой области.

Выбираем пустую область, относящуюся к детали максимальной длины в уровне.

Шаг 3. Объединение смежных боковых пустот.

Шаг 4. Пересчет параметров смежных областей.

Шаг 5. Заполнение выделенной боковой пустоты по стратегии полного перебора неразмещенных деталей.

Шаг 6. Проверка целесообразности объединения смежных областей: если ни одна деталь не уместилась в пустую область, то объединенные области возвращаются к исходному виду.

Шаг 7. Если есть неисследованные пустоты, то переход к шагу 2.

Пример заполнения вертикальных боковых пустот приведен на рис. 3, *б*.

В зависимости от типа прямоугольников в исходных данных хорошее решение может быть получено горизонтальным или вертикальным способом заполнения боковых пустот. Целесообразно, следуя мультиметодной технологии, случайно определять тип пустоты и размещать в ней деталь, выбранную полным перебором из π^-.

Численный эксперимент 2. Используем для эксперимента примеры численного эксперимента 1, включив в алгоритм процедуру заполнения пустых боковых областей. Результаты вычислений приведены в табл. 2. Расчет каждого примера повторялся 10 раз, выбрано лучшее из полученных решений.

Из табл. 2 видно, что включение процедуры заполнения боковых пустот позволило получить значительно лучшие размещения. В первом примере было достигнуто оптимальное решение.

Проведенный эксперимент также показал, что при заполнении боковых пустот эволюционный алгоритм поиска в окрестности (1 + 1)-EA(2) позволяет существенно улучшить результаты по сравнению с результатами, полученными однопроходной эвристикой FFDL. Наивный эволюционный алгоритм (1 + 1)-EA(m) показывает результаты хуже, чем поиск в локальной окрестности.

Рассмотренные примеры доказывают целесообразность использования эволюционных методик одновременно с включением в уровневые алгоритмы процедуры заполнения пустых областей.

Заключение

В статье рассмотрена задача размещения крупногабаритных грузов в отсеках самолетов, на палубах судов, железнодорожных и автомобильных платформах и других транспортных средствах (ТС). В простейшем случае она описывается моделью гильотинного раскроя полубесконечной полосы. Длина ТС учитывается только на конечном этапе погрузки. Уже это позволяет игнорировать ее на начальных этапах и далее учитывать как дополнительное ограничение. Кроме того, расчет карты размещения ведется только для одного ТС, и план загрузки является окончательным.

Таблица 2

Результаты численного эксперимента 2

№	Уровневый алгоритм (1 + 1)-EA(2)	Заполнение боковых пустот				
		Конструктивная эвристика	Эволюционные эвристики			
		FFDL	Случайный список		Упорядоченный список	
			(1 + 1)-EA(m)	(1 + 1)-EA(2)	(1 + 1)-EA(m)	(1 + 1)-EA(2)
1	0,741	0,800	**1,000**	**1,000**	**1,000**	**1,000**
2	0,690	0,833	**0,909**	**0,909**	**0,909**	**0,909**
3	0,870	0,909	0,909	0,909	**0,952**	**0,952**
4	**0,750**	**0,750**	**0,750**	**0,750**	**0,750**	**0,750**
5	0,789	0,882	**0,938**	**0,938**	**0,938**	**0,938**
6	0,652	0,652	**0,714**	**0,714**	**0,714**	**0,714**
7	0,750	**0,857**	**0,857**	**0,857**	**0,857**	**0,857**
8	0,714	0,811	**0,833**	**0,833**	**0,833**	**0,857**
9	0,698	**0,789**	**0,789**	**0,789**	**0,789**	**0,789**
10	0,811	0,882	0,923	0,923	0,923	**0,968**
11	0,811	0,896	0,896	0,909	0,909	**0,952**
12	0,750	0,938	0,923	0,938	0,938	**0,968**
13	0,891	0,918	0,947	0,947	0,938	**0,978**
14	0,849	0,882	0,918	0,928	0,918	**0,957**
15	0,849	0,909	0,918	0,938	0,938	**0,968**
16	0,882	0,938	0,938	0,945	0,938	**0,976**
17	0,828	0,952	0,902	0,923	0,916	**0,968**
18	0,863	0,923	0,923	0,930	0,930	**0,968**
19	0,920	0,945	0,923	0,934	0,945	**0,972**
20	0,851	0,956	0,920	0,923	0,952	**0,976**
21	0,879	0,952	0,916	0,927	0,930	**0,976**

Для расчета размещения грузов подходящим оказался уровневый подход. Однако при несомненном удобстве гильотинного размещения, он оставляет в каждом уровне много свободного места, т. е. не является рациональным. Предложена модификация уровневой технологии, сводящаяся к последовательному заполнению боковых пустот, образованных размещением предметов в данный уровень. На этом этапе предложено использовать процедуры комбинирования эвристик: случайный выбор направления формирования пустот и полный перебор предметов для его заполнения. Проведены численные эксперименты, подтверждающие эффективность предложенной комбинации. Она удобна и для комфортной загрузки и выгрузки предметов в пунктах назначения.

В статье не рассмотрены вопросы, связанные с обходом препятствий, которыми изобилуют палубы судов и других ТС. Дальнейшая модификация метода комбинации технологий будет рассмотрена в специальной статье.

Список литературы

1. **Канторович Л. В.** Математические методы организации планирования производства. Л.: Изд-во ЛГУ, 1939. 68 с.
2. **Канторович Л. В., Залгаллер В. А.** Рациональный раскрой промышленных материалов. Новосибирск: Наука СО, 1971. 299 с.
3. **Мухачева Э. А.** Рациональный раскрой прямоугольных листов на прямоугольные заготовки // Оптимальное планирование: Сб. научных трудов СО АН СССР. 1966. Вып. 6. С. 43—115.
4. **Романовский И. В.** Алгоритмы решения экстремальных задач. М.: Наука, 1977.
5. **Gilmore P., Gomory R.** A Linear Programming approach to the Cutting Stock Problem // Operation Research. 1961. Vol. 9. P. 849—859.
6. **Terno J., Lindeman R., Scheithauer G.** Zuschnitprobleme und ihre praktische losung. Leipzig: WEB Fachbuchverlag, 1987.
7. **Мухачева Э. А.** Рациональный раскрой промышленных материалов. Применение в АСУ. М.: Машиностроение, 1984. 176 с.
8. **Гэри М., Джонсон Д.** Вычислительные машины и труднорешаемые задачи. М.: Мир., 1979. 416 с.
9. **Liu P., Teng H.** An improval BL-algorithm for genetic algorithm for the orthogonal packing of rectangles // European Journal of Operation Research. 1999. N 112. P. 413—420.
10. **Норенков И. П.** Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. 1999. № 1. С. 2—7.
11. **Филиппова А. С.** Моделирование эволюционных алгоритмов решения задач прямоугольной упаковки на базе технологии блочных структур // Информационные технологии. 2006. № 6. Приложение. 32 с.
12. **Lodi A., Martello S., Vigo D.** Recent advances on two-dimensional bin packing problems // Discrete Applied Mathematics. 2002. N 123. P. 379—396.
13. **Adamovich A., Albano A.** Nesting two-dimensional shapes in rectangular Modules // Comput. Aided Design. 1976. N 8 (1). P. 27—33.
14. **Мухачева Э. А., Верхотуров М. А., Мартынов В. В.** Модели и методы расчета раскроя-упаковки геометрических объектов. Уфа: Изд-во УГАТУ, 1998. 216 с.
15. **Филиппова А. С.** Проблемы декодирования прямоугольных упаковок: краткий обзор современных технологий // Информационные технологии. 2005. № 12. С. 13—20.
16. **Мухачева Э. А., Бухарбаева Л. Я., Филиппов Д. В., Карипов У. А.** Оптимизационные проблемы транспортной логистики: оперативное размещение контейнеров при транспортировке грузов // Информационные технологии. 2008. № 7 (143). С. 17—22.
17. **Hopper E., Turton H.** A review of the application of meta-heuristic algorithms to 2D regular and irregular strip packing problems // Artifical Intelligence Rev. 2001. V. 16. P. 257—300.

УДК 519.23

Б. Н. Поляков, д-р техн. наук, проф.

Эффективный метод ранжирования независимых переменных и отбрасывания несущественных параметров при многофакторном статистическом анализе*

Приводятся обоснования и предлагаются надежные критерии ранжирования независимых переменных и отбрасывания несущественных параметров при многофакторном статистическом анализе, эффективность которых иллюстрируется конкретным примером и подтверждается более чем 30-летней практикой успешного проведения статистических исследований в машиностроении, металлургии и медицине.

Ключевые слова: *многофакторный статистический анализ, ранжирование, отбрасывание несущественных параметров, коэффициент частной корреляции, коэффициент множественной корреляции.*

В работе [1] отмечалось, что для нахождения криволинейного уравнения множественной регрессии методом Брандона независимые переменные необходимо ранжировать, т. е. располагать последние в порядке уменьшения силы их влияния на зависимую переменную. Ранжировать независимые переменные можно на основе линейного регрессионного анализа, как первого этапа многофакторного анализа, несколькими способами: по коэффициенту полной корреляции d_{1i}, по коэффициенту частной корреляции r_{1i} или по стандартизованному коэффициенту регрессии β_i.

Коэффициент полной корреляции d_{1i} характеризует тесноту связи между зависимой переменной x_1 и независимой x_i вне зависимости от того, чем обусловлена эта связь — действительным влиянием x_i либо влиянием других независимых переменных, корреляционно связанных с x_i и, вследствие этого, искажающих силу влияния рассматриваемой независимой переменной на зависимую. Особенно это отмечается в том случае, когда независимые переменные сильно коррелируют между собой.

Стандартизованный коэффициент регрессии β_i является количественной характеристикой силы влияния независимой переменной, выраженной в единицах среднего квадратического отклонения зависимой переменной при устранении другой линейной связи с остальными независимыми переменными. Но в анализе связи двух переменных обычно принято пользоваться не абсолютными оценками, а относительными, т. е. более общими характеристиками.

Коэффициент частной корреляции r_{1i} является относительной характеристикой силы влияния независимой переменной на зависимую при постоянстве других, участвующих в анализе, т. е. выражает влияние, очищенное от действия других независимых факторов. Из определения коэффициента частной корреляции ясно, что последний является наилучшей оценкой силы связи между независимой и зависимыми переменными на основе линейного приближения к эмпирическим данным. И поэтому принято в последовательном многофакторном анализе [1] независимые переменные располагать по мере убывания $|r_{1i}|$.

Рассмотрим теперь вопрос отбрасывания несущественных параметров. Обычно исследователь стремится зафиксировать как можно больше независимых переменных, которые, по его мнению, каким-то образом влияют на изучаемое явление. При этом многие "независимые" факторы могут быть на самом деле тесно взаимосвязаны друг с другом. Большое число независимых переменных иногда искажает физический смысл определяемого уравнения, к тому же коэффициенты регрессии, вычисленные для сильно коррелированных переменных, малонадежны и имеют широкие доверительные интервалы, поэтому часть независимых переменных на основе какого-либо критерия необходимо исключить из рассмотрения. Таким образом, возникает вопрос о критерии отбрасывания несущественных параметров.

О существенности влияния независимой переменной на зависимую переменную можно судить уже по доверительному интервалу коэффициента регрессии. Если доверительный интервал проходит через нуль, то вопрос о существенности влияния соответствующей независимой переменной ставится под сомнение. Поэтому выполнение неравенства (1) будет являться естественным критерием существенности независимого фактора:

$$t_{i\text{I}} = \left| \frac{a_i}{S_{a_i}} \right| > t_\alpha, \qquad (1)$$

где a_i — коэффициент регрессии; S_{a_i} — его среднеквадратическое отклонение; t_α — квантиль нормированного нормального распределения, соответствующий вероятности $(1 - \alpha)$. Данный критерий (неравенство (1)) будем называть первым, на

* Разработан совместно с канд. техн. наук Ю. Д. Макаровым и инж.-математиком Ф. М. Карлинской.

что указывает римская цифра I у индекса в обозначении критерия $t_{\alpha I}$. Но очень часто по многим причинам (неправильно выбраны независимые параметры, недостаточны выборка и точность экспериментальных данных и, вследствие этого, небольшая точность результатов анализов и т. д.) почти для всех коэффициентов регрессии несправедливо неравенство (1) или доверительные интервалы этих коэффициентов проходят через нуль. И одновременное отбрасывание несущественных независимых переменных по критерию **I** может резко уменьшить коэффициент множественной корреляции, увеличить стандартную ошибку оценки и вообще привести к неправильному объяснению изучаемого процесса.

Число же всевозможных вариантов отбрасывания независимых переменных на основе критерия **I** растет по закону $C_k^1 + C_k^2 + ... + C_k^{k-1} + C_k^k$, где k — число независимых переменных, которые нужно было бы отбросить по критерию **I**.

В. Визорке и др. [2] предложили метод одновременного отбрасывания нескольких независимых переменных на основе собственного опыта, полученного при обработке большого числа экспериментальных данных. Суть этого метода в следующем.

Рассмотрим величину $t_i = a_i / S_{a_i}$, эквивалентная формула которой имеет вид

$$t_i = \frac{\beta_i \sqrt{1 - R_i^2}}{\sqrt{1 - R^2}} \sqrt{n - m},$$

где β_i — стандартизованный коэффициент регрессии; R — коэффициент множественной корреляции; R_i — коэффициент множественной корреляции i-й независимой переменной с остальными независимыми переменными; n — число значений зависимой переменной; m —число рассматриваемых параметров, включая зависимый. Если x_i коррелирует только с одной независимой переменной и если последняя является несущественной по критерию **I**, то после ее отбрасывания t_i возрастает в $1/\sqrt{1 - R_i^2}$ раз, что не учитывает рассмотренный выше критерий.

Поэтому необходимо относиться очень осторожно к независимым переменным, которые тесно взаимосвязаны с другими независимыми переменными, и, в первую очередь, рекомендуется отбросить те независимые переменные, несущественные по критерию **I**, которые слабо коррелируют с другими, так как данный критерий не учитывает взаимной корреляции между независимыми переменными. Но независимая переменная может коррелировать с несколькими независимыми переменными и t_i может возрасти [2] приблизительно в $1 + R_i\sqrt{2} / \sqrt{1 - R_i^2}$ раз после отбрасывания несущественных параметров и тогда

$$t_{i\text{II}} = |t_{i\text{I}}| \frac{1 + R_i\sqrt{2}}{\sqrt{1 - R_i^2}} > t_\alpha. \qquad (2)$$

Экспериментальная проверка показала, что критерий **II** отбрасывания нескольких независимых параметров очень слаб, что будет проиллюстрировано в приведенном ниже примере. Для практического применения этого критерия можно его усилить. В алгоритме многофакторного статистического анализа [1] заложен следующий критерий:

$$t_{i\text{III}} = |t_{i\text{I}}| \frac{1}{\sqrt{1 - R_i^2}} > t_\alpha. \qquad (3)$$

Критерий отбрасывания **II** отличается от критерия **III** сомножителем $1 + R_i\sqrt{2}$ в числителе правой части выражения (2).

На рис. 1 и 2 показаны зависимости $1 + R_i\sqrt{2} / \sqrt{1 - R_i^2}$ и $1/\sqrt{1 - R_i^2}$ от R_i. Из рассмотрения этих графиков и неравенств (2) и (3) можно сделать вывод о том, что в первую очередь будут отброшены те независимые переменные, для которых мало значение R_i, т. е. слабо связанные с другими независимыми переменными.

После работы критерия **II** могут остаться еще несколько независимых переменных, существенных по критерию **III**, но не существенных по критерию **I**. В этом случае предлагается на втором

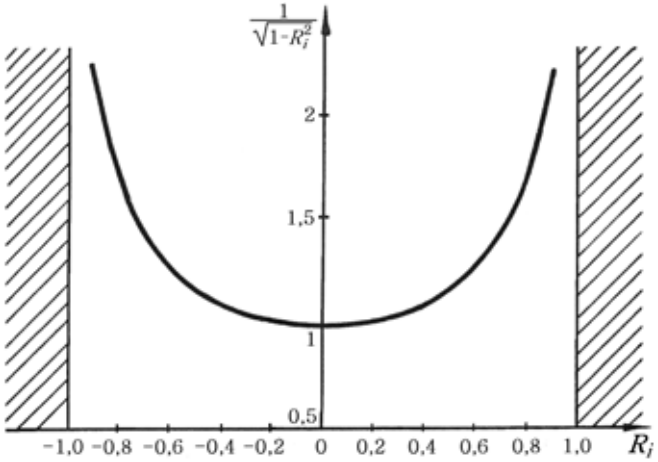


Рис. 1. Зависимость $\frac{1}{\sqrt{1 - R_i^2}}$ от R_i

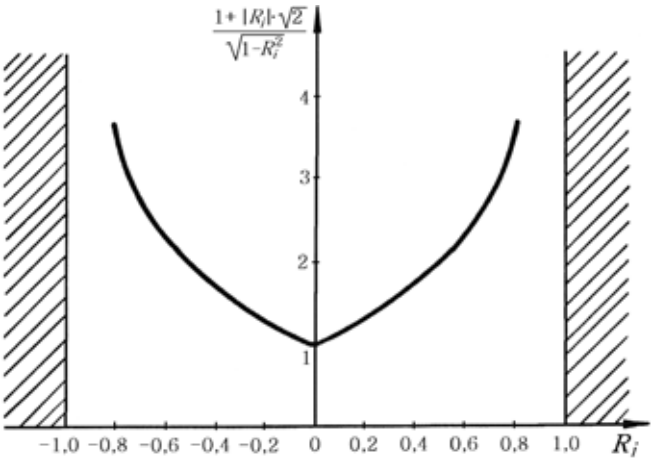


Рис. 2. Зависимость $\frac{1+|R_i|\sqrt{2}}{\sqrt{1-R_i^2}}$ от R_i

этапе исследования несущественные переменные отбрасывать по следующему критерию:

$$t_{i\text{IV}} = |t_{i\text{I}}|\frac{1+\left|r_{1i}^2\right|\sqrt{2}}{\sqrt{1+r_{1i}^2}} > t_\alpha. \qquad (4)$$

Сопоставляя критерии **II** и **IV** отбрасывания независимых переменных, можно заметить, что они имеют одинаковую структуру. Поэтому график функции $1 - R_i\sqrt{2}/\sqrt{1-R_i^2}$ полностью совпадает с графиком функции $1 + |r_{1i}|\sqrt{2}/\sqrt{1-r_{1i}^2}$. Из неравенства (4) следует, что в первую очередь будут отбрасываться те несущественные независимые переменные, которые слабо влияют на зависимую переменную. Практическое применение критерия **IV** дало положительный результат. Критерий **IV** сильнее критерия **III**, поэтому он применяется уже на втором этапе отбрасывания независимых переменных.

Действия приведенных выше критериев проиллюстрированы примером. Опытные данные для примера взяты из источника [3], в котором обобщены технологические режимы прокатки на блюмингах ряда отечественных металлургических заводов и дополнены данными по схемам и режимам обжатий, применяющимися на блюминге 1300 меткомбината "Криворожсталь".

Пример. На основе опытных данных определяется регрессионная зависимость числа пропусков N от среднего обжатия в пропуске за цикл прокатки $\Delta h_{\text{ср}}$, площади поперечного сечения слитка $F_{\text{сл}}$, площади поперечного сечения конечной заготовки $F_{\text{пр}}$, числа кантовок K, массы слитка $Q_{\text{сл}}$, числа пропусков до первой кантовки $n_{\text{к1}}$, числа пропусков между первой и второй кантовкой $n_{\text{к2}}$, числа пропусков между второй и третьей кантовкой $n_{\text{к3}}$, числа пропусков между третьей и четвертой кантовкой $n_{\text{к4}}$:

$$N = 13{,}81 - 0{,}094\Delta h_{\text{ср}} - 0{,}3\cdot 10^{-4}F_{\text{пр}} + \\ + 0{,}88\cdot 10^{-5}F_{\text{сл}} + 0{,}216n_{\text{к3}} + 0{,}291n_{\text{к4}} + \\ + 0{,}235Q_{\text{сл}} + 0{,}278K + 0{,}082n_{\text{к1}} + 0{,}086n_{\text{к2}}.$$

Первое приближение приведено в табл. 1.

Доверительные интервалы коэффициентов регрессии следующих параметров проходят через нуль: K, $Q_{\text{сл}}$, $n_{\text{к1}}$, $n_{\text{к2}}$, $n_{\text{к3}}$, $n_{\text{к4}}$. По критерию **II** отбрасывается только $n_{\text{к2}}$, по критерию **III** должно отбрасываться $n_{\text{к1}}$ и $n_{\text{к2}}$, а по критерию **IV** — $n_{\text{к1}}$, $n_{\text{к2}}$, K. Так как в программе многофакторного анализа сначала работает критерий **III**, то второе приближение после отбрасывания $n_{\text{к1}}$ и $n_{\text{к2}}$ будет следующим (см. также табл. 2):

$$N = 14{,}66 - 0{,}094\Delta h_{\text{ср}} - 0{,}31\cdot 10^{-4}F_{\text{пр}} + \\ + 0{,}94\cdot 10^{-5}F_{\text{сл}} + 0{,}202n_{\text{к3}} + 0{,}249n_{\text{к4}} + \\ + 0{,}206Q_{\text{сл}} + 0{,}231K.$$

Таблица 1

Первое приближение

Параметр	a_i	$a_i^{(1)}$	$a_i^{(2)}$	r_{1i}	$t_{i\text{I}}$	$t_{i\text{II}}$	$t_{i\text{III}}$	$t_{i\text{IV}}$	$\sqrt{1-R_i^2}$
N	13,81	13,99	13,63	—	7,522	—	—	—	—
$\Delta h_{\text{ср}}$, мм	−0,094	−0,077	−0,110	−0,857	−10,95	19,08	12,033	−47,04	0,910
K	0,278	0,770	−0,214	0,166	1,108	6,547	2,843	1,388	0,389
$Q_{\text{сл}}$, т	0,235	0,517	−0,047	0,241	1,630	11,47	4,916	2,253	0,332
$F_{\text{сл}}$, мм^2	$0{,}88\cdot 10^{-5}$	$0{,}13\cdot 10^{-4}$	$0{,}45\cdot 10^{-5}$	0,521	3,999	25,36	10,946	8,134	0,365
$F_{\text{пр}}$, мм^2	$-0{,}3\cdot 10^{-4}$	$-0{,}19\cdot 10^{-4}$	$-0{,}40\cdot 10^{-4}$	−0,640	−5,462	17,61	8,463	−13,54	0,645
$n_{\text{к1}}$	0,082	0,231	−0,068	0,161	1,068	2,825	1,462	1,329	0,731
$n_{\text{к2}}$	0,086	0,325	−0,154	0,106	0,700	1,735	0,929	0,810	0,753
$n_{\text{к3}}$	0,216	0,434	−0,001	0,285	1,949	6,534	3,110	2,836	0,627
$n_{\text{к4}}$	0,291	0,604	−0,023	0,267	1,818	8,873	3,936	2,600	0,462

Остаточное среднеквадратичное отклонение $S = 0{,}604$. Коэффициент множественной регрессии $R = 0{,}944$.

Таблица 2

Второе приближение

Параметр	a_i	$a_i^{(1)}$	$a_i^{(2)}$	r_{1i}	$t_{i\text{I}}$	$t_{i\text{II}}$	$t_{i\text{III}}$	$t_{i\text{IV}}$	$\sqrt{1-R_i^2}$
N	14,66	14,84	14,48	—	7,985	—	—	—	—
$\Delta h_{\text{ср}}$, мм	−0,094	−0,077	−0,110	−0,855	−11,05	19,19	12,13	47,03	0,911
K	0,231	0,704	−0,243	0,141	0,954	5,454	2,376	1,155	0,402
$Q_{\text{сл}}$, т	0,206	0,478	−0,065	0,217	1,493	10,14	4,355	1,999	0,343
$F_{\text{сл}}$, мм^2	$0{,}94 \cdot 10^{-5}$	$0{,}135 \cdot 10^{-4}$	$0{,}53 \cdot 10^{-5}$	0,556	4,498	27,30	11,83	9,665	0,380
$F_{\text{пр}}$, мм^2	$-0{,}31 \cdot 10^{-4}$	$-0{,}21 \cdot 10^{-4}$	$-0{,}41 \cdot 10^{-4}$	−0,667	−5,981	17,84	8,773	15,59	0,682
$n_{\text{к3}}$	0,202	0,386	0,016	0,303	2,132	5,618	2,913	3,195	0,732
$n_{\text{к4}}$	0,249	0,551	−0,053	0,234	1,617	7,622	3,397	2,215	0,476

$S = 0{,}615$. $R = 0{,}942$.

Таблица 3

Третье приближение

Параметр	a_i	$a_i^{(1)}$	$a_i^{(2)}$	r_{1i}	$t_{i\text{I}}$
N	15,38	15,56	15,20	—	8,377
$\Delta h_{\text{ср}}$, мм	−0,095	−0,079	−0,112	−0,861	−11,64
$Q_{\text{сл}}$, т	0,229	0,496	−0,037	0,241	1,684
$F_{\text{сл}}$, мм^2	$0{,}93 \cdot 10^{-5}$	$0{,}134 \cdot 10^{-4}$	$0{,}52 \cdot 10^{-5}$	0,547	4,429
$F_{\text{пр}}$, мм^2	$-0{,}32 \cdot 10^{-4}$	$-0{,}23 \cdot 10^{-4}$	$-0{,}42 \cdot 10^{-4}$	−0,698	−6,969
$n_{\text{к3}}$	0,249	0,407	0,091	0,414	3,089
$n_{\text{к4}}$	0,367	0,547	0,188	0,509	3,998

$S = 0{,}621$. $R = 0{,}941$.

При этом коэффициент множественной регрессии R уменьшается с 0,944 до 0,942, а остаточное среднеквадратическое отклонение S увеличивается с 0,604 до 0,615.

Во втором приближении доверительные интервалы коэффициентов регрессии параметров K, $Q_{\text{сл}}$, $n_{\text{к4}}$ проходят через нуль, но критерии **II** и **III** дают отрицательный ответ на отбрасывание этих независимых параметров. По критерию **IV** можно отбросить независимый параметр K. Тогда в третьем приближении уравнение регрессии запишется следующим образом (см. также табл. 3):

$$N = 15{,}38 - 0{,}095\Delta h_{\text{ср}} - 0{,}32 \cdot 10^{-4} F_{\text{пр}} +$$
$$+ 0{,}93 \cdot 10^{-5} F_{\text{сл}} + 0{,}367 n_{\text{к4}} + 0{,}249 n_{\text{к3}} + 0{,}229 Q_{\text{сл}}.$$

Хотя после третьего приближения доверительный интервал у независимого параметра $Q_{\text{сл}}$ проходит через нуль, но ни один из выше рассмотренных критериев (**II**, **III**, **IV**) не отбрасывает его, так как $t_{i\text{I}}$, равный 1,684, достаточно близок к $t_\alpha = 1{,}96$ и влияние на зависимую переменную N значительно ($r_{14} = 0{,}241$).

После третьего приближения коэффициент множественной корреляции равен 0,941, а остаточное среднеквадратическое отклонение равно 0,621.

Отбрасывание независимых параметров $n_{\text{к1}}$, $n_{\text{к2}}$, K согласуется с физическим процессом прокатки на блюминге, потому что среднее обжатие до первого ящичного калибра (где прокатка идет со стесненным уширением) определяется только размерами слитка и его положением при начальном обжатии, а в дальнейшем — размерами поперечного сечения конечного раската, последовательностью расположения и шириной калибров, которые в какой-то степени характеризуются параметрами $n_{\text{к3}}$ и $n_{\text{к4}}$. Отсюда очевидно, что и число кантовок K мало влияет на среднее обжатие за цикл прокатки.

Таким образом, предлагаемые критерии ранжирования независимых переменных и отбрасывания несущественных параметров, проверенные на многочисленных примерах из более чем 30-летней практики успешного проведения статистических исследований в машиностроении, металлургии и медицине [4, 5], подтверждают их надежность и эффективность для проведения разнообразных многофакторных статистических исследований.

Список литературы

1. **Статистические** методы в алгоритмах и примерах (из практики прокатного производства): Учеб. пособ. СПб.: Реноме, декабрь 2007. 182 с.
2. **Von H. Knüppel, Stumpf A., Wiezorke B.** Mathematische Statistik in Eisenhüttenwerken // Archive fürdas Eisenhüttenwesen. 1958. № 8. Перевод № 1492. НИИТЯЖМАШ Уралмашзавода, 1968.
3. **Логоватовский А. А.** Нормирование процессов на блюминге. М.: Металлургия, 1966. 220 с.
4. **Статистический** анализ и математическое моделирование блюминга / С. Л. Коцарь, Б. Н. Поляков, Ю. Д. Макаров, В. А. Чичигин. М.: Металлургия, 1974. 280 с.
5. **Поляков Б. Н.** Повышение качества технологий, несущей способности конструкций, долговечности оборудования и эффективности автоматических систем прокатных станов. СПб.: Реноме, 2006. 528 с.

БАЗЫ ДАННЫХ

УДК 004.4.068

В. Г. Мокрозуб, канд. техн. наук, доц.,
Тамбовский государственный
технический университет,
e-mail: makr@mail.gaps.tstu.ru

Таксономия в базе данных стандартных элементов технических объектов

Предложена структура базы данных стандартных и типовых элементов технических объектов, отличающаяся тем, что содержит таксономию в виде иерархической структуры терминов предметной области и правил на SQL, параметрические 3D-модели элементов и их базовые геометрические компоненты, предназначенные для позиционирования элемента в 3D-модели сборки технического объекта.

Ключевые слова: *база данных, стандартные элементы, таксономия.*

Введение

Технические объекты (ТО) в значительной степени состоят из стандартных и типовых элементов (СТЭ). Под элементами здесь понимаются конкретные физические объекты (болты, фланцы, редукторы), имеющие уникальное обозначение, например: "Болт М10-6gх20.46.019 ГОСТ 7796—70", "Мотор-редуктор МПО2-10Ф (U = = 23.1,3/63,45100S)".

Стандартные элементы определены нормативными документами, имеющими государственное или отраслевое действие (ГОСТ, ОСТ), типовые элементы определены в стандартах предприятий (СТП), в каталогах продукции предприятий и других аналогичных документах.

В настоящее время существует множество баз данных СТЭ (БДСТЭ), предназначенных для различных целей (проектирование ТО, составление заявок на приобретение комплектующих, ведение складского учета и др.). Известные фирмы АСКОН [1], АРМ WinMashine [2] и др. имеют собственные БДСТЭ. Некоторые предприятия разрабатывают свои БДСТЭ, учитывающие особенности этих предприятий и задачи, для которых эти базы предназначены (проектирование, снабжение, маркетинг и др.) [3]. Требования к информации, хранящейся в БДСТЭ, и к обмену этой информацией определены в стандартах ISO 13584 и 10303.

БДСТЭ при проектировании ТО применяются при разработке структуры ТО, расчете размеров элементов ТО, построении рабочих чертежей. В большинстве случаев это реляционные базы данных, в которых хранится информация о типоразмерах СТЭ.

В современных условиях одним из направлений развития информационных систем (ИС), в том числе и систем автоматизированного проектирования (САПР), является их интеллектуализация. В настоящее время активно развивается интеллектуализация ИС на основе инженерии знаний [4].

Целью настоящей работы является разработка структуры реляционной БДСТЭ, которая должна позволить:

а) подобрать необходимый тип СТЭ ТО для заданных условий эксплуатации (на основе таксономии предметной области (ПО));

б) выбрать конкретный физический СТЭ;

в) автоматически построить 3D-модель и 2D-чертеж выбранного СТЭ;

г) автоматически вставить (позиционировать) выбранный элемент в 3D-сборку ТО.

Следует подчеркнуть, речь не идет о том, что БДСТЭ должна уметь выполнять функции в) и г), она должна предоставить информацию для выполнения этих функций. Выполнять пункты в) и г) должен графический редактор.

Структура традиционной БДСТЭ

Традиционная БДСТЭ, используемая для создания ТО, содержит текстовую и графическую информацию. Текстовая информация представлена таблицами размеров и других свойств элементов (масса, материал изготовления) и классификатором типов элементов (как правило, в виде дерева "классы — подклассы"). Графическая информация представлена параметрическими 3D- и 2D-моделями или программами построения геометрических образов элементов в среде определенного графического редактора. Кроме того, БДСТЭ содержит интерфейс пользователя, который позволяет осуществить вручную выбор элемента и передачу параметров этого элемента в модули построения графического изображения элемента. После этого пользователь в среде выбранного графического редактора вручную вставляет построенный элемент в 3D-модель сборки ТО.

Описанная технология используется в графических редакторах Inventor, Компас, Solid Works, T-Flex.

Таким образом, традиционная БДСТЭ может быть представлена тройкой $B = <R, K, M>$, где $R = \{r_i\}$, $i = \overline{1, I}$, — множество типоразмеров (например, таблица на фланцы по ГОСТ 18821, таблица на опоры-лапы по ГОСТ 26296); K — классификатор типов элементов в виде дерева "класс—подкласс" (например, опоры: опоры вертикальных аппаратов — опоры лапы); $M = \{m_i\}$, $i = \overline{1, I}$, — множество 3D-параметрических моделей. Каждой таблице r_i соответствует своя модель m_i.

Для создания множества $R = \{r_i\}$, $i = \overline{1, I}$, в реляционной базе данных могут быть использованы следующие способы:

- для каждого r_i создать собственную таблицу с размерами и другими характеристиками элементов (масса, материал);
- создать широкую таблицу (например, таблицу с числом полей 100— 200), в которой будут размеры всех типовых элементов (одна строка соответствует одному физическому типовому элементу, один столбец соответствует одному размеру);
- хранить информацию о размерах в виде текстовой строки вида "наименование_размера = = значение_размера", например, "D1 = 1000, D2 = 500, S = 8".

Каждый из способов имеет свои преимущества и недостатки, рассмотрение которых не является предметом данной статьи.

Структурная схема традиционной БДСТЭ, когда для каждого r_i создана собственная таблица, и параметрическая графическая модель представлены на рис. 1, схема данных — на рис. 2.

Классификатор представлен таблицами "Дерево_классов" и "Типоразмеры". Таблица "Дерево_классов" отражает уровни 2 − n. Поле "Ключ_сортировки_дерева" позволяет выводить данные из таблицы "Дерево_классов" на экран клиента в нужной иерархической последовательности.

Записи таблицы "Типоразмеры" являются вершинами уровня 1. Имя таблицы с размерами элементов хранится в поле "Имя_таблицы". Поле "Запрос_на_представление_типоразмера" содержит SQL-запрос для получения всех записей из таблицы типоразмеров для их последующего вывода на экран монитора клиента. Например, "**select** Обозначение, Dy **as** Диаметр_фланца ... **from** Фланцы_ГОСТ 18821". Наличие этого поля позволяет значительно упростить клиентское приложение и не модифицировать его при появлении новой таблицы типоразмеров.

Структура БДСТЭ с таксономией предметной области

Для того чтобы подобрать необходимый тип СТЭ ТО для заданных условий эксплуатации, БДСТЭ должна иметь соответствующую информацию. Условия эксплуатации задаются, как правило, выражениями вида "приварные встык фланцы аппаратов следует применять при давлении больше, чем 2,5 МПа и температуре меньше, чем 300 °С". В ИС это выражение трансформируется в продукционное правило "Если давление больше 2,5 МПа и температура меньше 300 °С, то тип фланца аппарата — приварной встык".

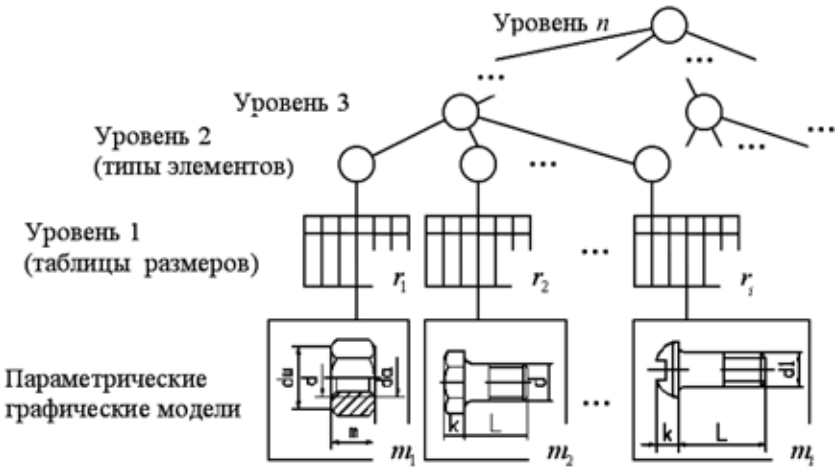


Рис. 1. Структурная схема традиционной БДСТЭ

Рис. 2. Схема данных традиционной БДСТЭ

Могут быть и более сложные правила, включающие зависимость применения элемента не только от условий эксплуатации, но и от других элементов ТО. Под *таксономией* предметной области (ПО) в данной работе понимается система терминов и связей между ними, которая позволяет использовать БДСТЭ для автоматического подбора СТЭ по заданным условиям эксплуатации.

Современные графические редакторы имеют развитые средства создания 3D-моделей сборок ТО. Технология создания таких моделей заключается в указании определенных сопряжений (соосность, параллельность плоскостей и др.) между элементами ТО. Несмотря на множество сервисов, представляемых графическими редакторами при создании 3D-моделей сборок ТО, процесс этот трудоемкий и требует повышенного внимания проектировщика (конструктора). Кроме того, хотя идеология интерфейсов, предлагаемых различными разработчиками графических редакторов, похожа, при практической работе требуется определенное время, чтобы адаптироваться при переходе от одного графического редактора к другому. В этом плане представляется интересным использование математической модели позиционирования элементов ТО в пространстве (далее — просто модель позиционирования) [5], по которой графический редактор автоматически строит 3D-модель ТО. Эта модель инвариантна к среде ее обработки.

На рис. 3 представлена 3D-модель сборки ТО, состоящего из двух втулок, болта и гайки. У каждой детали имеются базовые графические компоненты (БГК) — грани и оси. Оси на рис. 3 обозначены символом O_1, грани — символами S_1 и S_2. Положение элемента в 3D-модели сборки ТО задается соотношениями типа "ось втулки O_1 совпадает с осью болта O_1", "грань первой втулки S_1 совпадает с гранью болта S_1".

