

ИНФОРМАЦИОННО- УПРАВЛЯЮЩИЕ СИСТЕМЫ

НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ



3(34)/2008

3(34)/2008

ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ

РЕЦЕНЗИРУЕМОЕ ИЗДАНИЕ

Учредитель

ОАО «Издательство «Политехника»»

Главный редактор

М. Б. Сергеев,
доктор технических наук, профессор

Зам. главного редактора

Г. Ф. Мощенко

Редакционный совет:

Председатель А. А. Оводенко,
доктор технических наук, профессор
В. Н. Васильев,
доктор технических наук, профессор
В. Н. Козлов,
доктор технических наук, профессор
Ю. Ф. Подоплекин,
доктор технических наук, профессор
Д. В. Пузанков,
доктор технических наук, профессор
В. В. Симаков,
доктор технических наук, профессор
А. Л. Фрадков,
доктор технических наук, профессор
Л. И. Чубраева,
доктор технических наук, профессор, чл.-корр. РАН
Р. М. Юсупов,
доктор технических наук, профессор, чл.-корр. РАН

Редакционная коллегия:

В. Г. Анисимов,
доктор технических наук, профессор
Е. А. Крук,
доктор технических наук, профессор
В. Ф. Мелехин,
доктор технических наук, профессор
А. В. Смирнов,
доктор технических наук, профессор
В. И. Хименко,
доктор технических наук, профессор
А. А. Шалыто,
доктор технических наук, профессор
А. П. Шепета,
доктор технических наук, профессор
З. М. Юлдашев,
доктор технических наук, профессор

Редактор: А. Г. Ларионова

Корректор: Т. В. Звертановская

Дизайн: М. Л. Черненко, А. Н. Колешко

Компьютерная верстка: С. В. Барашкова

Ответственный секретарь: О. В. Муравцова

Адрес редакции: 190000, Санкт-Петербург,

Б. Морская ул., д. 67, ГУАП, РИЦ

Тел.: (812) 494-70-36

Факс: (812) 494-70-18

E-mail: 80x@mail.ru; ius@aanet.ru

Сайт: www.i-us.ru

Журнал зарегистрирован в Министерстве РФ по делам печати, телерадиовещания и средств массовых коммуникаций. Свидетельство о регистрации ПИ № 77-12412 от 19 апреля 2002 г.

Журнал входит в «Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертации на соискание ученой степени доктора и кандидата наук».

Журнал распространяется по подписке. Подписку можно оформить через редакцию, а также в любом отделении связи по каталогам: «Роспечать»: № 48060, № 15385; «Пресса России»: № 42476.

© Коллектив авторов, 2008

ОБРАБОТКА ИНФОРМАЦИИ И УПРАВЛЕНИЕ

Мальцев Г. Н., Стогов Г. В. Двумерное преобразование Хаара и особенности его вычисления при обработке оптических изображений 2

МОДЕЛИРОВАНИЕ СИСТЕМ И ПРОЦЕССОВ

Гордеев А. В., Кобзарев Н. А. Метод получения оценок времени выполнения на ранних этапах проектирования сложных распределенных приложений 5

ПРОГРАММНЫЕ И АППАРАТНЫЕ СРЕДСТВА

Ковальчук С. В., Иванов С. В., Кольхматов И. И., Бухановский А. В. Особенности проектирования высокопроизводительных программных комплексов для моделирования сложных систем 10

Лившиц И. Л., Пашковский М. А., Бронштейн И. Г., Сергеев М. Б., Унчун Чо. Характеристики программного обеспечения для проектирования дифракционно-ограниченных оптико-электронных систем 19

КОДИРОВАНИЕ И ПЕРЕДАЧА ИНФОРМАЦИИ

Федоренко С. В. Простой алгоритм декодирования алгебраических кодов 23

ИНФОРМАЦИОННЫЕ КАНАЛЫ И СРЕДЫ

Колбанев М. О., Татарникова Т. М., Малков К. О. Подход к организации адаптивного согласующего центра корпоративной сети 28

СИСТЕМНЫЙ АНАЛИЗ

Ведерников Ю. В., Сафронов В. В. Метод многокритериального ранжирования сложных систем при различных видах неопределенности исходных данных 32

УПРАВЛЕНИЕ В МЕДИЦИНЕ И БИОЛОГИИ

Ковалев Д. В., Шаповалов В. В. Архитектура процессора метаданных 39

Сотников А. Д. Инфокоммуникационные системы и их модели для здравоохранения 46

КРАТКИЕ СООБЩЕНИЯ

Семенов Н. Н. Фрактальный обнаружитель эхо-сигнала 54

СВЕДЕНИЯ ОБ АВТОРАХ

57

АННОТАЦИИ

61

ЛР № 010292 от 18.08.98.

Сдано в набор 05.05.08. Подписано в печать 10.06.08. Формат 60×841/8.

Бумага офсетная. Гарнитура SchoolBookC. Печать офсетная.

Усл. печ. л. 7,5. Уч.-изд. л. 9,0. Тираж 1000 экз. Заказ 280.

Оригинал-макет изготовлен в редакционно-издательском центре ГУАП. 190000, Санкт-Петербург, Б. Морская ул., 67.

Отпечатано с готовых диапозитивов в редакционно-издательском центре ГУАП. 190000, Санкт-Петербург, Б. Морская ул., 67.

УДК 621.391

ДВУМЕРНОЕ ПРЕОБРАЗОВАНИЕ ХААРА И ОСОБЕННОСТИ ЕГО ВЫЧИСЛЕНИЯ ПРИ ОБРАБОТКЕ ОПТИЧЕСКИХ ИЗОБРАЖЕНИЙ

Г. Н. Мальцев,

доктор техн. наук, профессор

Военно-космическая академия им. А. Ф. Можайского

Г. В. Стогов,

доктор техн. наук, профессор

Санкт-Петербургский государственный университет аэрокосмического приборостроения

Рассмотрены особенности вычисления преобразования Хаара при обработке оптических изображений, которые описываются дискретными и непрерывными двумерными функциями. Приводятся обобщенная матрица преобразования Хаара и ее модифицированная форма. Показано, что у групп функций Хаара, соответствующих одинаковой эквивалентной секвенте, дисперсии коэффициентов разложения равны между собой.

При цифровой обработке сигналов и изображений широкое распространение получило использование дискретных ортогональных преобразований в различных базисах [1, 2]. При этом в случае двумерных сигналов, описывающих оптические изображения, двумерные ортогональные системы функций строятся на основе соответствующих одномерных систем. Свойством одномерного и двумерного преобразований Хаара является локальная определенность большинства базисных функций. Это делает неудобным их использование при обработке сигналов, определенных на всем анализируемом интервале (в случае изображений — сцен). В то же время при анализе локальных свойств сигналов, а также при обработке сигналов, локально определенных в области анализа (изображений объектов конечных размеров), свойства функций Хаара могут быть полезными. Локальная определенность и связанная с ней нормировка функций Хаара приводят к особенностям вычисления двумерного преобразования Хаара, которые рассматриваются в настоящей работе.

Одномерные функции Хаара определяются на интервале $0 \leq x \leq X$ следующим образом [1–3]:

$$har_1(x) = har_{00}(x) = \frac{1}{\sqrt{X}};$$

$$har_m(x) = har_{ki}(x) =$$

$$= \begin{cases} \frac{1}{\sqrt{X}} 2^{\frac{k-1}{2}}, & \frac{(i-1)X}{2^{k-1}} \leq x < \frac{(i-\frac{1}{2})X}{2^{k-1}} \\ -\frac{1}{\sqrt{X}} 2^{\frac{k-1}{2}}, & \frac{(i-\frac{1}{2})X}{2^{k-1}} \leq x < \frac{iX}{2^{k-1}} \\ 0, & 0 \leq x < \frac{(i-1)X}{2^{k-1}}, \quad \frac{iX}{2^{k-1}} \leq x \leq X \end{cases} \quad (1)$$

Для одномерных функций Хаара (1) выполняются условия ортонормированности и замкнутости [3], а индексы обычной и двойной нумерации функции при $m \leq 2$ связаны соотношением $m = 2^{k-1} + i$, где $i = 1, \dots, 2^k$. Вся система функций Хаара (1) образуется путем сжатия и сдвига функций, получаемых из функции $har_{11}(x)$. При этом степень сжатия определяется индексом k , а величина сдвига — индексом i . Аналогичные функции с индексами n и lj могут быть введены по координате y в интервале $0 \leq y \leq Y$.

Определим двумерные функции Хаара в области $0 \leq x \leq X, 0 \leq y \leq Y$ в виде произведений одномерных функций вида (1):

$$har_{mn}(x, y) = har_m(x)har_n(y). \quad (2)$$

При таком подходе образованная из исходных ортонормированных и замкнутых одномерных функций система двумерных функций (2) также оказывается ортонормированной и замкнутой [4]

и может быть использована в обобщенном спектральном анализе. Следует отметить, что в работе [5] даны примеры построения на основе процесса сжатия и сдвига систем функций хааровского типа, отличающихся от приведенных. Мы будем рассматривать только «классические» функции Хаара, определяемые выражениями (1) и (2).