Если использовать принятую для объектно-ориентированного программирования нотацию *объект—свойство*, то для рассматриваемого примера модель позиционирования можно записать в следующем виде:

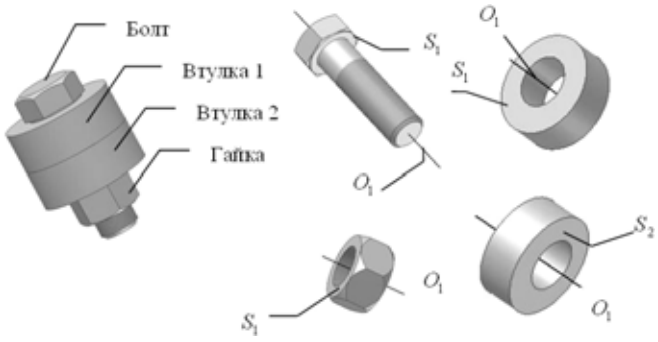


Рис. 3. Пример базовых графических компонентов

$$Втулка1.O_1 \odot Болт.O_1,\ Втулка1.S_1 \in Болт.S_1,$$
$$Втулка2.O_1 \odot Болт.O_1,\ Втулка2.S_1 \in Втулка1.S_2,$$
$$Гайка.O_1 \odot Болт.O_1,\ Гайка.S_1 \in Втулка2.S_2,$$

где $\odot$ — соосность, $\in$ — совпадение граней.

Таким образом, БДСТЭ, с помощью которой можно выполнить все обозначенные во введении цели, может быть представлена, как $BO = <T, R, M, P>$, где T — таксономия ПО, $R = \{r_i\}$, $i = \overline{1, I}$, — мно-

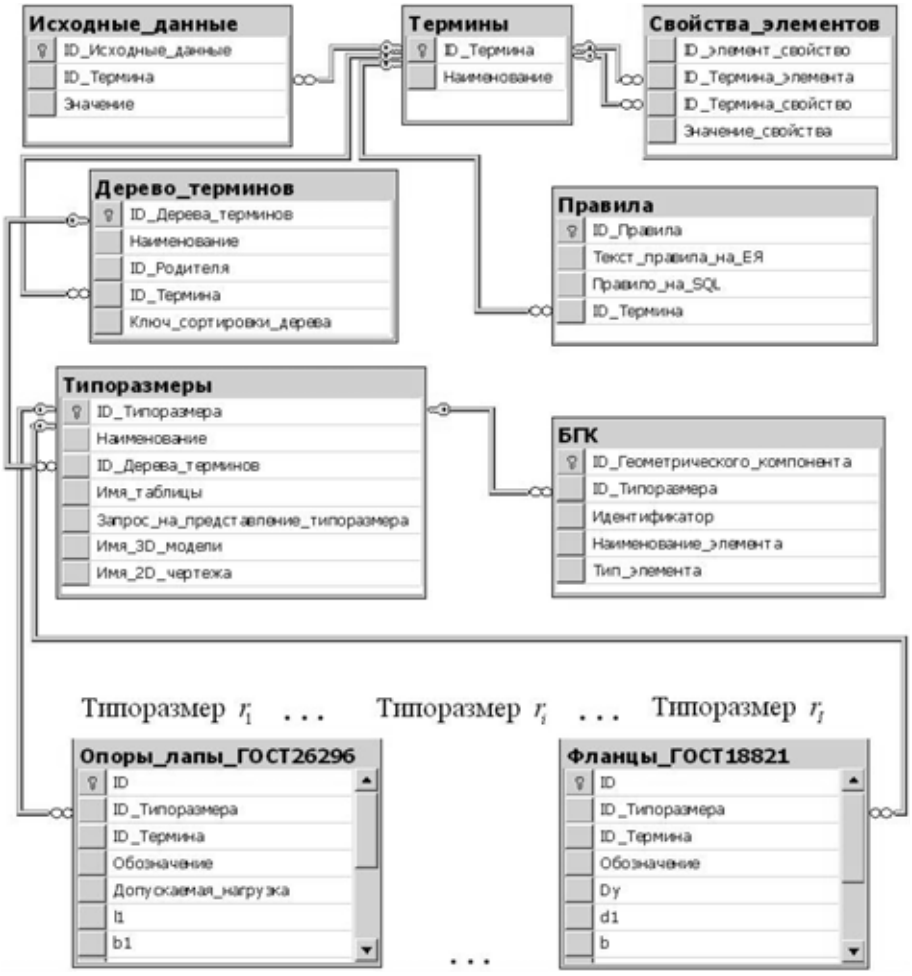


Рис. 4. Схема данных БДСТЭ с таксономией предметной области

Пример записей таблицы "БГК"

ID_Геометрического_компонента	ID_Типоразмера	Идентификатор	Наименование	Тип_элемента
100	211	S_1	Нижняя грань головки болта	Грань
101	211	O_1	Ось болта	Ось

жество типоразмеров элементов; $M = \{m_i\}$, $i = \overline{1, I}$, — множество 3D-параметрических моделей; $P = \{p_{iz}\}$, $i = \overline{1, I}$, $z = \overline{1, Z_i}$, — множество БГК; Z_i —число БГК для i-го типоразмера.

Таксономия ПО $T = <E, D, Pr>$, где $E = \{e_x\}$, $x = \overline{1, X}$, — множество терминов ПО, включая обозначения СТЭ; D — множество связей между терминами типа "класс—подкласс"; $Pr = \{pr_l\}$, $l = \overline{1, L}$, — множество продукционных правил, составленных из терминов ПО.

Схема данных БДСТЭ с таксономией предметной области представлена на рис. 4.

Таблицы "Термины", "Дерево терминов" и "Правила" представляют таксономию ПО. Поле ID_Термина в таблице "Правила" позволяет в дальнейшем сократить список используемых правил при выполнении конкретной задачи поиска элементов для заданных условий.

Таблица "Свойства элементов" содержит информацию о СТЭ, которая позволяет выбрать элемент заданного типа по его определяющим характеристикам, например, опору по допускаемой нагрузке.

Поля таблицы "БГК" для болта на рис. 3 принимают значения, приведенные в таблице.

Таблица "Исходные_данные" содержит условия эксплуатации, для которых необходимо подобрать СТЭ. В представленной структуре эти условия объединены логическим "И".

Таблица "Правила" содержит продукционные правила на SQL, позволяющие подобрать тип СТЭ в зависимости от условий эксплуатации.

Рассмотрим текст правила на SQL на следующем примере. Имеется правило "***Если P > 2,5 МПа и T < 300 °C, то фланец аппарата приварной встык***". Предположим, что для термина "Давление" ID_Термина = 7, для термина "Температура" ID_Термина = 6, для термина "Фланец аппарата приварной встык" ID_Термина = 2. Тогда текст правила в виде SQL предложения будет следующим:

select *TE.ID_Термина* ***from*** *Термины TE* ***where*** *TE.ID_Термина = 2* ***and exists*** *(select * **from** Исходные_данные Dan **where** Dan.ID_Термина = 7 **and** Dan.Значение > 2.5)* ***and exists*** *(**select** * **from** Исходные_данные Dan **where** Dan.ID_Термина = 6 **and** Dan.Значение < 300)*.

Схема данных на рис. 4 служит для иллюстрации предлагаемого метода. В реальной базе данных таблицы, представленные на рис. 4, имеют дополнительные поля, такие как время ввода записи, источник правила, кто ввел запись и др.

Обработка таксономии

Предположим, что надо выбрать тип фланца для аппарата при известных температуре и давлении. Таблица "Исходные_данные" будет содержать записи для значений температуры и давления. Фрагмент дерева терминов представлен на рис. 5. Узел "Фланцы аппаратов" ID_Дерева_ Терминов = 8 назовем стартовым узлом.

Ниже приведен пример одной из возможных процедурных моделей обработки таксономии. Эта модель может быть и другой, например, в зависимости от того, хотим ли мы выполнить правила только для стартового узла дерева терминов или для всех узлов, являющихся потомками (и не только прямыми) стартового узла. В примере выполняются правила для всех потомков стартового узла. Фрагменты программ выполнены в среде MS-SQL 2005.

Шаг 1. По заданному ID_Дерева_Терминов = = 8 выбрать стартовый узел "Фланец аппаратов" и все узлы дерева терминов, находящиеся ниже этого узла (рис. 5), результат — в таблице #*Дерево*, .

```
with Tree (ID_Дерева_терминов, ID_Родителя, ID_Термина)
   as
      (select ID_Дерева_терминов, ID_Родителя,
      ID_Термина
         from Дерево_терминов where
           ID_Дерева_терминов = 8
      union all
      select a.ID_Дерева_терминов, a.ID_Родителя,
      a.ID_Термина
```

Рис. 5. Фрагмент дерева терминов

```
        from Дерево_терминов a
        inner join Tree b on
        a.ID_Родителя = b.ID_Дерева_терминов)
    select * into #Дерево from Tree
```

Здесь используется рекурсивный запрос. Для систем управления базами данных, которые не позволяют составлять рекурсивные запросы, можно рекомендовать алгоритмы, описанные в [6].

Шаг 2. Выбрать правила для всех терминов, выбранных на шаге 1, результат — *Cursor1.*

```
declare @str varchar(max)
create table #Результат (ID_Термина int)
fetch next from Cursor1 into @str
while @@fetch_status = 0
    begin
        set @str = 'Insert #Результат (ID_Термина) ('+ @str+')'
        execute (@str)
        fetch next from Cursor1 into @str
end
close Cursor1; deallocate Cursor1
```

Шаг 3. Выполнить выбранные на шаге 2 правила, результат — таблица *#Результат.*

```
declare @str varchar(max); create table #Результат
(ID_Термина int)
fetch next from Cursor1 into @str
while @@fetch_status = 0
    begin
        set @str = 'Insert #Результат (ID_Термина) ('+ @str+')'
        execute (@str)
        fetch next from Cursor1 into @str
end
close Cursor1; deallocate Cursor1
```

Шаг 4. Из дерева терминов выбрать все узлы, находящиеся ниже узлов, полученных на шаге 3. Результат — таблица *#Дерево_Результата.*

```
with Treel (ID_Дерева_терминов, ID_Родителя, ID_Термина)
    as
      (select a.ID_Дерева_терминов, a.ID_Родителя,
      a.ID_Термина
          from Дерево_терминов a, #Результат b
          where a.ID_Термина = b.ID_Термина
      union all
        select c.ID_Дерева_терминов, c.ID_Родителя,
        c.ID_Термина
          from Дерево_терминов c
          inner join Treel d on
          c.ID_Родителя = d.ID_Дерева_терминов)
    select * into #Дерево_Результата from Tree1
```

Шаг 5. Из полученной на шаге 4 таблицы *#Дерево_Результата* выбрать:

а) узлы, являющиеся листьями. В этом случае мы получим множество конкретных физических элементов (т. е. фланцы всех размеров найденного типа). Выбор нужного фланца осуществляется по его определяющему размеру (здесь не представлено)

```
select a. * from #Дерево_Результата a
        where not exists (select * from #Дерево_Результата b
                          where b.ID_Родителя = a.ID_Дерева_терминов);
```

б) или узлы, являющиеся дочерними стартового узла. В этом случае мы получим только типы подходящих фланцев

```
select * from #Дерево_Результата where ID_Родителя = 8
```

Следует отметить, что предложенная структура БДСТЭ и алгоритм обработки правил не являются экспертной системой. Кроме того, при практической реализации необходимо разрабатывать программные элементы, позволяющие вводить правила в форме "Если ..., то" на языке, близком к естественному, и трансформировать правила в предложения SQL. Эти программные элементы (серверная и клиентская части) могут быть отдельной темой и не являются предметом настоящей статьи.

Заключение

Достоинством предложенного подхода является то, что он не требует дополнительного программного обеспечения в виде оболочки экспертной системы. Современные предприятия уже имеют необходимое программное обеспечение и программистов нужной квалификации для его внедрения.

Описанная структура БДСТЭ применена автором для дальнейшего развития системы автоматизированного расчета и конструирования химического оборудования [7], которая используется студентами Тамбовского государственного технического университета специальности 240801 "Машины и аппараты химических производств" при курсовом и дипломном проектировании. Удаленный вариант представлен по адресам www.gaps.tstu.ru и www. 170514.tstu.ru. Кроме того, элементы этой системы используются в ЗАО "завод Тамбовполимермаш", ОАО "Пигмент" и ОАО "Тамбовский завод Комсомолец им. Н. С. Артемова".

Список литературы

1. **Начитов Ю., Боровицкая В**. Построение единых справочников стандартных изделий. САПР и графика. 2006. № 3 [Электронный ресурс] — Режим доступа: http://www.sapr.ru/article.aspx?id = 16075&iid = 734.

2. **Базы** данных информационные массивы для хранения стандартных изделий и справочных данных. [Электронный ресурс] — Режим доступа: http://www.apm.ru/rus/machinebuilding/components/apmdata.

3. **Мокрозуб В. Г.** База мотор-редукторов Тамбовского завода полимерного машиностроения // Инновации в науке и образовании (Телеграф отраслевого фонда алгоритмов и программ). — 2008. № 9 (44). С. 68. [Электронный ресурс] Режим доступа: http://ofap.ru/portal/newspaper/2008/9_44.doc.

4. **Гаврилова Т. А., Муромцев Д. И.** Интеллектуальные технологии в менеджменте: инструменты и системы: Учеб. пособие. 2-е изд. СПб.: Изд-во "Высшая школа менеджмента"; Издат. дом С.-Петерб. гос. ун-та, 2008. 488 с.

5. **Мокрозуб В. Г., Мариковская М. П., Красильников В. Е.** Методологические основы построения автоматизированной информационной системы проектирования технологического оборудования // Системы управления и информационные технологии. 2007. № 1, 2 (27). С. 259—262.

6. **Мещеряков С. В., Иванов, В. М.** Эффективные технологии создания информационных систем. СПб.: Политехника, 2005. 309 с.

7. **Малыгин Е. Н., Карпушкин С. В., Мокрозуб В. Г., Краснянский М. Н.** Система автоматизированного расчета и конструирования химического оборудования // Информационные технологии. 2000. № 12. С. 19—21.

УДК 004.414.28

В. П. Сурпин, аспирант, науч. сотр.
Института проблем передачи информации
им. А. А. Харкевича РАН
e-mail: vadim@iitp.ru

Разработка подсистемы ведения классификаторов корпоративной информационной системы

Практически в каждой корпоративной информационной системе (КИС) присутствует необходимость работы со справочниками и их ведения. В данной работе проводится классификация справочников по ряду параметров, выделяются основные задачи при работе со справочниками и предлагается способ построения подсистемы ведения справочной информации в КИС.

Ключевые слова: *корпоративные информационные системы (КИС), справочники, объектно-ориентированный анализ и проектирование.*

При создании большинства корпоративных информационных систем важное место занимает задача ведения справочников системы. Справочник (или иначе классификатор) — это набор элементов, представляющих множество значений атрибута того или иного класса из предметной области системы. Ведение справочника представляет собой процесс задания и изменения данного множества в соответствии с правилами работы информационной системы. В информационных системах, работающих в разных областях человеческой деятельности, основные принципы ведения справочников схожи. Поэтому актуальными являются задачи выявления основных видов справочников, их свойств и операций при работе с ними, а также создания унифицированной программной модели для ведения справочников.

Классификация справочников

Для того чтобы спроектировать гибкую систему ведения справочников, необходимо определить, по каким параметрам отличаются справочники друг от друга, какие основные сценарии использования справочников существуют, какие структуры справочников наиболее часто применяются на практике. На рис. 1 приведена классификация справочников по ряду характеристик, определяющих назначение справочника и влияющих на способы обработки справочных данных программным обеспечением модуля ведения справочников. Раскроем содержание каждого пункта классификации более подробно.

Рис. 1. Классификация справочников

По связи между элементами. Структура справочника определяется связями между его элементами. Частным видом связей является их отсутствие, в таком случае мы получаем плоский справочник. Другим наиболее часто применяемым видом является древовидная структура, в этом случае применяется связь вида "родитель—потомок". Эти два вида справочников покрывают большую часть всех возможных применений, так как они наиболее просты в реализации и использовании. Более общим представлением связей между элементами является направленный граф, однако на практике выделяются несколько основных видов сложных связей, определяемых бизнес-логикой корпоративного приложения. Рассмотрим их подробнее.

1. При объединении двух или более иерархий получаем сетевую структуру, которая является более сложной. Такие связи часто возникают при необходимости отразить одновременно организационную и территориальную структуру предприятия.

2. Свойства элементов одного справочника задаются другим справочником. Примером может служить справочник территориального деления страны на федеральные округа, области, районы, города и т. д., когда элементами справочника являются названия соответствующих территориальных единиц, а дополнительным свойством — вид территориальной единицы: округ, область, район, город. Вид представляет собой предопределенное множество значений и задается отдельным справочником.

3. Между элементами двух справочников существует связь "многие ко многим". В качестве примера можно привести справочники медицинских диагнозов и лабораторных анализов, которые не-

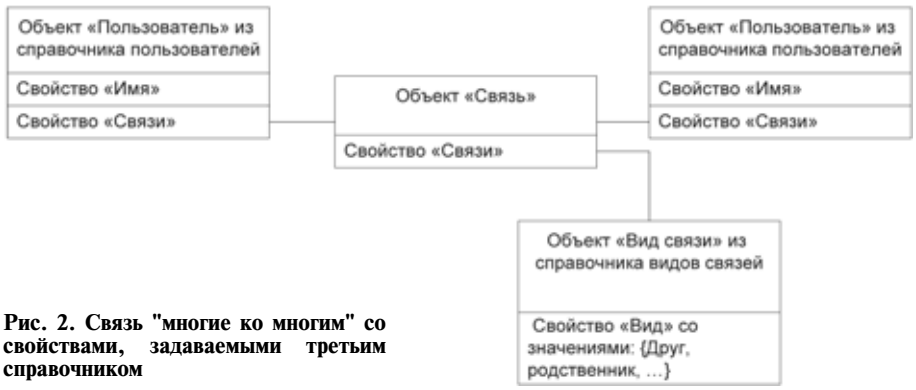


Рис. 2. Связь "многие ко многим" со свойствами, задаваемыми третьим справочником

обходимо провести, чтобы подтвердить диагноз. В этом случае мы имеем два древовидных справочника, между элементами которых задается соответствие вида "многие ко многим". Отличием этого случая от первого, где рассматривается участие элемента в двух или более деревьях, является то, что элементы каждого из двух справочников могут использоваться независимо друг от друга. В свою очередь, ассоциация между элементами справочников также может иметь свойства, заданные еще одним справочником. Сказанное пояснено рис. 2.

Именно вид связи между элементами справочника играет ведущую роль в выборе визуальных компонентов для представления справочника пользователю и его ведения. Далее в статье будут приведены часто применяемые модели графических интерфейсов пользователя для различных видов справочников.

По полноте. По полноте содержащейся информации справочники можно поделить на две группы:

1) полностью описывающие все множество значений;

2) с возможностью расширения этого множества путем детализации некоторых элементов. Необходимость детализации возникает в случае, когда заранее невозможно указать множество элементов с точностью до самого низкого уровня иерархии, однако для выполнения задач автоматизированной системы имеющейся классификации достаточно.

Второй случай часто возникает, когда часть информации, необходимой для составления полного справочника, находится в ведении другой организации и доступ к ней ограничен. В качестве примера рассмотрим создание информационной системы для крупной организации и справочник адресатов для пересылки корреспонденции. Мы можем полностью отразить структуру организации, своевременно вносить в нее все изменения, но не можем делать то же самое для структуры других организаций, хотя адресаты есть и там. В таком случае приходится создавать иерархию, в которой структура своей организации заполнена точно, а другие организации входят только на уровне самих организаций (без подразделений), с возможностью ввести конкретные подразделения и фамилии во время работы пользователей справочника.

Данная характеристика справочника оказывает влияние на структуру базы данных (БД) информационной системы, а также на структуру программных классов элементов справочника при использовании популярных средств объектно-реляционного отображения (ORM, [1]). Далее в статье мы рассмотрим построение соответствующих классов.

По способу заполнения. Информационное наполнение справочников может вестись централизованно администратором системы, либо проводиться в процессе работы пользователями. Первый способ наиболее распространен, так как исключает большинство ошибок ведения справочника, наиболее распространенной из которых является наличие дубликатов значений. Дубликаты могут возникнуть вследствие опечатки при вводе, поэтому наполнение справочника пользователями, для которых ведение справочника не является основной задачей, может привести к порче справочника и неверной работе сервисов информационной системы, которые используют справочники (часто ими являются модули формирования отчетов и сбора статистики). Однако в некоторых случаях пользователи могут самостоятельно формировать справочники. Примером может служить справочник партнеров фирмы, если в нем задается не только название фирмы-партнера, но и ее уникальный ключ — ИНН. В этом случае справочник служит для автоматического заполнения множества реквизитов фирмы, если известен ее ИНН. Пользователь понимает последствия опечатки при вводе реквизитов, поэтому вероятность ошибок мала.

По времени заполнения. Справочник может заполняться в процессе работы системы, либо при ее создании разработчиком. Как правило, если справочник может быть составлен на этапе разработки системы, то это служебный справочник со значениями вида "Да", "Нет", "Неизвестно", и его ведение в процессе эксплуатации системы не требуется.

По возможности удаления элементов. Удаление элементов справочника является опасной операцией с точки зрения целостности данных информационной системы. Если на элемент справочника ссылается другой информационный объект системы, то его состояние может стать недопустимым в

терминах логики работы системы и вызвать многочисленные системные и логические ошибки. Поэтому справочники по возможности удаления элементов подразделяются на следующие:

- с возможностью удаления любого элемента справочника. Достаточно редкий случай, как правило, удаление происходит вместе со всеми связанными данными. Пример — справочник пользователей системы мгновенных сообщений, когда с элементом справочника пользователей не связаны дополнительные информационные объекты.
- с возможностью удаления неиспользованных элементов. Как правило, верно для большинства справочников.
- с невозможностью удаления. Удаление может быть невозможно по техническим причинам, когда создание записи в справочнике сопровождается работой других информационных систем, которой мы не управляем.

Являющейся потенциальным источником большого числа ошибок, трудных для отладки, операции удаления из справочника должно уделяться особое внимание во время разработки системы. На уровне баз данных рекомендуется устанавливать триггеры, предотвращающие удаление использованных в системе элементов [2].

По возможности хранения версий. Зачастую справочники системы отражают динамично меняющуюся информацию, такую как организационная структура предприятия, список сотрудников и т. д. Существует несколько подходов к ведению таких справочников, когда условия задачи требуют сохранения предыдущего состояния справочника, либо лишь текущего. Выделим три подхода к этой проблеме.

1. Справочники всегда содержатся в актуальном состоянии. То есть элементы справочников можно добавлять, удалять и менять название и другие важные для предметной области свойства без всяких ограничений.

2. Возможно наличие "отмененных" элементов. Отмена элемента означает, что этот элемент не будет доступен пользователю для выбора после момента его отмены, но ссылки на него сохранятся во всех объектах, где они были использованы до отмены.

3. Возможны наследники. Наличие наследников означает, что при отмене элемента справочника его функции, обозначенные в условии задачи проектирования, будет выполнять наследник. Примером может служить справочник структуры организации и задача передачи данных одного подразделения новому при реорганизации.

По источнику информации. При построении корпоративных информационных систем необходимо учитывать, достаточно ли у организации информации для ведения справочника. Это влияет на выбор типа справочника между допускающими уточнение и не допускающими. Пример был приведен выше. Таким образом, при классификации по источнику информации можно выделить два случая:

- у организации достаточно информации для ведения справочника;
- необходимо использовать справочники других организаций.

На основании приведенной классификации справочников можно выделить основные требования к программной реализации подсистемы ведения справочников, определить круг операций, выполняемых со справочниками, и разработать архитектуру подсистемы. Далее в статье рассматривается проект подсистемы ведения справочников с использованием UML-схем [3] для иллюстрации принятых конструкторских решений.

Проект подсистемы ведения справочников

Сценарии использования. На приведенной ниже схеме сценариев использования (рис. 3) приведены действия, которые выполняет пользователь и администратор системы со справочниками.

Связь пользователя со сценарием "Добавление элемента справочника" необходима в случае справочника с возможностью пользовательского заполнения. Все сценарии использования подразумевают возможность проверки правил целостности данных и ограничений предметной области.

Структура базы данных. Абсолютное большинство корпоративных информационных систем для хранения информации используют реляционные базы данных [4], поэтому рассмотрим структуру таблицы, содержащую набор необходимых для справочника полей (рис. 4).

Рис. 3. Схема сценариев использования

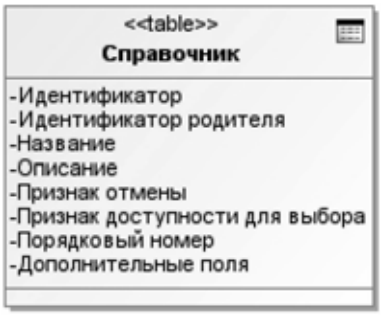


Рис. 4. Структура таблицы справочника в базе данных

Поле "Идентификатор" представляет собой первичный ключ таблицы, для ссылки на родительский элемент в дереве используется внешний ключ "Идентификатор родителя". Поле "Название" представляет собой заголовок элемента справочника, показываемый пользователю при выборе элемента, поле "Описание" используется для расшифровки краткого текста названия во всплывающей подсказке. С помощью поля "Признак отмены" в процессе работы системы помечаются записи, которые не должны использоваться во вновь создаваемых объектах. Признак доступности выбора служит для создания в дереве отдельных элементов для группировки связанных значений, но которые сами по себе не имеют смысла с точки зрения логики задачи и не должны быть выбраны пользователем. Как правило, элементы справочника имеют некоторое упорядочивание между собой, поэтому должно быть предусмотрено поле "Порядковый номер" или для упорядочивания должны быть использованы специфичные для задачи поля (например различные даты). Набор необходимых полей не исключает наличия дополнительных полей, необходимых с точки зрения задачи. Например, в справочнике пользователей системы кроме уникального имени пользователя в системе часто предусматривают поля для контактной информации.

Структура классов системы. На рис. 5 представлена структура основных классов, отвечающих за работу подсистемы справочников. Кратко опишем назначение наиболее важных классов на схеме.

Как было описано выше, все справочники системы имеют минимальный набор полей, который может расширяться в зависимости от предназначения каждого из справочников в системе. Для того чтобы обеспечить доступ к каждому из справочников независимо от его вида, выделены два интерфейса, группирующие все типовые операции с элементами справочника, — это DirectoryEntry и MutableDirectoryEntry. Первый предоставляет доступ к элементу справочника только на чтение, а второй добавляет методы модификации данных. На их основе создается класс AbstractDirectoryEntry, который реализует большинство методов указанных интерфейсов. Абстрактным класс объявлен для того, чтобы для каждого из справочников системы программист создавал свой конкретный класс. Такое решение принято в связи со спецификой работы инструментов объектно-реляционного отображения [5], которые в большинстве случаев требуют соответствия один к одному между классами и таблицами баз данных.

Для каждого справочника в системе необходимо реализовать интерфейс Directory, отвечающий за представление справочника пользователю. Помимо названия и описания справочника, этот интерфейс содержит ссылку на конкретный класс элемента справочника, что позволяет связать справочник с таблицей в базе данных. Также интерфейс Directory содержит метод для получения действий, которые могут быть выполнены с элементами справочника. Сюда входят как стандартные методы — добавление/удаление, перемещение и т. д., так и специфичные для каждого конкретного справочника. Эти действия отображаются на

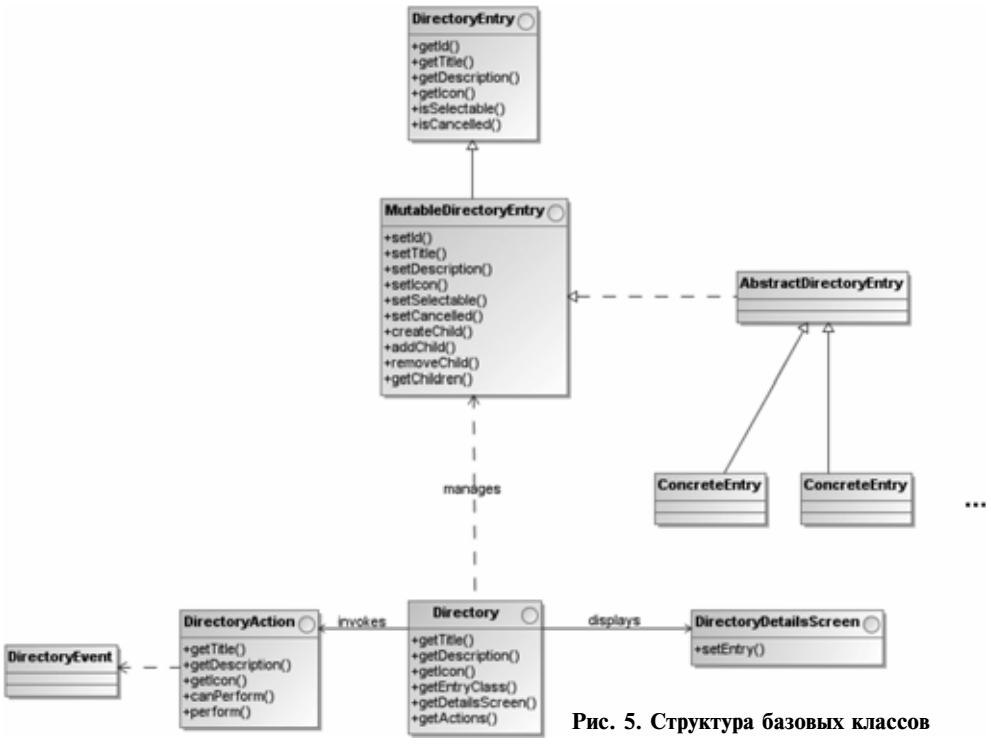


Рис. 5. Структура базовых классов

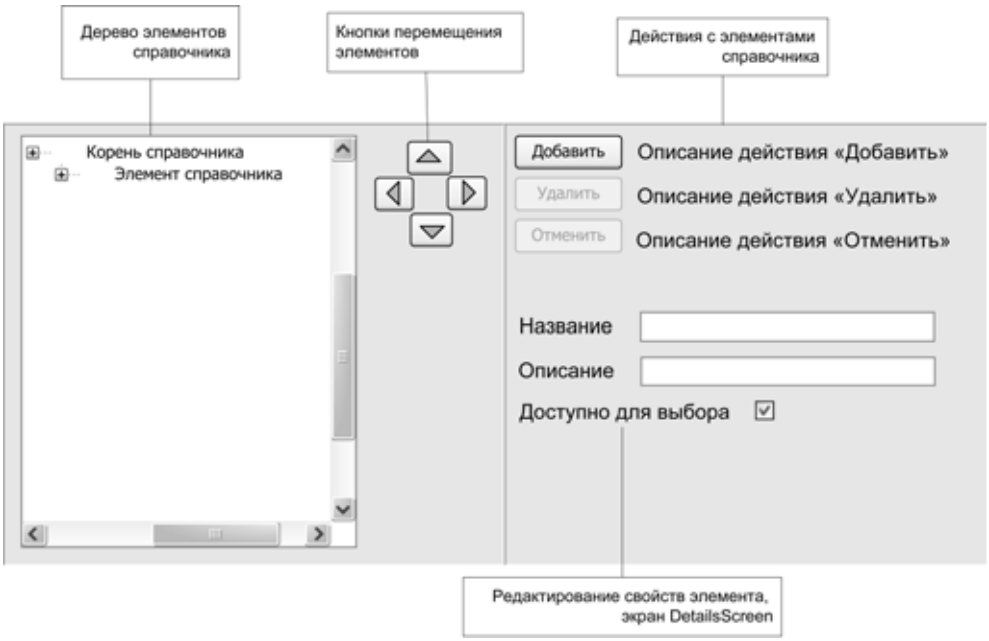


Рис. 6. Экран ведения справочника

экране ведения справочника, представленного на рис. 6.

Экран редактирования свойств элемента определяется исходя из связанного с каталогом объекта DetailsScreen. При использовании подсистемы в качестве веб-приложения может быть указан адрес страницы для редактирования свойств элемента, контролером для которой будет DetailsScreen (см. архитектуру Model-View-Controller, [6]). На экране редактирования деталей также может отображаться элемент управления для связывания справочников, если логика задачи требует связывания.

При проектировании подсистемы также следует создать набор шаблонных компонентов для выбора из справочника:

- выбор из древовидного справочника:
  - с поиском;
  - с возможностью множественного выбора;
- автозаполнение;
- выбор из плоского справочника;
- древовидный справочник с показом от определенного узла;
- древовидный справочник с фильтрацией элементов по фасетам (например, выбор файла с указанием списка расширений).

Эти компоненты покрывают большую часть взаимодействия пользователя со справочником и требуют расширения только в случае справочников, имеющих множество заданных требованиями задачи функций. Часто к ним относится справочник адресов, так как он предусматривает множество режимов поиска и подсказок для пользователя.

Выводы

Созданная с использованием описанного подхода подсистема реализована с использованием платформы Java [7] и технологии построения веб-страниц JSF [8]. Она успешно работает в проектах, реализуемых ИППИ им. А. А. Харкевича РАН, для государственных и коммерческих заказчиков. Опыт использования системы показал, что ее функционала хватает для решения большинства задач, возникающих в процессе разработки и поддержки создаваемых программных продуктов. Таким образом, выделение данной подсистемы в отдельный модуль и принятые проектные решения подтвердили свою эффективность.

Список литературы

1. **Bauer Christian, King Gavin.** Hibernate in Action. Manning Publications Co., 2005.
2. **Ульман Дж. Д., Уидом Д.** Основы реляционных баз данных. М.: Лори, 2006.
3. **Фаулер М.** UML. Основы. СПб.: Символ-Плюс, 2006.
4. **Эмблер С. В., Садаладж П. Дж.** Рефакторинг баз данных. Эволюционное проектирование. М.: Вильямс, 2007.
5. http://hibernate.org
6. **Ladd S., Davison D., Devijver S., Yates C.** Expert Spring MVC and Web Flow (Expert). APRESS, 2006.
7. http://iava.sun.com
8. http://iava.sun.com/javaee/javaserverfaces/

УДК 004.056

В. И. Левандовский, докторант, преподаватель, Экономическая академия Молдовы, г. Кишинев, e-mail: Depechevl70@mail.ru

Обеспечение защиты программных приложений от нелицензионного доступа

Рассматривается методика создания программного модуля защиты системы управления базой данных от нелицензионного использования.

Ключевые слова: *алгоритм защиты программы от нелицензионного доступа, защита программных приложений, нелицензионный доступ, лицензионный доступ, защита от нелицензионного использования, лицензионное использование.*

В статье рассмотрена защита локальной системы управления базой данных от нелицензионного использования, разработанная на базе языка программирования Borland DELPHI. Основная идея защиты состоит в следующем.

При каждой загрузке приложения происходит подсчет числа запусков и отработанных часов. Данные о числе запусков и часах хранятся в реестре Windows. При запуске также выполняется проверка, не превышают ли число запусков и число отработанных часов критических значений, установленных разработчиком. Если одно из значений выше критического, то приложение выдаст сообщение о завершении лицензии и закроется, не давая возможности работать с базой данных.

Особенность данного механизма защиты, в отличие от существующих аналогичных систем, заключается в том что данные о числе запусков хранятся сразу в нескольких разделах реестра Windows. То есть если взломщик отследит и обнулит регистр в реестре Windows, все равно доступ к информации не будет получен, так как приложение перед проверкой условий запуска внесет в обнуленный раздел реестра значение одного из резервных регистров.

Для создания дополнительной степени защиты можно оперировать всей информацией о числе запусков и отработанном времени в зашифрованном виде. Тогда в программу нужно будет добавить функции шифровки и дешифровки данных. Функция шифрования должна активизироваться перед передачей данных в реестр, а функция дешифровки — после получения данных из реестра.

Помимо того, что приложение использует реестр, оно также создает резервный файл на локальном диске с теми же данными о числе отработанных часов запусков. То есть если все регистры будут обнулены, то в такой ситуации приложение воспользуется значениями из резервного файла с локального диска. Таким образом, мы создаем дополнительную степень защиты, обезопасив себя от ситуации, когда злоумышленник успешно отследит и обнулит значения во всех регистрах реестра.

Алгоритм защиты программы от нелицензионного доступа

Рассмотрим более детально суть алгоритма. Приложение после установки все-таки дает возможность пользователю поработать с программой какое-то время, к примеру 100 ч. По истечении этого периода приложение блокируется, выдавая сообщение о нелицензионном использовании продукта. Параллельно с подсчетом числа наработанных часов система подсчитывает число запусков. Для примера предположим, что

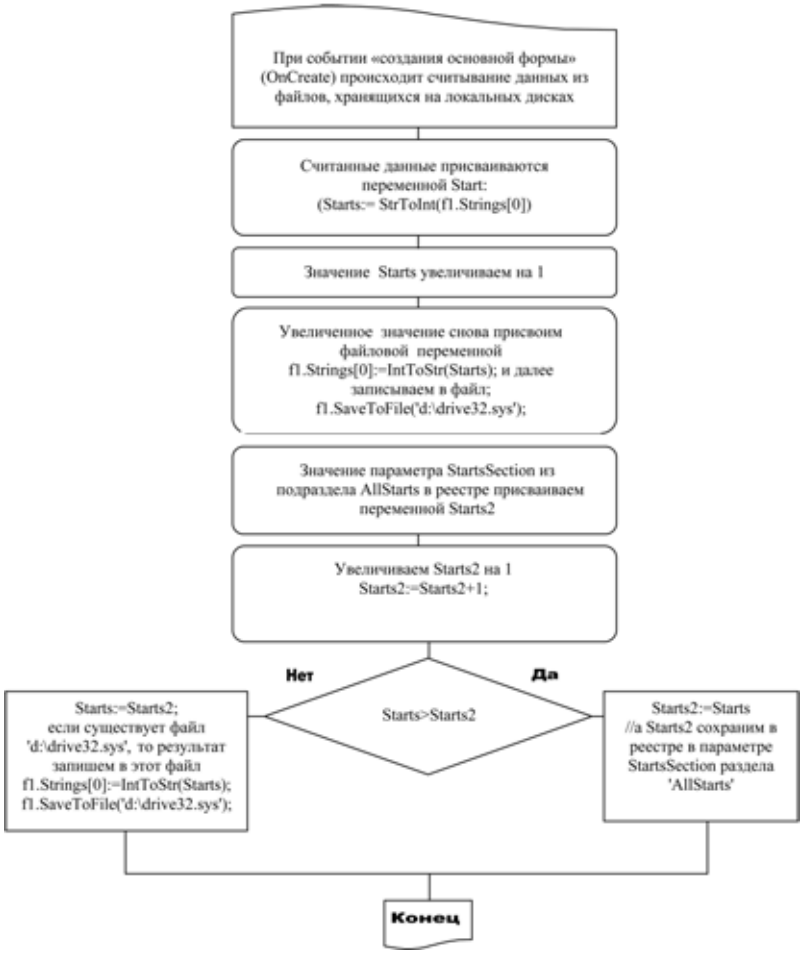


Рис. 1. Алгоритм процедуры при событии "создание основной формы" (OnCreate): Starts — число запусков; Starts2 — число запусков (резервная переменная); FileVar, AllTicks и Watch — число отработанных и резервных переменных; Max — максимальное значение

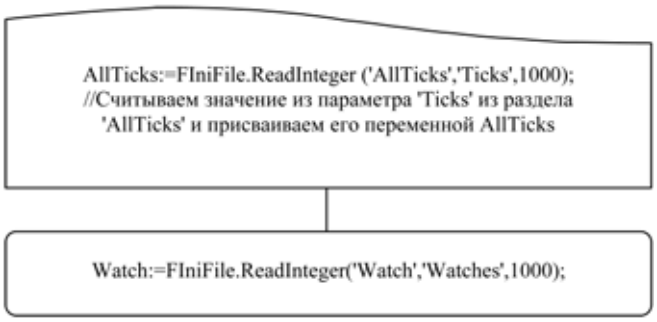


Рис. 2. Алгоритм функции LoadProtectParam

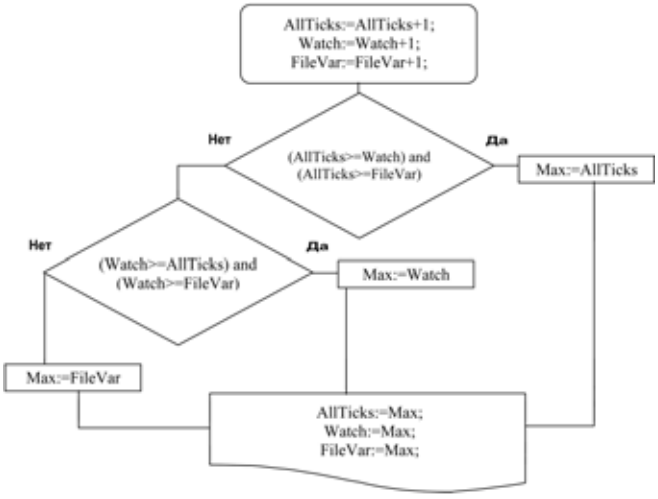


Рис. 3. Алгоритм процедуры, считающей число отработанных секунд

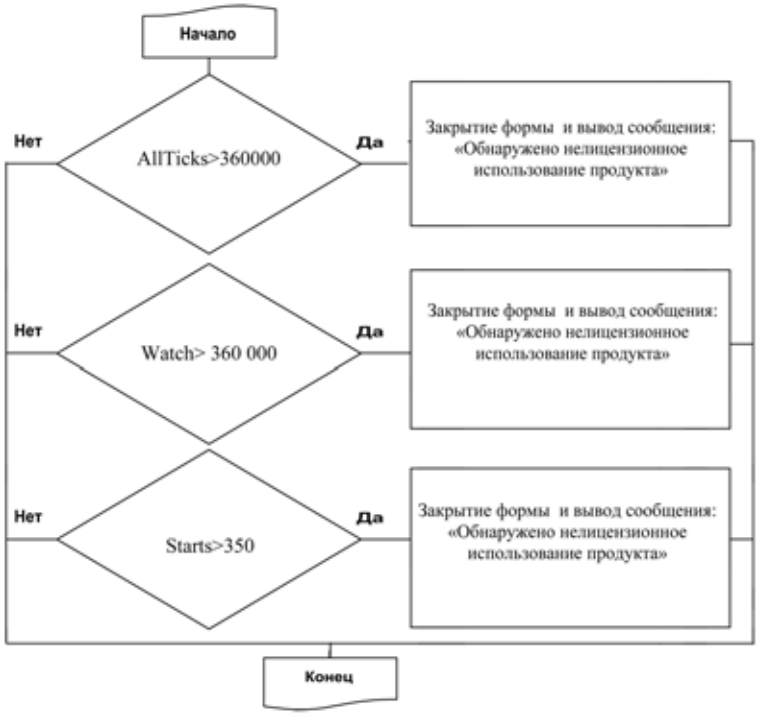


Рис. 4. Часть алгоритма процедуры SaveProtectParam, выполняющего проверку, не закончился ли лицензионный срок

число запусков не должно превышать 350. Если хотя бы одно из значений превысит установленный лимит, то приложение работать не будет. В этой ситуации лицензионный пользователь сможет воспользоваться специальной программой-ключом, с помощью которой сможет обнулить накопленные значения числа отработанных часов и запусков, а следовательно, разблокирует приложение.

При каждой загрузке приложение увеличивает значение числа запусков системы на единицу (рис. 1).

За событием создания формы (OnCreate) Delphi генерирует событие-процедуру появления формы (OnShow), в алгоритме которой имеется строка с вызовом функции LoadProtectParam, предназначенной для открытия реестра и считывания значений числа запусков и отработанного времени. Эти значения затем присваиваются глобальным переменным AllTicks и Watch (рис. 2), а также, если существует локальный диск D, to из файла 'd:\sys.drv' считанное значение присваивается глобальной переменной FileVar.

Добавленный на главную форму таймер TimerCountTicks по событию OnTimer с интервалом в 1 с увеличивает на единицу значения переменных, хранящих значение числа отработанных секунд (AllTicks, Watch и FileVar), затем определяет из них максимальное значение и присваивает его всем трем переменным (рис. 3).

Таким образом, если злоумышленник обнулит один из регистров, приложение не примет во внимание его обновленное нулевое значение, так как алгоритм будет принимать за основное не минимальное, а максимальное значение числа отработанных секунд или запусков.

В программе имеется еще один таймер TimerSendMessageTimer с интервалом действия 10 с. По событию OnTimer таймера TimerSendMessageTimer вызывается функция SaveProtectParam, по названию которой нетрудно догадаться, что она предназначена для сохранения данных в реестре и в файлах. В этой же процедуре по событию OnTimer выполняется проверка, не истек ли лицензионный срок использования. Если срок истек, то приложение закроется с соответствующим сообщением о нелицензионном использовании (рис. 4).

Список литературы

1. **Гофман В. Э., Хомоненко А. Д.** Delphi — быстрый старт. СПб.: БХВ-Петербург. 2003. С. 69.

2. **Фленов М.** Библия для программиста в среде Delphi. Copyright. 2002. www.cydsoft.com/vr-online. С. 236—242.

3. **Информационные** технологии современной России. http://www.rbc.ru.

4. **Дарахвелидзе П. Г., Марков Е. П.** "Программирование в Delphi 7. СПб.: БХВ-Петербург. 2003. С. 112—125.

УДК 004.051:007.51

Н. С. Белова, аспирант,
Московский государственный университет
приборостроения и информатики,
e-mail: ndovost@mail.ru

Методика приоритетного автовосстановления встраиваемых баз данных

Рассмотрены основные проблемы методов автовосстановления встраиваемых баз данных. Предложен новый метод автовосстановления, а также методика приоритетного автовосстановления с учетом разработанной качественной оценки эффективности встраиваемых баз данных.

Ключевые слова: *встраиваемые базы данных, автовосстановление, моделирование, эффективность, приоритет.*

Введение

Развитие современной техники и текущее состояние бизнеса привели к тому, что внедрение новых технологий с использованием мобильных вычислительных устройств очень удачно вписывается в рамки автоматизированных корпоративных программных систем, помогая наиболее оптимально использовать человеческие ресурсы и добиваться максимальной производительности и прибыли. В связи с этим популярность интеллектуальных мобильных устройств, которые используют базы данных (БД) специального типа, так называемые *встраиваемые БД* (*ВБД*), постоянно растет. Встраиваемые базы данных существенно отличаются от обычных БД следующим:

- работают с приложениями, как правило, в одном адресном пространстве;
- обладают высокой производительностью и надежностью при ограниченных ресурсах;
- функционируют прозрачно для пользователей;
- не требуют дополнительного администрирования.

ВБД уже заняли свою нишу на рынке и продолжают доказывать свое право на существование и развитие. К основным преимуществам ВБД относят простоту эксплуатации и существенно меньшую стоимость [1]. Простота эксплуатации, с точки зрения пользователя, достигается за счет прозрачности функционирования БД, администрирования ВБД и, в частности, осуществления процесса автовосстановления ВБД.

В настоящее время применяются два метода автовосстановления:

- метод анализа трасс;
- метод восстановления из резервной копии [2].

Однако сфера применения каждого из этих методов ограничена: в одном случае — требованием априорной известности номенклатуры распознаваемых сбоев, а в другом — требованием невысокой скорости внесения изменений в данные.

Новый метод автовосстановления встраиваемых баз данных

Для решения указанных выше проблем предлагается новый метод автовосстановления встраиваемых баз данных, который, в отличие от методов, описанных во введении, позволяет проводить анализ и лечение возникающих некорректных ситуаций (далее ошибок) в полном объеме. Это достигается за счет того, что обработка ошибок осуществляется не только встроенным модулем, реагирующим на заранее определенную номенклатуру ошибок, но и экспертами специального сервисного центра.

Метод может быть использован, например, для автовосстановления ВБД карманных персональных компьютеров (КПК) сотрудников предприятия. Такие ВБД содержат фрагменты корпоративной базы данных, синхронизация которых с центральной БД происходит с некоторой периодичностью. Будем считать, что все КПК имеют возможность сеансового подключения к сети Интернет для связи с центральной БД.

В момент возникновения ошибки встроенный модуль собирает всю информацию о ней и анализирует полученные данные. Если встроенному модулю не удалось исправить ошибку в автоматическом режиме, данные об ошибке фиксируются в файле ошибок. К данным об ошибке относятся:

- причина возникновения сбоя;
- тип ошибки, определяемый СУБД;
- имя таблицы, к которой было обращение;
- тип запроса;
- параметры запроса;
- данные о действии пользователя, совершенном в момент возникновения ошибки.

При ближайшем подключении пользователя к сети Интернет файл ошибок передается в сервисный центр, где выполняется автоматический анализ файлов ошибок, полученных со всех КПК. Результаты этого анализа становятся доступны специалистам, которые формулируют способ исправления ошибок данного типа и передают его в таблицу способов устранения ошибок в БД сервисного центра, а затем, в процессе регулярного обновления ВБД, передаются на все обслуживаемые КПК. Обновление не только исправляет некорректную ситуацию в базе данных, но и сохра-

няет все данные об ошибке и способе ее устранения в базе прецедентов, хранящейся на конкретном КПК. Благодаря этому при повторном появлении подобной ошибки действия по ее исправлению будут сгенерированы автоматически, что позволит повысить эффективность и надежность работы ВБД прозрачно для пользователя.

Методика приоритетного автовосстановления ВБД

Для того чтобы обработка и устранение ошибок, поступающих в единый сервисный центр, специалистами осуществлялись наиболее оперативно, разработана методика приоритетного автовосстановления ВБД, которая позволяет автоматически расставлять приоритеты ошибкам в режиме реального времени на основании разработанной системы качественной оценки эффективности работы ВБД.

Качественная оценка эффективности работы ВБД рассчитывается для конкретного интервала времени в зависимости от общего числа запросов к ВБД, числа запросов, выполненных без ошибок и числа запросов, выполненных с какими-либо ошибками. Эффективность работы ВБД отображается с помощью числа с плавающей запятой в диапазоне от 0 до 1.

Математическая модель, с помощью которой выполняется оценка эффективности работы ВБД, должна удовлетворять следующим требованиям:

- при небольшом числе критических ошибок на значительное число запросов эффективность должна резко снижаться;
- при значительном числе запросов время расчета эффективности должно быть минимально, таким образом, модель должна быть максимально простой и обладать допустимой точностью;
- модель должна учитывать различные типы ошибок.

Согласно перечисленным выше требованиям предложена следующая математическая модель оценки эффективности работы ВБД:

$$E_{\text{ВБД}}(t) = 1 - \frac{1}{1 + x(t)K}, \qquad (1)$$

где $E_{\text{ВБД}}(t)$ — эффективность работы ВБД за период времени t; $x(t)$ — общее число запросов за период времени t; K — коэффициент эффективности, который отражает зависимость эффективности работы ВБД от ошибочных запросов.

Коэффициент эффективности K рассчитывается по формуле

$$K = \sum_{i=1}^{n} k_i = \sum_{i=1}^{n} \frac{R_i W_i}{R_{\text{общ}}}, \qquad (2)$$

где k_i — коэффициент эффективности, который отражает зависимость эффективности работы ВБД от ошибочных запросов конкретного типа; n — число различных типов ошибок; R_i — число ошибочных запросов данного типа; W_i — значимость ошибочных запросов данного типа; $R_{\text{общ}}$ — общее число запросов за отчетный период.

Под значимостью W_i понимается то, насколько сильно снижает эффективность данный тип ошибки.

Определение коэффициента эффективности ограничено условием, что число ошибок не будет превосходить число запросов. Другими словами, будут отслеживаться лишь одиночные ошибки. Это обусловлено тем, что возникновение двойных и более ошибок на один запрос маловероятно и реализация их обработки является нецелесообразной.

Также необходимо учитывать, что значимость одной конкретной ошибки W_i должна уменьшаться с ростом общего числа запросов, т. е.

$$W_i = R_i / R_{\text{общ}}. \qquad (3)$$

Согласно (3) коэффициент эффективности будет рассчитываться по формуле

$$k_i = \frac{R_i^2}{R_{\text{общ}}^2}. \qquad (4)$$

Формула расчета эффективности работы ВБД за период времени t будет иметь вид

$$E_{\text{ВБД}}(t) = 1 - \frac{1}{1 + x(t) \sum_{i=1}^{n} k_i}. \qquad (5)$$

Благодаря описанной выше модели качественной оценки эффективности работы ВБД расстановка приоритетов ошибкам может осуществляться в режиме реального времени автоматически, при этом сервисным центром обслуживаются сразу все доступные ВБД.

Алгоритм расстановки приоритетов:

1. Анализ каждого файла ошибок, поступающего на вход сервисного центра от обслуживаемых КПК. На основании информации об общем числе запросов в ВБД, типах ошибочных запросов и методике качественной оценки эффективности работы ВБД рассчитывается эффективность работы конкретной ВБД.

2. На основании полученного результата эффективности работы расстановка приоритетов ошибкам — от высшего к низшему, в зависимости от типа ошибок, которые наиболее негативно сказываются на эффективности работы ВБД. Другими словами, определяются те ошибки, коэффициент эффективности которых минимален.

3. Объединение всех файлов ошибок с уже расставленными приоритетами.

4. Вычленение однообразных ошибок. Их приоритеты суммируются и делятся на общее число ошибок данного типа.

5. Сравнение полученных результатов и расстановка окончательных приоритетов.

Ошибки поступают для исправления специалистам по очереди, согласно автоматически расставленным приоритетам. Устраненные ошибки из очереди удаляются. Вновь поступившие файлы ошибок обрабатываются согласно шагам с 1-го по 3-й, а затем объединяются с уже имеющимися, и происходит перерасстановка приоритетов согласно шагам с 4-го по 5-й. Таким образом, приоритеты ошибок всегда остаются актуальными, за счет чего достигается максимальная оперативность решения ошибок.

Заключение

В работе проведен анализ существующих методов автовосстановления встраиваемых баз данных. Отмечены их основные недостатки. Предложена новая модель автовосстановления ВБД, которая включает в себя методику приоритетного автовосстановления ВБД. Данная методика на основе модели оценки эффективности работы ВБД позволяет расставлять приоритеты ошибкам в режиме реального времени, тем самым сокращая время анализа и увеличивая скорость обработки ошибок специалистами сервисного центра.

Список литературы

1. **Olson M. A.** Selecting and Implementing an Embedded Database System // IEEE Computer, September 2000. P. 27—34.

2. **Bowman I. T., Bumbulis P., Farrar D., Goel A. K., Lucier B., Nica A., Paulley G. N., Smirnilos J., Young-Lai M.** SQL Anywhere: An Embeddable DBMS // Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, September 2007. Vol. 30, N 3.

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 004.9:55

Д. Н. Кобзаренко, ст. науч. сотр.,
Институт проблем геотермии Дагестанского НЦ РАН, г. Махачкала,
e-mail: kobzarenko_dm@mail.ru,
А. М. Камилова, ст. преподаватель,
Р. Н. Гаджимурадов, ст. преподаватель,
Дагестанский государственный технический университет, г. Махачкала

Концепция построения системы трехмерного геоинформационного моделирования

Предлагается концепция построения системы трехмерного геоинформационного моделирования для решения задач, связанных с пространственными данными в науках о Земле. Концепция построения системы излагается с позиций: постановки задачи, основных определений, структуры и компонентов, данных проекта и принципа функционирования.

Ключевые слова: *геоинформационные технологии, геоинформационные системы.*

Введение

В науках о Земле существует множество задач, в которых исходные и результирующие данные имеют пространственное позиционирование, поэтому де-факто вопросы сбора, обработки, расчетов и вывода результата лежат в области геоинформационных систем (ГИС) и технологий. Тенденция в развитии современных ГИС такова, что можно выделить так называемые системы общего назначения (*ArcGIS, MapInfo*), направленные на решение широкого круга общих задач, и специализированные системы. Специализированные системы, в свою очередь, ориентированы либо на решение конкретных задач (ГИС-ИНТЕГРО в природопользовании [1, 2] или *3DGeomodeller* для построения трехмерных геологических моделей [3]), либо на предоставление пользователю некоторого вычислительного инструмента (интерполятор на регулярной сетке *Surfer* [4]).

Если понимать под решением научной задачи получение принципиально новой информации (как количественной, так и качественной) на базе некоторой модели и исходных данных, то можно смело утверждать, что системы общего назначения, строго говоря, не решают научных задач. Они, скорее, предоставляют инструментальные средства, позволяющие преобразовывать, хранить и визуализировать пространственную информацию для ее последующей интерпретации специалистом. Но это не умаляет достоинств этих систем, иначе они бы не получили столь массового распространения.

Специализированные системы направлены на решение конкретных научных задач и, как правило, разрабатываются в сугубо научных организациях. В России, пожалуй, наиболее преуспевшей организацией в области специализированных ГИС является ВНИИгеосистем. Разработанная во ВНИИгеосистем ГИС-ИНТЕГРО является очень мощным инструментом геолого-геофизического моделирования и картографирования, не имеющим аналогов ни в России, ни в мире.

Геоинформатика является активно развивающейся отраслью науки. Ее развитию способствуют новые возможности вычислительной техники и коммуникаций, новые задачи, связанные с пространственными данными, и новые подходы в геоинформационном моделировании.

В процессе постановки и решения задач в области оценки геотермальных энергоресурсов в Дагестане [5] у нас возникла идея создания мощного ГИС-проекта, позволяющего эффективно решать подобные научные задачи. Проект назван *системой трехмерного геоинформационного моделирования* (СТГМ). Идея проекта возникла не с нуля. Разработки в области ГИС нами ведутся с 2001 г. Вначале подобная система называлась технологией построения и визуализации цифровых картографических 3D-моделей и тематических модулей (2001—2004 гг.) [6], затем технологией построения электронного 3D-атласа (2004—2005 гг.) [7]. В данное время проект получил новый толчок в развитии. Мы взяли все лучшее от предыдущих разработок и обновили его принципиально новыми идеями и подходами. В результате разработана концепция построения системы трехмерного геоинформационного моделирования для решения задач в науках о Земле (сохраняется и старое рабочее название — электронный 3D-атлас).

Постановка задачи построения СТГМ

Решение любой задачи, связанной с пространственными данными в науках о Земле, можно описать следующим образом. Имеется объект моделирования (территория), который задается границами и уровнем генерализации или разрешением модели (точность представления информации). Имеется методика расчета (математическая модель) задачи, которая, в свою очередь, может предполагать разбиение задачи на дерево подзадач. Расчетная модель имеет свои параметры, варьируя которыми можно получать разный результат. Решение задачи предполагает набор исходных данных. Естественно, все исходные данные имеют пространственное позиционирование. Их можно разбить на четыре группы: картографический материал, дистанционные данные, наземные измерения и предположения ученых (там, где имеются существенные пробелы в данных) (рис. 1, см. четвертую сторону обложки).

Возьмем пример реальной задачи. Согласно [8], потенциальные геотермальные ресурсы характеризуют тепловой потенциал толщи пород на прогнозируемую глубину бурения до 10 км. Плотность распределения ресурсов определяется исходя из предпосылки, что массив можно охладить до температуры окружающей среды:

$$Q^{\text{р}} = kc_{\gamma}(H_{\text{пр}} - h_{\text{н.с}})(t_{\text{из}} - t_{\text{о.с}}), \qquad (1)$$

где $Q^{\text{р}}$ — плотность распределения ресурсов, т · у.т./м^2 (у.т. — условное топливо); k — коэффициент пересчета, т · у.т./Дж; c_{γ} — объемная теплоемкость пород, Дж/(м^3 · °С); $H_{\text{пр}}$ — прогнозная глубина бурения, м; $h_{\text{н.с}}$ — мощность нейтрального слоя, м; $t_{\text{о.с}}$ — температура окружающей среды, °С; $t_{\text{из}}$ —средняя температура массива, °С; $t_{\text{из}} = 0{,}5(t_{\text{пр}} + t_{\text{н.с}})$; $t_{\text{пр}} = G(H_{\text{пр}} - h_{\text{н.с}}) + t_{\text{н.с}}$; $t_{\text{н.с}}$ — температура нейтрального слоя, °С; $t_{\text{пр}}$ — температура пород на прогнозируемой глубине, °С; G — геотермический градиент, °С/м.

Из приведенного примера следует, что для полноценного решения задачи по оценке плотности распределения потенциальных геотермальных ресурсов необходимо вначале построить геологическую модель, чтобы рассчитать объемную теплоемкость пород, а затем и модель распределения температурного поля для расчета геотермического градиента. Таким образом, задача содержит в себе две подзадачи.

Анализ вышесказанного показывает, что СТГМ должна отвечать трем основным требованиям:

- обеспечение единого информационного поля для определенного объекта моделирования. Оно заключается в создании собственной системы координат объекта моделирования (с привязкой к географическим координатам) и в создании собственных банков данных;
- унифицированные структуры данных, позволяющие хранить разнородную информацию и оперировать в рамках любого объекта моделирования для решения любой пространственной задачи;
- единые унифицированные правила функционирования системы по управлению объектом моделирования, библиотеками данных, визуализацией и операционными модулями, решающими задачи.

Основные определения СТГМ

Концепция построения СТГМ базируется на следующих определениях.

Система трехмерного геоинформационного моделирования (*СТГМ*) направлена на решение задач, связанных с пространственными данными в области наук о Земле.

Проект СТГМ — совокупность объединенных данных, ассоциируемая с объектом моделирования — определенной территорией и масштабом.

Библиотека рабочих данных (*БРД*) — хранилище всей пространственно-атрибутивной информации в рамках текущего проекта, включающее как базовую (исходную) информацию, так и информацию, сгенерированную в процессе решения некоторой задачи.

Библиотека визуализируемых данных (*БВД*) — хранилище информации, адаптированной для визуализации с использованием графической библиотеки *OpenGL*, и ее интерпретации (легенда).

Генератор рабочих данных (*ГРД*) — самостоятельное приложение, осуществляющее выполнение совокупности операций (решение некоторой задачи) над группой данных из БРД текущего проекта и сохраняющее результат операций в той же БРД текущего проекта. Приложение ГРД может запускаться только из основной ГИС и функционирует в рамках параметров текущего проекта. Функциональность системы определяется совокупностью генераторов рабочих данных. Расширение функциональности СТГМ — разработка и добавление новых ГРД.

Генератор визуализируемых данных (*ГВД*) — самостоятельное приложение, осуществляющее преобразование группы данных из библиотеки БРД в формат, адаптированный для визуализации, и сохраняющее результат в БВД текущего проекта.

Структура и компоненты СТГМ

СТГМ состоит из двух частей: программная оболочка (ядро системы), объединяющая ГИС в единое целое, и библиотека генераторов данных (рис. 2, см. четвертую сторону обложки). Программная оболочка состоит из нескольких программных модулей, где каждый модуль отвечает за конкретную задачу. В настоящее время определены семь модулей.

1. *Модуль создания проекта.* Выполняет генерацию следующей информации: идентификатор проекта, система координат проекта, уровень генерализации (разрешение в метрах), опорные узлы привязки к географическим координатам, контур проекта. Для генерации проекта используется картографическая основа определенного масштаба. Вместе с проектом создаются библиотеки рабочих и визуализируемых данных. Наличие связи проекта с библиотеками данных определяется через идентификатор проекта.

2. *Модуль импорта данных.* Выполняет импорт данных в текущий проект. Данные импортируются в библиотеку рабочих данных. Модуль состоит из множества конвертеров.

3. *Модуль работы с библиотекой рабочих данных.* Предоставляет возможности просмотра списка и некоторого редактирования информации в БРД.

4. *Модуль работы с генераторами рабочих данных.* Предоставляет возможности просмотра имеющихся в системе генераторов рабочих данных, добавления в систему нового генератора и запуска на выполнение любого ГРД в рамках текущего проекта.

5. *Модуль работы с генераторами визуализируемых данных.* Аналогично предыдущему модулю, только для ГВД.

6. *Модуль выборки данных для визуализации.* Предоставляет возможность выбора данных из БВД для их комплексной визуализации в рамках текущего проекта.

7. *Модуль визуализации.* Осуществляет визуализацию выбранной информации в рамках текущего проекта.

Библиотека генераторов данных состоит из множества ГРД (рис. 3) и ГВД (рис. 4). Генератор данных программно независим от проекта СТГМ (в том смысле, что может функционировать в рамках любого проекта), но в то же время для его запуска и успешного функционирования необходима передача основных параметров текущего проекта.

Рис. 3. Схема функционирования ГРД:
$Д_i$ — файлы с исходными данными; $Р_i$ — файлы с данными результата

Рис. 4. Схема функционирования ГВД:
$Д_i$ — файлы с исходными данными; Р — файл с данными результата

ГРД может быть разработан для решения задач общего плана (двумерная и трехмерная интерполяция на регулярных сетях без учета специфики предметной области, математические операции над сетями и т. п.) или специализированных задач (интерполяция на регулярных сетях с учетом специфики предметной области, построение трехмерной модели геологического строения, температурного поля и т. п.).

Число ГВД в системе соответствует числу типов визуализируемых структур данных: по одному генератору на каждый тип данных.

Данные проекта СТГМ

Проект СТГМ состоит из группы параметров, библиотеки рабочих данных и библиотеки визуализируемых данных. Разделение данных в проекте на рабочие и визуализируемые объясняется тем, что информация, которую предстоит визуализировать, отличается от информации, используемой в расчетах. Такое разделение, насколько нам известно, не предусматривается в современных ГИС. Но мы считаем, что оно очень эффективно с точки зрения, с одной стороны, повышения скорости и качества визуализации — пожалуй, самого важного аспекта в трехмерной системе, с другой — придает системе особую гибкость в решении задач.

Для библиотеки рабочих данных предусмотрены три структуры (формата):

- "векторные объекты";
- "регулярная сеть со значениями в узлах";
- "регулярная сеть с индексами в узлах".

Структура "*векторные объекты*" содержит информацию о массиве объектов, где каждый объект состоит из N вершин с координатами (x, y, z) и M атрибутов (строкового, целого или вещественного типа). Такая структура позволяет импортировать в проект любую разнородную информацию: картографические объекты, скважинные данные, данные космических снимков и т. п.

Структура "*регулярная сеть со значениями в узлах*" содержит регулярное распределение вещественной величины в границах интервалов по X и Y. Это классическая структура данных в современных ГИС. В ней можно хранить рассчитанную или интерполированную величину (например, температуру), распределенную на определенной глубине (2D-модель). Для хранения рассчитанной величины, распределенной в объеме (3D-модель), используется множество сетей.

Структура "*регулярная сеть с индексами в узлах*" отличается от предыдущей тем, что в узле хранится не вещественное значение, а целочисленная ссылка на некоторый объект, определенный в таблице объектов, хранящейся в этой же структуре данных. Она позволяет хранить данные о принадлежности узла объекту (например, геологической формации) на определенной глубине (2D-модель). Для хранения данных об объектах в объеме (3D-модель) используется множество сетей.

Файлы библиотеки визуализируемых данных содержат информацию, адаптированную для визуализации с помощью механизмов графической библиотеки *OpenGL*. Они содержат: легенду отображаемой информации, геометрию визуализируемых объектов (в большинстве случаев объекты разбиваются на треугольники) и условный цвет объектов (согласно легенде). Основные типы визуализируемых структур: триангуляционная сеть, картографические объекты (рис. 5, *а*, см. четвертую сторону обложки), текстуры (рис. 5, *б*), трехмерное поле (рис. 5, *в*). В процессе функционирования СТГМ не исключено добавление новых типов визуализируемых структур при необходимости.

Функционирование СТГМ

СТГМ может функционировать и как решающая система, и как система комплексной визуализации данных. Геоинформационное моделирование выполняется в рамках проекта, создаваемого в СТГМ. Проект является информационной основой для решения задач и комплексной визуализации данных.

Любая задача решается в СТГМ путем разбиения ее на дерево подзадач и последовательного их решения снизу вверх. На первом уровне выполняется импорт исходных данных, которые заносятся в БРД текущего проекта с помощью модуля импорта данных. Затем, снизу вверх по дереву, решаются подзадачи с использованием библиотеки ГРД. Все промежуточные решения заносятся в БРД. Когда получен окончательный результат, из БРД выбираются данные, которые мы хотим визуализировать. На основе этих данных с помощью ГВД создаются визуализируемые структуры и заносятся в БВД. А уже из БВД мы формируем необходимые нам картины комплексной визуализации в проекте.

Вернемся к задаче о плотности распределения потенциальных геотермальных ресурсов (см. формулу (1)). Если предположить, что глубина нейтрального слоя константа, то для рассматриваемой задачи дерево подзадач выглядит, как показано на рис. 6. Все подзадачи в СТГМ условно разделены на четыре уровня: импорт, расчет, подготовка визуализации и визуализация. Вершины дерева соответствуют следующим задачам:

Z_{11} — импорт данных о залегании стратиграфических горизонтов;

Z_{12} — импорт данных о глубинных температурах;

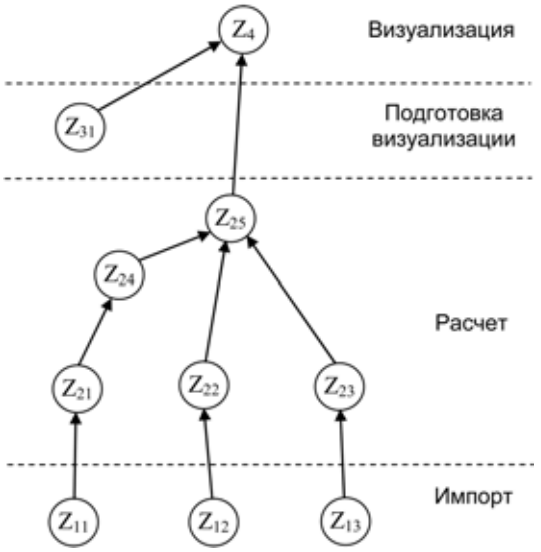


Рис. 6. Пример дерева подзадач при решении задачи в СТГМ

Z_{13} — импорт данных о приповерхностной температуре воздуха;

Z_{21} — расчет трехмерной модели геологической среды, в результате которого получаем набор регулярных сетей с данными о мощностях горизонтов;

Z_{22} — расчет регулярной сети распределения средней температуры массива ($t_{из}$) на основе данных о глубинных температурах;

Z_{23} — расчет регулярной сети распределения приповерхностной температуры воздуха ($t_{о.с}$);

Z_{24} — расчет регулярной сети распределения объемной теплоемкости пород (c_γ) по формуле

$$c_\gamma = \frac{\sum_{i=1}^{n} c_i h_i}{nH}, \qquad (2)$$

где n — число стратиграфических горизонтов; c_i — средняя объемная теплоемкость пород i-го горизонта; h_i — мощность i-го горизонта; H — суммарная мощность всех горизонтов;

Z_{25} — расчет регулярной сети распределения плотности потенциальных геотермальных ресурсов Q^p по формуле (1) на основе уже вычисленных регулярных сетей (c_γ, $t_{из}$, $t_{о.с}$);

Z_{31} — задание шкалы плотности потенциальных геотермальных ресурсов и псевдоцветов для каждого диапазона;

Z_4 — генерирование текстуры (по примеру, изображенному на рис. 5, *б*), отображающей в псевдоцветах распределение Q^p.

Каждая единица данных (файл), находящаяся в библиотеках рабочих и визуализируемых данных проекта, помимо полей "Название" и "Ключевое слово", используемых для быстрого поиска, содержит также поле, именуемое "История". Это поле аккумулирует (в хронологическом порядке) всю информацию, связанную с текущими данными. Таким образом, для каждой единицы данных доступна информация:

- об источниках и кратких описаниях всех данных, которые участвовали в цепочке получения текущего файла;
- о генераторах (моделях и методах), которые участвовали в цепочке получения текущего файла.

Механизм учета "Истории данных" дает возможность пользователю в любой момент ответить на вопрос: что смоделировано, какие данные и какая методика вложены в модель?

Заключение и выводы

Разработанная концепция построения системы трехмерного геоинформационного моделирования сориентирована на решение задач, связанных с пространственными данными в области наук о Земле. Концепция предполагает создание ГИС-проектов регионального (область, край) и локального (город и окрестности) масштаба.

Система не только "визуализирует", но и "решает", т. е. может получать принципиально новую информацию с помощью специализированных методик и моделей, вложенных в генераторы данных.

Принцип построения системы предоставляет гибкую возможность ее адаптации к новым задачам путем разработки и добавления новых генераторов данных. Гибкость системы заключается еще и в возможности разбить любую задачу на дерево подзадач и их последовательного решения. Полезной для ученого является возможность визуализации данных на всех стадиях решения задачи.

Список литературы

1. **Финкельштейн М. Я., Деев К. В.** Развитие инструментальных средств ГИС ИНТЕГРО // Геоинформатика. 2003. № 2. С. 49—51.

2. **Черемисина Е. Н., Никитин А. А.** Геоинформационные системы в природопользовании // Геоинформатика. 2006. № 3. С. 5—20.

3. **Calcagno P., Chiles J. P., Courrioux G., Guillen A.** Geological modelling from field data and geological knowledge. Part I. Modelling method coupling 3D potential-field interpolation and geological rules // Physics of the Earth and Planetary Interiors. 2008. N 11.

4. http://www.goldensoftware.com/

5. **Алхасов А. Б., Кобзаренко Д. Н.** Потенциальные геотермальные ресурсы Республики Дагестан // Матер. Междунар. конф. "Возобновляемая энергетика: проблемы и перспективы". Т. 2. Махачкала, 19—22 сентября 2005 г. С. 4—7.

6. **Кобзаренко Д. Н.** Цифровые картографические 3D-модели для решения геолого-геофизических задач. Автореферат дисс. на соискание ученой степени канд. техн. наук. М.: МГГРУ. 2004. 20 с.

7. **Kobzarenko D. N., Bulaeva N. M., Osmanov R. Sh., Magomedov B. I.** Project of the electronic 3D-atlas of the Republic Daghestan // II International conference "GIS in geology", extended abstracts, Vernadsky State Geological Museum of RAS, Moscow, 15—19 November, 2004. P. 56—57.

8. **Богуславский Э. В.** Тепловые ресурсы недр России // Теплоэнергетика 2004. № 6. С. 25—32.

9. **Булаева Н. М., Кудрявцева К. А., Кобзаренко Д. Н., Аскеров С. Я.** Трехмерное моделирование и анализ теплового поля Махачкалинского месторождения термальных вод // "Физика Земли". 2004. № 7. С. 65—70.

УДК 004.942

А. В. Черняев, д-р техн. наук, проф.,
А. А. Павлов, аспирант,
"МАТИ" — Российский государственный технологический университет им. К. Э. Циолковского,
e-mail: PavlovAndrey@list.ru

Моделирование процессов осаждения нефтяных загрязнений на береговую поверхность малых рек

Рассмотрены вопросы построения математической модели и основанной на ней информационной системы прогнозирования и поддержки принятия решений в сфере экологической и техногенной безопасности. Особое внимание уделено учету влияния процессов осаждения нефтепродуктов на растениях и их фильтрации в почву.

Ключевые слова: *разлив нефти, математическое моделирование, береговая поверхность, информационные системы, техногенная безопасность, поддержка принятия решений.*

С увеличением числа аварий при транспортировке нефтепродуктов возникает необходимость в разработке систем информационного обеспечения структур экологической и техногенной безопасности. Это требует комплексного моделирования последствий аварийных разливов [1]. Наименее изученной в данной области является проблема загрязнения береговых поверхностей малых рек [2], что связано с разнообразием влияющих факторов и их сложным взаимодействием.

Существующие в настоящее время модели распространения загрязнения по водотокам основаны на использовании эмпирических формул вида [2]

$$V_p = (1 - \exp(-K_i L))V,$$

где V_p — объем нефти, который будет потерян при прохождении каждого участка реки длиной L; V — объем нефти, попавшей в начало данного участка реки; K_i — интегральный коэффициент, зависящий от ширины реки, сорбционной способности береговой поверхности и физико-химических процессов преобразования нефтяного пятна.

Как видно из приведенной расчетной формулы, в модели не учитываются физико-химические процессы преобразования нефтяного пятна, а также влияние сорбционной способности основных видов береговой растительности и грунтов.

Необходимость построения имитационной математической модели обусловливается невозможностью проведения полномасштабных экспериментальных исследований происходящих процессов. Математические модели, описывающие процессы трансформации нефтяных разливов, необходимы для построения прогноза перемещения нефтяного пятна, правильной реакции на аварийные разливы, оценки воздействия на окружающую среду, планирования чрезвычайных ситуаций и обучения персонала.

Существует несколько процессов, обусловливающих накопление нефтяного загрязнения на береговой поверхности, а именно — процессы фильтрации нефтепродуктов в грунт береговой поверхности и осаждения на произрастающих растениях.

Целью настоящей работы было построить математическую модель, позволяющую оценить объем нефтепродуктов, осевших на береговой поверхности. Достижение поставленной цели сопряжено с решением следующих задач:

- моделирование процесса осаждения нефтепродуктов на береговую растительность;
- моделирование процесса фильтрации нефтяных продуктов в грунт береговой поверхности;
- построение модели информационной системы оценки объемов осаждающейся нефти.

Нами для решения данных задач была разработана математическая модель массопереноса в почвах, а также рассмотрен вопрос осаждения нефтепродуктов на условные типы береговой растительности. Ниже изложены базовые подходы, использованные при разработке модели, а также результаты определения основных характеристик и параметров, необходимых для численной реализации модели.

Моделирование процесса осаждения нефтепродуктов на береговую поверхность

Рассмотрим процесс осаждения нефтепродуктов на береговую растительность. Осаждение нефти на растительность обусловлено соприкосновением береговой растительности с нефтяным пятном. Для определения объема нефти, который может быть осажден на растительности, следует выяснить тип и видовой состав растений, произрастающих на береговой поверхности водоемов средней полосы России. Согласно исследованиям, проведенным в [3], все многообразие береговых растений можно классифицировать следующим образом:

- пояс береговых растений;
- прибрежная растительность мелководий;

- пояс высокорослых воздушно-водных растений;
- пояс водных растений с плавающими листьями;
- зона фитопланктона.