В общем случае для любой квадратично-интегрируемой в области $0 \leq x \leq X, 0 \leq y \leq Y$ функции $f(x, y)$ коэффициенты Фурье—Хаара

$$C_{mn} = \int_0^x \int_0^y f(x, y) har_{mn}(x, y) dx dy \quad (3)$$

образуют равномерно сходящийся двумерный ряд

$$\lim_{\substack{M \rightarrow \infty \\ N \rightarrow \infty}} \sum_{m=1}^M \sum_{n=1}^N C_{mn} har_{mn}(x, y) = f(x, y), \quad (4)$$

где коэффициенты C_{mn} при функциях $har_{mn}(x, y)$ определяются выражением [2]

$$C(M, N) = H(M) \cdot F(M, N) \cdot H^T(N), \quad (5)$$

где $C(M, N)$ — матрица коэффициентов Хаара C_{mn} размерностью $M \times N$; $F(M, N)$ — матрица отсчетов исходной функции $f(x, y)$ размерностью $M \times N$; $H(M)$ и $H(N)$ — матрицы преобразования Хаара размерностью $M \times M$ и $N \times N$ соответственно. Интервалы X и Y при соответствующей нормировке отсчетов функции $f(x, y)$ полагаются единичными, а число элементов разбиения выбирается равным $M = 2^a, N = 2^b$, где a и b — целые числа.

Элементы матрицы преобразования Хаара определяются значениями базисных функций (1). Так, при $M = 8$ ($a = 3$) матрица преобразования Хаара имеет вид

$$H(8) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\sqrt{2} & -\sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}. \quad (6)$$

Можно выделить следующие особенности матрицы $H(M)$, существенные с точки зрения вычисления двумерного преобразования Хаара. Во-первых, в отличие от матриц преобразований Фурье и Уолша, она является несимметричной, и в выражении (5) обязательно транспонирование второй матрицы преобразования. Во-вторых, элементы матрицы $H(M)$ принимают значения 0 и $\pm 2^{\frac{r}{2}}$, где r — целое число, что создает определенные неудобства при реализации быстрых алгоритмов вы-

числения. В-третьих, при получении обобщенных выражений для матрицы преобразования Хаара приходится использовать нетрадиционные математические операции, что также затрудняет реализацию алгоритмов вычислений.

Обобщенное выражение для матрицы преобразования Хаара при $M = 2^a$ может быть записано в виде [6]

$$H(M) = H_r^a \left(2^{\frac{r-1}{2}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{[r-1]} \otimes [1 \ -1] \right)^{\delta(r)} \otimes [1 \ 1]^{a-r}, \quad (7)$$

где H_r^a — знак вертикальной суммы ($a + 1$) матриц, нумеруемых по r сверху вниз; \otimes — знак кронекеровского произведения двух матриц; $[G]^{[r]}$ — матрица G в r -й кронекеровской степени;

$$\delta(r) = \begin{cases} 0, & r = 0 \\ 1, & r \neq 0 \end{cases}. \text{ Операции вертикального суммирования матриц, кронекеровского умножения и возведения матриц в степень определены в работах [1, 6].}$$

Первые две из вертикально суммируемых матриц в (7), соответствующие $r = 0$ и $r = 1$, имеют размерности $1 \times M$, а последующие ($a - 1$) матрицы, соответствующие $r = 2, \dots, a$, имеют размерность $2^{r-1} \times M$. В результате получается квадратная матрица размерностью $M \times M$.

Выполнение условия ортонормированности для локально определенных функций Хаара приводит к неодинаковым их уровням на интервалах ненулевых значений у различных групп из 2^{r-1} функций, $r = 1, \dots, a$, и, соответственно, к различным значениям элементов матрицы преобразования Хаара $H(M)$. В то же время при реализации быстрых алгоритмов вычислений желательно использовать матрицы, элементы которых принимают значения 0 и ± 1 . Этому условию удовлетворяет модифицированная матрица преобразования Хаара $H^0(M)$, которая получается при делении элементов матрицы (7) на $2^{\frac{r-1}{2}}$. При $M = 8$ ($a = 3$) модифицированная матрица преобразования Хаара имеет вид

$$H^0(8) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}. \quad (8)$$

С помощью модифицированной матрицы преобразования Хаара $H^0(M)$ вычисляется матрица коэффициентов C_{mn}^0 :

$$C^0(M, N) = H^0(M)F(M, N)H^{0T}(N), \quad (9)$$

а истинные коэффициенты Хаара равны: $C_{mn} = 2^{\frac{k+l-2}{2}} C_{mn}^0$ ($C_{11} = C_{11}^0$). На основе нормировочных коэффициентов $2^{\frac{r}{2}}$ могут быть составлены диагональные матрицы размерностями $M \times M$ и $N \times N$ так, что вычисление матрицы коэффициентов Хаара будет представлять собой умножение пяти матриц: матрицы значений исходной функции, двух модифицированных матриц преобразования и двух нормировочных матриц.

Выбор числа базисных функций в разложении (4) осуществляется следующим образом. Если функция $f(x, y)$ дискретная и число точек по координатам x и y соответственно $M = 2^a$ и $N = 2^b$, то в силу полноты базисной системы функций Хаара [3] отсчеты функции $f(x, y)$ могут быть однозначно представлены суммой MN двумерных функций $har_{mn}(x, y)$. Если же функция $f(x, y)$ непрерывная, то число элементов разложения определяется, исходя из требуемой точности ее аппроксимации конечной суммой Фурье—Хаара. При этом элементы матрицы $F(M, N)$ в выражении (5) соответствуют средним значениям функции $f(x, y)$ на интервалах дискретизации размером $\left(\frac{X}{M}\right) \times \left(\frac{Y}{N}\right)$ по координатам x и y соответственно.

Среднеквадратическая ошибка аппроксимации стационарной случайной функции $f(x, y)$ конечным числом MN двумерных функций Хаара равна:

$$\sigma^2(M, N) = \sigma_f^2 - \sum_{m=1}^M \sum_{n=1}^N \sigma_{mn}^2, \quad (10)$$

где σ_f^2 — дисперсия случайной функции $f(x, y)$; σ_{mn}^2 — дисперсии случайных коэффициентов разложения C_{mn} (3); $-$ — логический символ исключения. Коэффициент C_{11} , исходя из определения функции Хаара $har_{11}(x, y)$, равен среднему значению функции $f(x, y)$ в области анализа и поэтому исключается из рассмотрения.

Для всех коэффициентов C_{mn} , образующих группы, соответствующие фиксированным значениям пар индексов k и l при двойной нумерации функций Хаара в выражении (1), дисперсии σ_{mn}^2 равны между собой. Это следует из выражения

$$\sigma_{mn}^2 = \int_0^X \int_0^Y \int R(\Delta x, \Delta y) har_{mn}(x, y) \times har_{mn}(x + \Delta x, y + \Delta y) dx dy d\Delta x d\Delta y, \quad (11)$$

где $R(\Delta x, \Delta y)$ — автокорреляционная функция стационарной случайной функции $f(x, y)$. Функции Хаара, соответствующие одинаковым значениям индексов k и l в двойной нумерации, отличаются друг от друга только сдвигом по соответствующей координате. Можно показать, что их произведе-

ния $har_{mn}(x, y) har_{mn}(x + \Delta x, y + \Delta y)$, входящие в выражение (11), зависят только от величины сдвигов Δx и Δy . В результате все дисперсии σ_{mn}^2 , соответствующие фиксированным значениям пар индексов k и l , равны между собой.

Используем понятие секвенты как средней частоты пересечения функцией нулевого уровня [5]. Тогда выделенные группы функции Хаара с одинаковыми индексами k и l характеризуются одинаковой эквивалентной секвентой. Следовательно, дисперсии коэффициентов функций Хаара при аппроксимации стационарных случайных функций равны для групп функций, соответствующих одинаковой эквивалентной секвенте. Число таких

групп $\left[a(b+1) - \frac{b}{2}(b-1) \right]$, где $a = \log_2 M$, $b = \log_2 N$, $a \geq b$, а число функций в каждой группе

$$S = \begin{cases} 2^{k+l-2}, & k=l, k \neq 0, l \neq 0 \\ 2^{k+l-2}, & k \neq l, k \neq 0, l \neq 0. \\ 2^{k+l}, & k \neq l, k=0, l=0 \end{cases} \quad (12)$$

Они вносят одинаковый вклад в разложение функций $f(x, y)$.

Таким образом, в работе приведены соотношения для двумерного преобразования Хаара, обеспечивающие реализацию на их основе простых вычислительных процедур. Даются рекомендации по составлению обобщенных матриц преобразования Хаара и выбор их размерности при анализе двумерных случайных функций. Опыт практического использования двумерного преобразования Хаара показывает, что вычислительные матрицы коэффициентов разложения удобно выводить в виде двумерных диаграмм, приведенных в работе [2]. Такие диаграммы, упорядоченные по каждой координате по возрастанию номеров коэффициентов разложения m и n от 0 до M и N , характеризуют как пространственный спектр двумерной функции, так и форму описываемого ею объекта.

Литература

1. Ахмед Н. Д., Рао К. Р. Ортогональные преобразования при обработке цифровых сигналов. М.: Связь, 1980. 248 с.
2. Обработка изображений и цифровая фильтрация / Под ред. Т. Хуанга. М.: Мир, 1979. 320 с.
3. Соболев И. М. Многомерные квадратурные формулы и функции Хаара. М.: Наука, 1969. 288 с.
4. Петров В. П. Прикладная спектральная теория оценивания. М.: Наука, 1982. 432 с.
5. Хармут Х. Теория секвентного анализа. Основы и применения. М.: Мир, 1980. 576 с.
6. Ярославский Л. П. Цифровая обработка сигналов в оптике и голографии. Введение в цифровую оптику. М.: Радио и связь, 1987. 296 с.