Однако наиболее распространены воздушно-водные (прибрежные, земноводные) растения, погруженные в воду лишь своими основаниями, а большая часть их представлена надземной фитомассой, располагающейся высоко над поверхностью воды. Среди всего многообразия растений, произрастающих в прибрежной зоне, для построения математической модели необходимо отобрать наиболее характерные. Принимая во внимание то обстоятельство, что береговая растительность представлена несколькими ярусами, условно примем, что объем осевшей нефти на любом растении яруса будет мало отличаться от объема осевшей нефти на характерном элементе яруса. Выберем для каждого яруса растений характерного представителя, объем осевшей нефти на котором позволил бы провести оценку объема нефти, осевшей на элементах растительности данного яруса. Так, наиболее характерным растением кустарникового яруса прибрежной зоны является кустарник рода "ива", крупно-травянистого — рогоз широколистный и травянистого — тростник обыкновенный.

После определения характерных видов береговых растений был проведен расчет плотности этих растений, т. е. количества растений на квадратный метр. Методика подсчета количества растений на единице площади представлена в работах [4]. В данном исследовании был использован следующий метод счета растений. Была выделена элементарная площадка — элемент береговой поверхности площадью 0,2 м^2, в границах которого проводился счет растений. Согласно [5], было подсчитано число растений на 10 площадках в каждом из четырех районов Московской области, а именно — в Ленинском, Мытищинском, Дмитровском и Раменском. Результаты измерений представлены в табл. 1.

Следующий этап исследования — получение оценки объема нефти, осевшей на одном растении. Для этого был проведен эксперимент. В ходе эксперимента в емкость, наполненную нефтью плотностью 860 кг/м^3, погружали растения на глубину от 5 до 15 см. При этом измерялась разница массы растения до и после погружения. Значения массы осевшей нефти на единице растения умножались на число растений, произрастающих на 1 м^2. Таким образом, нами была получена оценка массы нефти, осевшей на растительности береговой поверхности. Результаты расчета представлены в табл. 2.

На основании данных, приведенных в табл. 2, был проведен регрессионный анализ. Очевидно, что представленные данные можно описать с помощью линейной зависимости. Уравнения зависимости массы осажденной нефти от глубины погружения условных типов растительности представлены в табл. 3.

Таблица 1

Число растений на береговой поверхности водоемов Московской области

Название растения	Число побегов у растений на 1 м^2
Тростник обыкновенный	68 ± 20
Рогоз широколистный	84 ± 13
Кустарник рода "Ива"	104 ± 11

Таблица 2

Масса осевшей на растениях нефти

Глубина погружения, см	Масса нефти, осевшей на растениях на 1 м^2, г		
	Тип растения		
	Осока	Рогоз	Ива
5	20 ± 4,3	44 ± 12,7	93 ± 34
10	52 ± 5	84 ± 19	186 ± 48
15	90 ± 5	133 ± 32	282 ± 65

Таблица 3

Уравнения зависимости массы осаждающейся нефти от толщины нефтяной пленки

Условный тип растения	Регрессионное уравнение
Осока	$m_о = -23{,}3 + 7{,}6h_н$, где $h_н$ — толщина нефтяной пленки
Ива	$m_и = -9{,}6 + 19{,}5h_н$
Рогоз	$m_р = -13{,}7 + 9{,}8h_н$

Таким образом, была построена математическая модель, позволяющая оценить массу осаждающейся нефти на прибрежные растения различных условных типов.

Объем нефти, осаждающейся на береговой поверхности, можно вычислить по следующей формуле:

$$V_{б\_раст} = \rho_{раст}(BH_{раст})\frac{K_{раст}}{\rho_н},$$

где $\rho_{раст}$ — число растений на единицу площади; B — протяженность нефтяного разлива; $H_{раст}$ — протяженность нефтяного разлива; $K_{раст}$ — нефтеемкость единицы растительного покрова; $\rho_н$ — плотность нефти.

Моделирование процесса фильтрации нефтепродуктов в грунт береговой поверхности

Сущность процесса фильтрации заключается в проникновении нефтепродуктов в почву береговой поверхности. Очевидно, что скорость проникновения нефтепродуктов в толщу грунта зависит от нескольких параметров, а именно: от типа грунта, температуры воды, плотности нефтепродукта и динамической вязкости нефти. Математически данную зависимость отражает следующее соотношение, представленное в работе [6]:

$$V_{\text{грунт}} = K_{\text{н}} BH \frac{h_{\text{н}}}{l} K_{\text{ф}} t,$$

где $K_{\text{н}}$ — нефтеемкость грунта; H — ширина полосы контакта нефтяного пятна с береговой поверхностью; $h_{\text{н}}$ — толщина слоя нефти; l — толщина грунта в направлении фильтрации; $K_{\text{ф}}$ —коэффициент фильтрации; t — время соприкосновения нефтяного пятна с данным участком береговой поверхности.

Схема комплексной модели осаждения нефтяных загрязнений на береговую поверхность

Построение информационной системы

Для реализации описанной выше математической модели необходима довольно разнообразная исходная информация, которую можно получить, во-первых, по данным дешифровки аэрофотоснимков (типы почв и микроландшафтов, сетка линий стекания и т. д.), во вторых, на основе лабораторных и натурных исследований (физика взаимодействия загрязнителя с почвой, фильтрационные свойства почв, тип растительности и т. д.).

Список входных параметров включает в себя: $T_{\text{грунта}}$ — тип грунта; $T_{\text{растит}}$ — условный тип растительности; $P_{\text{загрязн}}$ — геометрические размеры нефтяного загрязнения; $T_{\text{нефти}}$ — физико-химические характеристики нефтяного пятна; $t_{\text{разлива}}$ — продолжительность соприкосновения нефтяного загрязнения с береговой поверхностью.

Предварительно участок береговой поверхности, на котором прогнозируется распространение загрязнителя, разбивается пространственной регулярной сеткой на квадратные ячейки. В поле ячейки задаются тип почвы и условный тип растительного покрова. Программа позволяет рассчитывать динамику увеличения объема нефтяного загрязнения, контур пятна загрязнения и динамику площади аварийного загрязнения. Процессы моделирования и прогнозирования распространения загрязнения выполняются в последовательности, определяемой комплексной моделью (рис. 1):

- на фотоплане или карте водотока указываются: геометрические размеры нефтяного загрязнения, тип почвы и растительности для участков береговой поверхности;
- вводятся параметры расчета, в том числе толщина нефтяного загрязнения, температура воды, воздуха, время прохождения нефтяного загрязнения и др.;
- информация о физических свойствах почвы и береговой растительности в узлах расчетного квадрата поступает на вход описанной выше математической модели;
- рассчитываются динамика концентрации нефтяного загрязнения в почве и объем нефтепродукта, осевшего на прибрежных растениях;
- расчетные характеристики выводятся в виде таблицы и выполняется визуализация пятна загрязнения на фотоплане или карте.

На рис. 2 (см. вторую сторону обложки) показан экран дисплея с рабочим окном программы расчета объема осажденного нефтепродукта. В данном окне представлена информация, необходимая для построения математических моделей осаждения загрязнения, а также показана карта местности с нанесенной информацией об объеме осевшей нефти.

Пример расчета объема осажденной на береговой поверхности нефти

В качестве примера для расчета используем следующие данные:

Протяженность нефтяного разлива, м	1000
Ширина пятна контакта нефтяного разлива с растительностью береговой поверхности, м	0,35
Ширина пятна контакта нефтяного разлива с грунтом берега, м	0,2
Толщина нефтяного пятна, м	0,05
Плотность нефти, кг/м^3	860
Время прохождения нефтяного пятна, ч	0,56
Нефтеемкость грунта (песок (диаметр частиц 0,05...2 мм))	0,06
Коэффициент фильтрации	0,21

Тип береговых растений . . . . . Рогоз
Число растений на 1 м^2 84
Нефтеемкость 1 м^2 береговой растительности, кг 0,044

Оценка объема нефти, осевшей на береговой поверхности,

$$V = 2\left(K_{\text{н}} BH_{\text{берег}} \frac{h_{\text{н}}}{l} K_{\text{ф}} t + \rho_{\text{раст}} (BH_{\text{раст}}) \frac{K_{\text{раст}}}{\rho_{\text{н}}} \right) = 0{,}4 \text{ м}^3.$$

Обсуждение результатов

Анализ соотношений, описывающих процесс осаждения нефтепродуктов на береговую поверхность, показывает, что отсутствует значимая зависимость между объемом осаждения и объемом нефтяного загрязнения. В то же время примечательно проанализировать результаты примера расчета. Как видно, объем осажденного нефтепродукта составил всего 0,4 м^3 на протяжении 1 км реки. Таким образом, можно заключить, что только незначительная часть нефтяного загрязнения остается на береговой поверхности. Приведенные результаты расчетов позволяют предположить, что большая часть нефтяного загрязнения распространяется вниз по течению реки. Повышение точности оценки процессов трансформации нефтяного загрязнения дает возможность строить более точные планы локализации и ликвидации нефтяных разливов.

Выводы

В рамках данной работы была построена математическая модель осаждения нефтяного загрязнения на береговую поверхность водотока. Отличительной особенностью данной модели является совокупное рассмотрение процессов осаждения на растениях прибрежной зоны и фильтрации в почву береговой поверхности. На основе приведенной методики было разработано программное обеспечение, позволяющее выполнять расчеты и осуществлять визуализацию результатов. Полученные данные показывают, что применение данной модели целесообразно в системах поддержки принятия решений подразделений, обеспечивающих экологическую и техногенную безопасность при прогнозировании распространения нефтяного загрязнения по акваториям водотоков. Результаты апробации способствовали тому, что ряд компаний проявили интерес к представленной информационной системе.

Список литературы

1. **Дмитренко А. В., Черняев А. В.** Обобщенная модель системы информационной поддержки процессов мониторинга и анализа рисков аварийных разливов нефти // Информационные технологии моделирования и управления. 2007. № 1. С. 145—150.
2. **Дмитренко А. В., Павлов А. А., Черняев А. В.** Комплексная модель для компьютерного анализа последствий аварийных разливов нефти из трубопроводов // Информационные технологии моделирования и управления. 2007. № 8. С. 970—975.
3. **Неронов В. В.** Полевая практика по геоботанике в средней полосе Европейской России: методологическое пособие. М.: Изд-во Центра охраны дикой природы, 2002. С. 78—83.
4. **Еремеева В. Г., Пирогова Т. И.** Полевое изучение растений и почв: Учебно-метод. пособие. Омск: Изд-во ОмГПУ. 1998. С. 3—19.
5. **Сергиенко Л. А.** Методика изучения прибрежно-водной флоры // Методы полевых и лабораторных исследований растений и растительных сообществ: Сб. ст. / Отв. ред. Е. Ф. Марковская. Петрозаводск: ПетрГУ, 2001. С. 261—264.
6. **Ларионов В. А.** Моделирование аварийных разливов нефти на суше с применением ГИС-технологий: методика. М.: МНТЦ БЭСТС, 2004. С. 11—13.

КОМПЬЮТЕРНАЯ ГРАФИКА

УДК 004.383.5

М. Я. Сафин, инж.-програм., аспирант,
Институт проблем информатики АН Республики Татарстан, г. Казань, e-mail: maratsafin@list.ru

Геометрический процессор для определения видимости, теней и освещенности

Рассматриваются аппаратные способы определения процедуры трассировки лучей, реализованные в геометрическом процессоре. Подробно описываются все шаги процедуры. Рассматриваются схема и элементы, с помощью которых данная процедура может быть реализована аппаратно. Данная статья позволяет понять принцип работы геометрического процессора и его преимущества.

Ключевые слова: *трехмерные объекты, геометрический процессор, модуль трассировки лучей, значение глубины, модуль текстурирования, растровая линия, пиксель, область, поверхность.*

Введение

В настоящей работе рассматриваются аппаратные способы определения процедуры трассировки лучей, которые позволяют распараллелить этот процесс. Концепция построения систолических процессоров, предложенная в работе [2], предполагает, что все объекты, включая затененные и световые объемы, представляются, как множество плоскостей. Определение и способы вычисления параметров затененных и световых объемов приведены в работе [1].

Понятие геометрического процессора

На рис. 1 изображена общая схема устройства определения видимости непересекающихся поверхностей в области экрана, в которой показано место предлагаемого геометрического (систолического) процессора. Предполагается, что экран поделен на конечное число прямоугольных областей, каждая из которых будет обрабатываться на этом процессоре. На первой стадии геометрический процессор преобразовывает координаты всех объектов сцены. Поверхности освещенных и затененных объемов строятся на основании новых координат источников света и трехмерных объектов сцены. Непрозрачные поверхности трехмерных объектов, обращенные внутренней стороной к зрителю, отбрасываются [1]. Для трехмерных объектов с прозрачными частями, а также для затененных и освещенных объемов специальные атрибуты обозначают внутреннюю и внешнюю поверхности.

Принцип работы процессора

Геометрический процессор (см. рис. 1) формирует базу данных сцены, которая хранится в модуле памяти. Эта база данных содержит массив структур, соответствующих поверхностям объектов сцены. Предполагается, что все поверхности представлены треугольниками. Каждая структура треугольника состоит из двенадцати полей (см. таблицу).

Три значения A, B и C определяют глубину D для плоскости треугольника вдоль луча проекции, проходящего через точку (x, y) экрана, следующим образом:

$$D = Ax + By + C. \quad (1)$$

Рассмотрим рис. 2, на котором показана проекция *110* треугольника на экран *112*. Стороны проекции *110* принадлежат прямым линиям *114*, *116* и *118* на экране. Коэффициенты a_1, b_1, a_2, b_2, a_3, b_3 определяют уравнения этих линий на экране следующим образом:

$$x = a_1 y + b_1; \quad (2)$$

$$x = a_2 y + b_2; \quad (3)$$

$$x = a_3 y + b_3. \quad (4)$$

Рис. 1

Поле	Значение
A, B, C	Определяют значение глубины D для плоскости
a_1, b_1, a_2, b_2, a_3, b_3	Коэффициенты в уравнениях прямых линий, соответствующих сторонам проекции треугольника на экране
s_1, s_2, s_3	Определяют относительную позицию проекции треугольника на экране

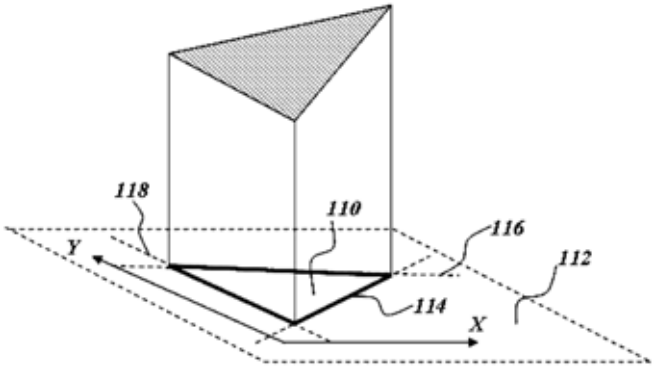


Рис. 2

Далее рассмотрим рис. 3, на котором изображена та же самая проекция *110* треугольника на экран. Три булевы переменные s_1, s_2 и s_3 определяют на какой стороне линий *114*, *116* и *118* расположена проекция треугольника *110* в направлении оси *X*. На рис. 3 треугольник *110* расположен на правой стороне линии *118* и на левой стороне линий *114* и *116* в направлении оси *X*. Та же самая структура связана открытыми поверхностями затененных и световых объемов.

Уравнения (2), (3) и (4) для проекции произвольного треугольника обеспечивают возможность определить, пересекается ли луч проекции, проходящий через точку (x, y) с этой проекцией.

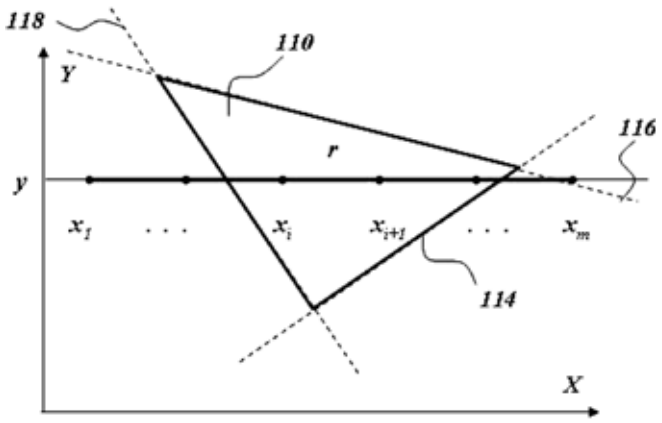


Рис. 3

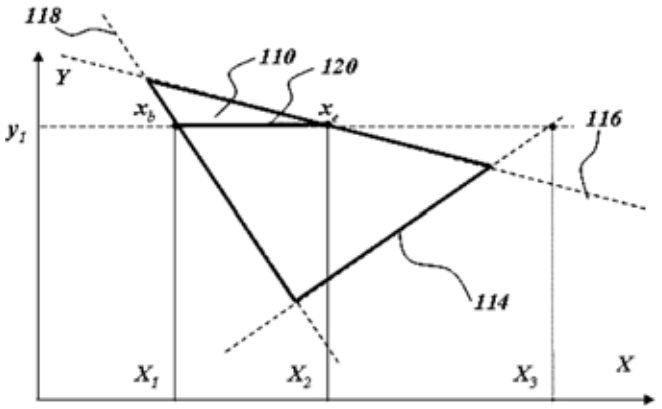


Рис. 4

Значение глубины D для точки пересечения (x, y) рассчитано из уравнения (1).

Точки x_1, ..., x_i, x_{i+1}, ..., x_m на экране имеют равную горизонтальную координату y (см. рис. 3) и расположены на равных расстояниях r друг от друга. Значение глубины D_{i+1} для плоскости треугольника в произвольной точке (x_{i+1}, y) может быть рассчитано из значения глубины D_i в точке (x_i, y) с помощью рекуррентного выражения, которое получено ниже:

$$D_{i+1} = A(x_{i+1}) + By + C = A((x_i) + r) + By + C;$$
$$D_{i+1} = Ax_i + By + C + Ar;$$
$$D_{i+1} = D_i + Ar. \quad (5)$$

Предполагается, что расстояние r является степенью 2.

Для каждой горизонтальной линии $y = y_1$ все точки этой линии, принадлежащей проекции треугольника, расположены в пределах интервала $\{x_b, x_e\}$ (*120* на рис. 4). Для каждой горизонтальной линии $y = y_1$ крайние точки x_b и x_e интервала определяются из уравнений (2)—(4) с помощью следующей процедуры.

1. Определение координаты X точек пересечения линий *114*, *116*, *118* с линией $y = y_1$ и сохранение результатов в переменных X_1, X_2 и X_3 (рис. 4):

$$X_1 := a_1y_1 + b_1; \quad (6)$$

$$X_2 := a_2y_1 + b_2; \quad (7)$$

$$X_3 := a_3y_1 + b_3. \quad (8)$$

2. Выбор линий (*114* и *116*), для которых проекция треугольника расположена на левой стороне.

3. Выбор координат X их пересечения с линией $y = y_1$ (X_3 и X_2).

4. Определение минимальной из этих координат $X_{\min}$ ($X_{\min} := X_2$).

5. Выбор линий (*118*), для которых проекция треугольника расположена на правой стороне.

6. Выбор координат X их пересечения с линией $y = y_1$ (X_1).

7. Определение максимальной из этих координат $X_{\max}$ ($X_{\max} := X_1$).

8. Если $X_{\max}$ меньше, чем $X_{\min}$, то $x_b := X_{\max}$; $x_e := X_{\min}$, в противном случае интервал пустой.

Та же самая процедура используется для того, чтобы определить интервал для поверхностей затененных и освещенных объемов. Обратимся теперь к рис. 5, который показывает проекцию $QUVW$ поверхности затененного или освещенного объема на экране, сегмент UV и лучи UQ и VW, ограничивающие проекцию открытой поверхности затененного или освещенного объема.

В предлагаемом геометрическом процессоре используется следующий набор обрабатывающих элементов. Обрабатывающий элемент *130* (рис. 6) вычисляет выражение $D = Ax + By + C$, где A, B и C — константы. Кроме того, элемент *130* состоит из двух элементов: *132* и *134*, вычисляющего результат выражения $Ax + C$, который может использоваться независимо. Значение r в выражении (5) является степенью 2. Обрабатывающие элементы *136* (рис. 7) используются для расчета

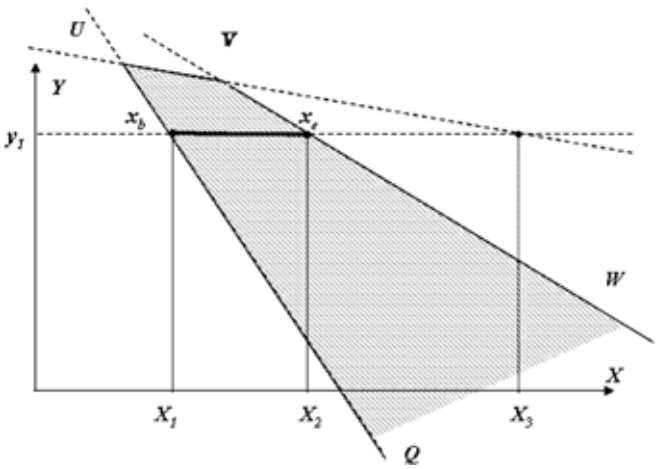


Рис. 5

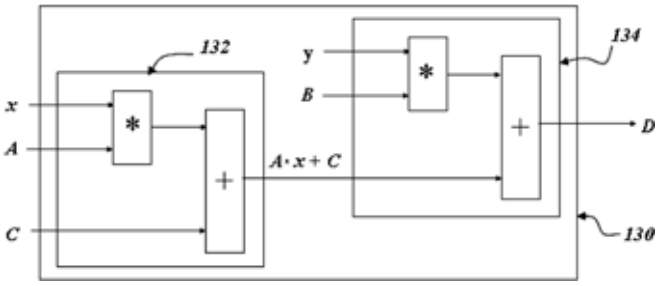


Рис. 6

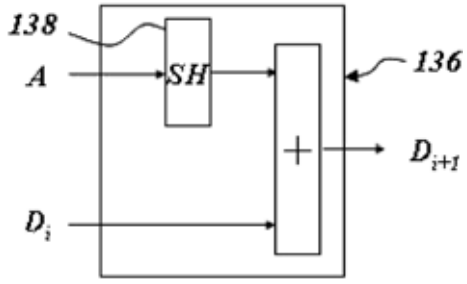


Рис. 7

этого выражения. Сдвигающее устройство *138* используется вместо умножителя в этом обрабатывающем элементе.

Обрабатывающий элемент на рис. 8 определяет интервал $\{x_b, x_e\}$ согласно процедуре, описанной выше.

Выражения (2), (3) и (4) для X_1, X_2 и X_3 — координаты X точек пересечения линий *114*, *116*, *118* с линией $y = y_1$ вычисляются в элементах *132*.

Интервал рассчитывается в вычислительном блоке *140*. Элементы *142* и *144* реализуют шаги 2, 3 и 5, 6 процедуры. Элемент *142* выбирает координаты X точек пересечения с линиями, на левой стороне которых расположена проекция треугольника, элемент *144* — на правой стороне. Значения s_1, s_2 и s_3 используются для того, чтобы определить позицию проекции треугольника относительно границ. Каждый из элементов *142* и *144* имеет только два выхода, потому что произвольная прямая линия пересекается только с двумя сторонами треугольника. Элементы *146* и *148* определяют минимальный (шаг 4) и максимальный (шаг 7) значения вывода. Значения на выходе элементов *146* и *148* — граничные точки интервала x_b и x_e.

Если поверхности объекта представлены произвольными плоскими выпуклыми многоугольниками, тогда вычислительный блок *140* должен быть заменен. Новый блок должен вычислить интервал, используя число точек пересечения, равное числу граней в многоугольнике. Дополнительная информация о гранях многоугольника также появится в структурах, соответствующих поверхностям объектов сцены. Структура для треугольника должна быть расширена за счет добавления пар коэффициентов для уравнений прямых линий, соответствующих дополнительным граням многоугольника, и булевых переменных s_4, ..., s_n, определяющих относительную позицию проекции. Соответствующие изменения очевидны.

Теперь рассмотрим рис. 9, который описывает обрабатывающий элемент *150* для сохранения значения глубины и сравнения. Входной сигнал I имеет значение TRUE, когда элементарную область (пиксель) рассматривают внутри проекции поверхности, и значение FALSE в противном случае; D — значение глубины в текущей элементарной области; N — запись, состоящая из двух полей.

Запись N содержит информацию о текущей обработанной поверхности и будет описана далее в подробностях. Первое поле соответствует идентификации информации о поверхности непосредственно. Второе поле содержит код операции, который инструктирует обрабатывающий блок о типе обработки, которая должна быть выполнена. Различные процедуры используются, чтобы сформировать конечные параметры пикселя для различных типов поверхностей. Значения глубины прозрачных и непрозрачных поверхностей, поверхностей затененных и освещенных объемов используются различным способом для формирования цвета пикселя. Код операции во втором поле записи N управляет надлежащим использованием значения глубины.

Регистры REG_1 и REG_2 могут быть доступными или недоступными для записи входных данных

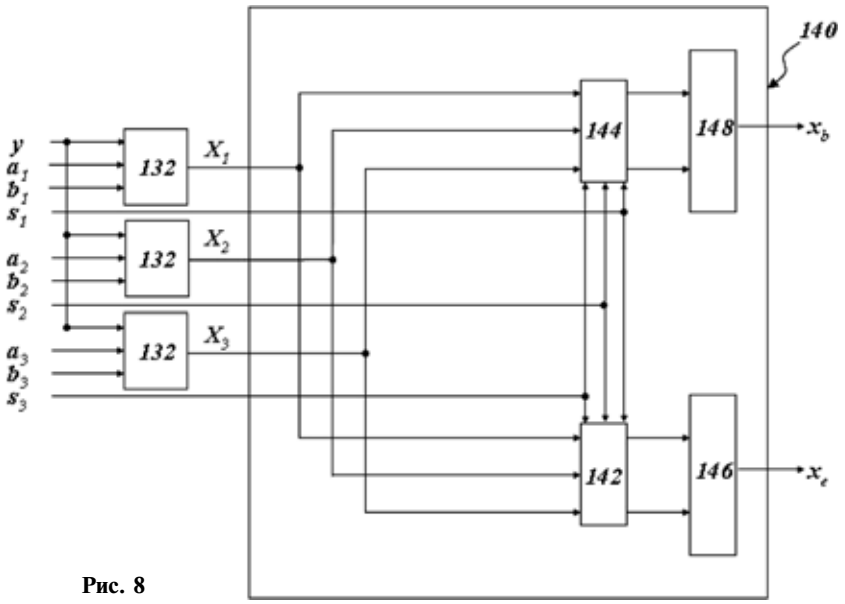


Рис. 8

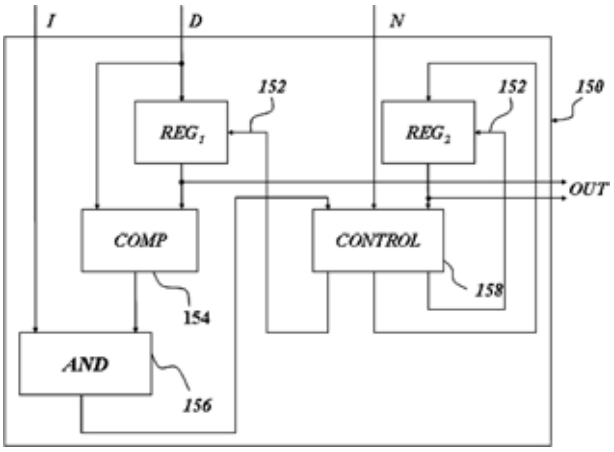


Рис. 9

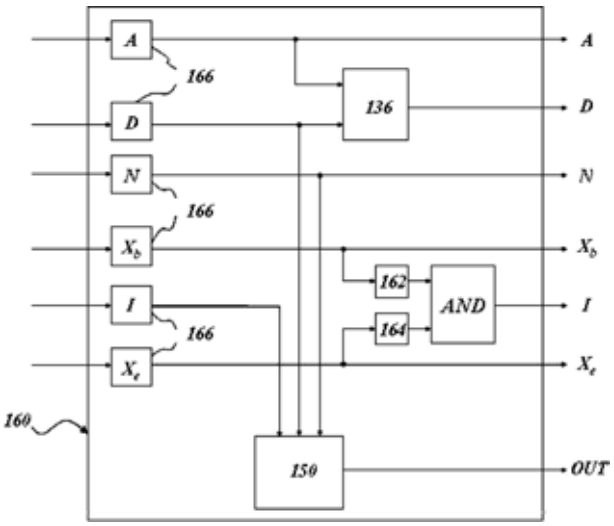


Рис. 10

со входа *152*. Они хранят значение глубины D и информацию записи N. Для непрозрачных треугольников регистр REG_1 хранит значение глубины D самого близкого к экрану треугольника, а REG_2 идентифицирует информацию этого треугольника. Обработку в процессорном элементе, хранение и сравнение значения глубины наиболее просто можно объяснить на примере непрозрачных треугольников. Новое значение глубины D сравнивается в компараторе *154* со значением глубины, уже сохраненным в REG_1. Если новое значение глубины является меньшим, чем глубина, сохраненная в REG_1, и значение I является TRUE, то значение на выходе элемента *156* тоже TRUE. В этом случае элемент *158* допускает запись данных в регистры REG_1 и REG_2. Новое значение глубины D сохраняется в REG_1 и новая идентифицирующая информация о треугольнике сохраняется в REG_2. Для других типов поверхностей процессорный элемент сохраняет идентифицирующие и промежуточные данные о поверхности в REG_2. Эти данные зависят от значения глубины поверхности. Выходы OUT используются для чтения конечной информации о пикселе модулем текстурирования.

Функциональные возможности процессорного элемента *160* (рис. 10) включают функции элемента *150*, описанные выше, и вычисления значения D_{i+1} из D_i с помощью рекуррентного отношения (5) в элементе *136*. Элементы *162* и *164*, а также элемент AND определяют булево значение I. Экранные координаты интервала $\{x_b, x_e\}$ преобразовываются к координатам обрабатываемой прямоугольной области $\{X_b, X_e\}$, чтобы уменьшить число битов в координатном представлении. Элементы *162* и *164* сравниваются со входными значениями X_b и X_e и с собственным числом процессорного элемента *160*. Если число процессорного элемента *160* принадлежит интервалу $\{X_b, X_e\}$ тогда I (выходное значение AND) равно TRUE, в противном случае — FALSE. На рис. 10 элементы *166* (A, D, N, X_b, I, X_e) — промежуточные конвейерные регистры для хранения соответствующих значений.

Последовательность процессорных элементов на рис. 11 состоит из элемента *180*, и множества процессорных элементов *160* $L_1, L_2, ..., L_m$. Эта последовательность вычисляет значение глубины D для отдельной растровой линии, параллельной оси X в m равноотстоящих точках. Значение глубины в точках $x_1, ..., x_i, x_{i+1}, ..., x_m$ (см. рис. 3) рассчитывается одновременно в элементах $L_1, ..., L_i, L_{i+1}, ..., L_m$. Каждый обрабатывающий элемент *160* имеет собственный номер. Этот номер равен номеру пикселя в исходной растровой линии. Элемент L_i в последовательности *170* имеет свой

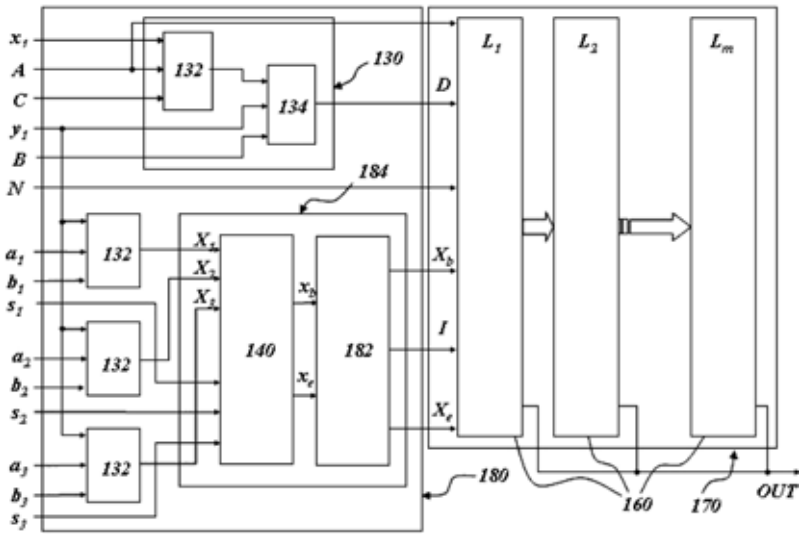


Рис. 11

собственный номер *i*. Элемент *180* на рис. 11 получает на вход полную информацию о текущей обработанной поверхности и координаты (x_1, y_1) начального пикселя растровой линии. Элемент *130* вычисляет значение глубины для плоскости поверхности в начальной точке. Элементы *132* вычисляют точки пересечения горизонтальной растровой линии $y = y_1$ со всеми прямыми линиями, формирующими проекцию поверхности.

Элемент *184* вычисляет приведенные координаты интервала и булево значение *I*, определяющее, принадлежит ли начальная точка этому интервалу. Элемент *140* вычисляет значения интервала x_b и x_e, а элемент *182* преобразует эти координаты к X_b и X_e. Последовательность *170* обрабатывающих элементов *160* вычисляет и хранит значение глубины и идентифицирующую информацию поверхности для каждого пикселя. Результат выполнения вычислительных элементов *160* — значение глубины самого близкого треугольника и его идентифицирующая информация в каждой из точек $x_i, x_{i+1}, ..., x_m$ растровой линии. Выходная шина используется для чтения конечной информации о пикселе модулем текстурирования (см. рис. 1).

Далее представлены последовательности вычислительных элементов, предназначенных для обработки набора горизонтальных растровых линий области. Каждая растровая линия области может быть обработана с помощью последовательности вычислительных элементов, изображенных на рис. 11. Некоторые инициализирующие действия обычны для всех последовательностей и могут быть выполнены только однажды для всех последовательностей. Начальные значения *x* в уравнении (1) равны для всех обработанных растровых линий. Поэтому выражение $Ax + C$, рассчитанное в элементе *132* (см. рис. 6), общее для всех последовательностей. Для каждой последовательной растровой линии области с номером *j*, значение *y* рассчитано следующим образом:

$$y = y_1 + (j - 1)r,$$

как это показано на рис. 12 для $r = 1$. Элемент *136* используется для вычисления *y*.

Значение *y* рассчитано с помощью регистра $j - 1$, в котором сохранен номер растровой линии минус 1 (рис. 13).

На рис. 14 матрица процессора состоит из *k* последовательностей $m + 1$ процессорного элемента *200*. Процессорные элементы *160* с номерами от 1 до *m* показаны на рис. 10. Процессорные элементы *190* с индексом 0 рассматриваются на рис. 13. Процессорный элемент *132* (рис. 14) инициализирует элемент. Для области, состоящей из *kn* пикселей, мат-

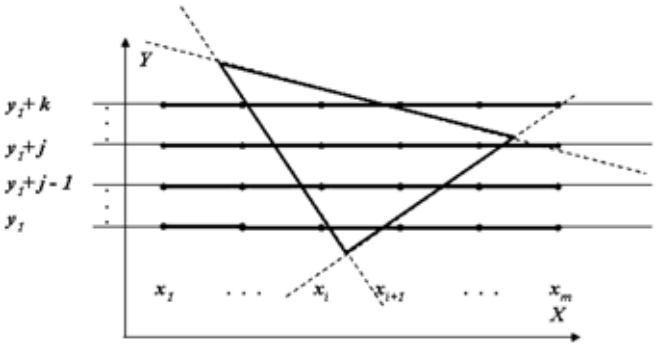


Рис. 12

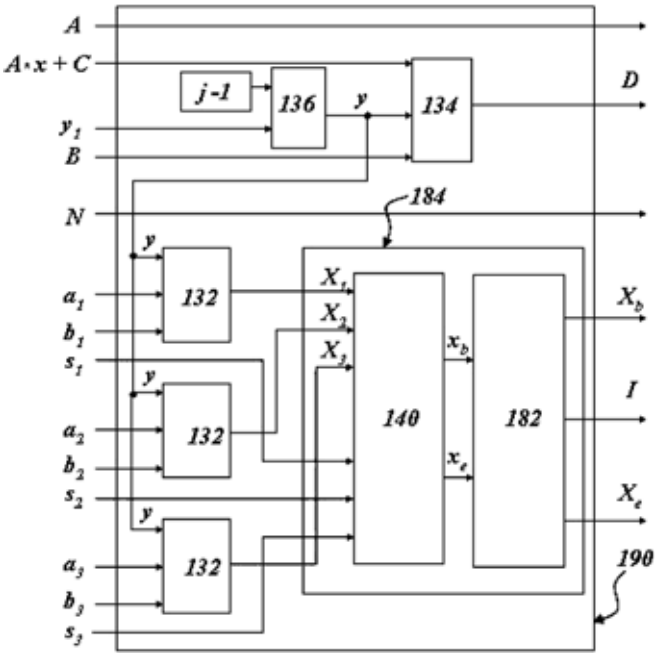


Рис. 13

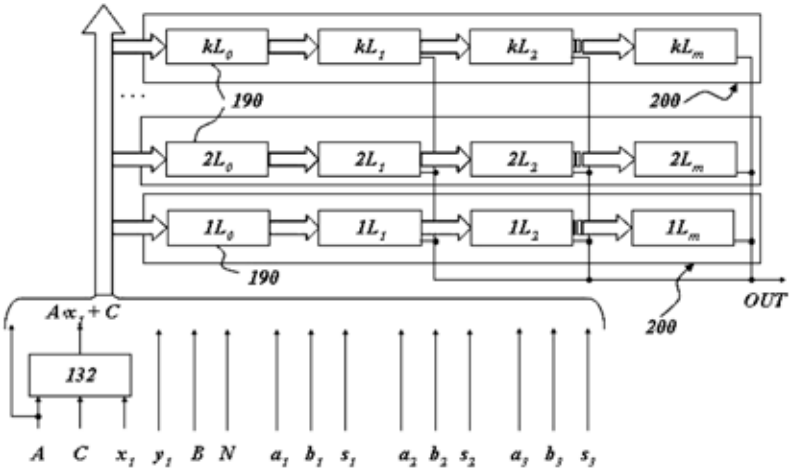


Рис. 14

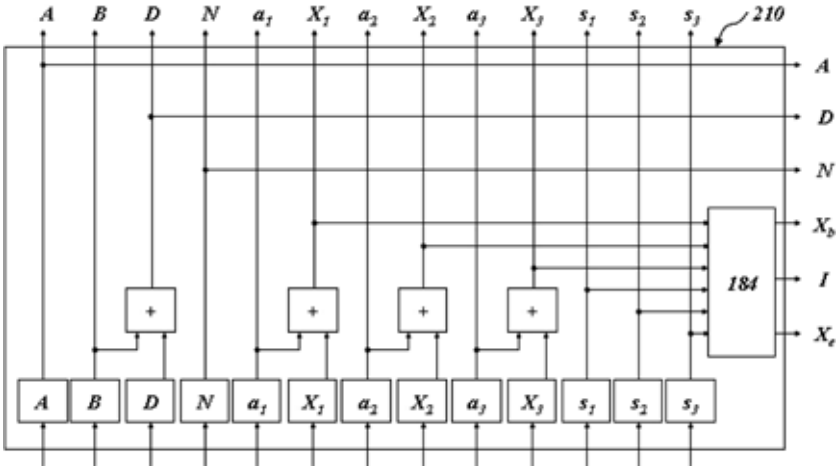


Рис. 15

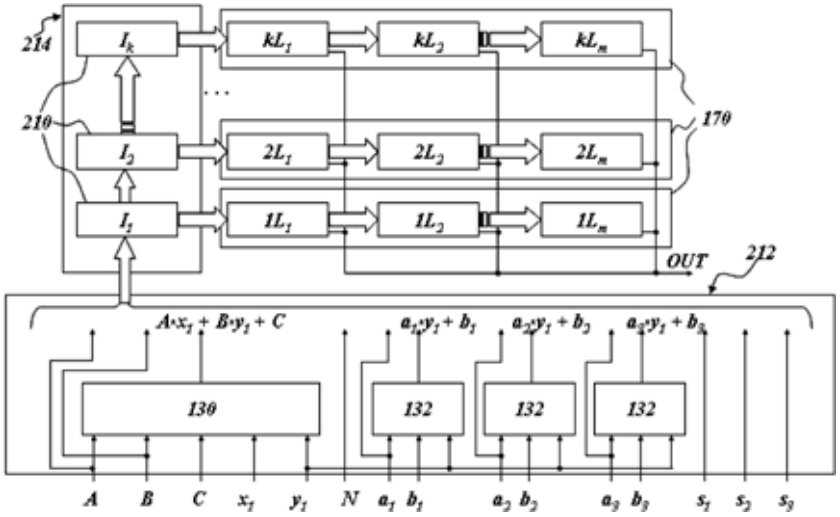


Рис. 16

рица процессора вычисляет значение глубины в каждом пикселе области. Выходная шина используется для чтения конечной информацию о пикселе модулем текстурирования.