УДК 004.421.4

МЕТОД ПОЛУЧЕНИЯ ОЦЕНОК ВРЕМЕНИ ВЫПОЛНЕНИЯ НА РАННИХ ЭТАПАХ ПРОЕКТИРОВАНИЯ СЛОЖНЫХ РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЙ

А. В. Гордеев,

доктор техн. наук, профессор

Санкт-Петербургский государственный университет аэрокосмического приборостроения

Н. А. Кобзарев,

зам. начальника отдела разработки программного обеспечения

ОАО «РИВЦ-Пулково»

Рассмотрены основные средства построения распределенных приложений, возможности средств моделирования. Предлагается новый подход к анализу моделей приложений, позволяющий определять потенциальные ошибки проектирования до выполнения стадии программирования и тестирования, не усложняя работу системного архитектора. Данный метод заключается в преобразовании исходных UML-диаграмм проекта к диаграммам Ганта. Необходимые данные для построения диаграмм Ганта предлагается получать на основе метаданных исходных UML-диаграмм.

Совершенствование технологий разработки программного обеспечения привело к тому, что текстовые документы уже давно не являются единственным описанием как технического задания, так и самого проекта, а широко используются различные графические образы и схемы.

В качестве основных средств графического отображения модели нового проекта можно выделить UML-диаграммы. Многие системы программирования предоставляют свои системы моделирования на основе UML. Все их перечислять не имеет смысла, функциональные возможности у них примерно одинаковы. Для примера можно обратить внимание на продукты Visual Studio Team System или Rational Software Architect как наиболее распространенные и передовые в смысле используемых технологий. Чтобы не пересказывать многочисленную литературу, лишь отметим, что приложения моделирования достигли большой гибкости, поддерживают различные технологии как моделирования, так и архитектуры распределенных приложений [1].

Чем масштабнее приложение, тем сложнее для понимания и восприятия его модель, и модели становятся многоуровневыми, представляющими проект в различных разрезах — например, модель приложений, модель данных, модель публикации. Все эти модели тесно взаимосвязаны и представляют собой достаточно подробное описание про-

екта. Средства моделирования позволяют создавать код из диаграмм и наоборот — диаграммы из кода [1]. Элементы диаграмм не ограничены минимальным набором объектов, а поддерживают высокий уровень абстракций и необходимое количество скрытой вспомогательной информации — метаданных [2]. На основе метаданных производятся различные проверки, сопоставления и, в конечном счете, генерация шаблонов исходного кода. Без использования метаданных такие возможности были бы недоступны, но, тем не менее, диаграммы остаются инструментом проектирования и повышения наглядности. Многие системы позволяют производить тестирование модулей, но такие тесты возможны только после завершения процесса разработки. При этом тестируется работа программиста; произвести автоматизированный анализ архитектуры проекта на ранних этапах разработки с целью получения качественных характеристик не представляется возможным.

После рассмотрения возможностей пакетов моделирования становится ясно, что требуется новый подход к обработке диаграмм, к работе с диаграммами. Простых проверок на корректность схемы явно не достаточно. Очень важным моментом является то, что в рамках создания UML-моделей имеется возможность закладывать в модель дополнительную информацию — метаданные. Наличие метаданных выводит простую и наглядную диа-

грамму на новый уровень, позволяя применить к этой диаграмме автоматизированные средства проверок и преобразований [2].

Для получения системным архитектором интересующих его оценок того или иного варианта проекта фактически требуется выполнить моделирование проекта. Иными словами, нужно определить поведение разрабатываемого проекта в реальной среде. Однако для получения таких оценок требуется ответить на вопрос, как моделировать проект, если он еще не закончен. Если (для примера) в качестве средства моделирования предложить имитационное моделирование на основе сетей Петри и их модификаций [3], то детализированная сетевая модель проекта может стать достаточно большой. Известно, что в то время, как сеть Петри растет, сложность работы с такой моделью растет намного быстрее, и в конечном итоге произвести моделирование и анализ станет практически невозможно [4]. Очевидно, что решение о применении имитационного моделирования не всегда приемлемо, накладные расходы на проведение моделирования и анализ свойств создаваемого проекта могут потребовать слишком много вычислительных ресурсов и времени.

В данном случае требуется так упростить задачу, чтобы достоверность результатов оставалась высокой, а накладные расходы на анализ и моделирование — низкими. Для этого предлагается выполнить такое преобразование исходной модели проекта, чтобы с преобразованной версией модели можно было выполнять относительно несложные операции с целью получения искомым оценок.

Поскольку очень часто системного архитектора интересуют временные характеристики разрабатываемого программного обеспечения, и сетевая структура системы наиболее естественно представляется ориентированными графами, то для работы с временными промежутками как нельзя лучше подходят средства, применяемые для диаграмм Ганта [5].

Итак, мы уже отмечали, что в своей повседневной работе системный архитектор использует средства моделирования, позволяющие строить некоторые UML-диаграммы. Необходимо, не усложняя процесс проектирования и не требуя от архитектора дополнительных действий, произвести преобразование некоторого фрагмента проектируемой диаграммы к диаграмме Ганта, на основе которой можно рассчитать критические пути и потенциальные критические пути в будущем приложении [6].

В качестве исходных данных для преобразования служат связи в исходной диаграмме, а также набор метаданных исходной диаграммы. В процессе формирования диаграммы Ганта в качестве элементов диаграммы выступают некоторые предопределенные процессы из библиотеки процессов.

Важным в методике работы с исходной диаграммой является тот факт, что предполагается

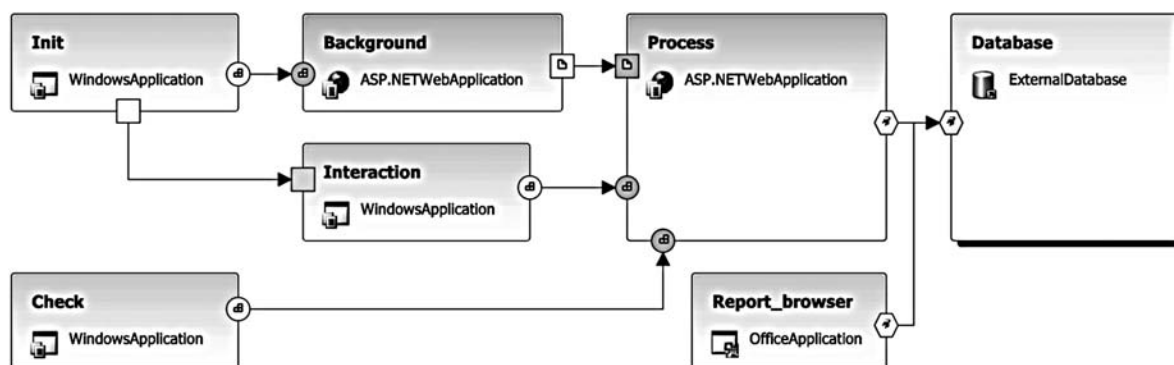
использование экспертных оценок. Экспертные оценки позволят исключить излишние вычисления и необходимость многократного повторного моделирования повторяющихся процессов. В качестве экспертных оценок могут выступать любые источники данных, которым вы доверяете, в том числе опыт предыдущих разработок и собственные измерения. Для каждого процесса в библиотеке метаданных имеется известная функция от некоторого набора параметров, позволяющая получить временные характеристики процесса.

Таким образом, становится возможным применение математического аппарата анализа диаграмм Ганта для исходных UML-диаграмм, и, выделив в UML-диаграмме проекта некоторый путь данных, можно получить для него количественные оценки, на основе которых системный архитектор будет принимать решение относительно приемлемости или неприемлемости рассматриваемого варианта решения.

Диаграммы Ганта выбраны нами по той причине, что они ориентированы на анализ параллельно выполняющихся процессов в течение некоторого промежутка времени; они позволяют перейти от структуры проектируемого приложения к измерению оценки времени выполнения приложения. Поскольку требуется выполнить анализ модели объекта, на основе которого будет создано реальное приложение, то задачу можно описать как необходимость выполнения моделирования проекта приложения или самого приложения, так как проект должен быть подобен результату разработки.

В качестве примера предлагаем рассмотреть диаграмму гипотетической распределенной программной системы. Диаграмма представлена в виде Application Design [2]. Приложение обычно выполняет десятки функций. Для того чтобы проанализировать работу конкретной функции, необходимо выделить эту функцию на модели приложения. Фрагмент приложения, реализующий некоторую функцию, представлен на рис. 1. Каждый из приведенных блоков — процессов может реализовывать некоторый набор других функций, отличных от исследуемой, и хотя эти функции и отсутствуют на схеме, они все же учитываются, поскольку данные о них сохранены в виде метаданных. Опишем предлагаемый подход по шагам.

Итак, на первом шаге требуется для каждого элемента UML-диаграммы построить матрицу зависимостей. Поскольку данное действие предполагается производить автоматически, то оно не приведет к дополнительным трудозатратам системного архитектора и будет являться всего лишь другим представлением исходной диаграммы. Построение такой матрицы предусматривает наличие в диаграмме метаданных. Поскольку некоторый набор метаданных в диаграмме уже присутствует, его придется расширить для хранения дополнительных данных, связанных с информацией о вы-



■ Рис. 1. Схема приложения

полнении процесса. В таблице по диагонали расположены элементы, определяющие оптимистические и пессимистические временные характеристики процессов. Выше диагонали расположены элементы, характеризующие взаимосвязи процессов.

В данном примере в таблице отмечено наличие или отсутствие связи между процессами. Кроме связей в таблице зависимостей указываются метод взаимодействия между процессами, определяющий технологию, и протокол взаимодействия, функция получения временных характеристик процесса взаимодействия и некоторые другие параметры, определяющие связь и передачу данных. Фактически, на данном шаге уже получены исходные данные для построения диаграммы Ганта.

Второй шаг — получить на исходной диаграмме путь прохождения и обработки данных. В нашем примере имеется три функции приложения, которые можно обозначить как функции ввода данных — Init, проверки — Check и просмотра — Report_browser. Требуется выделить только одну функцию, которая в данный момент интересует системного архитектора. Этот процесс назовем процессом редуцирования исходной схемы приложения. Полная схема приложения представлена на рис. 1. Процесс редуцирования предполагает получение пути от исходного процесса диаграммы до конечного посредством анализа взаимосвязей между процессами по метаданным диаграммы. Выделив на схеме начальный процесс — Init и конечный — Database, архитектор имеет возможность добавлять или убирать необходимые для анализа процессы.

Таким образом, процессы Check и Report_browser не попадают в редуцированную схему, поскольку не участвуют в передаче данных от начального процесса к конечному, однако данные об их присутствии все равно остаются в редуцированной схеме в виде метаданных. Редуцированная схема содержит только задействованные в рассматриваемой функции процессы.

На следующем шаге для всех задействованных процессов производятся вычисления необходимых параметров по функциям f_a и f_b . Следует отметить, что для процесса Interaction один из параметров недетерминирован. Это связано с тем, что для продолжения выполнения вычислений требуется вмешательство пользователя. В нашем примере — это время реакции пользователя на запрос системы. Поэтому результат вычисления будет функцией от времени реакции пользователя.

На основе экспертных оценок и полученных выражений связи исходных параметров (метаданных) с временными оценками для каждого процесса требуется получить оптимистические и пессимистические оценки времени выполнения f_a и f_b . Таким образом, рассчитывается ожидаемая продолжительность работы как

$$t_{ij} = \frac{3a_{ij} + 2b_{ij}}{5},$$

где a — оптимистическая оценка длительности операции; b — пессимистическая оценка [6]. Индексы i и j обозначают номера процессов, тогда $t_{i,j}$ — это время выполнения процесса i и переход

■ Взаимные связи между элементами UML-диаграммы

	Init	Background	Interaction	Process	Database
Init	$f_{1a}()$; $f_{1b}()$	+	+	-	-
Background		$f_{2a}()$; $f_{2b}()$	-	+	-
Interaction			$f_{3a}()$; $f_{3b}()$	+	-
Process				$f_{4a}()$; $f_{4b}()$	+
Database					$f_{5a}()$; $f_{5b}()$

к выполнению процесса j . Процессы нумеруются от 1 до ω , а путь между событиями i и j обозначается L_{ij} . Вид диаграммы Ганта для рассматриваемого примера представлен на рис. 2.

На основе этих данных можно рассчитать границы временного промежутка выполнения операции — раннего (р) и позднего (п) срока начала операции и резерва времени выполнения:

$$T_i^p = T_1 + \sum_{i,j \in L_{1i}} t_{ij}, \quad T_i^n = T_\omega - \sum_{i,j \in L_{i\omega}} t_{ij}, \quad R_i = T_i^n - T_i^p.$$

Ранний срок начала выполнения зависит только от времени выполнения предыдущих процессов текущего пути, поздний же срок зависит от всех путей от начального процесса до конечного.

Для оценки выполнения процессов также вычисляются величины ранних сроков начала и окончания процесса, поздних сроков начала и окончания работы, а также свободный и полный резерв времени выполнения:

$$t_{i,j}^{p,h} = T_i^p, \quad t_{i,j}^{p,o} = T_i^p + t_{i,j},$$

$$t_{i,j}^{n,o} = T_j^n, \quad t_{i,j}^{n,h} = T_j^n - t_{i,j},$$

$$r_{i,j}^n = T_j^n - T_i^p - t_{i,j}, \quad r_{i,j}^{cb} = T_j^p - T_i^n - t_{i,j}.$$

Данные границы рассчитываются на основе определения критического пути $L_{кр}$ и времени наступления завершающей операции T_ω . Таким образом, для каждого события рассчитываются ранний и поздний сроки начала и окончания событий, резерв времени события. При этом события, лежащие на критическом пути, имеют нулевой резерв времени выполнения. Для решения поставленной задачи получения оценок параметров разрабатываемого проекта определение критических путей, а также определение временных резервов процессов являются чрезвычайно важными; на их основе возможно формирование как предупреждений и рекомендаций относительно необходимости оптимизации структуры, так и рекомендаций относительно взаимодействия с другими функциями приложения.

Наиболее важным параметром является критическое время выполнения $T_{кр}$ — минимальное

время, за которое может быть выполнен весь путь передачи и обработки данных. Эта величина определяется по параметрам критического пути:

$$T_{кр} = \sum_{i,j \in L_{кр}} t_{ij}.$$

Кроме того, полученные данные позволяют провести оценки продолжительности критического пути с заданной вероятностью. При расчете параметров выполнения операции передачи информации от одного узла диаграммы к другому потребуются следующие величины:

— дисперсия реальной продолжительности операции

$$D_{ij} = \left(\frac{b_{ij} - a_{ij}}{5} \right)^2;$$

— дисперсия критического пути передачи информации

$$D_{кр} = \sum_{i,j} D_{ij};$$

— среднеквадратическое отклонение критического пути

$$\sigma_{кр} = \sqrt{D_{кр}}.$$

Вероятность того, что продолжительность критического пути не выходит за пределы t_{min} и t_{max} :

$$P(t_{min} \leq T_{кр} \leq t_{max}) = \Phi \left(\frac{t_{max} - T_{кр}}{\sigma_{кр}} \right) - \Phi \left(\frac{T_{кр} - t_{min}}{\sigma_{кр}} \right),$$

где Φ — функция Лапласа.

Например, для того чтобы вероятность успешного выполнения операции в заданный срок была в диапазоне $0,25 \div 0,65$, требуется, чтобы

$$t_{min} = T_{кр} - 0,7\sigma_{кр}, \quad t_{max} = T_{кр} + 0,4\sigma_{кр}.$$

Таким образом, на этом шаге нашего алгоритма анализа архитектуры проекта мы осуществляем следующие вычисления:

— на первом проходе анализа полученной диаграммы Ганта для всех процессов вычисляем время выполнения каждого процесса;

— на втором проходе вычисляем ранние времена начала процессов и определяем критическое время выполнения всего пути;



■ Рис. 2. Вид диаграммы Ганта

— на третьем проходе вычисляем поздние времена начала выполнения процессов и, соответственно, определяем критические пути и временные резервы операций.

Следующим шагом является структуризация информации и передача ее из полученной диаграммы Ганта на исходную UML-диаграмму.

Предлагаемый подход позволяет системному архитектору наравне с мощными средствами моделирования архитектуры проекта воспользоваться средствами, позволяющими предсказывать поведение проекта в реальной работе. Однако следует учесть, что данный метод не может использоваться для полной автоматизации анализа проектов. Основная роль, как и раньше, принадлежит человеку. Можно выделить следующие ограничения.

— Анализировать можно только один поток за раз, при этом выбор предмета для анализа остается за человеком. Данный метод является вспомогательным инструментом, позволяющим облегчить работу системного архитектора, указывая на потенциальные проблемы и на потенциальные свободные ресурсы.

— В соответствии с принципами диаграмм Ганта нет возможности анализировать циклические процессы, можно лишь развернуть путь данных в виде совокупности параллельных путей на оси времени.

— Системный архитектор может не обратить внимание на тот самый путь данных, который приведет в конечном счете к неработоспособности всей системы или к очень большим задержкам. Ответственность за это, как и прежде, лежит на человеке. Обнаружив одно из узких мест системы и решив проблему, архитектор может не заметить другой проблемы, вызванной сложными процессами передачи данных, и, как следствие, не решить всех проблем приложения на этапе проектирования.

— Системный архитектор остается ответственным за определение ключевых параметров, отвечающих за функционирование каждого процесса, и должен определять, какие параметры могут привести к худшему сценарию выполнения.

— Системный архитектор отвечает за формирование метаданных диаграмм, и это дополнительно усложняет его работу. При использовании рассматриваемого метода анализа проектов распределенных программных систем предполагается наличие библиотеки процессов, для которых уже определены ключевые параметры процессов, а также их отношение к оценке временных параметров.

Однако, несмотря на наличие ограничений и необходимость задавать параметры для каждого объекта диаграмм, системный архитектор имеет возможность автоматизированно получать предварительную информацию о поведении системы в реальных условиях еще на этапе проектирова-

ния, что, безусловно, перевешивает любые минусы накладных расходов.

В рассмотренном примере одним из параметров, как отмечалось, является недетерминированная величина. Поскольку она одна, то в результате время выполнения получается как функция от этой величины. А результатом анализа всего пути будет ответ на вопрос, в каких границах должна лежать эта величина и какое влияние она окажет на процессы. Если величина меньше некоторого параметра, то процесс Interaction лежит не на критическом пути и нужно обратить внимание на оптимизацию процесса Background. В противном случае процесс Background является не критичным и его можно, например, исполнять с пониженным приоритетом.

В заключение можно оценить накладные расходы на проведение оценочных расчетов и задания расширенных параметров диаграммы приложения. В начале статьи было показано, что для успешного решения поставленной задачи требуется применять метод, который будет иметь по возможности минимальные накладные расходы на анализ, либо эти расходы будут расти не очень быстро с ростом масштабов системы. Для системы из N элементов требуется вычислить:

- N продолжительностей работ;
- $N \cdot 4$ ожидаемых границ времени выполнения;
- $N \cdot 2$ резервов времени выполнения.