Каждый элемент с индексом 0 (*190* на рис. 13) в каждой последовательности содержит четыре множителя. Можно упростить набор этих элементов с помощью инициализирующей последовательности, состоящей из процессорных элементов, один из которых (*210*) изображен на рис. 15.

Элемент *210* содержит только четыре сумматора вместо четырех множителей и тот же самый элемент *184*. Элементы *210* объединены в инициализирующие последовательности *214*, как показано на рис. 16.

Процессорный элемент *212* — стартовый элемент инициализирующей последовательности. Элемент *130* на рис. 16 вычисляет значение глубины в точке (x_1, y_1) в соответствии с уравнением (1). Элементы 132 вычисляют координаты X точек пересечения растровой линии $y = y_1$ с прямыми строками, соответствующими сторонам поверхности. Промежуточный конвейерный регистр N (см. рис. 10) содержит инициализирующую информацию об этой поверхности. Выходная шина используется для чтения конечной информации модулем текстурирования (см. рис. 1). Инициализирующая последовательность *214* загружает инициализирующие данные в каждую последовательность *170* процессорных элементов последовательно, а не одновременно, как это показано на рис. 14.

Заключение

В заключении отметим, что при обработке прямоугольных областей, на которые делится экран, через систолический процессор (см. рис. 10) достаточно пропустить только те треугольники (многоугольники), проекции которых пересекаются с этой областью. Поэтому, предварительно, необходимо с каждой из прямоугольных областей связать множество треугольников (многоугольников), проекции которых имеют не пустое пересечение с этой областью. Для этих целей подходит процедура, предложенная в работе [3]. Эта процедура имеет очень хорошие временные параметры несмотря на простоту описания. Последнее позволяет получить высокоскоростную и простую аппаратную реализацию, которая может быть использована в совокупности с предлагаемым геометрическим процессором.

Применение данного геометрического процессора для определения видимости, теней и освещенности позволит существенно сократить объем вычислений. Большим преимуществом рассматриваемого процессора является возможность его аппаратной реализации в системах с параллельной обработкой информации. Для этого не потребуется специальных модификаций предлагаемого процессора.

Список литературы

1. **Фоули Д. Д. и др.** Введение в компьютерную графику. Издательство Addison-Wesley Publishing Corp., 1997.

2. **Эштон Мартин и др.** Метод трассировки лучей и аппарат для проецирования лучей через объект, представленный рядом бесконечных поверхностей. Патент США 5,729,672.

3. **Сафин М. Я.** Прямой метод локализации объектов в двумерном пространстве // Исследования по информатике. Вып. 12. Казань: Отечество, 2007. С. 124—130.

УДК 004.9:519.226.3

М. А. Кораблин, д-р техн. наук, проф., зав. каф.,
e-mail: ist@psati.ru,
А. А. Салмин, канд. техн. наук, ст. препод.,
e-mail: alx63@bk.ru,
О. И. Бедняк, аспирант, **С. С. Таев,** студент,
Поволжский государственный университет
телекоммуникаций и информатики, г. Самара

Категориальный анализ и оценка поведения клиентов для прогнозирования рыночных отношений

Рассматриваются вопросы оценки поведения клиентов в компании на основе их индивидуальных свойств и качеств. Приводится категориальный анализ свойств клиента на основе байесовского подхода. Представлена система категориального анализа оценки платежеспособности клиента.

Ключевые слова: *категориальный анализ, индивидуальный подход к клиенту, сегментация, свойства клиента, качества клиента, скоринг.*

Введение

Формирование отношений с клиентами на рынке товаров и услуг в настоящее время неразрывно связано с анализом свойств и качеств клиентов, формированием их стереотипов и проверкой гипотез о принадлежности клиента соответствующему сегменту рынка. Такая ситуация связана с самыми разнообразными задачами, такими как: скоринг (в банковском деле), оценка лояльности клиентов (телекоммуникационные компании), анализ причин оттока клиентов из фирмы и т. д. Эти и подобные им задачи характеризуют общую тенденцию формирования отношений с клиентами — анализ их реальных и потенциальных возможностей поведения в системе "клиент — компания".

Такой анализ тесно связан с использованием новых информационных технологий, формирующих оценки гипотез о принадлежности клиента к тому или иному сегменту рынка. В первую очередь, такие технологии связаны с концепциями CRM (Customer Relation Management), Data Mining и т. д. [1, 2]. Онтология отношений в системе "клиент — компания" как альтернатива гносеологии изобилует многочисленными практическими деталями, далеко выходящими за рамки чисто математических моделей. В этом смысле аспекты онтологического знания не имеют четкой формализации, часто опираются на эвристический подход, и, к сожалению, лишены строгой математической основы.

Ниже описываются основы категориального анализа, позволяющего манипулировать информацией о клиенте (гипотезами и фактами) на разных уровнях и на строго математической основе, формируя оценки реального и гипотетического развития событий.

В качестве иллюстрирующего материала используется задача скоринга, который формирует оценки платежеспособности клиента-заемщика банка.

Категориальный анализ отношений в системе "клиент — компания"

Отношения в системе "клиент — компания" рассматриваются с двух точек зрения: индивидуальные (личностные) и массовые (групповые). Индивидуальные отношения характеризуются **свойствами,** которые описывают отдельные черты, присущие клиенту. Например, свойство "отцовство" — клиент имеет детей (является отцом), при этом качество "отцовства" характеризуется числом детей. Такого рода качество имеет несколько **категорий:** многодетный отец, малодетный отец, бездетный отец и т. п.

Свойства клиента могут быть тесно связаны с интересами компании (например, материальное состояние клиента) или представляют второстепенный интерес для компании (является ли клиент "совой" или "жаворонком"). Важность изучения свойств клиента заключается в том, что второстепенные на первый взгляд свойства неожиданно в определенных ситуациях могут стать важнее тех, которые считаются основными. В этом заключаются скрытые связи между свойствами.

Массовые (групповые) отношения определяют совокупность клиентов, обладающих общими свойствами и "близкими" качествами. Такая совокупность обычно рассматривается как сегмент рынка. При этом совокупность свойств и качеств определяет не конкретного клиента, а некоторый **стереотип,** который может формироваться менед-

Таблица 1

Свойства и категории качества

Свойства	Категории качества		
Возраст	Молодой	Пожилой	Старый
Образование	Начальное	Среднее	Высшее
Пол	М	Ж	—
Район проживания	Октябрьский	Ленинский	Промышленный

Таблица 2

Индивидуальная характеристика клиента

Свойства	Категории качества		
Возраст	Молодой	Пожилой	Старый
Образование	Начальное	Среднее	Высшее
Пол	М	Ж	—
Район проживания	Октябрьский	Ленинский	Промышленный

жером из собственных соображений полезности для компании.

В любом случае основу категориального анализа клиентов определяет таблица свойств и категорий качества (табл. 1).

Индивидуальная характеристика клиента (ИХК) формируется путем подчеркивания нужных или вычеркивания ненужных категорий в табл. 1. Такая таблица может выглядеть как анкета, вопросник и т. д.

В табл. 2 характеризуется молодой мужчина со средним образованием, проживающий в Промышленном районе:

ИХК = (Возраст = молодой) &
& (Образование = среднее) & (Пол = м) &
& (Район проживания = Промышленный).

Здесь ИХК представляет собой набор из четырех фактов, связанных конъюнктивной связью (&).

Любая строка таблицы свойств и качеств может быть расширена путем введения более "тонких" категорий или сокращена введением более "грубых". Например, детализация категорий:

Возраст = (10 ÷ 30 лет) OR (30 ÷ 40 лет) OR
(40 ÷ 60 лет) OR (> 60 лет).

Здесь "OR" — логическая дизъюнкция.

Обратный пример (интеграции категорий):

Возраст = [(Не старый = (молодой OR пожилой)) OR (старый)].

Если ИХК всегда ассоциируется с конкретным клиентом, то стереотип (и соответствующий сегмент) идентифицируется групповым именем, например: Пенсионер с достатком = [Образование = = (Высшее OR Среднее) & Возраст = (Пожилой OR Старый) & (Доход > 10 000)].

Такая двойственность в интерпретации таблицы свойств и категорий позволяет, с одной стороны, идентифицировать "место" конкретного клиента в процессе сегментации, а с другой стороны, формировать тот стереотип, качества которого позволяют реализовать сегмент с хорошим наполнением. Таким образом, индивидуальность и массовость в отношениях с клиентами следует рассматривать как своеобразное "единство и борьбу противоположностей" в бизнесе.

Для менеджера, которого интересует наполнение сегмента и, соответственно, его финансовые показатели, в конечном счете, важны те свойства и качества потенциальных клиентов, которые определяют такое наполнение.

Структура табл. 1 в программной реализации обладает большой гибкостью. В процессе эксперимента по оценке поведения клиентов можно добавлять новые свойства, интегрировать категории (столбцы), детализировать их, добавляя новые столбцы и т. п.

Для дальнейшей формализации введем таблицу свойств и качеств с использованием буквенно-индексных обозначений (табл. 3).

Верхний индекс в записи K_j^i идентифицирует свойство, нижний — категорию качества, присущую клиенту (или сегменту), подчеркивание определяет выбранную категорию соответствующего свойства.

Категории в строке альтернативны (несовместные события), т. е. в каждой строке может быть выбрана только одна категория качества. Выбор такой категории и включение ее в структуру ИХК рассматривается как **достоверный факт,** связанный с конкретным клиентом [3]. Категории по столбцам конъюнктивны (совместные события — факты). Совокупность таких выбранных категорий по всем строкам таблицы определяет ИХК в целом. Например, для табл. 3

$$\text{ИХК} = (K_1^1 \ \& \ K_2^2 \ \& \ \ldots \ \& \ K_m^N).$$

Символ "K" формально определяет числовое или лингвистическое значение категории, это может быть терм (например, "молодой" в свойстве Возраст) или числовое ограничение (например, (10 <) & (< 30) в том же свойстве).

Таблица 3

Свойства и категории качеств

Свойства	Категории					
	1	2	...	m	...	M
1	K_1^1	K_2^1	...	K_m^1	...	K_M^1
2	K_1^2	K_2^2	...	K_m^2	...	K_M^2
...	...	...	...	...	...	...
N	K_1^N	K_2^N	...	K_m^N	...	K_M^N

Рис. 1. Общая схема категориального анализа на основе байесовского пересчета вероятностей

Выдвижение и оценка гипотез о том, попадает ли конкретный клиент в тот или иной сегмент рынка, связана с задачей классификации, при этом количественное значение такой оценки определяется вероятностной мерой. В описываемом подходе используется байесовская вероятность [4, 5, 6], которая интерпретируется как оценка влияния свойств и категорий клиента на его "поведение" в процессе сегментации.

Основная схема описываемого подхода связана с переопределением байесовских вероятностей на основе получения фактов, определяющих ИХК: "априорная ситуация" → "констатация фактов" → → "апостериорная ситуация" → "новая априорная ситуация".

В практических реализациях эта схема приводит к устойчивой оценке гипотез [5] и практически снимает проблему априоризма и субъективизма.

Общая схема категориального анализа на основе байесовского пересчета вероятностей иллюстрируется рис. 1.

В качестве исходных данных рассматриваются:

- индивидуальная характеристика клиента (ИХК);
- общая посегментная статистика клиентов (или априорные вероятности попадания клиента в сегменты $P(S_k)$;
- посегментная статистика категорий клиентов (или условные вероятности принадлежности клиента к категориям K_j^i, составляющим ИХК, для всех сегментов S_k, $k = 1, 2, ..., K$).

Пересчет апостериорных вероятностей реализуется по формуле Байеса [7]:

$$P(S_k|\text{ИХК}) = \frac{P(\text{ИХК}|S_k)P(S_k)}{\sum\limits_{k=1}^{K} P(\text{ИХК}|S_k)P(S_k)}, \qquad (1)$$

где $P(S_k)$ — априорные вероятности попадания клиента (тяготения) в сегмент S_k; $P(S_k|\text{ИХК})$ — условные вероятности тяготения к сегменту S_k при условии, что ИХК является фактом; $P(\text{ИХК}|S_k)$ — обратная вероятность (вероятность наличия фактов ИХК в сегменте S_k).

Схема рис. 1 интерпретируется в условиях тривиальных ограничений:

1) $\sum\limits_{k=1}^{K} P(S_k) = 1$;

2) для всех S_k и для любого i $\sum\limits_{j=1}^{M} P(K_j^i|S_k) = 1$;

3) если ИХК = K_1^1 & K_m^2 & K_k^3 & ... & K_r^N; то для любого S_k:

$$P(\text{ИХК}|S_k) = P(\underline{K_1^1}|S_k) \cdot P(\underline{K_m^2}|S_k) \cdot P(\underline{K_k^3}|S_k) \cdot ... \cdot ... \cdot P(\underline{K_r^N}|S_k);$$

4) обновление для всех k:

$$P(S_k) := P(S_k|\text{ИХК}).$$

Последний оператор обновляет априорные вероятности сегментов, заменяя их апостериорными, полученными на основе фактов, заключенных в ИХК.

Итерационный процесс переопределения байесовских вероятностей может развиваться различными путями: итерациями по свойствам, итерациями по новым клиентам, итерациями по группам клиентов. На этой основе формируется устойчивое тяготение клиента в системе сегментов.

Основой категориального анализа является общая и посегментная статистика клиентов, дифференцированная по всем свойствам и категориям. Наличие такой статистики определяет достоверность оценки проверяемых гипотез на основе наличия фактов, свойственных клиентам как по отдельности, так и в совокупности. Как правило, такая статистика базируется на предыстории деятельности компании, на прецедентах, характерных для различных ситуаций. Все перечисленные выше вероятности определяются как частости (отношения числа клиентов, обладающих присущими свойствами и категориями, к общему числу клиентов в сегменте).

Отсутствие такой информации можно заменить только субъективными умозаключениями о взаимовлияниях свойств клиентов на процессы сегментации. Этот подход фактически реализует концепцию "What if", свойственную многим системам поддержки принятия управленческих решений.

Формирование оценки платежеспособности клиента-заемщика банка

Скоринг, как процесс выдачи и возврата кредитов, наиболее ярко подчеркивает достоинства и особенности критериального анализа. В простейшем случае в этом процессе можно рассматривать только два сегмента — "неплательщики" и "надежные заемщики". Реальные неплательщики, попавшие в историю со всеми их атрибутами, свойст-

Таблица 4

Пример свойств и категорий

Свойства	Категории					
	1	2	3	4	5	6
1. Должность	ИП, руководитель	Специалист				
2. Стаж общий	Более 5 лет	От 3 до 5 лет	От 1 до 3 лет			
3. Стаж после выдачи кредита	Более 1 года	От 4 месяцев до 1 года	Менее 4 месяцев			
4. Образование	Уч. степень, 2 высших образования	Высшее	Незак. высшее	Среднеспец.	Среднее	
5. Иждивенцы	Нет	1	2	3	4	5 и более
6. Возраст	От 22 до 25	От 26 до 45	От 46 до 65	—		
7. Кредитная история	Полож. (>2/3)	Полож. (<2/3)	Нет	Удовлетв.	Отриц.	
8. Пол	М	Ж				
9. Семейное положение	Замужем (женат)	Не замужем (не женат)				

вами и качествами образуют реальную базу данных. В этих данных скрыты закономерности отношений, которые можно рассматривать как своеобразные шаблоны для оценки свойств потенциального заемщика на предмет его тяготения к одному из двух упомянутых сегментов.

В табл. 4 представлены свойства и категории для описываемого примера.

Посегментная статистика категорий выстраивалась на основе кредитных историй, включающих в себя 500 анкет (индивидуальных характеристик клиентов). Такая статистика содержит вероятности (частости) всех категорий клиентов $P(K_j^i|S_k)$ для каждого из двух сегментов. Структура таблицы, содержащей посегментную статистику $P(K_j^i|S_k)$, аналогична табл. 4 с той разницей, что вместо спецификаций категорий K_j^i в соответствующую клетку данной таблицы вписывается соответствующая вероятность.

Рис. 2 (см. третью сторону обложки) иллюстрирует формирование ИХК и результаты расчета байесовской вероятности, формирующей оценку испытания клиента (в нашем случае это правдоподобие принадлежности клиента сегменту); рис. 2 также связан с разрабатываемой системой категориального анализа.

Заключение

Наиболее интересным является не собственно результат испытания, а динамика изменения правдоподобия в процессе анализа влияния той или иной категории клиента на этот результат. При этом удается выделить: **наиболее информативные категории,** резко меняющие общую оценку принадлежности клиента; второстепенные категории, которые могут быть изъяты из анкеты без снижения правдоподобия и т. п. Большое значение для категориального анализа имеют также взаимозависимости между свойствами, такая корреляция может констатировать избыточность свойств в системе "свойства — категории".

В конечном счете, описываемый инструмент, основанный на весьма простом анализе фактов — категорий, присущих заемщикам, может быть успешно использован не только для формирования оценок надежности заемщиков, но и для выявления главных факторов, влияющих на эту надежность.

Специфика использования байесовской вероятности для оценок правдоподобия соответствующего поведения клиентов отличается от других подходов тем, что результатом является не только возможность классификации клиентов, но и формирование категориальных траекторий, которые определяют эту классификацию.

Список литературы

1. **Кораблин М. А., Пальмов С. В.** Сравнение прогностических возможностей алгоритмов поддержки принятия решений при определении лояльности клиента оператора сотовой связи // Мобильные системы. 2005. № 8. С. 32—35.

2. **Кораблин М. А., Пальмов С. В.** Применение технологии Data Mining для прогнозирования лояльности клиентов // Материалы V Международной научно-технической конференции "Проблемы техники и технологии телекоммуникаций". — Самара: ПГАТИ, 2004. С. 76—78.

3. **Бернштейн С. Н**. Теория вероятностей. Изд. второе доп. — М.: Л.: 1934. 200 с.

4. **Гнеденко Б. В., Хинчин А. Я.** Элементарное введение в теорию вероятностей. М.: Наука, 1976. 168 с.

5. **Секей Г.** Парадоксы в теории вероятностей и математической статистике: пер. с англ. М.: Мир, 1990. 240 с.

6. **Байесовские** процедуры классификации: Вводный обзор. [Электронный ресурс]. — URL: http://www.spc-consulting.ru/DMS/Machine %20 Learning/ Machine Learning/Overvie, свободный.

7. **Кораблин М. А., Мелик-Шахназаров А. В., Салмин А. А.** Оценка лояльности клиентов телекоммуникационной компании на основе байесовского подхода // Информационные технологии. 2006. № 4. С. 63—67.

УДК 330.43

Р. Р. Рзаев, канд. физ.-мат. наук, доц.,
вед. науч. сотр.,
Институт кибернетики НАН Азербайджана,
г. Баку,
e-mail: raminrza@yahoo.com,
Э. Р. Алиев, канд. техн. наук,
президент компании, SINAM Ltd (ICT Company),
e-mail: elchin@sinam.net

Оценка профессиональных качеств служащих компании методом нечеткого логического вывода

Рассматривается задача оперативной оценки работы персонала в компаниях. Соответствующая информационная база данных формируется и периодически обновляется на основе предлагаемой схемы оценки профессиональных качеств. В результате применения метода нечеткого логического вывода синтезируются электронные "портреты" сотрудников компании и устанавливаются тренды их профессионального роста за отчетные периоды.

Ключевые слова: *лингвистическая переменная, нечеткое множество, нечеткая импликация, нечеткое отношение.*

Введение

Периодическая оценка эффективности работы персонала завоевывает в современных компаниях все большую популярность. Сегодня это мощный инструмент кадровой политики. Традиционная методология ее проведения довольно проста: собрать информацию о каждом сотруднике и, обработав ее, получить кадровый "портрет" компании на данном этапе. Оценка деятельности персонала, как постоянная и регулярная процедура, служит интересам компании и характеризуется постоянным наблюдением за трудовым процессом в целом и работой каждого сотрудника в отдельности. Ежедневный контроль и "работа над ошибками" являются самым простым и надежным способом оценки эффективности персональной деятельности. Результат оценки — это установление соответствия конкретного сотрудника занимаемому им рабочему месту.

Ввиду того, что современный потребительский рынок предельно жесткий и конкурентный, каждой компании необходимо регулярно "инвентаризировать" свои кадровые ресурсы на соответствие динамики развития персонала динамике роста компании. Прежде всего необходимо уточнить, как при текущей динамике развития компании меняется рабочее место и что тогда происходит с сотрудником:

- адекватен ли он или не успевает за развитием компании?
- соответствует ли его зарплата тому, что реально показала оценка его деятельности?

Как правило, некоторые сотрудники не успевают за развитием компании. В результате происходит естественный отбор и ротация кадров.

На сегодняшний день процедура оценки и управления эффективностью деятельности является одной из самых проблемных в большинстве компаний, которые ее используют. Каждую компанию интересует: исполняется ли принятая стратегия сейчас и будет ли она выполняться далее. Ее руководству необходимы гарантии в том, что каждый сотрудник на своем месте выполнит стоящие перед ним цели и задачи в полном объеме и в строгом соответствии с принятыми планами. Поэтому для компании очень важно иметь адекватные информационные "портреты" своих сотрудников, которые предлагается формировать на основе оперативной (автоматизированной) обработки текущей персональной информации, упорядоченной по методике [1].

Методика оценки текущей работы сотрудников компании

Оценке подлежат:

- эффективность выполнения должностных обязанностей;
- текущий уровень профессиональных знаний и навыков;
- профессионально важные качества;
- дисциплинированность;
- лояльность и деловой внешний вид сотрудников компании.

В свою очередь, каждый из перечисленных параметров складывается в результате обработки соответствующих показателей. После этого полученная информация упорядочивается в виде табл. 1.

Своевременность выполнения должностных обязанностей сотрудником оценивается начальником их отдела путем установления процента выполнения незапланированных поручений в срок и процента выполнения плана работ. Оценка их эффективности, а также профессионально-важных качеств, профессиональной компетентности, дисциплинированности и прочих показателей сотрудников проводится начальником подразделения, директорами по направлениям, начальниками других подразделений. Показатели

Таблица 1

Оценки текущей работы сотрудников компании

Критерии оценки			Числовая оценка на [0; 5]
1. 1.1.	**Эффективность выполнения должностных обязанностей** Своевременность		
1.1.1. 1.1.2.	Соблюдение сроков выполнения незапланированных поручений Соблюдение плана работ на месяц	% выполнения незапланированных поручений в срок (средний показатель за 1 мес.)/20 % выполнения плана работ на месяц в рамках подразделения (средний показатель за 1 мес.)/20	
1.1.3.	Работник всегда в срок выполняет работу, при необходимости — досрочно		
Итого:			
1.2. 1.2.1.	Качество/полнота Работник качественно и ответственно выполняет свои обязанности. Выполняет обязанности с выраженным умением работать на конечный результат		
1.2.2.	Соблюдение требований к работе	Число нарушений (средняя оценка за аудит)	
Итого:			
2. 2.1. 2.2. 2.3.	**Уровень профессиональных знаний и навыков** Способен заменить любого коллегу в своем подразделении Достаточно профессиональных знаний и навыков для выполнения должностных обязанностей Я доверяю мнению, оценке, профессионализму сотрудника: его оценки взвешенны, обоснованны и аргументированны		
Итого:			
3. 3.1. 3.2. 3.3. 3.4.	**Профессионально важные качества** Мне легко и удобно с ним работать: мы говорим на одном "языке", мне не приходится повторять и объяснять по несколько раз (понятливость) Инициативность: инициирует решение проблем, предлагает несколько вариантов решения, проявляет инициативу в разработке и внедрении нового Речь: хорошо развитая речь, четкая, правильная, без слов "паразитов" Коммуникативные навыки: открыт для общения, умеет доступно, понятно и логично изложить свою точку зрения, способен найти контакт с любым сотрудником		
Итого:			
4. 4.1. 4.2. 4.3. 4.4.	**Дисциплинированность** Работник приходит на работу вовремя Работник не уходит с работы раньше времени Работник не отлучается с работы Работник не нарушает режим обеденного перерыва		
Итого:			
5. 5.1. 6. 6.1.	**Лояльность** Принимает и разделяет ценности компании, приверженность интересам фирмы имеет для него большое значение **Деловой внешний вид** Преобладает деловой стиль в одежде, опрятен		

фиксируются в ежемесячных отчетах по образцу табл. 2.

Ежемесячная оценка *качества* выполнения работником своих обязанностей осуществляется путем проведения внутреннего аудита начальником подразделения или аудиторами по системе качества. Оценки за соблюдение требований к работе могут выставляться по принципу:

- 1 балл присваивается, если число нарушений разных элементов больше 4 или одного элемента больше 7;
- 2 балла — если число нарушений разных элементов 3, 4 или одного элемента 5—7;
- 3 балла — число нарушений разных элементов 2 или одного элемента 3—4;
- 4 балла — число нарушений одного элемента не более 2;
- 5 баллов — нарушений нет.

Количественная оценка профессиональных качеств сотрудников компании методом нечеткого логического вывода

Оценка работы персонала компании, по существу, является многокритериальной процедурой, подразумевающей применение композиционного правила агрегирования оценки каждого сотрудника на

основе информации о предпочтениях лиц, ответственных за ее выполнение. Как правило, эти предпочтения формулируются в виде нечетких суждений. Поэтому для реализации данной задачи целесообразно использовать правила нечеткого вывода.

Рассмотрим задачу точечной оценки альтернатив в нечеткой информационной среде рассуждений. Для ее реализации воспользуемся методом нечеткого вывода, сущность которого состоит в следующем [2].

Пусть U является множеством альтернатив, а $\tilde{A}$ — его нечетким подмножеством, принадлежность к которому элементов из U определяется соответствующими значениями из [0,1] функции принадлежности. Предположим, что нечеткие множества $\tilde{A}_j$ описывают возможные значения (термы) лингвистической переменной x. Тогда множество решений (альтернатив) можно характеризовать совокупностью критериев — значениями лингвистических переменных $x_1, x_2, ..., x_p$, например, в нашем случае значением "ЭФФЕКТИВНО" лингвистической переменной $x_1 =$ $=$ *Исполнение текущей работы* или термом "НЕОБХОДИМОЕ" лингвистической переменной

Таблица 2

Пятибалльная система оценивания служащих компании

Критерий	5 — Да, всегда	4 — Да, в большинстве случаев	3 — в 50/50 % случаев	2 — Нет, в большинстве случаев	1 — Нет, практически всегда
Выполнение должностных обязанностей					
Качество/полнота: работник качественно и ответственно выполняет свои обязанности. Выполняет обязанности с выраженным умением работать на конечный результат					
Своевременность: работник всегда в срок выполняет работу, а в случае необходимости — досрочно					
Уровень профессиональных знаний, навыков					
Способен заменить любого коллегу в своем подразделении					
Достаточно профессиональных знаний и навыков для выполнения должностных обязанностей					
Я доверяю профессионализму работника: его оценки взвешенны, обоснованны и аргументированны					
Профессионально важные качества					
Мне легко и удобно с ним работать: мы говорим на одном "языке", мне не приходится повторять и объяснять по несколько раз (понятливость)					
Инициативность: инициирует решение проблем, предлагает несколько вариантов решения, проявляет инициативу в разработке и внедрении нового					
Речь: хорошо развитая речь, четкая, правильная, без слов "паразитов"					
Коммуникативные навыки: открыт для общения, умеет доступно, понятно и логично изложить свою точку зрения, способен найти контакт с любым сотрудником					
Дисциплинированность					
Сотрудник приходит на работу вовремя					
Сотрудник не уходит с работы раньше времени					
Сотрудник не отлучается с работы					
Сотрудник не нарушает режим обеденного перерыва					
Прочие показатели					
Лояльность: принимает и разделяет ценности компании, приверженность интересам фирмы для него имеет большое значение					
Деловой внешний вид: (деловой стиль в одежде, опрятность)					

x_2 = *Уровень профессиональных знаний*. Совокупность лингвистических переменных (критериев), принимающих подобные значения, могут характеризовать представления ответственного за аттестацию лица о достаточности рассматриваемых альтернатив. Тогда, полагая S = *достаточность* также лингвистической переменной, типовое импликативное правило может выглядеть следующим образом:

e_1: "Если x_1 = НИЗКОЕ и x_2 = ХОРОШЕЕ, тогда S = ВЫСОКАЯ".

В общем виде импликативные рассуждения ответственного за оценку персонала можно представить в следующем виде:

$$e_1\text{: "Если } x_1 = \tilde{A}_{1i} \text{ и } x_2 = \tilde{A}_{2i} \text{ и ... и } x_p = \tilde{A}_{pi}, \text{ тогда } S = \tilde{B}_i\text{".} \quad (1)$$

Далее обозначим пересечение $x_1 = \tilde{A}_{1i} \cap x_2 = \tilde{A}_{2i} \cap ... \cap x_p = \tilde{A}_{pi}$ в виде $x = \tilde{A}_i$. Как известно, в дискретном случае операция пересечения нечетких множеств определяется как

$$\mu_{\tilde{A}_i}(v) = \min_{v \in V}(\mu_{\tilde{A}_{1i}}(u_1), \mu_{\tilde{A}_{2i}}(u_2), ..., \mu_{\tilde{A}_{pi}}(u_p)), \quad (2)$$

где $V = U_1 \times U_2 \times ... \times U_p$; $v = (u_1, u_2, ..., u_p)$; $\mu_{\tilde{A}_{ji}}(u_j)$ — степень (функция) принадлежности элемента u_j нечеткому множеству $\tilde{A}_{ji}$. Тогда высказывания (1) можно представить в более компактном виде:

$$e_1\text{: "Если } x = \tilde{A}_i\text{, тогда } S = \tilde{B}_i\text{".} \quad (3)$$

В целях обобщения обозначенных высказываний представим базовые множества U и V в виде множества W. Тогда $\tilde{A}_i$ соответственно будет нечетким подмножеством базового множества W, а $\tilde{B}_i$ — нечетким подмножеством единичного интервала $I = [0, 1]$.

Для реализации правил используется операция импликации. В принятых обозначениях выберем импликацию Лукасевича:

$$\mu_{\tilde{H}}(w, i) = \min_{w \in W}(1, 1 - \mu_{\tilde{A}}(w) + \mu_{\tilde{B}}(i)), \quad (4)$$

где $\tilde{H}$ — нечеткое подмножество на $W \times I$; $w \in W$ и $i \in I$. Тогда правила $e_1, e_2, ..., e_q$ транспонируются в соответствующие нечеткие множества $\tilde{H}_1$, $\tilde{H}_2$, ..., $\tilde{H}_q$, на пересечении которых $\tilde{D} = \tilde{H}_1 \cap \cap \tilde{H}_2 \cap ... \cap \tilde{H}_q$ для каждой пары $(w, i) \in W \times I$ получим $\mu_{\tilde{D}}(w, i) = \min_{w \in W}(\mu_{\tilde{H}_j}(w, i)), j = \overline{1, q}$. В этом случае вывод об удовлетворительности альтернативы, описанной нечетким множеством $\tilde{A}$ из W, можно определить через композиционное правило $\tilde{G} = \tilde{A} \circ \tilde{D}$, где $\tilde{G}$ является нечетким подмножеством единичного интервала I с функцией принадлежности $\mu_{\tilde{G}}(i) = \max_{w \in W}(\min \mu_{\tilde{A}}(w), \mu_{\tilde{D}}(w, i))$.

Сравнение альтернатив осуществляется на основе их точечных оценок. Вначале для нечеткого подмножества $\tilde{C} \subset I$ определяются α-уровневые множества ($\alpha \in [0,1]$) в виде $C_\alpha = \{i | \mu_{\tilde{C}}(i) \geqslant \alpha, i \in I\}$. Затем для каждого из них находятся средние значения соответствующих элементов $M(C_\alpha)$. В общем случае для множества, состоящего из n элементов, $M(C_\alpha) = \sum_{j=1}^{n} \frac{i_j}{n}$; $i \in C_\alpha$. В итоге точечную оценку альтернативы $\tilde{C}$ можно получить как

$$F(\tilde{C}) = \frac{1}{\alpha_{\max}} \int_0^{\alpha_{\max}} M(C_\alpha) d\alpha, \quad (5)$$

где $\alpha_{\max}$ — максимальное значение на $\tilde{C}$.

Теперь предположим, что руководству компании необходима оценка деятельности своих сотрудников по имеющейся текущей информации, причем за основу оно приняло следующие высказывания:

e_1: "Если работник выполняет свои обязанности, имеет необходимый уровень профессиональных знаний и навыков, а также является дисциплинированным, то он — удовлетворяющий";

e_2: "Если работник к тому же обладает профессионально важными качествами, то он более чем удовлетворяющий";

e_3: "Если работник дополнительно к условиям e_2 проявляет лояльность и имеет деловой внешний вид, то он — безупречный";

e_4: "Если работник имеет все оговоренное в e_3, кроме профессионально важных качеств, то он очень удовлетворяющий";

e_5: "Если работник выполняет свои должностные обязанности, имеет необходимый уровень профессиональных знаний и навыков, обладает профессионально важными качествами, но в то же время не является дисциплинированным, то он все-таки будет удовлетворяющим";

e_6: "Если работник не выполняет свои обязанности и не обладает профессионально важными качествами, то он — неудовлетворяющий".

Анализ приведенных высказываний позволяет выявить по отношению к предлагаемой модели шесть экзогенных критериев, используемых для оценки работников: X_1 — выполнение должностных обязанностей; X_2 —уровень профессиональных знаний и навыков; X_3 — профессионально важные качества; X_4 — дисциплинированность;

Таблица 3

Оценка базовых показателей сотрудников

Условное обозначение сотрудника	Эффективность выполнения должностных обязанностей	Уровень профессиональных знаний и навыков	Профессионально важные качества	Дисциплинированность	Лояльность	Деловой внешний вид
	X_1	X_2	X_3	X_4	X_5	X_6
u_1	2,56	3,34	2,89	3,87	2,93	4,78
u_2	3,58	2,67	3,25	3,57	3,79	2,10
u_3	4,14	4,23	2,28	4,90	3,41	4,57
u_4	2,20	3,67	3,12	2,68	4,25	4,95
u_5	4,75	4,82	4,88	3,56	4,83	2,87

X_5 — лояльность; X_6 —деловой внешний вид, и один эндогенный признак Y — удовлетворительность. Тогда, определив возможные значения (нечеткие термы) лингвистических переменных X_i ($i = \overline{1, 6}$) и Y, на базе приведенных высказываний построим импликативные правила в следующем виде:

e_1: "Если X_1 = ЭФФЕКТИВНО и
X_2 = НЕОБХОДИМЫЙ и
X_4 = ДИСЦИПЛИНИРОВАННЫЙ, то
Y = УДОВЛЕТВОРЯЮЩИЙ;
e_2: "Если X_1 = ЭФФЕКТИВНО и
X_2 = НЕОБХОДИМЫЙ и X_3 = ДОСТАТОЧНЫЕ и
X_4 = ДИСЦИПЛИНИРОВАННЫЙ, то Y = БОЛЕЕ ЧЕМ УДОВЛЕТВОРЯЮЩИЙ";
e_3: "Если X_1 = ЭФФЕКТИВНО и
X_2 = НЕОБХОДИМЫЙ и X_3 = ДОСТАТОЧНЫЕ и
X_4 = ДИСЦИПЛИНИРОВАННЫЙ и
X_5 = ПРОЯВЛЯЕТ и X_6 = ОТМЕННЫЙ, то
Y = БЕЗУПРЕЧНЫЙ";
e_4: "Если X_1 = ЭФФЕКТИВНО и
X_2 = НЕОБХОДИМЫЙ и
X_4 = ДИСЦИПЛИНИРОВАННЫЙ и
X_5 = ПРОЯВЛЯЕТ и X_6 = ПРИГЛЯДНЫЙ, то
Y = ОЧЕНЬ УДОВЛЕТВОРЯЮЩИЙ";
e_5: "Если X_1 = ЭФФЕКТИВНО и
X_2 = НЕОБХОДИМЫЙ и X_3 = ДОСТАТОЧНЫЕ и
X_4 = НЕДИСЦИПЛИНИРОВАННЫЙ, то
Y = УДОВЛЕТВОРЯЮЩИЙ;
e_6: "Если X_1 = НЕ ЭФФЕКТИВНО и
X_3 = НЕДОСТАТОЧНЫЕ, то
Y = НЕУДОВЛЕТВОРЯЮЩИЙ".