Таким образом, для получения ключевых параметров диаграммы требуется провести $N \cdot 7$ операций. Процессы, имеющие нулевой резерв выполнения, образуют критический путь диаграммы; таким образом, задача вычисления критического пути становится тривиальной. Объем вычислений относительно размеров системы растет линейно, что более чем удовлетворительно, и преимущество перед методами имитационного моделирования очевидно.

Литература

1. Kunal Mittal. Introducing IBM Rational Software Architect. http://www.ibm.com/developerworks/rational/library/05/kunal/?S_TACT=105AGX99&S_CMP=CP
2. Hundhausen R. Working with Microsoft visual studio 2005 team system. Microsoft Press, 2005. 320 p.
3. Гордеев А. В. Моделирование дискретных параллельных систем с использованием F-сетей Петри // Информационно-управляющие системы и сети. Структуры, моделирование и алгоритмы: Сб. ст. СПб.: Политехника, 1999. С. 56–69.
4. Hurson A. R., Zelkowitz M. V. Advances in Computers. Academic Press, 2005. 312 p.
5. Program Evaluation and Review Technique. <http://en.wikipedia.org/wiki/Pert>
6. Ковылев Ю. И., Тимошин С. И., Тихомиров О. И. Организация, планирование и управление приборостроительным предприятием / ЛИАП. Л., 1986. 70 с.

УДК 681.3.069, 681.324

ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ПРОГРАММНЫХ КОМПЛЕКСОВ ДЛЯ МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ

С. В. Ковальчук,

младший научный сотрудник

С. В. Иванов,

младший научный сотрудник

И. И. Колыхматов,

стажер-исследователь

А. В. Бухановский,

доктор техн. наук, профессор

Санкт-Петербургский государственный университет информационных технологий,
механики и оптики

Рассматриваются особенности проектирования высокопроизводительных программных комплексов для моделирования сложных систем на примере задачи моделирования экстремальных явлений в атмосфере и океане. Обсуждается подход к формализации структуры комплекса, сочетающий в себе специфику концепций SOA и REST, в соответствии с задачей отображения компонентов сложной системы на параллельную вычислительную архитектуру. Предложен способ оценки архитектуры на основе анализа параметрических моделей параллельной производительности.

Введение

Высокоуровневое проектирование (или разработка архитектуры ПО — программного обеспечения) ставит своей целью формализацию и обоснование внутренней структуры ПО, которая наилучшим образом удовлетворяет проектным требованиям при заданном наборе ограничений в области работоспособности, безопасности, безотказности, защищенности [1].

Эволюция способов формализации структуры ПО определяется, в первую очередь, потребностью снижения сложности процессов разработки, тестирования и поддержки программных продуктов на фоне увеличения общей сложности решаемых задач. Это последовательно порождает структурный [2], объектно-ориентированный [3] и компонентно-ориентированный [4] подходы в инженерии ПО. Дальнейшая тенденция функциональной изоляции компонентов отражается в архитектурах на базе SOA (Service Oriented Architecture) [5], REST (Representational State Transfer) [6], AOA (Aspect Oriented Architecture) [7] и др. Эти подходы достаточно успешно используются при разработке коммерческих приложений и часто могут

определять весь технологический процесс создания ПО [8].

При разработке наукоемкого ПО (scientific computing software engineering) необходимо учитывать, что приоритеты проектных требований, их структура, процесс проектирования, создания, проверки, внедрения, поддержки и использования обладают рядом индивидуальных особенностей [9]. Во многих случаях важным критерием работоспособности ПО является производительность (productivity), характеризующая получение результата вычислений с приемлемой точностью за заданное время. Она обеспечивается за счет применения технологий высокопроизводительных вычислений в многопроцессорных и (или) распределенных средах.

Использование традиционных компонентно-ориентированных подходов к проектированию распределенных программных систем (на основе DCOM [10], CORBA [11], Enterprise JavaBeans [12] и пр.) допустимо не во всех случаях, поскольку они не ориентированы непосредственно на достижение наибольшей производительности. Несмотря на то что для наукоемкого ПО предлагаются специфические проблемно-ориентированные подхо-

ды, в частности ССА (Common Component Architecture) [13], в общем случае архитектура таких систем определяется особенностями решаемой задачи, спецификой предметной области и характеристиками параллельной вычислительной системы. В данной статье обсуждаются особенности проектирования наукоемкого ПО для моделирования сложных систем на примере высокопроизводительного программного комплекса моделирования экстремальных явлений в атмосфере и океане, разрабатываемого в рамках проекта ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2012 годы».

Специфика компьютерного моделирования сложных систем

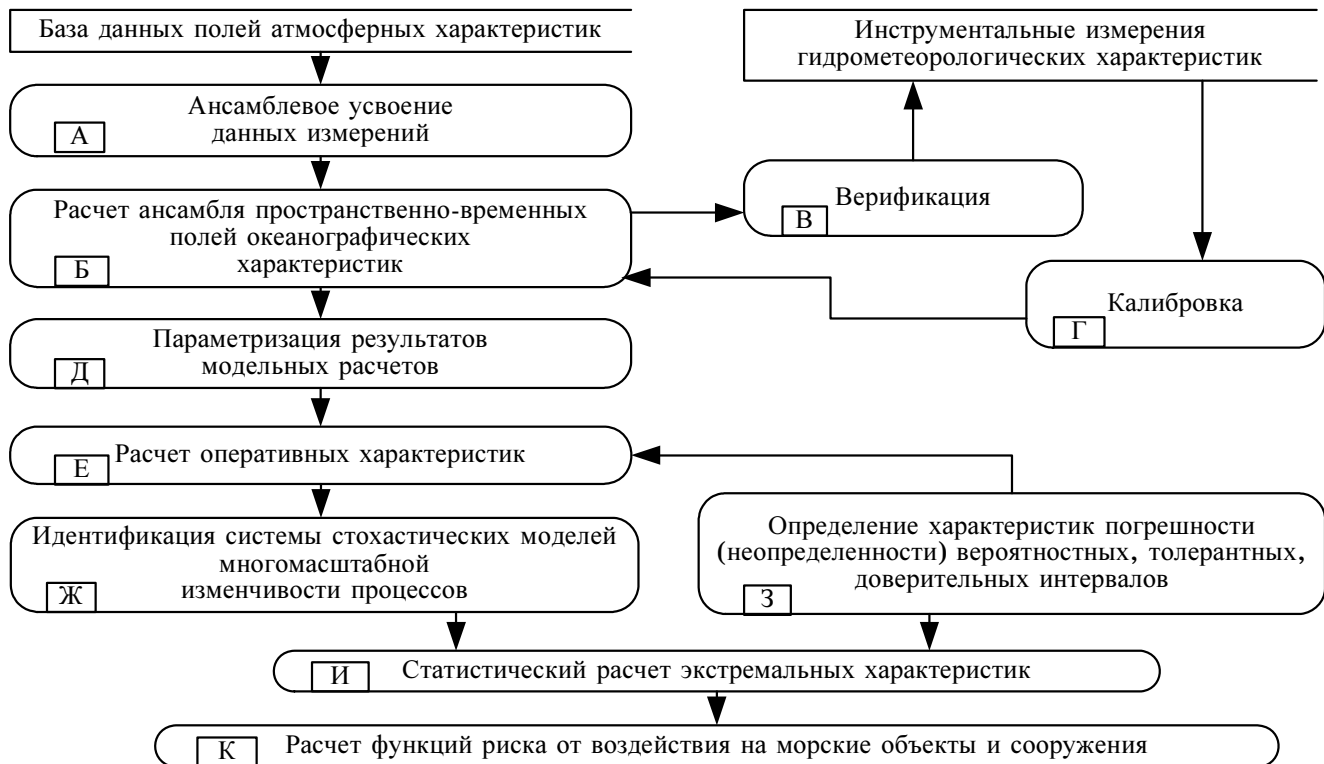
Моделирование сложных систем (complex system simulation) является одним из приоритетных направлений развития наукоемкого ПО, порождающим новый этап развития «электронной» науки (e-Science) [14]. Под сложной системой [15] понимается система, которая:

- а) состоит из очень большого количества элементов;
- б) допускает «дальние» связи между элементами;
- в) обладает многомасштабной (пространственно-временной) изменчивостью. Как следствие, поведение системы описывается целым комплексом взаимодействующих моделей (в общем случае ос-

нованных на различном математическом аппарате и соответствующих различным областям знания) и требует существенных вычислительных затрат.

Характерным примером сложной системы является система «океан-атмосфера», рассматриваемая в диапазонах мелкомасштабной, синоптической, сезонной, межгодовой изменчивости (временные масштабы от секунд до десятилетий и пространственные — от метров до тысяч километров). Она характеризуется гидрометеорологическими полями атмосферного давления, скорости ветра, температуры воздуха, морского волнения, температуры и солености, течений и уровня моря. Необходимость исследования такой системы связана с задачей проектирования судов и сооружений на шельфе, что требует оценок экстремальных гидрометеорологических явлений как сочетаний характеристик, возможных один раз в T лет. Величина T определяется инженерными требованиями к конкретному объекту или сооружению и составляет от 10 до 10 000 лет. Экстремальные характеристики обычно не обеспечены данными наблюдений и, как следствие, должны определяться на основе компьютерного моделирования [16].