Далее зададим лингвистическую переменную на дискретном множестве $J = \{0; 0{,}1; 0{,}2; ...; 1\}$. Тогда используемые в импликативных правилах ее значения — нечеткие термы — можно задать с помощью следующих функций принадлежности [3]:

- $\tilde{S}$ = УДОВЛЕТВОРЯЮЩИЙ как:
  $\mu_{\tilde{S}}(x) = x$, $x \in J$;
- $M\tilde{S}$ = БОЛЕЕ ЧЕМ УДОВЛЕТВОРЯЮЩИЙ как: $\mu_{M\tilde{S}}(x) = \sqrt{x}$, $x \in J$;
- $\tilde{P}$ = БЕЗУПРЕЧНЫЙ как:
  $\mu_{\tilde{P}}(x) = \begin{cases} 1, \text{ if } x = 1, \\ 0, \text{ if } x < 1, \end{cases}$, $x \in J$;
- $V\tilde{S}$ = ОЧЕНЬ УДОВЛЕТВОРЯЮЩИЙ как:
  $\mu_{V\tilde{S}}(x) = x^2$, $x \in J$;
- $U\tilde{S}$ = НЕУДОВЛЕТВОРЯЮЩИЙ как:
  $\mu_{U\tilde{S}}(x) = 1 - x$, $x \in J$.

Предположим, что необходимо провести оперативную оценку пяти сотрудников компании, т. е. на конечном множестве $U = \{u_1, u_2, u_3, u_4, u_5\}$. Оперативная информация об их деятельности за истекший период упорядочена в виде табл. 3.

Далее на основе оценок базовых показателей построим нечеткие множества рассматриваемых шести критериев оценки по опорному вектору $(u_1, u_2, u_3, u_4, u_5)$ с гауссовскими функциями принадлежности $\mu_{\tilde{X}}(t) = \exp(-(t - 5)^2/\sigma_k^2)$, где σ_k ($k = \overline{1, 6}$) — плотность, подбираемая с учетом приоритетности соответствующего критерия. При этом оценка сотруднику по каждому критерию выставляется в соответствии со значением соответствующей гауссовской функции, изменяющейся на интервале [0; 5]. Например, оценка u_1 по первому критерию будет: $\mu_{\tilde{X}_1}(2.56) = 0.3857$.

Таким образом, полагая $\sigma_1 = 2.5$, $\sigma_2 = 1.4$, $\sigma_3 = 2$, $\sigma_4 = 2.6$, $\sigma_5 = 2.8$ и $\sigma_6 = 2$, принятые критерии оценки будут иметь следующий вид:

- ЭФФЕКТИВНО (работает):
$$\tilde{A}_1 = \frac{0.3857}{u_1} + \frac{0.7242}{u_2} + \frac{0.8884}{u_3} + \frac{0.2852}{u_4} + \frac{0.99}{u_5};$$
- НЕОБХОДИМЫЕ (знания):
$$\tilde{A}_2 = \frac{0.2451}{u_1} + \frac{0.0627}{u_2} + \frac{0.7390}{u_3} + \frac{0.4056}{u_4} + \frac{0.9836}{u_5};$$
- ДОСТАТОЧНЫЕ (профессиональные качества):
$$\tilde{A}_3 = \frac{0.3286}{u_1} + \frac{0.465}{u_2} + \frac{0.1573}{u_3} + \frac{0.4133}{u_4} + \frac{0.9964}{u_5};$$
- ДИСЦИПЛИНИРОВАННЫЙ:
$$\tilde{A}_4 = \frac{0.8279}{u_1} + \frac{0.739}{u_2} + \frac{0.9985}{u_3} + \frac{0.451}{u_4} + \frac{0.7358}{u_5};$$

- ПРОЯВЛЯЕТ (лояльность):

$$\tilde{A}_5 = \frac{0.5789}{u_1} + \frac{0.8297}{u_2} + \frac{0.7244}{u_3} + \frac{0.9308}{u_4} + \frac{0.9963}{u_5};$$

- ПРИГЛЯДНЫЙ (внешний вид):

$$\tilde{A}_6 = \frac{0.988}{u_1} + \frac{0.1222}{u_2} + \frac{0.9548}{u_3} + \frac{09994}{u_4} + \frac{0.3217}{u_5}.$$

С учетом этих формализмов нечеткие правила сформулируем следующим образом:

e_1: "Если $X_1 = \tilde{A}_1$, $X_2 = \tilde{A}_2$, $X_4 = \tilde{A}_4$, то $Y = \tilde{S}$";

e_2: "Если $X_1 = \tilde{A}_1$, $X_2 = \tilde{A}_2$, $X_3 = \tilde{A}_3$, $X_4 = \tilde{A}_4$, то $Y = M\tilde{S}$";

e_3: "Если $X_1 = \tilde{A}_1$, $X_2 = \tilde{A}_2$, $X_3 = \tilde{A}_3$, $X_4 = \tilde{A}_4$, $X_5 = \tilde{A}_5$, $X_6 = \tilde{A}_6$, то $Y = \tilde{P}$;

e_4: "Если $X_1 = \tilde{A}_1$, $X_2 = \tilde{A}_2$, $X_4 = \tilde{A}_4$, $X_5 = \tilde{A}_5$, $X_6 = \tilde{A}_6$, то $Y = V\tilde{S}$";

e_5: "Если $X_1 = \tilde{A}_1$, $X_2 = \tilde{A}_2$, $X_3 = \tilde{A}_3$, $X_4 =$ не $\tilde{A}_4$, то $Y = \tilde{S}$";

e_6: "Если $X_1 =$ не $\tilde{A}_1$ и $X_3 =$ не $\tilde{A}_3$, то $Y = U\tilde{S}$".

Далее для левых частей этих правил вычислим функции принадлежности $\mu_{\tilde{M}_i}(a)$ $(i = \overline{1, 6})$. В частности:

$$e_1\colon \mu_{\tilde{M}_1}(a) = \min\{\mu_{\tilde{A}_1}(a), \mu_{\tilde{A}_2}(a), \mu_{\tilde{A}_4}(a)\},$$

$$\tilde{M}_1 = \frac{0.2451}{u_1} + \frac{0.0627}{u_2} + \frac{0.739}{u_3} + \frac{0.2852}{u_4} + \frac{0.7358}{u_5};$$

$$e_2\colon \mu_{\tilde{M}_2}(a) = \min\{\mu_{\tilde{A}_1}(a), \mu_{\tilde{A}_2}(a), \mu_{\tilde{A}_3}(a), \mu_{\tilde{A}_4}(a)\},$$

$$\tilde{M}_2 = \frac{0.2451}{u_1} + \frac{0.0627}{u_2} + \frac{0.1573}{u_3} + \frac{0.2852}{u_4} + \frac{0.7358}{u_5};$$

$$e_3\colon \mu_{\tilde{M}_3}(a) = \min\{\mu_{\tilde{A}_1}(a), \mu_{\tilde{A}_2}(a), \mu_{\tilde{A}_3}(a), \mu_{\tilde{A}_4}(a), \mu_{\tilde{A}_5}(a), \mu_{\tilde{A}_6}(a)\},$$

$$\tilde{M}_3 = \frac{0.2451}{u_1} + \frac{0.0627}{u_2} + \frac{0.1573}{u_3} + \frac{0.2852}{u_4} + \frac{0.3217}{u_5};$$

$$e_4\colon \mu_{\tilde{M}_4}(a) = \min\{\mu_{\tilde{A}_1}(a), \mu_{\tilde{A}_2}(a), \mu_{\tilde{A}_4}(a), \mu_{\tilde{A}_5}(a), \mu_{\tilde{A}_6}(a)\},$$

$$\tilde{M}_4 = \frac{0.2451}{u_1} + \frac{0.0627}{u_2} + \frac{0.7244}{u_3} + \frac{0.2852}{u_4} + \frac{0.3217}{u_5};$$

$$e_5\colon \mu_{\tilde{M}_5}(a) = \min\{\mu_{\tilde{A}_1}(a), \mu_{\tilde{A}_2}(a), \mu_{\tilde{A}_3}(a), 1 - \mu_{\tilde{A}_4}(a)\},$$

$$\tilde{M}_5 = \frac{0.1721}{u_1} + \frac{0.0627}{u_2} + \frac{0.0015}{u_3} + \frac{0.2852}{u_4} + \frac{0.2642}{u_5};$$

$$e_6\colon \mu_{\tilde{M}_6}(a) = \min\{1 - \mu_{\tilde{A}_1}(a), 1 - \mu_{\tilde{A}_3}(a)\},$$

$$\tilde{M}_6 = \frac{0.6143}{u_1} + \frac{0.2758}{u_2} + \frac{0.1116}{u_3} + \frac{0.5867}{u_4} + \frac{0.0036}{u_5}.$$

В итоге правила можно записать в более компактной форме:

(e_1) "Если $X = \tilde{M}_1$, то $Y = \tilde{S}$";

(e_2) "Если $X = \tilde{M}_2$, то $Y = M\tilde{S}$";

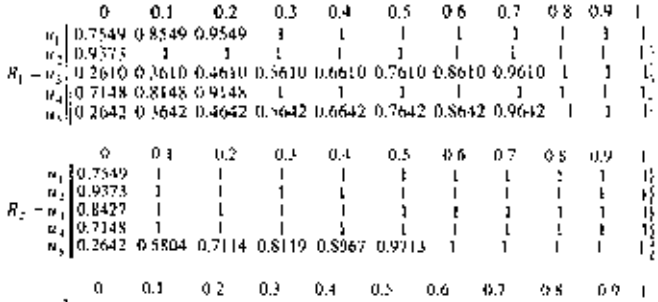
(e_3) "Если $X = \tilde{M}_3$, то $Y = \tilde{P}$";

(e_4) "Если $X = \tilde{M}_4$, то $Y = V\tilde{S}$";

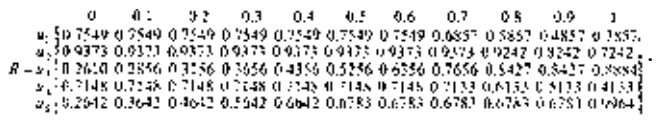
(e_5) "Если $X = \tilde{M}_5$, то $Y = \tilde{S}$";

(e_6) "Если $X = \tilde{M}_6$, то $Y = U\tilde{S}$".

Для преобразования этих правил воспользуемся импликацией (4). Тогда для каждой пары $(u, j) \in U \times Y$ получим следующие нечеткие отношения:



В результате пересечения отношений R_1, R_2, ..., R_6 в итоге получим общее функциональное решение:



Для вычисления степени удовлетворительности каждого из пяти сотрудников применим правило композиционного вывода в нечеткой среде:

$$\tilde{E}_k = \tilde{G}_k \circ R,$$

где $\tilde{E}_k$ — степень удовлетворения k-го сотрудника; $\tilde{G}_k$ — отображение k-то сотрудника в виде нечеткого подмножества на U.

Для $\tilde{E}_k$ имеем:

$$\mu_{\tilde{E}_k}(j) = \max_u (\min(\mu_{\tilde{G}_k}(u), \mu_R(u))),$$

где $\mu_{\tilde{G}_k}(u) = 0$, если $u \neq u_k$, и $\mu_{\tilde{G}_k}(u) = 1$, если $u = u_k$. Отсюда следует, что $\mu_{\tilde{E}_k}(j) = \mu_R(u_k, j)$, т. е. E_k есть k-я строка матрицы R.

Теперь применим описанную выше процедуру для получения точечных оценок каждого из рассматриваемых сотрудников. Итак, для первого сотрудника u_1 имеем оценку в виде нечеткого множества:

$$E_1 = \frac{0.7549}{0} + \frac{0.7549}{0.1} + \frac{0.7549}{0.2} + \frac{0.7549}{0.3} + \frac{0.7549}{0.4} + \frac{0.7549}{0.5} + \frac{0.7549}{0.6} + \frac{0.6857}{0.7} + \frac{0.5857}{0.8} + \frac{0.4857}{0.9} + \frac{0.3857}{1.0}.$$

Вычислим ее уровневые множества $E_{i\alpha}$ и соответствующие мощности $M(E_\alpha)$ по формуле $M(E_{j\alpha}) = \sum_{i=1}^{n} \frac{x_i}{n}$:

- для $0 < \alpha < 0.3857$; $\Delta\alpha = 0.3857$

$$E_{1\alpha} = \{0; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1\};$$
$$M(E_{1\alpha}) = 0.5;$$

- для $0.3857 < \alpha < 0.4857$; $\Delta\alpha = 0.1$

$$E_{1\alpha} = \{0; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9\};$$
$$M(E_{1\alpha}) = 0.45;$$

- для $0.4857 < \alpha < 0.5857$; $\Delta\alpha = 0.1$

$$E_{1\alpha} = \{0; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8\};$$
$$M(E_{1\alpha}) = 0.4;$$

- для $0.5857 < \alpha < 0.6857$; $\Delta\alpha = 0.1$

$$E_{1\alpha} = \{0; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7\};$$
$$M(E_{1\alpha}) = 0.35;$$

- для $0.6857 < \alpha < 0.7549$; $\Delta\alpha = 0.0692$

$$E_{1\alpha} = \{0; 0.1; 0.2; 0.3; 0.4; 0.5; 0.6\};$$
$$M(E_{1\alpha}) = 0.3.$$

Далее, по формуле (5) найдем точечную оценку удовлетворительности первого сотрудника ($\tilde{E}_1$):

$$F(\tilde{E}_1) = \frac{1}{0.7549} \int_0^{0.7549} M(E_{1\alpha}) d\alpha =$$

$$= \frac{1}{0.7549}(0.5 \cdot 0.3857 + 0.45 \cdot 0.1 + 0.4 \cdot 0.1 + 0.35 \cdot 0.1 + 0.3 \cdot 0.0692) = 0.4419.$$

Аналогичными действиями устанавливаем точечные оценки для остальных сотрудников: $F(\tilde{E}_2) = 0.4819$; $F(\tilde{E}_3) = 0.7031$; $F(\tilde{E}_4) = 0.4576$; $F(\tilde{E}_5) = 0.7133$. В качестве наиболее удовлетворяющего выбираем сотрудника, имеющего наибольшую точечную оценку. В приведенном примере это сотрудник u_5. На втором месте u_3, на третьем — u_2, на четвертом — u_4 и, наконец, на пятом — u_1.

Заключение

Полученные агрегированные точечные оценки профессиональных качеств пяти произвольно выбранных сотрудников не являются абсолютными, так как набор используемых лингвистических правил и параметры гауссовских функций принадлежности не оптимизированы. Да эту задачу, собственно, и не ставили перед собой авторы. На основе предлагаемой методики можно периодически оценивать профессиональные качества сотрудников компании и формировать динамические ряды, отражающие их профессиональное развитие. В свою очередь, ранжирование сотрудников можно осуществлять путем сравнения соответствующих им трендов.

Метод нечеткого логического вывода можно применить и для выявления итоговой оценки по каждому критерию в отдельности. Например, для установления эффективности выполнения должностных обязанностей можно аналогично использовать следующий набор правил:

e_1: "Если работник соблюдает план работы на месяц и старается соблюдать требования к работе, то он эффективен";

e_2: "Если к указанным выше требованиям он еще и соблюдает сроки выполнения незапланированных поручений, то он более чем эффективен";

e_3: "Если к указанным в приведенных выше двух правилах требованиям работник выполняет работу досрочно, то он очень эффективен";

e_4: "Если работник соблюдает план работы на месяц, а в случае необходимости — досрочно, соблюдает сроки выполнения незапланированных поручений, выполняет свои обязанности с выраженным умением работать на конечный результат и соблюдает все требования к работе, то он в высшей степени эффективен";

e_5: "Если работник соблюдает план работы на месяц, соблюдает требования к работе, качественно и ответственно выполняет свои обязанности, но не всегда соблюдает сроки выполнения незапланированных поручений, то он все равно эффективен";

e_6: "Если работник не соблюдает план работы на месяц и служебные требования, то он не эффективен".

Хотя приведенные правила и не являются оптимальными, но чисто произвольными их тоже нельзя назвать, так как их конструкция предполагает доминирование более существенных критериев над менее существенными. В конце концов, руководство каждой компании в праве само выбирать набор лингвистических правил исходя из специфики собственного предприятия.

Список литературы

1. **Васильев С. В., Жуковский А. И., Цуркер К.** Эффективность работы организаций государственного и муниципального управления и их служащих. Великий Новгород: АБУ-Консалт, 2002. 70 с.
2. **Андрейчиков А. В., Андрейчикова О. Н.** Анализ, синтез, планирование решений в экономике. М.: Финансы и статистика, 2000. 368 с.
3. **Zadeh L. A.** The concept of a linguistic variable and its application to approximate reasoning. New York: American Elsevier Publishing Company, 1974.
4. **Fuzzy** Sets, Neural Networks, and Soft Computing / Edited by Yager R. R., L. A. Zadeh. Van Nostrand Reinhold, 1994. P. 440.

УДК 621.3

Н. И. Лиманова, д-р техн. наук, проф.,
Тольяттинский государственный университет,
e-mail: N. Limanova@tltsu.ru,
М. Н. Седов, вед. специалист,
Муниципальное учреждение
"Городской информационный центр" мэрии,
г. Тольятти, e-mail: SedovMN@inbox.ru

Метод автоматизированного поиска персональных данных на основе нечеткого сравнения

В процессе межведомственного информационного обмена возникает проблема согласования основных реквизитов (ФИО, даты рождения, адреса, паспортных данных и т. п.) физических лиц в базах данных различных ведомств, обменивающихся информацией. Проблема персональной идентификации приобретает наибольшую актуальность для физических лиц, у которых частично или полностью не совпадают реквизиты. В статье описан новый алгоритм идентификации, позволяющий выполнять автоматизированный поиск таких физических лиц в базах данных на основе нечеткого сравнения. Алгоритм реализован на языке PL-SQL системы управления базами данных Oracle 9. Разработанное программное обеспечение, реализующее метод автоматизированного поиска персональных данных на основе нечеткого сравнения, внедрено и успешно работает в муниципальном учреждении "Городской информационный центр" мэрии г. Тольятти Самарской области.

Ключевые слова: *информационный обмен, реквизиты физических лиц, персональная идентификация, алгоритм идентификации, автоматизированный поиск, нечеткое сравнение.*

При организации информационного обмена между ведомствами возникает ряд проблем, главная из которых обусловлена отсутствием единого персонального номера, закрепленного за конкретным человеком. В подавляющем большинстве случаев персональный идентификационный номер (ПИН) вводится в пределах только одной базы данных, максимум — в пределах учреждения. Поэтому при передаче данных от одного учреждения и помещении их в базу другого возникает проблема привязки новых данных к ПИНу базы того ведомства, которое принимает информацию.

Традиционно данная проблема решается путем анализа тождественности основных реквизитов физического лица. Таких реквизитов несколько: фамилия, имя, отчество, дата рождения, серия, номер паспорта и адрес. Однозначно определив совпадение существующих и новых реквизитов, можно выполнить идентификацию физического лица в базе данных. Данный метод поиска выполняется вручную только в том случае, когда объем передаваемой информации невелик (число физических лиц не более 30). При больших объемах передаваемых данных используется автоматизированное сравнение тождественности реквизитов. Такой подход позволяет определить в среднем (50—60) % от общего числа идентифицируемых физических лиц. Оставшиеся (40—50) % представляют собой персональные данные, в которых частично или полностью не совпадают реквизиты. Такую информацию вручную обрабатывать еще сложнее. В связи с этим задача автоматизированного поиска распадается на три подзадачи в зависимости от типа исходных данных. В результате сравнения могут получиться следующие три типа результатов.

1. *Человек найден.* Этот вывод может сформироваться в результате прямого сравнения реквизитов, а также равенства совокупностей некоторых ключевых данных. В данном случае физическое лицо сразу привязывается к соответствующему ПИНу.

2. *Человек неоднозначно определен.* Этот результат выводится при наличии ошибок как в новых данных, так и в ранее полученных. Например, возможны ошибки оператора при ручном вводе основных реквизитов, порча данных при передаче, некорректная работа пакетных запросов при обработке информации и т. д. В данном случае выводится список ПИНов, основные реквизиты которых наиболее приближены к идентифицируемым данным.

3. *Человек не найден.* Этот случай свидетельствует о том, что данное физическое лицо отсутствует в базе и для привязки этого человека к ПИНу необходимо добавить его в имеющийся набор данных с присвоением ему нового ПИНа.

При создании автоматизированного комплекса программного обеспечения, который дает перечисленные выше результаты, наиболее важно достоверно определять границы между первым и вторым случаями, а также между вторым и третьим. Программное обеспечение, работающее без подобного разграничения, всем найденным лицам однозначно проставит ПИНы, а те, чьи данные определены неоднозначно, будут выведены в отчет для ручной обработки оператором. При этом все не найденные лица добавятся в базу с присвоением нового ПИНа.

Теперь представим себе, что при любом несоответствии основных реквизитов данные будут представлены в отчет или, что еще хуже, будут добавлены как новые. Например, женщину зовут Наталья, она вышла замуж, соответственно сменила фамилию, переехала на другое место жительства и поменяла паспорт. Кроме того, в базе она числится под именем Наталия, и в ее дате рождения допустили ошибку, неверно указав число. При обработке таких данных программа решит, что это новый человек и добавит их с присвоением нового ПИНа. Конечно, новому ПИНу поставят в соответствие какую-либо задачу. В итоге получается, что данные по одному физическому лицу удвоены, и разные ПИНы одного человека работают в разных задачах. Если данная ошибка не будет сразу исправлена, то количество некорректных сведений будет нарастать в геометрической прогрессии. На исправление последст-

Рис. 1. Упрощенная блок-схема алгоритма программы поиска физических лиц в базах данных на основе нечеткого сравнения

вий работы такого программного обеспечения уйдет немало времени и сил у большого числа компетентных служащих учреждения.

Неправильная идентификация может привести также к большому количеству данных в отчете для ручной отработки, к присвоению ПИНа не тому человеку и к добавлению излишних данных. Последствия таких ошибок в худшем случае могут полностью парализовать работу учреждения на неопределенное время, в лучшем — отнять более 10 % рабочего времени специалистов на исправление ошибок. Анализ существующих источников информации, алгоритмов и программного обеспечения показал, что единого идентификатора нет, универсальный алгоритм идентификации также отсутствует.

Цель данной работы состояла в том, чтобы создать оптимальный алгоритм идентификации, иначе говоря, научить программу правильно анализировать основные реквизиты и выполнять поиск физических лиц на основе нечеткого сравнения. Такой алгоритм был создан и реализован на языке PL-SQL системы управления базами данных (СУБД) Oracle 9. Упрощенная блок-схема алгоритма программы поиска физических лиц в базах данных на основе нечеткого сравнения приведена на рис. 1.

Особенностью разработанного алгоритма является использование в нем встроенной в СУБД процедуры SOUNDEX фонетического представления слова для увеличения качества идентификации и исключения возможных ошибок оператора. Во всех современных СУБД присутствует встроенная функция SOUNDEX. Она необходима для сравнения слов, правописание которых различно, но произношение одинаково. Проблема заключается в том, что данная функция работает только с латинским алфавитом. Для применения данной функции к кириллице была разработана процедура транслитирования (перекодирования) русских букв в латинские TO_TRANSLIT.

Разработаны также процедуры нечеткого сравнения двух русских слов (COMPARISON_STRING) и нечеткого сравнения двух чисел (COMPARISON_NUMBER). Все эти процедуры могут использоваться не только в рассматриваемом алгоритме, но также и при составлении любого запроса к базе данных, где нужно применить нечеткое сравнение двух величин подходящего типа. Например:

```
CURSOR persons
  IS
    SELECT p.person_id, p.lastname, p.firstname, p.patronymic,
           p.birthdate
    FROM work.person p
    WHERE (COMPARISON_STRING(p.lastname,
           fo_Lastname) = 1
      AND COMPARISON_STRING(p.firstname,
           fo_Firstname) = 1
      AND COMPARISON_NUMBER(p.birthdate, fo_Birthdate))
      OR (COMPARISON_STRING(p.lastname,
           fo_Lastname) = 1
      AND COMPARISON_STRING(p.patronymic,
           fo_Patronymic) = 1
      AND COMPARISON_NUMBER(p.birthdate, fo_Birthdate))
      OR (COMPARISON_STRING(p.firstname,
           fo_Firstname) = 1
      AND COMPARISON_STRING(p.patronymic,
           fo_Patronymic) = 1
      AND COMPARISON_NUMBER(p.birthdate,
           fo_Birthdate));
```

Таким образом, при использовании функции SOUNDEX совместно с TO_TRANSLIT, COMPARISON_STRING и COMPARISON_NUMBER можно составлять универсальные запросы по выбору похожих людей. Такие запросы возвращают примерно 200—300 человек, отдаленно похожих своими основными реквизитами на идентифицируемое физическое лицо. Далее в цикле перебора полученного набора данных можно сравнивать различные вариации реквизитов. Например, совпадения имени и отчества, адреса и номера паспорта могут говорить об идентичности физических лиц. Подобный подход позволяет максимально проработать все возможные варианты, отслеживающие ошибки ввода и смену реквизитов физического лица.

Программа формирует удобный отчет для ручной отработки наиболее спорных ситуаций, в котором несколькими щелчками мыши можно добавить нового человека в базу или выбрать физическое лицо из предлагаемых вариантов. Кроме того, тщательная обработка массива похожих людей и удачная вариация наборов ключевых реквизитов позволяет выводить в отчет для ручной отработки только 0,01 % записей от всего объема идентифицируемых данных.

При детальном сравнении разработанного программного обеспечения с процедурой прямого сравнения для годового объема идентификации в 1 200 000 физических лиц были получены следующие данные: трудовые затраты на обра-

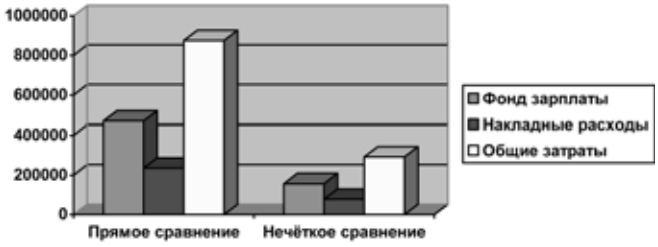


Рис. 2. Диаграмма для сравнительного анализа стоимостных показателей при использовании методов прямого и нечеткого сравнения

ботку информации по методу нечеткого сравнения по сравнению с методом прямого сравнения уменьшены в 6,7 раза, абсолютное снижение трудовых затрат составило 1446 ч, годовые затраты при использовании метода нечеткого сравнения уменьшились в 3 раза по сравнению с аналогичным периодом применения метода прямого сравнения, а годовой экономический эффект превысил 580 000 руб. Для наглядности некоторые стоимостные показатели, формирующиеся при использовании разработанного и применявшегося до настоящего времени программного обеспечения, отображены на диаграмме, приведенной на рис. 2. Затраты отложены по оси ординат в рублях.

Список литературы

1. **Международный** фонд автоматической идентификации. Технологии автоматической идентификации (доступны по адресу http://www.fond-ai.ru/artl/art223.html).
2. **Положение** о персональном идентификационном номере граждан Российской Федерации, проживающих или пребывающих на территории Санкт-Петербурга (доступно по адресу http://iac.spb.ru/shablon.asp?subpage = 171&id = 40&dir = 0.)
3. **Проект** "Социальная карта москвича" (доступен по адресу http://www.soccard.ru/soccard.shtml).
4. **Лиманова Н. И., Седов М. Н.** Проблема межведомственного информационного обмена и метод ее решения // Матер. научно-практической конференции "Инновации в условиях развития информационно-коммуникационных технологий". М.: МГИЭМ, 2008. С. 435—437.
5. **Аллен К.** Oracle PL/SQL / Пер. Т. Москалева. М.: Лори, 2001. 350 с.
6. **Чубукова И. А.** Data Mining: Учеб. пос. М.: Интернет-ун-т информ. технологий. Бином, 2006. 382 с.

КОДИРОВАНИЕ И ОБРАБОТКА СИГНАЛОВ

УДК 004.932.2

Е. В. Медведева, канд. техн. наук, доц.,
Вятский государственный университет,
e-mail: EMedv@mail.ru

Адаптивная нелинейная фильтрация цветных видеоизображений

Рассматривается метод адаптивной нелинейной фильтрации последовательности цветных видеоизображений при большой интенсивности белого гауссовского шума, полностью маскирующего передаваемое изображение. Последовательность цветных видеоизображений представлена в системе RGB, состоящей из трех отдельных полутоновых изображений — компонентов. Метод фильтрации цветных видеоизображений базируется на представлении последовательности цифровых полутоновых изображений трехмерным дискретно-значным марковским процессом с несколькими значениями. Особенностью приведенного алгоритма адаптивной нелинейной фильтрации является его высокая эффективность при небольшом времени адаптации.

Ключевые слова: *цветные видеоизображения, система RGB, нелинейная фильтрация, метод адаптации, марковский процесс, разрядное двоичное изображение, цветовое полутоновое изображение.*

Введение

Изображения в процессе передачи могут искажаться шумом большой интенсивности, возникающим в каналах связи. Примером могут служить изображения, передаваемые с беспилотного летательного аппарата на пульт управления и приема информации в реальном масштабе времени.

Особый интерес вызывает фильтрация цветных изображений, поскольку именно цвет является тем важным признаком, который облегчает распознавание и выделение объекта на изображении [1]. Большинство современных цветовых моделей ориентированы на определенные прикладные задачи. В данной работе рассмотрена модель RGB, в которой изображения состоят из трех отдельных полутоновых изображений — компонентов.

Учитывая характер статистической связи между элементами внутри и между кадрами видеопоследовательностей, можно предположить, что видеопоследовательность представляет собой трехмерный дискретно-значный марковский процесс (МП) с несколькими значениями, имеющий разделимую автокорреляционную функцию вида [2, 3]

$$r_{i,\,j,\,k} = \sigma_{\mu}^2 \exp\{-\alpha_1|f| - \alpha_2|s| - \alpha_3|t|\}, \qquad (1)$$

где i, j, k — дискретные пространственные координаты: по горизонтали, вертикали и времени (по кадру) соответственно; σ_{μ}^2 — дисперсия трехмерного дискретно-значного марковского процесса; α_1, α_2, α_3 — множители, зависящие от ширины спектральной плотности мощности случайных процессов по трем измерениям; f, s, t — шаг корреляции по горизонтали, вертикали и времени.

Если полутоновые изображения (компоненты) в видеопоследовательности представлены g-разрядными двоичными числами, то видеопоследовательность цифровых полутоновых изображений (ЦПИ) является трехмерным дискретно-значным МП с несколькими значениями, а последовательность разрядных двоичных изображений (РДИ) l-го разряда образует трехмерный МП с двумя равновероятными дискретными значениями [2—3].

На рис. 1 представлены два соседних кадра l-го разряда видеопоследовательности ЦПИ, разделенных на области $F_{ik}^{(l)}$ ($i = \overline{1, 4}$, $k = 1, 2, \ldots$), элементы которых являются цепью Маркова различной размерности. Алго-

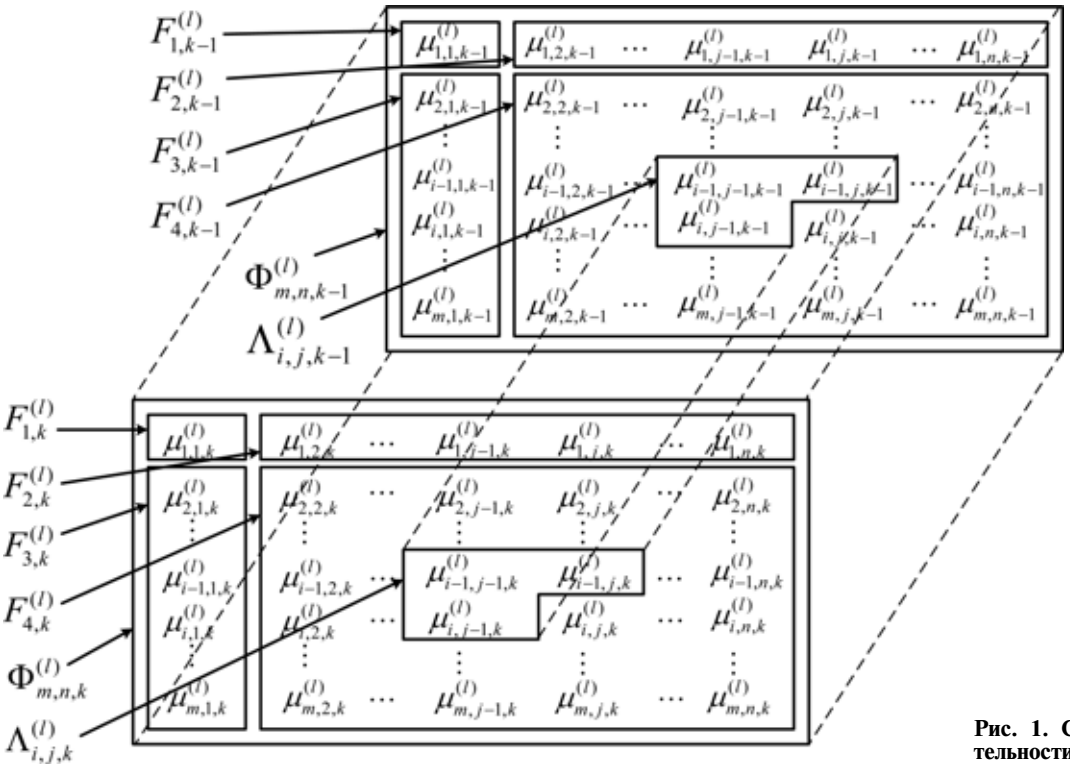


Рис. 1. Соседние кадры последовательности РДИ *l*-го разряда ЦПИ

ритмы фильтрации элементов первых трех областей известны и хорошо изучены [3]. Наибольшую сложность представляет алгоритм фильтрации элементов области $F_{4k}^{(l)}$. Фильтруемый элемент $v_4^{(l)} = \mu_{i,j,k}^{(l)}$ области $F_{4k}^{(l)}$ зависит от семи соседних элементов, входящих в его окрестность (рис. 2), где $v_1^{(l)} = \mu_{i,j-1,k}^{(l)}$, $v_2^{(l)} = \mu_{i-1,j,k}^{(l)}$, $v_3^{(l)} = \mu_{i-1,j-1,k}^{(l)}$, $v_1'^{(l)} = \mu_{i,j-1,k-1}^{(l)}$, $v_2'^{(l)} = \mu_{i-1,j,k-1}^{(l)}$, $v_3'^{(l)} = \mu_{i-1,j-1,k-1}^{(l)}$, $v_4'^{(l)} = \mu_{i,j,k-1}^{(l)}$.

При известных на приемной стороне статистических данных о фильтруемом процессе в работе [3] был синтезирован оптимальный алгоритм нелинейной фильтрации видеопоследовательности ЦПИ.

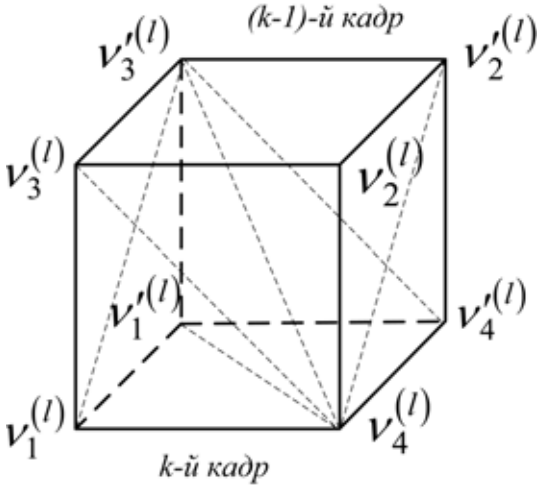


Рис. 2. Окрестность $\Lambda_{i,j,k}^{(l)}$ элемента $v_4^{(l)}$

В реальных условиях статистические данные о фильтруемом процессе полностью или частично неизвестны. В этом случае целесообразно применять адаптивные алгоритмы обработки видеопоследовательности ЦПИ, позволяющие непосредственно в процессе приема кадров вычислять недостающие статистические данные и использовать их для повышения качества восстановления изображений, разрушенных шумом. Повышение помехоустойчивости приема ЦПИ обеспечивается за счет большой информационной избыточности, заложенной в изображении.

Широко распространенные методы адаптации, например, основанные на методах среднеквадратичной ошибки или наименьших квадратов [4], требуют больших вычислительных ресурсов и труднореализуемы в реальном масштабе времени. Поэтому в адаптивном алгоритме обработки ЦПИ механизм адаптации должен быть простым, эффективным и сравнимым по вычислительной сложности с самим алгоритмом фильтрации.

В данной работе предложен метод адаптации, в котором процесс адаптации сведен к измерению, на основе принимаемого сигнала, неизвестных значений элементов одношаговых матриц вероятностей переходов (МВП) в сложной цепи Маркова с несколькими значениями для каждого цветового компонента и подстановки их в алгоритм фильтрации.

Метод адаптивной нелинейной фильтрации видеоизображений

Необходимо разработать адаптивный алгоритм нелинейной фильтрации цветных видеоизображений, представленных в системе RGB, состоящих из трех отдельных ЦПИ марковского типа при наличии белого гауссовского шума $\eta(t)$ с нулевым средним и дисперсией σ_η^2.