Процедура моделирования соответствует функциональной схеме, представленной на рис. 1. На первом этапе осуществляется подготовка метеорологических данных «А» (атмосферного давления и ветра) за климатический интервал (от 30 лет для выполнения непрерывных расчетов полей вол-



■ Рис. 1. Функциональная схема программного комплекса моделирования экстремальных гидрометеорологических явлений

нения, течений и уровня моря по гидродинамическим моделям «Б» с учетом верификации результатов расчетов «В» и их калибровки по данным измерений «Г». После этого выполняется параметризация «Д» и последующая статистическая обработка однородных выборок больших объемов данных «Е». Целью обработки является идентификация семейства стохастических моделей «Ж», по которым методом Монте-Карло (с использованием зависимых испытаний) воспроизводится ансамбль многомасштабной изменчивости гидрометеорологических полей. На основании этих данных выполняется оценивание экстремальных (ненаблюдаемых) явлений «И», например сочетаний скоростей ветра, высот волн и скоростей течений в шторме, возможном один раз в 100 и более лет, по которым рассчитывается степень риска для инженерных объектов «К». Подробнее применяемые методы и технологии описаны в работах [16, 17].

Ресурсоемкость расчетов существенно зависит от расчетной акватории и характеристик задачи (например, периода повторяемости T); в общем случае она составляет порядка нескольких петафлоп. Например, для Каспийского моря выполнение таких расчетов на одном процессоре AMD Opteron 275 занимает 446 суток [18]. Поэтому каждый из функциональных компонентов «А» — «К» должен предусматривать параллельное выполнение.

Поскольку экстенсивный подход на уровне автоматического распараллеливания программных кодов [19] далеко не всегда эффективен, разработка высокопроизводительного ПО моделирования сложных систем требует формализации параллельной структуры компонентов на этапе проектирования. В то же время применение традиционного для параллельных вычислений подхода на основе РСАМ-концепции [20] затруднено тем, что механизм моделирования сложных систем включает в себя иерархическую систему моделей, связанных между собой, основанных на различном математическом аппарате и имеющих различную ресурсоемкость. С точки зрения проектирования структуры ПО, это отражается в том, что:

- функциональные компоненты могут иметь ограничения по использованию — функциональные, платформенные, лицензионные и пр.;
- функциональные компоненты могут быть написаны на различных языках, могут использовать разнообразные системы ввода-вывода;
- структура исходных кодов компонентов (в случае их доступности и возможности внесения изменений) может быть трудномодифицируемой вследствие индивидуальных особенностей разработки;
- компоненты используют различные интерфейсы параллельного программирования и ориентированы в общем случае на разные вычислительные архитектуры.

Например, на этапе «Б» для гидродинамического моделирования волнения применяется свободно-распространяемая модель SWAN [21], ко-

торая реализована в стандарте языка Fortran 90 и распараллеливается с использованием интерфейсов OpenMP и MPI [22]. Эта модель признана мировым сообществом; любая ее модификация (содержательная или программная) требует проведения дополнительной сертификации. С другой стороны, на этапе «И» для расчета экстремальных характеристик гидрометеорологических процессов применяется авторская модель BOLIVAR. Несмотря на то что эта модель рекомендована Всемирной метеорологической организацией [23], коллектив разработчиков имеет права на модификацию ее кодов, в то время как модификация самого метода потребует дополнительных согласований.

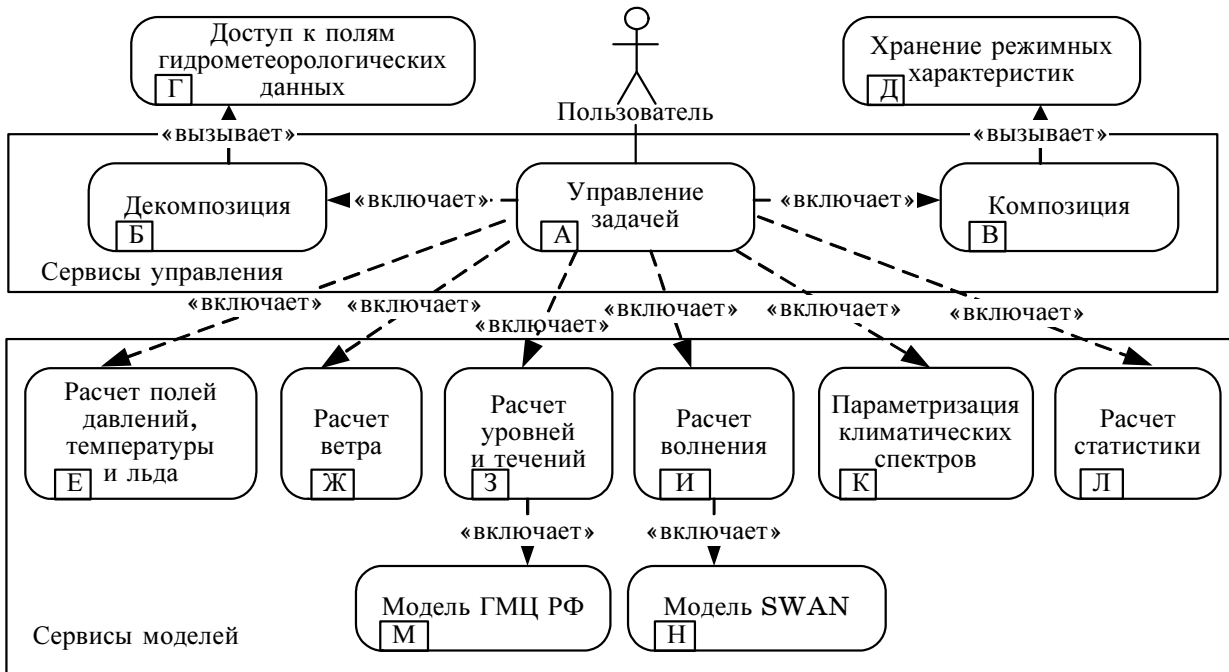
Специфика разработки архитектуры высокопроизводительного программного комплекса состоит в том, что он должен соответствовать архитектуре вычислительной системы. Ниже рассматривается решение, ориентированное на перспективный суперкомпьютерный вычислительный комплекс Росгидромета РФ, состоящий из двух кластеров: кластера SGI Altix 4700 с пиковой производительностью 11 ТФлопс и кластера SGI ICE 8200 с пиковой производительностью 16 ТФлопс.

Разработка программной архитектуры

Классический подход к разработке архитектуры программной системы предусматривает операции структурирования системы, описания свойств ее компонентов и установления связей между ними. Для ПО моделирования сложных систем структурные компоненты соответствуют различным функциональным компонентам, например «А» — «К» на рис. 1. Потому в данном случае принципиальной проблемой является организация взаимодействия (как между компонентами, так и внутри них) таким образом, чтобы обеспечивалось наиболее эффективное использование ресурсов вычислительной системы.

Сервисно-ориентированная архитектура. В общем случае сложная система описывается иерархией моделей, соответствующих различным диапазонам изменчивости и функциональным подсистемам. При этом диапазоны изменчивости связаны между собой параметрически, т. е. выходные данные, полученные по одной модели, являются управляющими данными для модели более низкого уровня. Как следствие, соответствующее ПО может быть интерпретировано в рамках сервисно-ориентированной архитектуры (SOA) [5]. Это позволяет добиться максимального уровня изоляции частей программного комплекса, тем самым делая более строгой его структурированность и облегчая процесс разработки и тестирования. На рис. 2 представлена общая структура взаимодействия сервисов, соответствующая функциональной схеме на рис. 1. Для нее характерна следующая типизация сервисов [24].

- Управляющие сервисы (композиции, декомпозиции, управления задачей).



■ Рис. 2. Сервисно-ориентированная архитектура программного комплекса моделирования экстремальных гидрометеорологических явлений в форме диаграммы вариантов использования

- Сервисы доступа к данным (входная гидрометеорологическая информация, результаты расчетов).
- Вычислительные сервисы (модели подготовки метеорологической информации, гидродинамические и стохастические модели, статистическая обработка и расчет экстремумов).

В общем случае функциональные компоненты, указанные на рис. 1, могут быть представлены в виде совокупности сервисов, исполняющихся на отдельных узлах вычислительной системы и обеспечивающих наиболее полное использование вычислительных ресурсов. Следует отметить, что часть сервисов присутствует в единственном экземпляре, обеспечивая централизованное управление комплексом моделей в целом, в то время как другая часть исполняется на каждом из узлов вычислительной системы.

Вычислительные модули представляют собой сервисы доступа к используемым моделям. Взаимодействие моделей организуется управляющим модулем вычислительной системы, на который ложится ответственность за определение потоков данных между сервисами моделей. В то же время потоки данных в рамках сервисов также обладают довольно сложной структурой, обусловленной фиксированным интерфейсом модели. Таким образом, каждый из сервисов моделей обязан предоставить интерфейс для унифицированного обращения с моделью, организуя потоки данных таким образом, чтобы эффективно использовать вычислительные ресурсы системы.

Использование SOA позволяет путем формализации интерфейса взаимодействия с сервисом уп-

ростить процедуру разработки управляющих модулей. При совмещении данного решения с применением паттерна проектирования «Адаптер» [25] становится возможной легкая модификация и замена одних моделей другими с аналогичной функциональностью. Изоляция сервисов и четкое описание интерфейса позволяют упростить схему взаимодействия, не учитывая при этом особенностей внутренней реализации каждого из разрабатываемых компонентов. В рамках решаемой задачи для реализации сервисов была использована среда .NET Framework, реализующая технологию Windows Communication Foundation (WCF), что в общем случае дает возможность строить кросс-платформенные сервисы различных видов.

Предоставление доступа к данным. Моделирование сложных систем связано с оперированием большими объемами информации. Потому в рамках данной задачи применяется технология REST, ориентированная на обеспечение эффективного представления ресурсов [6, 26]. Доступ к ресурсам осуществляется через фиксированный интерфейс, что обеспечивает унификацию ресурсов и упрощает взаимодействие с пользователем. Совместное использование технологий SOA и REST позволяет добиться организации унифицированного доступа к обрабатываемым данным. На рис. 2 часть рассмотренных сервисов («Г», «Д») отвечает за предоставление доступа к большим объемам данных, необходимых для работы программного комплекса.

В процессе организации взаимодействия системы с разделяемой памятью возникает необходи-

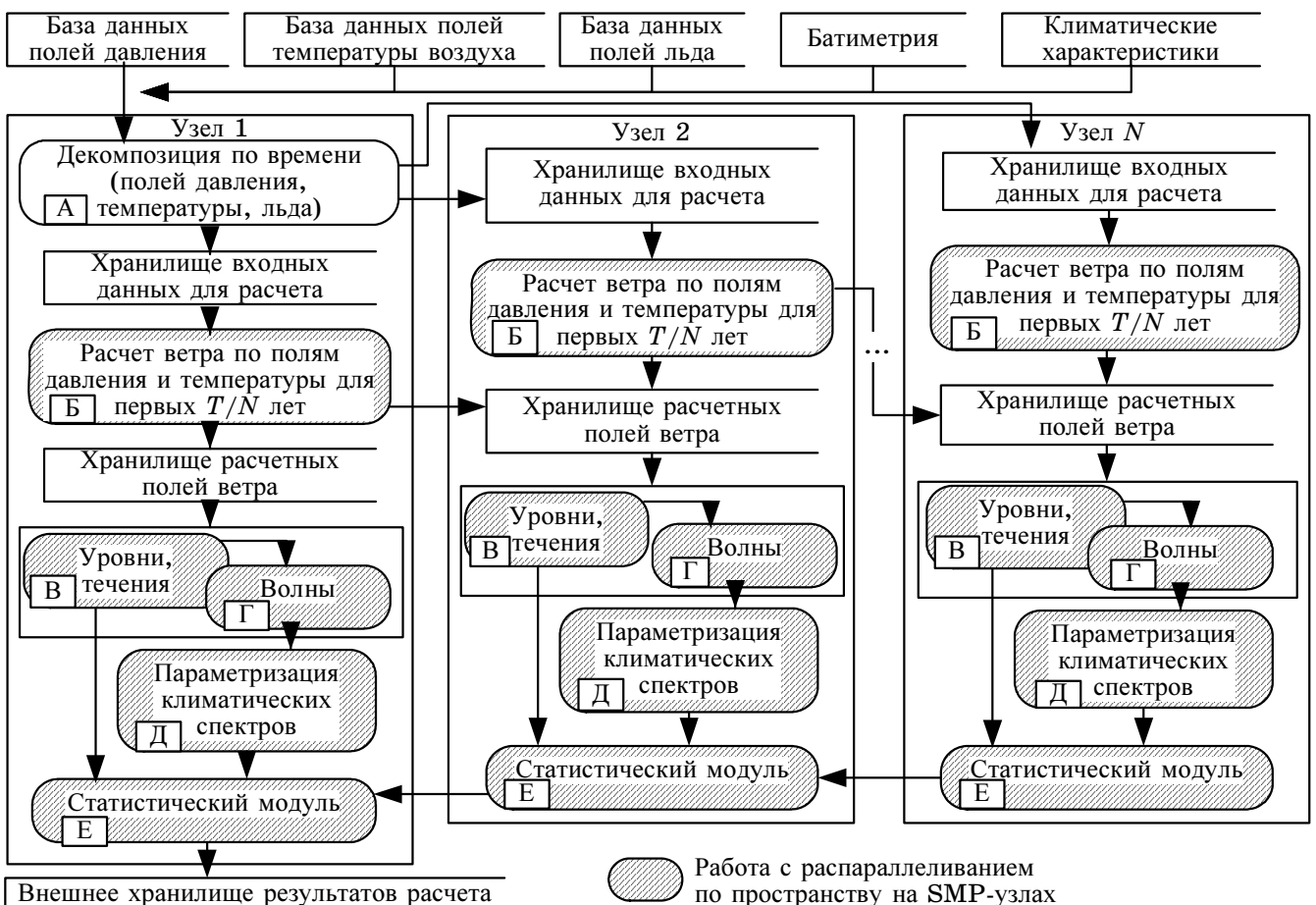
мость решения задачи эффективной коммуникации. При использовании среды исполнения .NET Framework возможна передача данных при помощи механизма сериализации/десериализации данных. Этот механизм позволяет осуществить преобразование объектов, доступных программному коду, к форме, удобной для передачи посредством механизма потоков. Это дает возможность оптимизировать процесс передачи данных по сети за счет сжатия, синхронизации и дополнительного контроля передачи данных путем как декларативного, так и императивного управления процессом сериализации/десериализации [27].

Несмотря на то что для коммерческого ПО традиционным способом хранения данных являются реляционные базы данных, для представления расчетной гидрометеорологической информации эффективнее использовать файловые форматы GRIB и NetCDF. Однако каждому из расчетных файлов сопоставляются метаданные (характеристики сетки, условия расчета и пр.), которые допустимо хранить в реляционной базе данных.

Отображение на параллельную вычислительную архитектуру. В процессе разработки программного комплекса необходимо учитывать осо-

бенности аппаратной архитектуры и среды исполнения, что позволяет добиться эффективной работы программного комплекса в целом.

Задача отображения архитектуры программного комплекса на параллельную вычислительную архитектуру определяет соотношение четырех уровней взаимодействия между компонентами системы. *На аппаратном уровне* взаимодействие происходит в рамках системы, состоящей из ряда соединенных коммуникационной средой вычислительных узлов, на каждом из которых установлен один или более процессоров (ядер), имеющих доступ к общей для них памяти. *Системный уровень взаимодействия* оперирует объектами операционной системы, процессами, потоками, нитями в пределах одного вычислительного узла и взаимодействиями на уровне сетевых коммуникаций при синхронизации различных узлов. *Программный уровень* предполагает применение библиотечных средств, предназначенных для коммуникации, синхронизации, балансировки и других задач организации параллельных вычислений. К средствам, традиционно относимым к реализации параллельных вычислений, можно причислить MPI (для систем с разделенной памятью)



■ Рис. 3. Схема отображения программной архитектуры на параллельную вычислительную архитектуру

и OpenMP (для систем с общей памятью). Средства, позволяющие осуществлять межпроцессное взаимодействие, обеспечивают возможность коммуникации между отдельными процессами в рамках локального узла. Возможны различные способы реализации IPC (Inter-Process Communication) как с использованием механизма потоков, каналов, очереди сообщений MSMQ (Microsoft Message Queuing) и пр., так и более высокоуровневые реализации вплоть до возможности работы .NET Remoting посредством IPC. Наконец, *логический уровень* представляет собой сам вычислительный алгоритм, который выполняется с использованием средств нижних уровней, обеспечивая оптимальную загрузку вычислительных ресурсов системы.

В рамках данного программного комплекса реализуется иерархическая схема распараллеливания. Она включает в себя компоненты трех категорий.

- Центральный управляющий модуль, предоставляющий внешним подсистемам сервис для осуществления вычислений и использующий сервисы узлов для непосредственного контроля процесса работы.

- Исполняющие модули, производящие вычисления на узлах и работающие с каждым из вычислительных компонентов.

- Адаптированные модели, реализующие параллельный вариант расчетных алгоритмов для систем с общей памятью.

На рис. 3 представлена схема отображения. Центральный управляющий модуль (ввода-вывода) выполняет предварительные операции с данными. Затем происходит распределение массива метеорологических характеристик (давление, температура воздуха) и льда объемом T лет между p вычислительными узлами. После этого выполняется процедура усвоения, а также по полям давления и температуры рассчитывается ветер, «Б». Эта процедура на каждом узле распараллеливается по пространству. Производится репликация данных между смежными узлами, что необходимо для последующей сшивки результатов расчетов по гидродинамическим моделям «В» — «Г». На каждом узле совместно (на каждом шаге) выполняется гидродинамическое моделирование волнения, течений и уровня моря, также распараллеленное по пространственной области в рамках общей памяти узла. Затем результаты расчетов передаются в модуль параметризации «Д». Полученные в итоге поля параметров передаются в статистический модуль, выполняется агрегация результатов. Статистические модули «Е» на каждом узле также допускают многоуровневое распараллеливание — по ансамблю, по индексующей переменной, по таксонам, на основе принципа перемешивания [28].

Использование гибридно-ориентированного подхода позволяет гибко реализовывать различные схемы параллельных вычислений, допуская как одновременную обработку независимых задач,

так и распараллеливание внутри каждой задачи, в зависимости от специфики вычислительной архитектуры. Это дает возможность организовывать иерархические схемы параллельных вычислений на системах гибридной (или NUMA, Non-Uniform Memory Architecture) архитектуры.