Полагая, что ЦПИ состоит из g РДИ, при разработке адаптивного алгоритма фильтрации видеоизображений воспользуемся уравнением, полученным в [3] для опти-

мальной нелинейной фильтрации последовательности l-го РДИ ($l \in g$) с известными МВП по горизонтали ($^1\Pi^{(l)}$), вертикали ($^2\Pi^{(l)}$) и кадрам ($^4\Pi^{(l)}$), и заменим в нем элементы всех МВП на их оценки:

$$u^{(l)}(v_4^l) = [f(M_1^{(l)}(v_4^{(l)})) - f(M_2^{(l)}(v_4^{(l)}))] + u^{(l)}(v_1^{(l)}) + z_1^{(l)}[u^{(l)}(v_1^{(l)}),\ {}^1\hat{\pi}_{ij}^{(l)}] + u^{(l)}(v_2^{(l)}) + z_2^{(l)}[u^{(l)}(v_2^{(l)}),\ {}^2\hat{\pi}_{ij}^{(l)}] + u^{(l)}(v_4'^{(l)}) + z_4^{(l)}[u^{(l)}(v_4'^{(l)}),\ {}^4\hat{\pi}_{ij}^{(l)}] + u^{(l)}(v_3'^{(l)}) + z_7^{(l)}[u^{(l)}(v_3'^{(l)}),\ {}^7\hat{\pi}_{ij}^{(l)}] - u^{(l)}(v_3^{(l)}) - z_3^{(l)}[u^{(l)}(v_3^{(l)}),\ {}^3\hat{\pi}_{ij}^{(l)}] - u^{(l)}(v_1'^{(l)}) - z_5^{(l)}[u^{(l)}(v_1'^{(l)},\ {}^5\hat{\pi}_{ij}^{(l)}] - u^{(l)}(v_2'^{(l)}) - z_6^{(l)}[u^{(l)}(v_2'^{(l)}),\ {}^6\hat{\pi}_{ij}^{(l)}] \geqslant H, \quad (2)$$

где $u^{(l)}(v_4^l) = \ln\dfrac{p_1^{(l)}(v_4^{(l)})}{p_2^{(l)}(v_4^{(l)})}$, ($l = \overline{1, g}$) — логарифм отношения апостериорных вероятностей значений двоичных элементов l-го РДИ в точке $v_4^{(l)}$ (рис. 2); $[f(M_1^{(l)}(v_4^{(l)})) - f(M_2^{(l)}(v_4^{(l)}))]$ — разность логарифмов функций правдоподобия значений двоичных элементов l-го РДИ в элементе изображения $v_4^{(l)}$; H — порог, выбранный в соответствии с критерием идеального наблюдателя (для алгоритма (2) $H = 0$);

$$z_p^{(l)}(\cdot) = \ln\frac{{}^p\hat{\pi}_{\alpha\alpha}^{(l)} + {}^p\hat{\pi}_{\beta\alpha}^{(l)}\exp\{-u^{(l)}(v_q^{(l)})\}}{{}^p\pi_{\beta\beta}^{(l)} + {}^p\pi_{\alpha\beta}^{(l)}\exp\{u^{(l)}(v_q^{(l)})\}};$$
$$p = q = 1, 2, 3;\ \alpha, \beta = \overline{1, 2}; \quad (3)$$

$$z_p^{(l)}(\cdot) = \ln\frac{{}^p\hat{\pi}_{\alpha\alpha}^{(l)} + {}^p\hat{\pi}_{\beta\alpha}^{(l)}\exp\{-u^{(l)}(v_q'^{(l)})\}}{{}^p\pi_{\beta\beta}^{(l)} + {}^p\pi_{\alpha\beta}^{(l)}\exp\{u^{(l)}(v_q'^{(l)})\}};$$
$$p = \overline{4, 7};\ q = \overline{1, 4}, \quad (4)$$

${}^r\hat{\pi}_{ii}$ ($i = \overline{1, 2}$; $r = \overline{1, 7}$) — оценки элементов МВП в одномерных цепях Маркова с двумя состояниями по горизонтали ($^1\Pi^{(l)}$), вертикали ($^2\Pi^{(l)}$), кадрам ($^4\Pi^{(l)}$) последовательности l-го РДИ и сопутствующих МВП:

$$^3\Pi^{(l)} = {}^1\Pi^{(l)} \cdot {}^2\Pi^{(l)};\ {}^5\Pi^{(l)} = {}^1\Pi^{(l)} \cdot {}^4\Pi^{(l)};$$
$$^6\Pi^{(l)} = {}^2\Pi^{(l)} \cdot {}^4\Pi^{(l)};$$
$$^7\Pi^{(l)} = {}^3\Pi^{(l)} \cdot {}^4\Pi^{(l)} = {}^1\Pi^{(l)} \cdot {}^2\Pi^{(l)} \cdot {}^4\Pi^{(l)}.$$

Оценки элементов МВП неизвестны на приемной стороне системы связи и должны быть вычислены и подставлены в уравнение адаптивной фильтрации (2).

Определение оценок элементов МВП $^1\Pi^{(l)}$ сводится к вычислению средней длины цуга $\hat{\chi}^{(l)}$ (последовательности элементов одного знака) в строке l-го РДИ. В этом случае точность оценок элементов ${}^1\hat{\pi}_{ii}^{(l)} = 1 - {}^1\hat{\pi}_{ij}^{(l)}$, ($i, j = \overline{1, 2}$; $i \neq j$) МВП по горизонтали ($^1\Pi^{(l)}$ оказывается ограниченной числом элементов одной строки l-го РДИ:

$${}^1\hat{\pi}_{ii}^{(l)} = 1 - \frac{2p_1^{(l)}}{\hat{\chi}^{(l)}},\ i = 1, 2, \quad (5)$$

где $p_i^{(l)}$ — априорная вероятность значения $M_i^{(l)}$, одинаковая для всех разрядов ($p_i^{(l)} = 0{,}5$; $i = \overline{1, 2}$; $l = \overline{1, g}$).

Для получения заданной точности оценки ${}^1\hat{\pi}_{ij}^{(l)}$ проводится усреднение оценок по нескольким строкам l-го РДИ.

Используя трехмерную модель (рис. 2), содержащую множества элементов $\psi_1 = \{v_1^{(l)},\ v_2^{(l)},\ v_3^{(l)},\ v_4^{(l)}\}$, $\psi_2 = \{v_1^{(l)},\ v_4^{(l)},\ v_1'^{(l)},\ v_4'^{(l)}\}$ и $\psi_3 = \{v_2^{(l)},\ v_4^{(l)},\ v_2'^{(l)},\ v_4'^{(l)}\}$, вероятности перехода для сложной цепи Маркова можно вычислить в каждом множестве ψ_i $\{i = \overline{1, 3}\}$ по формулам

$$\pi_{iii} = 1 - \frac{{}^1\pi_{ij} \cdot {}^2\pi_{ij}}{{}^3\pi_{ii}}; \quad (6)$$

$$\pi_{iii}^{*} = 1 - \frac{{}^1\pi_{ij} \cdot {}^4\pi_{ij}}{{}^5\pi_{ii}}; \quad (7)$$

$$\pi_{iii}^{**} = 1 - \frac{{}^2\pi_{ij} \cdot {}^4\pi_{ij}}{{}^6\pi_{ii}}. \quad (8)$$

Если известны оценки вероятностей перехода (${}^1\hat{\pi}_{ii}^{(l)}$ и $\hat{\pi}_{iii}^{(l)}$), то, подставив их в уравнение (6), можно вычислить оценку вероятности перехода по вертикали ${}^2\hat{\pi}_{ii}^{(l)}$:

$${}^2\hat{\pi}_{ii}^{(l)} = \hat{\pi}_{iii}^{(l)}\frac{1 - {}^1\hat{\pi}_{ii}^{(l)}}{{}^1\hat{\pi}_{ii}^{(l)} - {}^1\hat{\pi}_{iii}^{(l)}({}^12\hat{\pi}_{ii}^{(l)} - 1)}. \quad (9)$$

Аналогично, имея оценки ${}^1\hat{\pi}_{ii}^{(l)}$ и $\hat{\pi}_{iii}^{*(l)}$, из уравнения (7) или (8) можно вычислить оценку ${}^4\hat{\pi}_{ii}^{(l)}$ между кадрами по формулам

$${}^4\hat{\pi}_{ii}^{(l)} = \hat{\pi}_{iii}^{*(l)}\frac{1 - {}^1\hat{\pi}_{ii}^{(l)}}{{}^1\hat{\pi}_{ii}^{(l)} - {}^1\hat{\pi}_{iii}^{*(l)}({}^12\hat{\pi}_{ii}^{(l)} - 1)}; \quad (10)$$

$${}^4\hat{\pi}_{ii}^{(l)} = \hat{\pi}_{iii}^{**(l)}\frac{1 - {}^2\hat{\pi}_{ii}^{(l)}}{2\hat{\pi}_{ii}^{(l)} - \hat{\pi}_{iii}^{**(l)}(2^2\hat{\pi}_{ii}^{(l)} - 1)}. \quad (11)$$

Результаты исследования

Процесс адаптивной фильтрации исследовался на последовательности из десяти искусственных РДИ размером кадра 512 × 512 элементов (пикселей), с заданными вероятностями переходов ${}^1\pi_{ii}^{(l)} = {}^2\pi_{ii}^{(l)} = {}^4\pi_{ii}^{(l)} = 0{,}9$. Относительная погрешность вычисления оценок вероятностей перехода не превосходила 1 %. Качество фильтрации определялось объективной оценкой выигрыша по мощности сигнала:

$$\eta^{(l)} = 10\lg(\rho_{э\ вых}^2/\rho_э^2), \quad (12)$$

где $\rho_э^2$, $\rho_{э\ вых}^2$ — отношение сигнал—шум по мощности сигнала в элементе l-го РДИ на входе и выходе устройства фильтрации. Отношение сигнал—шум на входе $\rho_э^2$ априорно принято одинаковым для всех g РДИ, а $\rho_{э\ вых}^2$ определяется по средней ошибке различения двоичных элементов l-го РДИ и имеет различное значение для всех g РДИ.

Выигрыш $\eta^{(l)}$ фильтрации последовательности РДИ l-го разряда ЦПИ, осуществляемой оптимальным (из-

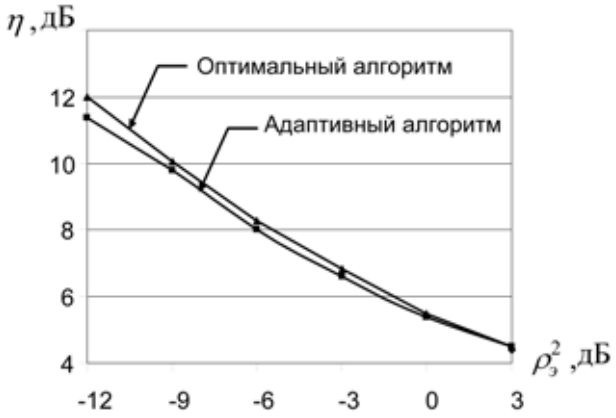


Рис. 3. Выигрыш по мощности при оптимальной и адаптивной фильтрации

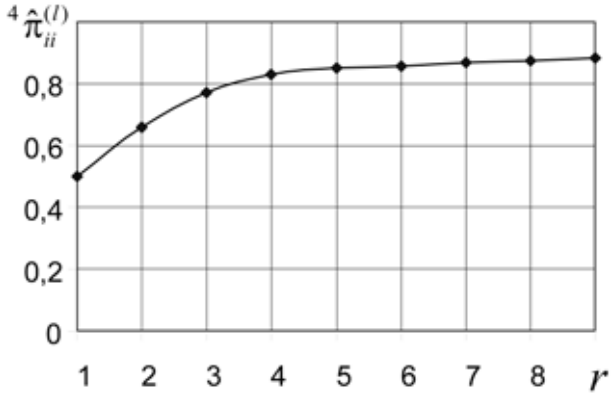


Рис. 4. Зависимость вероятности перехода $^4\hat{\pi}_{ii}^{(l)}$ от номера шага адаптации *r*

вестные априорные данные) и адаптивным алгоритмами, при различных $\rho_э^2$ отличаются не более чем на 1 дБ (рис. 3). Оценки элементов $^1\hat{\pi}_{ii}^{(l)}$, $^2\hat{\pi}_{ii}^{(l)}$ и $^4\hat{\pi}_{ii}^{(l)}$ МВП $^1\Pi^{(l)}$, $^2\Pi^{(l)}$ и $^4\Pi^{(l)}$ для каждого разряда корректировались с помощью формулы, полученной для различных отношений сигнал—шум на входе приемного устройства фильтрации ЦПИ:

$$^1\tilde{\pi}_{ii}^{(l)} = \frac{^1\pi_{ii}^{(l)} - 2p_{ош}^{(l)}(1-p_{ош}^{(l)})}{(1-p_{ош}^{(l)})^2}, \qquad (13)$$

где $p_{ош}^{(l)}$ — средняя вероятность ошибки распознавания элементов *l*-го РДИ.

На рис. 4 показана зависимость оценки $^4\hat{\pi}_{ii}^{(l)}$ элемента МВП $^4\Pi^{(l)}$ для одного из компонентов цветного видеоизображения от номера шага адаптации *r* (номер строки РДИ во втором кадре) при $\rho_э^2 = -6$ дБ.

Значение вероятности перехода $^4\hat{\pi}_{ii}^{(l)}$ после 14 шагов адаптации (строк во втором кадре) достигло 0,893, что соответствует погрешности не более 1 % от истинного значения $^4\pi_{ii} = 0{,}9$. При этом процесс адаптации по времени (по кадрам) завершается. При моделировании предполагалось, что начальное значение элемента МВП $^4\hat{\pi}_{ii} = 0{,}5$, что соответствует независимой цепи Маркова. В реальных условиях можно устанавливать среднестатистическое для телевизионных сигналов значение $^4\hat{\pi}_{ii} = 0{,}9$. В этом случае процесс адаптации ограничивается вторым кадром при отношениях сигнал—шум $\rho_э^2 = -3$ дБ.

На рис. 5 (см. третью сторону обложки) показан процесс адаптивной нелинейной фильтрации последовательности неподвижных цветных видеоизображений "Река", зашумленных белым гауссовским шумом, при $\rho_э^2 = -12$ дБ в 1-м и 25-м кадрах, полученных в реальном масштабе времени. При моделировании полагалось, что все цветовые компоненты имеют одинаковые искажения.

Заключение

- Разработанный алгоритм адаптивной нелинейной фильтрации цветных видеоизображений, представленных в системе RGB, обладает высокой эффективностью, особенно при малых отношениях сигнал—шум по мощности $\rho_э^2 < 0$ дБ.
- Скорость адаптации при фильтрации видеоизображений не превышает двух-трех кадров при среднестатистических значениях вероятности переходов по времени (кадрам) $^4\hat{\pi}_{ii} = 0{,}9$ и отношении сигнал—шум $\rho_э^2 \ll -3$ дБ.
- Адаптивный алгоритм, с учетом его однородной структуры, прост в реализации, особенно при малой разрядности представления ЦПИ ($g \leqslant 8$).

Список литературы

1. **Гонсалес Р., Вудс Р.** Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с.

2. **Трубин И. С.** Метод моделирования цифровых полутоновых изображений / И. С. Трубин, Е. В. Медведева, О. П. Булыгина // Инфокоммуникационные технологии. 2008. Т. 6. № 1. С. 94—99.

3. **Трубин И. С.** Нелинейная фильтрация видеопоследовательностей цифровых полутоновых изображений / И. С. Трубин, Е. В. Медведева, О. П. Булыгина // Инфокоммуникационные технологии. 2007. Т. 5. № 4. С. 29—35.

4. **Уидроу Б., Стирнз С.** Адаптивная обработка сигналов: пер с англ. / Под ред. В. В. Шахгильдяна. М.: Радио и связь, 1988. 440 с.

УДК 004.324

А. Ш. Сулейманов, канд. техн. наук,
Азербайджанский технический университет,
г. Баку, e-mail: akif@inbox.ru

Алгоритмы сжатия данных, основанные на применении логических шкал позиционного кодирования

Сделана попытка анализа машинных алгоритмов универсальной схемы сжатия (компрессии) и однозначного развертывания (декомпрессии) произвольной двоично-кодированной информации. Предлагаются методы сжатия, которые основываются на статистических данных, и применение логических шкал позиционного кодирования, вносящих единообразие в процессы сжатия и развертывания и позволяющих разработать относительно универсальные эффективные средства сжатия данных.

Ключевые слова: *сжатие данных, компрессия/декомпрессия, кодирование, избыточность.*

Введение

Развитие компьютерных сетей приводит к тому, что все большая часть информации, прежде всего научно-технической, экономической и социально-политической, перемещается в различные виды памяти. Большинство прогнозов сходятся на том, что в ближайшем будущем в технически развитых странах основная масса информации полностью будет храниться в безбумажном виде — в памяти компьютерных систем и сетей [1]. Подобные изменения способны повлиять на существующее положение и самым неожиданным образом выдвинуть на передний план то или иное научное направление или техническое решение, до сих пор остававшееся без внимания. Одним из таких направлений является сжатие дискретных данных, которое, на наш взгляд, занимает одну из ведущих позиций в безбумажной информатике. Это очень актуально, так как информационные ресурсы уже становятся основным национальным богатством, а эффективность их промышленной эксплуатации во все большей степени будет определять экономическую мощь в целом.

Одним из действенных путей удовлетворения требований к автоматизированной системе обработки, хранения и передачи информации и увеличения эффективности функционирования различных систем обработки, хранения и передачи данных является повышение компактности последних. Она обеспечивается за счет обнаружения и сокращения локально неиспользуемой избыточности данных и их соответствующего преобразования, получившего название "компрессия".

В работах [1—4] показывается, что дальнейшие исследования в области компрессии данных должны концентрироваться вокруг следующих четырех направлений: идентификация и характеристика избыточности данных; разработка моделей сжатия для различных схем компрессии; исследование методов адекватного кодирования Хаффмана при обновлениях баз данных; изучение возможности аппаратно-программной реализации различных схем компактного преобразования.

Постановка задачи

Уже разработано большое число различных методов компрессии данных, являющихся в определенной степени проблемно-ориентированными (специализированными). Большинство известных методов основываются на известном алгоритме Лемпеля—Зива и являются в некоторой степени его модификациями. То есть в основу алгоритмов реализации этих методов положены семантические, лингвистические, структурные, вероятностно-статистические и иные характеристики языка первичного представления информации. Они, в основном, не отвечают требованиям систем обработки, хранения и передачи данных и управления. Здесь необходимо отметить и такой немаловажный факт — отсутствие достаточно четкой теоретической базы системного решения этой задачи. В связи с этим в представленной работе предлагаются методы сжатия информации, которые основываются на локально-статистических характеристиках данных с применением логических шкал позиционного кодирования и позволяющих разработать относительно универсальные эффективные средства сжатия данных.

Основные области применения сжатия данных

Особенно остро проблема сжатия данных стоит в связи с необходимостью в относительно короткие сроки обрабатывать и передавать огромные массивы данных или же записывать и хранить их в различного рода устройствах памяти конечной емкости. Можно указать на следующие основные области применения сжатия данных.

1. Техника связи. Максимальное уменьшение объема передаваемого по каналу сообщения без потери для получателя его информационного содержания.

2. Система централизованной распределенной обработки данных. С ростом объемов обрабатываемой информации важно обеспечить высокий коэффициент использования памяти.

3. Повышение надежности передачи данных. В результате сжатия данных высвобождается определенное число битов, которое может быть использовано для целей обнаружения и исправления ошибочных битов или же пакетов ошибок [5].

4. Интеллектуальные информационные системы. Например, в информационно-поисковых системах при сжатии данных поисковый образ документа (ПОД) — модель текста — может быть представлен более полно (без значительного увеличения его объема) с автоматизацией процесса сжатия, в результате чего повышается релевантность поиска [8, 9].

5. Книгохранилище (создание электронных фондов). Все книги и рукописи определенных значительных работ могут быть подвергнуты сжатию и сохранены в таком виде на кристаллах или других носителях информации. Это очень важно как с технологической (безбумажной технологии) стороны, так и с экологической (сохранения лесов и т. д.) [1].

Естественно, с развитием и широким внедрением информационных технологий методы и средства сжатия и однозначного развертывания данных будут все более широко применять в различных системах обработки, хранения и передачи данных общего назначения и специализированного типа.

Алгоритмы сжатия и однозначного развертывания дискретных данных

Как стало ясно из сопоставительного анализа наиболее распространенных методов и алгоритмов компрессии и развертывания данных, для синтеза алгоритмов многоцелевого (относительно универсального) применения лучшими являются методы, основанные на применении логических шкал позиционного кодирования (ЛШПК) [6, 7]. Это обосновывается следующими основными характерными особенностями:

А. Указанные методы основываются на применении статистических характеристик (частоты встречаемости букв) алфавитов сжатия, в результате чего получается значительный эффект от сжатия.

Б. В качестве алфавита сжатия могут быть применены не только символы (буквы алфавита) языка первичного представления информации, а также и битовые группы различной длины (кванты).

В. Логические шкалы позиционного кодирования дают возможность разбить файл, содержащий любую информацию, на подфайлы для эффективности применения методов сжатия. Декомпозиция файла повышает вероятность получения эффекта от сжатия и увеличивает степень нерасшифруемости (секретности) его хранения или передачи по коммуникационным каналам.

Г. Методы, основанные на применении ЛШПК, дают возможность типизировать основные процедуры обработки данных в целях разработки модульных алгоритмов и аппаратно-программных средств сжатия и развертывания с возможностями программируемой коммутации последних.

В работе рассматриваются два метода сжатия данных. Первый метод (метод последовательного исключения букв), основанный на применении локальных статистических характеристик (частота встречаемости каждого кванта-буквы) сегмента данных, который рассматривается как стохастическая последовательность битов, а сжатие (преобразование) проводится путем последовательного исключения из обрабатываемого сегмента соответствующих букв алфавита сжатия с фиксацией их позиций автономными ЛШПК (число ЛШПК равно числу букв сегмента).

Второй метод (метод исключения группы букв) также основан на применении локальных статистических характеристик сегмента данных. Здесь сжатие происходит путем выделения и фиксации посредством ЛШПК позиции группы букв, имеющих сравнительно большую избыточность с точки зрения применяемого метода сжатия. Эти два метода основаны на технологии применения ЛШПК, а последняя позволяет проводить последовательную декомпозицию файла на эффективные подфайлы и разработать параллельные алгоритмы сжатия/развертывания.

Математическая модель алгоритма

Вначале вкратце рассмотрим некоторые основные понятия и определения. Сегмент стохастической последовательности битов назовем двоичным кортежем и представим в следующем виде

$$M_0 = \{x_1, x_2, ..., x_q\}, \quad (1)$$

где $x_q \in M$ и $M = \{0, 1\}$, q — число битов (символов) стохастической последовательности.

Обозначим непрерывную конечную последовательность битов длиной z через m_z и назовем эту величину шагом квантования:

$$m_z = (y_1, y_2, ..., y_z). \quad (2)$$

Необходимо значения m_z и q выбрать так, чтобы q/m_z было целым числом K_z, т. е.

$$K_z = q/m_z, \quad (3)$$

где z есть максимальная длина шага квантования в битах. После квантования M_0 с шагом квантования m_z получим K_z-компонентный кортеж:

$$M_0'(m_z) = (M_1^{(m_z)}, M_2^{(m_z)}, ..., M_{K_z}^{(m_z)}). \quad (4)$$

Предположим, что какие-то компоненты кортежа $M_0'(m_z)$ отличаются определенным свойством. Множество этих компонентов обозначим через $A^{\tau}(m_z)$. Тогда логической шкалой позиционного кодирования [ЛШПК (m_z)] можно выделить те компоненты кортежа $M_0'(m_z)$, которые входят в $A^{\tau}(m_z)$ (здесь τ — номер этапа разбиения компонент кортежа на два непересекающихся характеристических множества):

$$\left.\begin{aligned} &ЛШПК\,(m_z) = (\alpha_1^{(\tau)}, \alpha_2^{(\tau)}, ..., \alpha_{K_z}^{(\tau)}),\\ &пр_j M_0' = (M_j^{(m_z)}),\ j = 1, 2, ..., K_z,\\ &\alpha_j^{(\tau)} = \begin{cases} 1, \text{ если } M_j^{(m_z)} \in A^{(\tau)}(m_z),\\ 0, \text{ если } M_j^{(m_z)} \notin A^{(\tau)}(m_z). \end{cases} \end{aligned}\right\} \quad (5)$$

Естественно, что при квантовании M_0 с шагом квантования m_z, число различных m_z-битных букв (назовем множество этих букв алфавитом преобразования — в данном случае сжатия) будет 2^{m_z}. Тогда один и тот же кортеж M_0 можно квантовать $(z - 1)$ различными квантами (так как квантование двоичного кортежа с шагом $z = 1$ дает исходный кортеж M_0, то эта операция не имеет смысла) и получить $(z - 1)$ различных множеств алфавита сжатия. Обозначим конкретную букву определенного алфавита через β_{j*}:

$$\left.\begin{aligned} &\beta_{j*}(m_z) \in \{\beta_{j(m_z)}(m_z)\},\\ &j(m_z) = 0, 1, 2, ..., (2^{m_z} - 1) \text{ и } j^* \in j(m_z) \end{aligned}\right\}. \quad (6)$$

Естественно, что конкретный двоичный кортеж при квантовании с шагом квантования m_z может содержать не все разнообразные буквы $\beta_{j(m_z)}(m_z)$ в количестве 2^{m_z} и локальная частота их встречаемости будет для каждого алфавита разной. Частоту буквы $\beta_{j*}(m_z)$ обозначим через $\eta_{j*}(m_z)$. Тогда после квантования исходного кортежа M_0 с шагом квантования m_z образуем следующие две упорядоченные последовательности (для простоты записи не будем отмечать шаг квантования m_z):

$$\eta_{j1} \geqslant \eta_{j2} \geqslant \eta_{j3} \geqslant ... \geqslant \eta_{jQ}; \quad (7)$$

$$\beta_{j1}, \beta_{j2}, \beta_{j3}, ..., \beta_{jQ}, \quad (8)$$

где $j_1, j_2, j_3, ..., j_Q \in \{0, 1, 2, ..., (2^{m_z} - 1)\}$.

Сжатие двоичного кортежа назовем *R*-преобразованием. Будем пользоваться обозначением *RX*, где *X* будет отображать мнемонику конкретного применяемого метода компрессии данных. Как было сказано выше, в работе рассматриваются два метода компрессии данных — метод последовательного исключения букв (RPOS-преобразование) и метод исключения группы букв (RGRUP-преобразование). Оба метода основываются на вероятностно-статистических характеристиках букв алфавита сжатия. Общим для обоих методов является формирование последовательностей (7) и (8), т. е. определения частоты встречаемости каждой буквы ($\eta_{j^*}(m_z)$) и упорядочение, а также определения буквы сжатия $\beta_{j^*}(m_z)$. Эта процедура является "Статистическим анализом файла и упорядочением частоты встречаемости букв" и характеризуется следующими особенностями.

1. Переменная длина шага квантования выдвигает особые требования к длине исходного файла. Длина файла в битах должна быть нацело делимой на все принятые значения шага квантования m_z.

2. Из анализа выбранных методов выявилось, что вероятность получения максимальной эффективности от сжатия более значительна при возможно большей длине шага квантования m_z. Однако с увеличением m_z увеличивается и число букв 2^{m_z}, а это обстоятельство, в свою очередь, порождает другие трудности (при больших значениях m_z увеличивается общее число множеств алфавита сжатия, в результате чего увеличивается требуемый объем памяти для хранения букв).

3. С увеличением m_z увеличиваются параметры алгоритмов обработки. Следовательно, должны рассматриваться все перечисленные особенности в комплексе для принятия компромиссного решения. Для программной реализации алгоритмов выбранных методов максимальная длина шага квантования принята $m_z = 8$, т. е. $z = 8$. При этом $m_z = \{3, 4, 5, 6, 7, 8\}$, так как при $z = 2$ мало вероятно получение сколько-нибудь значительной эффективности от преобразования.

RPOS-преобразование

Исходными данными для работы указанного алгоритма являются последовательности (7) и (8), являющиеся результатом работы программы статистического анализа файла и упорядочения частоты встречаемости букв. Основными типовыми процедурами являются:

- выборка очередного кванта из сегмента обработки;
- сравнение выбранного кванта с соответствующим эталонным квантом;
- формирование логической шкалы позиционного кодирования;
- формирование выходных данных RPOS-преобразования (тега, шкал и т. д.).

Как видно, время работы программы в основном линейно зависит от длины обрабатываемого сегмента данных (в итоге и от размера файла обработки). Поэтому для достижения определенной высокой производительности необходимо применять методы параллельной обработки исходных квантов. Для этого должны быть разработаны типовые модули параллельной обработки квантов.

Рассмотрим пример. Пусть задана следующая последовательность букв (исходный сегмент): "acbaaadbacabaabaaacaabae". Процесс сжатия данной информации методом RPOS-преобразования выполняется следующим образом. Процедура формирования логической шкалы позиционного кодирования здесь выполняется для каждого неповторяющегося кванта (буквы a, b, c, d, e) исходного сегмента (см. рисунок) — массив ЛШПК SG(I). Они будут иметь переменную длину. Длина каждой шкалы зависит от статистики частоты встречаемости буквы, для которой формируется предыдущая шкала. Следовательно, для организации параллельного формирования ЛШПК необходимо учитывать, что предблоком для всех этих блоков будет расчетный блок определения длин отдельных ЛШПК и соответствующих букв исключения. Общее число блоков (I) формирования ЛШПК должно быть $(2^{m_z} - 1)$.

В выполнении процедуры формирования выходных данных (<a, b, c, d, e, массив ЛШПК — SG(I)>) программы RPOS-преобразования целесообразно применять стековые структуры организации памяти.

RGRUP-преобразование

Здесь также исходными данными для работы указанного алгоритма являются последовательности (7) и (8). Сначала из упорядоченной последовательности выбирается группа букв (наиболее часто встречающиеся) в количестве GRUP = 2^k (где $k \geqslant m_z - 2$), удовлетворяющая условиям эффективности сжатия. Должны быть сформированы следующие компоненты сжатого сегмента:

- логическая шкала позиционного кодирования (основная разница от предыдущего метода состоит в том, что здесь формируется всего одна шкала, которая обозначена через L(GRUP)),

$$L(GRUP) = (r_1, r_2, ..., r_{K_z});$$

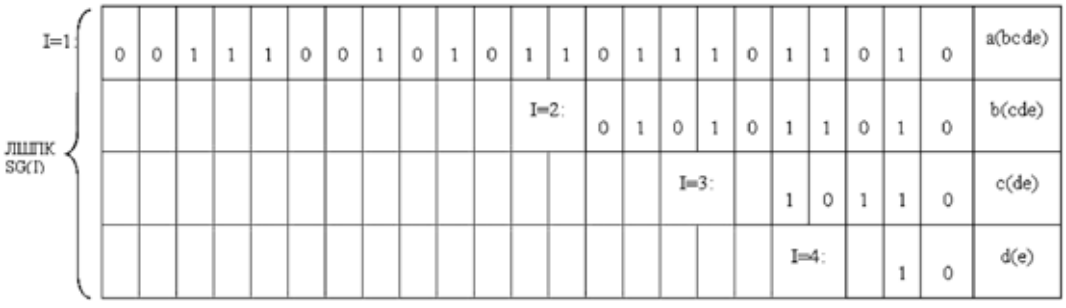
$$r_i = \begin{cases} 1, \text{ если } \beta_i(m_z) \in GRUP, \\ 0, \text{ если } \beta_i(m_z) \notin GRUP. \end{cases}$$

- кортеж (сжатый сегмент)

$$s_0^{(1)} = (y_1, y_2, ..., y_{K_z}), \quad (9)$$

$$y_i = \begin{cases} \beta_i(m_z), \text{ если } r_i = 0, \\ z_i \in Z_0, \text{ если } r_i = 1; \end{cases} \quad (10)$$

Массив ЛШПК



Процесс получения ЛШПК

Таблица 1

Буквы	Частоты встречаемости	Коды (*)
a	12	000
b	4	001
c	3	010
d	1	011
e	1	100
* Коды букв взяты условно.		

$$Z_0 = \{(b_1 b_2 b_3 ... b_k)_n\},\ n = 0,\ 1,\ 2,\ ...,\ (2^k - 1),$$
$$b^k = \{0,\ 1\}. \qquad (11)$$

Выходные данные программы RGRUP формируются в виде кортежа

$$s_1^{(1)} = (\text{тег},\ \text{L(GRUP)},\ s_0^{(1)}).$$

Необходимо специально указать на то, что после получения статистических данных (частоты встречаемости букв) файла в первую очередь должен быть определен эффективный, с точки зрения применения к конкретному файлу, алгоритм преобразования. Эта процедура является буферной между процедурами статистической обработки и непосредственного их преобразования. Как видно из сравнения данного алгоритма с предыдущим, они состоят приблизительно из аналогичных модулей типовых процедур.

Пример. Рассмотрим ту же последовательность букв "acbaaadbacabaaacaabae". Процесс сжатия производится в следующем порядке. Сначала вычисляются и упорядочиваются частоты встречаемости каждой буквы (табл. 1).

Предположим, что $m_z = 3$ (кванты — длина в битах каждой буквы) и $k = 1$ (длина сжатых букв). Тогда число выбранных букв, которое подлежит сжатию, равно 2^k, это будут буквы "a" и "b". Учитывая вышесказанное, формируется следующая последовательность преобразования (табл. 2).

Выходные данные формируются в виде:

$$S_1^{(1)} = (\underbrace{a,\ b,\ 1,\ 0}_{\text{тег}},\ \textit{ЛШПК, сжатый текст}).$$

В сжатом тексте вместо трехразрядного кода букв (a = 000, b = 001) используем одноразрядные коды $a_{сж} = 1$ и $b_{сж} = 0$.

Как уже отмечалось ранее, средства сжатия и развертывания по критерию эффективности сжатия могут быть разделены на два класса. К первому классу могут быть отнесены средства, обеспечивающие максимальную степень сжатия с возможно большей производительностью.

Если при этом используются только программные средства, то они могут занимать большой объем памяти, состоят из большого числа модулей и отличаются сложным алгоритмом обмена программными модулями между различными уровнями памяти. Конечно, при этом целесообразно использовать параллельные алгоритмы обработки данных. С этой точки зрения исследованные методы и алгоритмы сжатия и развертывания данных, основанные на применении логической шкалы позиционного кодирования, являются наиболее подходящими. Также предлагается метод декомпозиции файлов на основе его статических данных на более эффективные подфайлы с применением ЛШПК. При этом каждый подфайл как бы максимально адаптируется к наиболее подходящему методу (алгоритму) сжатия, а все остальные подфайлы подвергаются одновременной (параллельной) обработке, используя возможности мультизадачных режимов современных средств вычислительной техники. Можно наиболее эффективно реализовать декомпозиционный алгоритм сжатия только программными средствами.

Во втором классе для сжатия используются программно-аппаратные средства. При этом аппаратные расходы лимитируются параметрами производительности системы в целом и эффективности сжатия данных. Такая система будет отличаться большим аппаратно-программным наличием разнообразных специализированных модулей сжатия и развертывания данных и, в некоторой степени, сложным алгоритмом их коммутации.

На базе рассмотренных алгоритмов можно разработать программные и программно-аппаратные средства, относящиеся к обоим отмеченным классам. В результате анализа предложенных методов компрессии можно выделить следующие типовые укрупненные процедуры.

1. Процедура квантования двоичной стохастической последовательности с переменным шагом квантования $m = 3,\ 4,\ ...,\ 8$, получения статистических данных и их упорядоченной последовательности (букв алфавитов).

2. Процедура формирования тега и массива ЛШПК.

3. Процедура формирования преобразования данных, т. е. выходных данных — сжатие.

4. Процедура развертывания. Каждый метод компрессии имеет соответствующий метод обратного преобразования (развертывания).

5. Процедура определения эффективного метода (алгоритма) преобразования конкретного сегмента. Данная процедура в основном базируется на анализе локальных статистических данных.

В целях повышения эффективности работы всей системы эта процедура должна выполняться с максимально возможными совмещениями микро- и макроопераций. Как предлагается в данной работе, файлы уже будут иметь в своих паспортных тегах необходимые и заранее полученные статистические данные. Тогда процессы сжатия могут выполняться более производительно, без

Таблица 2

Исходный текст																			
a	C	b	a	a	a	d	b	a	c	a	b	a	A	a	C	a	a	b	a	e
Логическая шкала позиционного кодирования (ЛШПК)																			
1	0	1	1	1	1	0	1	1	0	1	1	1	1	1	0	1	1	1	1	0
Сжатый текст																			
0	010	0	1	1	1	011	0	1	010	1	0	1	1	1	010	1	1	0	1	100

необходимого привлечения для этого дополнительных программных средств. А типовые модули обработки данных дадут возможность коммутировать более эффективную систему сжатия и развертывания данных.

Заключение

В результате анализа представленных выше методов и средств показано, что каждый из них является в некоторой степени специализированным и не в полной степени подходит для широкого и универсального применения. Они должны применяться, и достаточно эффективно, на начальном этапе обработки данных.

Однако предложенные методы сжатия информации, основанные на применении абстрактного множества (алфавита) двоичных квантов без учета структуры его построения и семантики, позволяют найти системное решение задачи компрессии данных для их относительной универсализации и многократности сжатия. Недостатком данных алгоритмов является быстродействие операций сжатия вследствие предварительного статистического анализа входных файлов. Для устранения этого недостатка исследуются возможности аппаратно-программной реализации и распараллеливание обработки для различных шагов квантования при определении эффективности сжатия.

Список литературы

1. **Глушков В. М.** Основы безбумажной информатики. М.: Наука, 1987. 552 с.

2. **Грег Пейстрик.** Как удвоить, а то и утроить, емкость жесткого диска // PC Magazine / Russian Edition. 1992. N 2. С. 15—26.

3. **Chen Z., Gehrke J., Korn F.** Query Optimization in Compressed Database Systems // Proc. 2001 ACM-SIGMOD Int. Conf. Management. Santa Barbara, CA. May 2001. P. 271—282.

4. **Сулейманов А. Ш., Пашаев И. С. и др.** Обнаружение ошибок при передаче числовых массивов в сети ЭВМ с использованием упорядочения // Известия Высших учебных заведений. Серия "Нефть и газ". 1997. № 3—4. С. 52—54.

5. **Курбаков К. И.** Кодирование и поиск информации в автоматическом словаре. М.: Советское радио, 1968. С. 246.

6. **Сулейманов А. Ш., Дамадаев М. М.** Аппаратные и программные средства динамического сжатия данных // Научные труды Национальной академии авиации. 1999. № 1. С. 169—174.

7. **Сулейманов А. Ш., Касумов Н. К.** Об одном методе компрессии в решении проблем резервного копирования и восстановления данных на фоне сближения возможностей SAN и NAS // Известия Национальной академии наук Азербайджана. Серия физико-технических и математических наук. 2004. Т. XXIV. № 2. С. 35—39.

8. **Сулейманов А. Ш.** Семантическая близость и семантические расстояния между текстами // Адаптивные системы автоматического управления. Международный научно-технический сборник. Днепропетровск: Системные технологии, 2007. 164 с.

9. **Аббасов А. М., Сулейманов А. Ш.** Модульный принцип организации нейронных сетей для распознавания смысла слов и текста // Доклады Международной конференции "Информационные средства и технологии". М.: 2003. С. 54—57.

10. **Сулейманов А. Ш.** Интеллектуализация систем обработки информации (монография). Баку: Элм, 2004. 240 с.

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 50.41.29:73.37.81

Н. Н. Подольская, главный специалист,
Всероссийский НИИ радиоаппаратуры, г. Санкт-Петербург, e-mail: podolsky47@inbox.ru

Проектирование человеко-ориентированного программного обеспечения отображения воздушной обстановки

Объектно-ориентированное программирование в сочетании с рациональными способами формирования экранного изображения позволило создать комплекс программ отображения воздушной обстановки, который ориентирован на диспетчера в плане обеспечения высокой реактивности человеко-машинного взаимодействия и имеет структуру, ориентированную на специалиста по разработке и сопровождению программ.

Ключевые слова: *отображение информации, режим реального времени, объектно-ориентированное программирование, реактивность программного обеспечения, человеко-машинное взаимодействие, управление воздушным движением.*

Введение

Комплекс программ отображения воздушной обстановки в составе автоматизированной системы управления воздушным движением (АС УВД) призван в течение многих лет служить диспетчеру в целях обеспечения безопасности воздушного движения. Это большой комплекс, который должен надежно функционировать в режиме реального времени, успевая отображать часто обновляемую входную информацию, в первую очередь, о воздушных судах (ВС), и без задержек реагировать на диспетчерские вводы.

Кроме требований со стороны пользователей, существуют требования и со стороны разработчиков — тех,

кто будет расширять и модифицировать программное обеспечение данной АС УВД, а также тех, кто будет создавать новые программные комплексы, используя крепкую основу.

Таким образом, сам комплекс программ отображения должен быть ориентирован на пользователя, а его внутренняя структура — на разработчика.

Вначале уделим внимание предпочтениям разработчиков, хотя строгие, ясные, логичные программы одновременно являются ключевым условием высоких потребительских характеристик целевого комплекса. Такие программы не только хорошо понимаются другими программистами, но и повышают ясность мысли и эффективность труда их автора в стремлении обеспечить решение поставленных пользователем задач.

Затем рассмотрим метод формирования изображения, сокращающий временные затраты целевых программ с тем, чтобы пользователь не ощущал замедленной реакции на свои действия и, вообще, не ощущал "торможения" отображения.

Структура программ отображения

В настоящее время наиболее естественным способом написания высококачественных программ считается объектно-ориентированное программирование. Столь же естественно то, что в рассматриваемой предметной области одни программные объекты моделируют объекты реального мира (ВС, трассы, взлетно-посадочные полосы, зоны ограниченного использования воздушного пространства и др.), другие — их образы на экране. Так, инструментарий ODS Toolbox фирмы *Barco* [1] оперирует с концептуальными и презентационными объектами, а Graffica SDK фирмы *Graffica* [2] — с сущностями и объектами их реализации. Классы, которые целесообразно создать в языке C++ для объектов первого вида, можно назвать классами состояния, а для объектов второго вида — классами представления.

Каждому классу состояния может соответствовать набор классов представления. Детали реализации каждого класса скрыты за его открытым интерфейсом — свойствами и поведенческими характеристиками, доступными извне, — так реализуется принцип абстракции [3].

Классы, лежащие в основе комплекса программ графического отображения динамической информации о воздушной обстановке и относящиеся исключительно к ВС, показаны на рис. 1 в виде прямоугольников. Прямоугольник, охватывающий другой прямоугольник, соответствует классу, находящемуся с другим классом (классом-компонентом) в отношении *has-a* [3].

Вершиной иерархии классов представления, основанной на отношении *has-a*, является класс "База окон графического отображения динамической воздушной обстановки". В качестве компонента объект этого класса содержит вектор объектов класса "Окно графического отображения динамической воздушной обстановки".

Объект класса "Окно графического отображения динамической воздушной обстановки" (этот класс по-другому может называться "База образов ВС") содержит вектор объектов класса "Образ ВС". Кроме того, он содержит выделенный формуляр сопровождения (ФС) — объект, который отображается при наложении курсора мыши на обычный ФС любого ВС и который содержит дополнительную, по сравнению с обычным ФС, информацию. Класс "Образ ВС" содержит в качестве компонентов классы графических элементов образа ВС.

Опыт разработки показал, что такая система классов не является избыточной. К тому же, она естественным образом позволяет добавлять новые классы.

Применение объектно-ориентированного программирования позволяет в максимальной степени использовать уже созданный программный код. Проиллюстрируем это на примере особых окон представления радиолокационной информации — окон демонстрации ВС, совершающих посадку. Так же как основное окно представления динамической картины воздушной обстановки и окно лупы, они принадлежат классу "База образов ВС" Imagebase. По существу, они отличаются от обычных окон этого класса (Imagebase::type = DEFAULT_TYPE) своими системами координат. В окне глиссады (Imagebase::type = LAND_GLISS) отображается проекция на вертикальную плоскость, проходящую через ось взлетно-посадочной полосы; в окне курса (Imagebase::type = LAND_COURSE) — горизонтальная проекция, где ось абсцисс соответствует оси взлетно-посадочной полосы. Поэтому почти весь программный код, созданный для обычных окон, остается годным и для этих особых. Отличаются, грубо говоря, лишь открытые методы Imagebase::MetersToPixels() и Imagebase::PixelsToMeters(), делающие то, что следует из их названий.

Проектирование комплекса программ отображения воздушной обстановки можно характеризовать уровнем автоматизации. Низкому уровню соответствует низкий уровень абстракции программных компонентов; объектно-ориентированное программирование наличествует

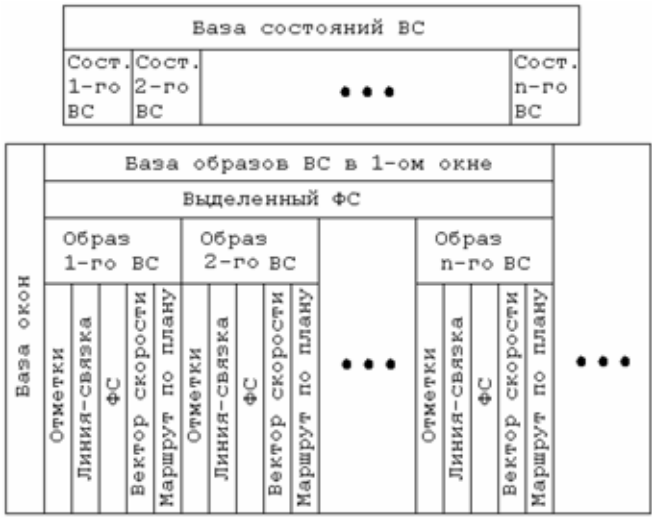


Рис. 1. Схематическое представление классов описания состояния ВС и их графического представления

лишь в виде слабо организованного множества графических объектов общего назначения. Большие трудозатраты таких проектов не приводят к созданию надежных и модифицируемых программ. Высокому уровню соответствует применение узкоспециализированных сред разработки [1]. Их недостаток — низкая вариативность проблемно-ориентированных объектов с высоким уровнем абстракции. Разработчику трудно создать что-либо новое, не имеющее прототипов в рамках среды.

Золотая середина связана с возможностью формирования программных объектов в соответствии с нуждами разработчика на основе легко расширяемого набора абстрактных объектов [2]. Интерфейс объектов такого набора необязательно должен иметь формат API (Application Programming Interface). Главное, чтобы обеспечивался барьер между кодом реализации объекта и кодом, который позволяет его использовать. Вполне допустимым интерфейсом являются заголовочные файлы языка C++, позволяющие создавать экземпляры классов и конструировать производные классы.

Формирование изображения

Операции рисования требуют больших затрат времени. В целях снижения возлагаемой на них работы рисование частично заменяют копированием. Для этого результат рисования сохраняют во вспомогательных пиксельных картах и в дальнейшем при необходимости многократно его используют, до тех пор, пока он не потребует обновления. При этом затраты вычислительных ресурсов перераспределяются: требуется меньше времени, но больше памяти.

В программах отображения воздушной обстановки чаще всего обновляются образы ВС. Реже обновляются образы географических карт и наземных объектов, зон ограниченного использования воздушного пространства и др. Образы ВС принято отображать на экране "на фоне" других образов.

Очевидно, что нерационально проводить ни полную, ни частичную перерисовку "фона" в том случае, если обновилась лишь информация о ВС. Поэтому кроме "результирующей" пиксельной карты, из которой информация копируется на экран, резервируются пиксельная карта для "фона" и по паре пиксельных карт для каждого образа ВС.

Далее приведем примерный алгоритм формирования графической информации (его идея описана в работе [4]) в случае поступления новой информации об i-м ВС (в предположении режима монитора TrueColor).

1. Рисование нового образа i-го ВС в первой пиксельной карте данного ВС $monoPixmap_i$ черным цветом на белом фоне.

2. Рисование нового образа i-го ВС во второй пиксельной карте данного ВС $colorPixmap_i$ требуемыми цветами на черном фоне.

3. Определение прямоугольника P минимальных размеров, охватывающего старый и новый образы i-го ВС.

4. Среди образов других ВС запоминание номеров j тех из них, для которых прямоугольники p_j минимальных размеров, охватывающие их образы, пересекаются с прямоугольником P.

5. Копирование в "результирующую" пиксельную карту фрагмента, соответствующего прямоугольнику P, из пиксельной карты "фона" без трансформации цветов.

6. Копирование в "результирующую" пиксельную карту фрагментов, соответствующих пересечениям прямоугольников p_j с прямоугольником P, из пиксельных карт $monoPixmap_j$ с трансформацией цветов, управляемой логической функцией *И*.

7. Копирование в "результирующую" пиксельную карту фрагментов, соответствующих пересечениям прямоугольников p_j с прямоугольником P, из пиксельных карт $colorPixmap_j$ с трансформацией цветов, управляемой логической функцией *ИЛИ*.

8. Копирование в "результирующую" пиксельную карту фрагмента, соответствующего новому образу i-го ВС, из пиксельной карты $monoPixmap_i$ с трансформацией цветов, управляемой логической функцией *И*.

9. Копирование в "результирующую" пиксельную карту фрагмента, соответствующего новому образу i-го ВС, из пиксельной карты $colorPixmap_i$ с трансформацией цветов, управляемой логической функцией *ИЛИ*.

10. Копирование на экран фрагмента "результирующей" пиксельной карты, соответствующего прямоугольнику P, без трансформации цветов.

Оценим объем памяти, необходимый для реализации алгоритма. Пусть с периодичностью в несколько секунд требуется обновлять на экране образы до 100 ВС. При этом индивидуальные пиксельные карты, пара которых резервируется для образа каждого ВС, имеют размеры 400×400 пикселей и цвет каждого пикселя кодируется 1 байтом. Тогда требуемая память составит $400 \text{ байт} \cdot 400 \cdot 100 \cdot 2 = 32$ Мбайт.

С увеличением (до разумного предела) числа рисованных "слоев" с собственными парами пиксельных карт, за счет сокращения рисования снижаются затраты времени комплекса отображения. Вместе с тем растет объем требуемой памяти.

Естественным представляется выделение в отдельный слой прозрачных [2] или квазипрозрачных (штрихованных) областей-образов зон опасных метеоявлений и ограниченного использования воздушного пространства, отображаемых "между" образами географических карт и образами ВС.

Применение индивидуальных пиксельных карт для образов объектов одного слоя оправдано, когда связанная с объектом изменяемая область мала по сравнению с областью всего изображения. Это условие соблюдается, когда, во-первых, мал размер образа объекта и, во-вторых, мало его перемещение. Для образов ВС вторая часть условия, как правило, выполняется. Для обеспечения выполнения первой части протяженные элементы всех образов ВС относят к отдельному слою.

Например, компонент "Маршрут по плану" класса "Образ ВС" (см. рис. 1) обладает той особенностью, что соответствующий графический элемент может иметь произвольный размер. Этим он отличается от других

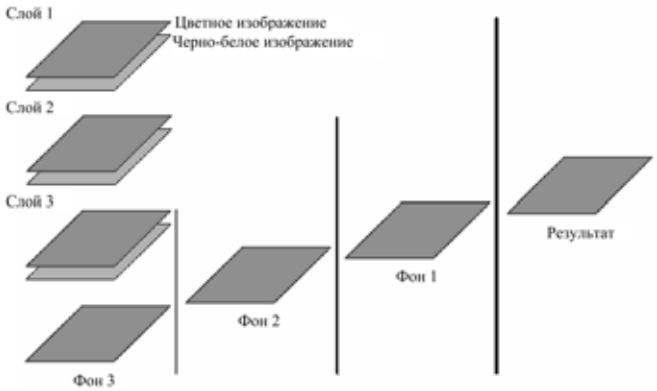


Рис. 2. Схематическое представление построения многослойного изображения

компонентов, для совокупности графических элементов которых можно указать предельный размер охватывающего прямоугольника, что и позволяет резервировать пиксельные карты для образов ВС. Из изображений маршрутов по плану формируется отдельный слой. Можно привязать эти изображения к слою прозрачных областей, но тогда вследствие того что изображения маршрутов по плану (вызываемые диспетчером эпизодически для отдельных ВС) изменяются с частотой обновления радиолокационной информации, существенно превышающей частоту изменения информации о таких областях, будет возникать необходимость непроизводительной перерисовки областей.

На рис. 2 схематически показана работа с пиксельными картами нескольких слоев. "Слоеный пирог" последовательно наращивают на один слой (слой k), каждый раз сохраняя промежуточный результат (фон $k-1$).

В комплексе программ отображения за временной ресурс нередко конкурируют две задачи, требующие обновления изображения:

- обработка информации о ВС, поступившей от комплекса обработки радиолокационной информации;
- обработка действия пользователя, выполненного, например, посредством манипулятора-мыши.

Реакция системы на действие пользователя может характеризоваться максимально допустимым временем отклика. Вообще, с позиций пользователя, допустимым может быть следующий отклик: 1) мгновенный; 2) незамедлительный; 3) сохраняющий непрерывность человеко-машинного взаимодействия; 4) задержанный [5]. Специфика деятельности диспетчера АС УВД требует, чтобы отклики относились к классу мгновенных. Чтобы реактивность системы в отношении ручных операций не становилась недопустимо низкой вследствие асинхронного режима поступления радиолокационной информации, целесообразно создавать очереди типа FIFO (First In, First Out), где поступающая информация ожидает обработки, и эту обработку проводить в фоновом режиме.

Если после обработки взятого из очереди сообщения выясняется, что очередь не пуста, можно без ущерба для актуальности картины воздушной обстановки, ради экономии времени, отложить экранное обновление изображения ВС, зафиксировав геометрию обновляемой области, до выявления опустошения очереди, и только после этого осуществить вывод на экран объединенного прямоугольника "результирующей" пиксельной карты.

Тогда множество копирований "мелких" прямоугольников заменяется одним копированием "крупного" [2].

Сопоставление длительностей операций формирования изображения

Для получения косвенной характеристики длительности формирования изображения проводился подсчет того, сколько раз программа успевала выполнить заданную работу в течение заданного интервала времени, — чем меньше времени требуется на однократное выполнение работы, тем больше оказывается число выполнений за заданное время.

Для сопоставления длительностей операций рисования и копирования в пиксельные карты выполнялись две серии экспериментов, каждая со своей парой процессов. В каждой серии один процесс базировался на предлагаемом подходе, другой вовсе не использовал копирование при построении изображения в пиксельной карте, предназначенной для вывода на экран.

В первой серии оба процесса проводили работу по обновлению визуальной информации об одном ВС, образ которого пересекается с образами девяти других ВС. Первый процесс был устроен в соответствии с изложенными выше принципами, поэтому оцениваемая работа включала одно рисование и 10 копирований. Второй процесс не использовал индивидуальных пиксельных карт для образов ВС, и его работа включала 10 рисований. Табл. 1 содержит данные о заданных интервалах времени и числе выполнений работы в течение этих интервалов для обоих процессов.

Во второй серии оба процесса выполняли работу по обновлению визуальной информации о 10 ВС, образы которых также взаимно пересекаются. Первый процесс был устроен в соответствии с изложенными выше принципами и содержал 10 рисований и 100 копирований. Второй процесс не использовал индивидуальных пиксельных карт и содержал 100 рисований. Результаты приведены в табл. 2.

Таблица 1

Временные характеристики процессов обновления информации об одном ВС

Заданный интервал времени, мс	Число выполнений программного блока, включающего	
	одно рисование, 10 копирований	10 рисований
50	24	10
70	34	13
100	47	15
150	71	19
200	95	22

Таблица 2

Временные характеристики процессов обновления информации о 10 ВС

Заданный интервал времени, мс	Число выполнений программного блока, включающего	
	10 рисований, 100 копирований	100 рисований
50	3	1
70	4	2
100	5	2
150	8	2
200	10	3

Из таблиц с очевидностью следует полезность замены операций рисования операциями копирования как при малом, так и при большом объеме работы по формированию изображения.

Выводы

1. Применение объектно-ориентированного программирования при рациональном построении системы классов позволяет создать большой программный комплекс отображения информации, структура которого обеспечивает простоту модификации и расширения и поэтому ориентирована на разработчика.

2. Рациональные способы формирования экранного изображения позволяют свести на нет снижение реактивности в отношении действий пользователя, которое провоцируется пиковой загруженностью системы.

3. Изложенный подход к проектированию позволил в считанные месяцы создать базовый программный комплекс отображения воздушной обстановки, оперирующий с реальной информацией. Наращивание комплекса с расширением его функциональных возможностей происходит без существенных трудностей.

Список литературы

1. **ODS Toolbox.** Development toolbox for operational display systems // URL: http://www.barco.com/corporate/en/products/product.asp?gennr = 1222

2. **Graffica.** eDEP Development Project. GSDK. Detailed Design Document. EUROCONTROL Development and Evaluation Platform Reference GL/DEP/DDD/1/2.0. 13 Aug. 2007. // URL: http://www.eurocontrol.fr/projects/edep/documents/eDEP_GSDK_DDD.pdf

3. **Солтер Н., Клепер Дж.** C++ для профессионалов. М.: Вильямс, 2006. 912 с.

4. **Эйткен П., Джерол С.** Visual C++ для мультимедиа. Киев: Диалектика, 1996. 384 с.

5. **Seow S. C.** Designing and Engineering Time: The Psychology of Time Perception in Software. Addison-Wesley Professional, 2008. 224 p. Электронная версия книги / URL: http://my.safaribooksonline.com/9780321562944

УДК 628.394(001.57)

С. Н. Коваленко, канд. техн. наук, доц., докторант, преподаватель,
Санкт-Петербургский государственный политехнический университет,
e-mail: kovalenko03@mail.ru

Методика определения допустимых сбросов дренажных вод с учетом стохастического процесса на основе математического моделирования концентрации биогенных загрязняющих веществ в малых реках (на примере Двинско-Печерского бассейна)

Биогенное загрязнение малых водотоков стоками, поступающими с мелиоративных осушительных систем, рассмотрено с точки зрения стохастического процесса. По результатам натурных наблюдений выделены лимитирующие периоды. Для построения кривых обеспеченности при недостатке натурной информации предусматривается удлинение рядов с помощью математического моделирования методом Монте-Карло. Предложена методика назначения концентрации нормативно-допустимых сбросов дренажных вод с учетом стохастического характера процесса загрязнения.

Ключевые слова: *нечерноземная зона РФ, малая река, мелиоративные осушительные системы, биогены, лимитирующие периоды, стохастический процесс, кривые обеспеченности, математическое моделирование, управление качеством водных ресурсов.*

На современном этапе исследований при оценке загрязненности поверхностных вод различными веществами доминирует детерминированный подход. Суть его заключается в сопоставлении результатов химического анализа содержания в пробе воды загрязняющих веществ с нормативно установленной концентрацией данного ингредиента в речной воде.

В процессе анализа натурных наблюдений удалось выделить два напряженных сезона, связанных с особенностями работы дренажных осушительных систем. Эти

системы являются основными загрязнителями природных вод малых рек биогенными веществами, которые поступают с сельскохозяйственных угодий, расположенных на площади бассейнов рек. Для условий нечерноземной зоны России указанные сезоны следующие: весенний (март—май) и летне-осенний (август—октябрь).

Гидрологические и гидрохимические характеристики малых рек в настоящее время изучены слабо. Данная категория рек относится к третьему и четвертому классам по классификации Росгидромета. На водных объектах третьей категории наблюдения проводятся один раз в месяц, на объектах четвертой категории — в определенные гидрологические фазы (в среднем 3—7 раз в год). Эти данные являются практически единственным материалом, который может быть использован для прогностических исследований.

В основу предлагаемого метода расчета допустимого уровня содержания загрязняющего вещества в речной воде положен стохастический анализ исходной информации, результатом которого является получение кривой обеспеченности максимального содержания загрязняющего вещества в створе полного смешения речных и сбросных вод с учетом фонового загрязнения речных вод. Однако, как это следует из сказанного выше, данных натурных наблюдений недостаточно для проведения такого полноценного анализа, поэтому метод базируется на использовании способа удлинения рядов наблюдений с помощью математического моделирования. На полученной тем или иным способом кривой обеспеченности устанавливается вероятность превышения, соответствующая предельно допустимой концентрации (ПДК) конкретного ингредиента. Расчетное значение обеспеченности предлагается назначать с учетом категории реки и ее загрязненности в среднем на 5—10 % меньше обеспеченности ПДК. Расчетное значение концентрации загрязняющего вещества в створе полного смешения речных и дренажных вод является нормативно-допустимым содержанием (НДС). Оно используется для расчетов предельно допустимых сбросов (ПДС) — предельно допустимого содержания загрязняющего вещества в дренажных водах в соответствии с балансовым уравнением.

Математическое моделирование концентрации загрязняющего вещества основано на использовании метода Монте-Карло (метод статистических испытаний). В основе метода лежит натурная информация о максимальной концентрации загрязняющих веществ в речных водах и средней арифметической за выбранный сезон наблюдений (весенний или осенний). Статистический анализ этих данных обнаружил довольно тесную корреляционную связь между ними. Естественно, что за такой короткий срок наблюдений при весьма малой частоте наблюдений сведения о максимальной концентрации загрязняющих веществ в речном стоке являются недостоверными. Между тем среднее арифметическое в статистическом анализе является наиболее устойчивой вероятностной характеристикой. В связи с этим удлинение рядов наблюдений за средней концентрацией загрязняющих веществ в пределах одного периода в речных водах математическим методом представляется достаточно надежной операцией. При этом должно приниматься во внимание наличие или отсутствие автокорреляционных связей в хронологических рядах наблюдений [1]. При отсутствии таких связей (равенстве коэффициента автокорреляции нулю) расширение ряда наблюдений реализуется непосредственно по сглаженной кривой обеспеченности, полученной по результатам натурных наблюдений.

При наличии автокорреляционных связей в хронологическом ряду наблюдений за средними значениями концентрации загрязняющих веществ в пределах одного сезона (весеннего или осеннего) математическое моделирование усложняется и реализуется в соответствии с формулой [1]

$$C_{ci} = [\,\overline{C}_c + r(C_{c_{i-1}} - \overline{C}_c)]K_{pi}(\xi_i,\ C_{v_i}^{\text{усл}}), \qquad (1)$$

где на первом шаге моделирования C_{ci} — это первая искомая средняя за сезон случайная концентрация загрязняющего вещества, распределенная по трехпараметрическому гамма-распределению с автокорреляционной связью; $\overline{C}_c$ — среднее арифметическое значение концентрации из средних величин, данных натурных наблюдений; r — коэффициент автокорреляции; $C_{c_{i-1}}$ — последнее значение концентрации загрязняющего вещества в хронологическом ряду; ξ_i — случайная нормально распределенная величина; K_{pi} — ордината, определенная по кривой обеспеченности в зависимости от выбранного генератором случайных чисел (ГСЧ) случайного числа и условного коэффициента вариации ($C_{v_i}^{\text{усл}}$). Последний рассчитывается по формуле

$$C_{v_i}^{\text{усл}} = \frac{\sigma\sqrt{1-r^2}}{\overline{C}_c + r(C_{c_{i-1}} - \overline{C})}, \qquad (2)$$

где σ — среднее квадратическое отклонение.

Уравнение прямой регрессии C_{mi} по C_{ci}, где C_{mi} — максимальное, а C_{ci} — среднее арифметическое значение концентраций конкретного ингредиента за один сезон i-го года ($i = 1, 2, 3, \ldots, n$, n — число лет наблюдений), запишется в следующем виде:

$$C_{mi} = \overline{C}_m + R\frac{\sigma_m}{\sigma_c}(C_{ci} - \overline{C}_c). \qquad (3)$$

Здесь R — коэффициент корреляции; σ_m и σ_c — средние квадратические отклонения C_{mi} и C_{ci}; $\overline{C}_m$ и $\overline{C}_c$ — соответственно среднее арифметическое из максимальных и средних за весь срок наблюдений значений концентрации загрязняющего вещества в течение осеннего или весеннего периода.

В соответствии с [1] уравнение прямой регрессии (3) используется при удлинении ряда наблюдений за максимальным значением концентрации загрязняющего вещества в выбранный весенний или осенний лимитирующий сезон путем математического моделирования. В процессе математического моделирования по величине C_{ci} определяется точка, лежащая на прямой регрессии (3), затем — случайное отклонение от нее, которое получается в результате розыгрыша. При этом предполагается, что случайные отклонения от прямой регрессии распределены по нормальному закону. Окончательно зависимость для моделирования случайных максимальных

значений концентрации загрязняющего вещества за лимитирующий сезон имеет вид [1]

$$C_{m_i} = \overline{C}_m + R\frac{\sigma_m}{\sigma_c}(C_{c_i} - \overline{C}_c) + \xi_i\sigma_m\sqrt{1-R^2}. \qquad (4)$$

Нормально распределенная случайная величина ξ_i моделируется с помощью кривой обеспеченности, которая находится в первой строчке таблицы биномиального закона распределения [2]. Значения обеспеченности на оси абсцисс моделируются по равномерно распределенному закону случайных чисел, находящихся в диапазоне от 0 до 1 ГСЧ, имеющегося в программном обеспечении. Каждому смоделированному значению обеспеченности с помощью квадратической интерполяции (по методу Бесселя) определяется значение ординаты по кривой нормального закона распределения. Таким образом, в предлагаемом методе математическое моделирование реализуется дважды: вначале моделируются средние значения концентрации загрязняющего вещества за весенний или осенний сезоны, а затем с учетом регрессионных связей — максимальное значение в выбранном сезоне.

Для выполнения математического моделирования концентрации биогенов в малых реках выбраны пять водотоков в бассейнах рек Северной Двины и Верхней Волги на территории Вологодской области [3]. В настоящей работе апробация методики осуществляется на натурной информации по реке Верхняя Ерга.

Река Верхняя Ерга является левым притоком реки Сухоны, впадающим в нее в среднем течении. По принятой классификации Верхняя Ерга относится к малым рекам. На ней установлен один гидрометрический пост наблюдений Росгидромета, совмещенный с гидрохимическим. Натурные наблюдения на данном посту относятся к четвертой категории и проводятся лишь в определенные гидрологические фазы водного объекта. Экспериментальный ряд наблюдений охватывает период с 1978 по 2006 гг. В общей сложности ряд хронологических данных натурных наблюдений за концентрацией аммонийного азота составляет 147 значений. Из них за весенний сезон объем выборки составляет 85 значений. Из натурного хронологического ряда наблюдений за весенний сезон были получены средние арифметические значения по каждому году наблюдений и отобраны максимальные значения соответственно для каждого года. Таким образом, сформировано два ряда величин объемом в 28 значений. Выполнена статистическая обработка средних арифметических *значений концентрации аммонийного азота* для весеннего сезона. Получены следующие статистические характеристики:

- максимальное значение в ряду . . . . . 1,18 мг/л;
- среднее арифметическое значение . . . . . 0,47 мг/л;
- коэффициент вариации C_V. 0,69;
- отношение коэффициента асимметрии к коэффициенту вариации C_S/C_V. . . . . 1,05.

Построена эмпирическая кривая обеспеченности, к которой с помощью статистического критерия согласия Пирсона подобрана аналитическая кривая трехпараметрического гамма-распределения с параметрами $C_V = 0{,}7$, $C_S/C_V = 1$. Расчетное значение критерия Пирсона равно 6,24, а критическое — 10,81.

Построена прямая регрессии между максимальными и средними за сезон концентрациями загрязняющего вещества по данным натурных наблюдений. Коэффициент линейной парной корреляции средних арифметических и максимальных значений за многолетний период для весеннего сезона оказался равным 0,95. Прямая регрессии приведена на рис. 1, где нанесены не только данные натурных наблюдений, но и данные результатов математического моделирования C_c и C_m. Число моделируемых значений равно 100.

На рис. 2 построены эмпирическая кривая обеспеченности и аналитическая кривая обеспеченности трехпараметрического гамма-распределения, подобранная по критерию согласия Пирсона. Эмпирическая кривая включает в себя 28 натурных значений и 100 значений, полученных математическим моделированием. По совместным данным, полученным в результате натурного эксперимента и моделирования, определены для максимальных концентраций аммонийного азота следующие статистические характеристики:

- максимальное значение в ряду . . . . . 2,31 мг/л;
- среднее арифметическое значение . . . . . 0,60 мг/л;

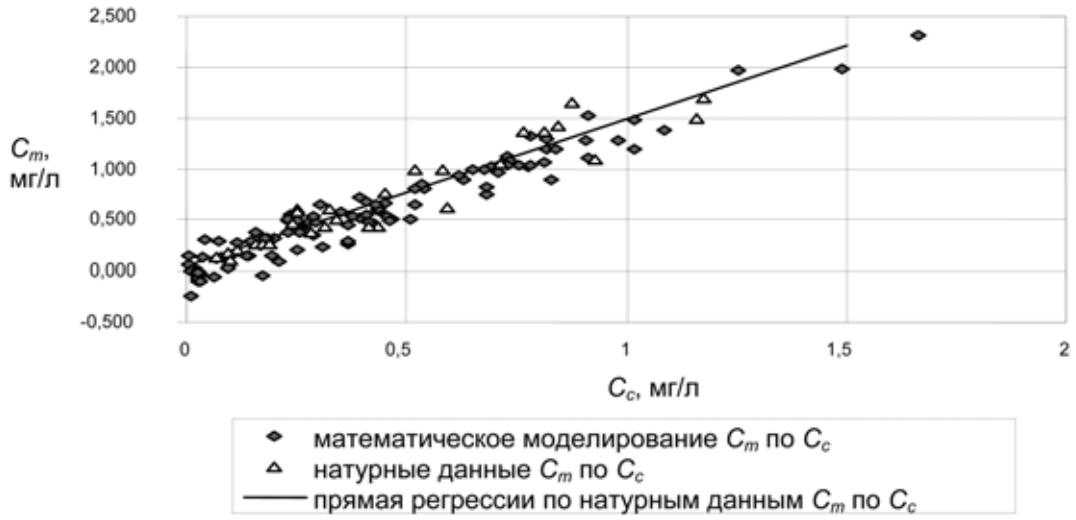


Рис. 1. Результаты математического моделирования максимальных значений концентрации аммонийного азота для весеннего сезона

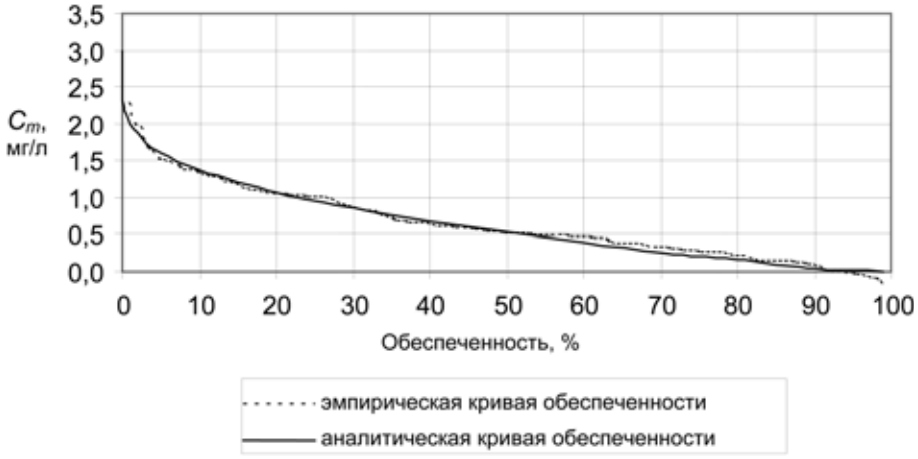


Рис. 2. Кривая обеспеченности максимальных значений многолетних данных за весенний сезон, удлиненных с помощью математического моделирования (штриховая линия), и аналитическая кривая трехпараметрического гамма-распределения (сплошная линия)

- коэффициент вариации C_V 0,81;
- отношение коэффициента асимметрии к коэффициенту вариации C_S/C_V . . . . . .1,12.

Аналитическая кривая обеспеченности трехпараметрического гамма-распределения имеет следующие параметры: $C_V = 0{,}8$; $C_S/C_V = 1$.

Методика определения допустимого уровня загрязняющего вещества в устье магистрального канала, впадающего в водоприемник, заключается в следующем. На графике рис. 2 находим значение, соответствующее предельно допустимой концентрации (ПДК), определяем его обеспеченность. ПДК для азота аммонийного для хозяйственно-питьевого водоснабжения составляет 1,5 мг/л. В данном случае обеспеченность ПДК составляет примерно 7 %. Согласно предлагаемой методике, значение обеспеченности необходимо увеличить. Принимаем обеспеченность допустимого уровня загрязнения равной 10 %, ей соответствует максимальная концентрация 1,3 мг/л. Полученное значение считается расчетным с учетом стохастического характера процесса загрязнения. Расчет концентрации загрязняющего вещества в дренажных водах следует проводить по балансовой зависимости, учитывающей расчетные характеристики расходов речного и дренажного стока, а также концентрации загрязняющего вещества в фоновом створе. В первом приближении можно рекомендовать следующие расчетные значения обеспеченности: 60 % для речного и 40 % для дренажного стоков. Обеспеченность концентрации загрязняющего вещества в фоновом створе рекомендуется принимать равной 50 % по кривой обеспеченности средних концентраций за сезон.

Список литературы

1. **Резниковский А. Ш., Александров А. Ю.** и др. Гидрологические основы гидротехники. М.: Энергия, 1979. 232 с.

2. **Михалев М. А.** Инженерная гидрология. СПб.: СПбГПУ, 2003. 360 с.

3. **Федеральная** служба России по гидрометеорологии и мониторингу окружающей среды. Северное территориальное управление по гидрометеорологии и мониторингу окружающей среды. Государственный водный кадастр. Раздел 1: Поверхностные воды. Серия 2: Ежегодные данные о качестве поверхностных вод. Часть 1: Реки и каналы. Том. 1 (28) РФ (Бассейны рек на территории Архангельской, Вологодской и республики Коми). Архангельск: Росгидромет. 1977. 150 с. 2006. 300 с.

CONTENTS

In article the technique, creations of the program module of protection of a control system by a database, from counterfeit use is considered.

Keywords: algorithm of protection of the program from counterfeit access; protection of program applications; counterfeit access; licence access; protection against counterfeit use; licence use.

The article deals with basic problems of the auto recovery embedded databases. A new method of the auto recovery and the method of priority taking into account the design quality evaluation of the effectiveness of embedded database is suggested.

Keywords: embedded databases, auto recover, modeling, effectiveness, priority.

The paper offers the concept to construct system of three-dimensional geoinformation modeling for decision problems, connected to spatial data and generated by sciences for the Earth. The concept to construct system is stated from the follow positions: problem statement, basic definitions, structures and components, project data and operation principle.

Keywords: geoinformation technologies, geoinformation systems.

In article questions of construction of system of mathematical modelling of sedimentation of mineral oil on a coastal surface are considered. The special attention is given, to the account of influence of processes of sedimentation of mineral oil on plants and their filtration in soil.

Keywords: oil flood, mathematical modeling, a coastal surface, information systems, technogenic safety, decision-making support.

This article is devoted to hardware ways of calculation of procedure of ray tracing, realized in the geometrical processor. All steps of procedure are described in detail. The scheme and elements, with which help the this procedure can be realized in hardware, are considered in this article. Also this article is considered the principle of work of the geometrical processor.

Keywords: three-dimensional objects, geometry processor, ray-tracing unit, depth value, texture shading unit, scan-line, pixel, tile, surface.

The questions of client's conduct estimation in a company on the basis of their individual properties and qualities are considered. A category analysis over of client's properties on the basis of Bayesian approach is produced. The system of category analysis of client's solvency estimation is presented.

Keywords: category analysis, personal service, segmentation, properties of the client, quality customer, scoring.

The problem of operative performance rating of the personnel in the companies is considered. According information database is formed and periodically updated based on the offered scheme of an estimation of professional qualities. By application of the fuzzy logic conclusion method the electronic

"portraits" of employees of the company are synthesised and trends of their professional growth for the accounting periods are determined.

Keywords: linguistic variable, fuzzy set, fuzzy implication, fuzzy relation.

During the data transmission from one department to another the problem of identification of such personal particulars as first name, last name, date of birth, address and etc comes up. The personal identification problem has the most actuality for the people with partial coincide personal particulars. In order to solve this problem the new algorithm and procedure of authentication on the base of indistinct comparison were elaborated for the automatic search of physical persons with partial coincide or synonymous determination of particulars in different databases during information interchange. The developed algorithm has the acceleration of the search speed in 6,7 times as compared with the existing self-training systems on the base of previous authentications, and requires nothing intervention from an operator. The operator hand working report prosecution is shorten to 2—5 minutes by the automatic search. The developed programs provision is used in Togliatti City Information Center.

Keywords: information interchange, personal particulars, identification, identification algorithm, personal identification, automatic search, indistinct comparison.

An adaptive nonlinear filtration method of color videoimages sequence is considered at the big intensity Gaussian noise. Color videoimages sequence is submitted in system RGB consisting of three separate half-tone pictures — a component. The filtration method of color videoimages is based on representation of digital half-tone pictures sequence by three-dimensional discontinuous Markov process with a number of values. Feature of the resulted algorithm of an adaptive nonlinear filtration is its high efficiency at small adaptation time.

Keywords: color videoimages, system RGB, nonlinear filtration, method of adaptation, markov process, the digit binary image, digital half-tone picture.

In the given work was attempt to review of machine algorithms of the universal scheme of a compression and unequivocal unpacking of any binary-coded information is made. Methods of compression which are based on the static data and application of logic scales of the item coding bringing uniformity in processes of compression and unpacking and allowing to develop effective remedy of compression of the data, concerning the universal are offered.

Keywords: compression of data, compression/decompression, coding, redundancy.

The employment of object-oriented programming in combination with efficient screen picture creation procedures allows creating advanced visualization software for air traffic control systems. This software is user-oriented in respect of high real-time computer responsiveness in human-machine interaction. On the other hand, its structure is oriented on software engineers needs.

Keywords: real-time information visualization, object-oriented programming, software responsiveness, human-computer interaction, air traffic control.

Biogenic small water sewage pollution coming from land reclamation drainage systems, considered from the viewpoint of a stochastic process. According to the results of field observations made during sunset. To construct the curves of security with little information provided naturnoy lengthening series with the help of mathematical modeling by the Monte-Carlo method. The method of appointment of legal and allowable discharges of drainage water, taking into account the stochastic nature of the contamination.

Keywords: nechernozemnaya zone of the Russian Federation, a small river, drainage system, drainage, nutrients, sunset times, a stochastic process, supply curves, mathematical modeling, quality management of water resources.

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *О. А. Ефремова*.

Корректор *Е. В. Комиссарова*

Сдано в набор 27.08.2009. Подписано в печать 16.10.2009. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.

Усл. печ. л. 9,8. Уч.-изд. л. 10,98. Заказ 962. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"

142110, Московская обл., г. Подольск, ул. Кирова, 15