Оценка архитектуры

Оценка программной архитектуры устанавливает степень соответствия предложенной архитектуры поставленным требованиям. Для многих классов бизнес-приложений оценка архитектуры строится на неформализованных подходах (например, оценка сценариев). Однако с точки зрения проектирования высокопроизводительных программных систем предоставляется возможность количественной оценки основной проектной характеристики — параллельной производительности системы (времени работы, ускорения, эффективности и пр.).

Для оценки архитектуры на этапе проектирования необходимым становится прототипирование и моделирование разрабатываемого ПО для прогнозирования особенностей его использования и формализации требований. Ниже представлены оценки, полученные на примере моделирования климатической системы Каспийского моря на мини-суперкомпьютере TForge-Mini.

В рамках схемы на рис. 3 можно приближенно описать время выполнения произвольного сервиса на p вычислителей с общей памятью (внутри одного узла) в форме

$$T_{calc}(p, \alpha, \gamma) = T_0 \left[\left[\alpha(p-1) + \frac{1}{p} \right] \cdot \gamma + (1-\gamma) \right], \quad (1)$$

где α — параметр, характеризующий рост накладных расходов с увеличением количества вычислителей, а γ — доля параллельных вычислений. Например, применительно к акватории Каспийского моря процедуре подготовки метеорологической информации соответствует $\gamma = 0,96$, расчету волн — $\gamma = 0,95$, расчету течений и уровня моря — $\gamma = 0,59$, параметризации климатических спектров — $\gamma = 0,98$.

Общее время коммуникаций между узлами определим как сумму времени рассылки и сбора данных, а также времени на передачу перекрытий между независимыми блоками данных:

$$T_{comm} = t_w \left[(3L_0\tau_0 + \Pi\mu L) + \Pi L \vartheta + \left(\eta\xi L + \nu(\xi L)^2 \right) (\Pi - 1) \right], \quad (2)$$

где τ_0 — число полей метеорологических (входных) данных по времени; L_0, L — число узлов расчетной сетки по пространству для метеорологических (входных) и океанографических (выходных) данных; Π — количество узлов; η, ν, ξ, μ — коэффициенты, характеризующие кратность использования данных в моделях статистики; ϑ — коли-

чество реплицируемых полей между узлами. Число полей океанографических (выходных) данных по времени τ не влияет на объем собираемых с удаленных узлов данных, так как модели статистической обработки результатов применяются непосредственно к тем океанографическим полям, которые были рассчитаны на данном узле. Величина t_w соответствует времени передачи единицы информации по коммуникационной сети. Предполагается, что время работы T_0 каждой задачи (расчета ветра, волн, течений, параметризации спектров и расчета статистики) определяется только производительностью вычислителя (например, ядром CPU) и объемом обрабатываемых данных. Например, для одного ядра AMD Opteron 275 можно записать:

$$T_{0(wind)} = \left[1,2 \cdot 10^{-5} \cdot \frac{L_0 \tau_0}{\Pi} + 60 \cdot 10^{-5} \cdot \frac{L \tau}{\Pi} \right];$$

$$T_{0(waves)} = \left[8,25 \cdot 10^{-4} \cdot L \cdot \left(\frac{\tau}{\Pi} + \vartheta_{wave} \right) \right];$$

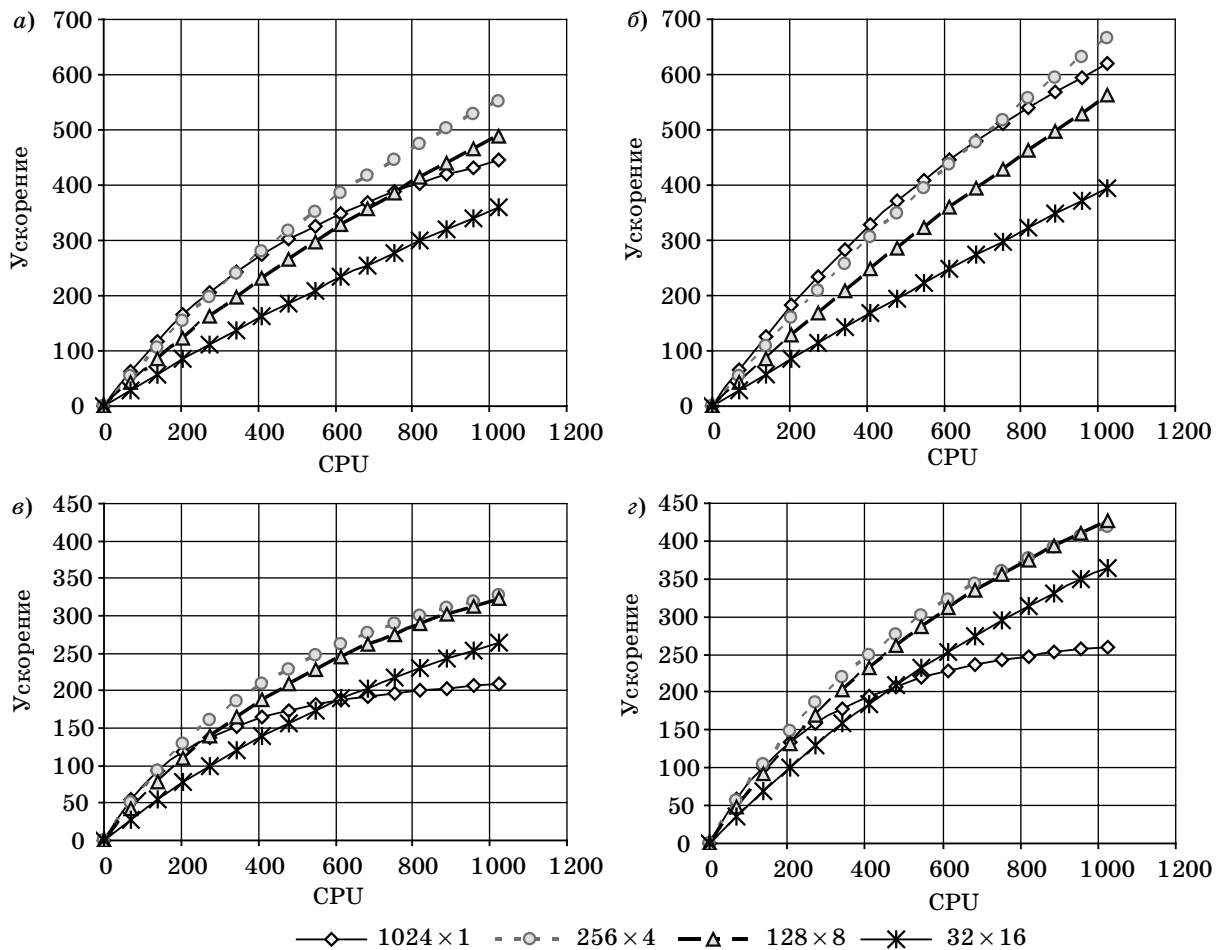
$$T_{0(current_level)} = \left[21,6 \cdot 10^{-5} \cdot L \cdot \left(\frac{\tau}{\Pi} + \vartheta_{level} \right) \right];$$

$$T_{0(spectra)} = \left[\xi \cdot 0,16 \cdot L \cdot \left(\frac{\tau}{\Pi} \right) \right];$$

$$T_{0(statistics)} = \frac{5 \cdot 10^{-5} \cdot \xi \cdot (L \tau + 0,07 \cdot L^2 + 0,15 \cdot L \tau \ln(\tau))}{\Pi r} \quad (3)$$

В выражениях (3) численные коэффициенты перед объемами выполняемых операций имеют смысл времени расчета элементарной операции (в пересчете на одну точку исходных или модельных данных, в секундах); объем обрабатываемых данных делится пропорционально между Π узлами. Учитывая, что общее время вычислений является суммой времен, определяемых (2) и (3), этого достаточно, чтобы оценить параллельное ускорение как функцию от параметров.

На рис. 4 показано влияние архитектуры программной системы на общее ускорение вычислений. При небольшом количестве узлов оптимальным является распараллеливание по времени



■ Рис. 4. Параллельное ускорение в зависимости от конфигурации вычислительных узлов, длины входного массива и размера перекрытия независимых блоков данных: а — 15 лет, перекрытие 1 месяц; б — 30 лет, перекрытие 1 месяц; в — 5 лет, перекрытие 3 месяца; г — 5 лет, перекрытие 1 месяц

с использованием отдельных процессоров узлов с общей памятью как независимых вычислителей. Однако с увеличением числа процессоров тенденция изменяется и многопроцессорные узлы становятся более предпочтительными. С другой стороны, чем большее число лет T обрабатывается, тем большее число процессоров необходимо для получения эффекта от применения многопроцессорных узлов. При этом очевидно, что чем большее число процессоров используется в узлах, тем дальше будет находиться точка заката кривой ускорения. Стоит отметить, что с увеличением размера входного массива (число моделируемых лет, рис. 4, а, б) в результате увеличения гранулярности общее ускорение также возрастает.

Рис. 4, в, г иллюстрирует влияние перекрытия независимых блоков данных (что является обязательным при распараллеливании по времени) на производительность алгоритма. При уменьшении величины перекрытия с трех месяцев до одного ускорение в среднем вырастает примерно в полтора раза, а эффективность от использования многопроцессорных узлов сдвигается в зону большего числа узлов и меньшего числа процессоров на узле. Эта характеристика может меняться в зависимости от специфики акватории (океанские процессы обладают большим «временем жизни», чем процессы в закрытых морях), а также от особенностей моделируемого процесса (синоптический интервал атмосферных процессов и волнения составляет несколько суток, а эволюция вихрей водных масс может длиться несколько месяцев).

Результаты проведенного анализа могут рассматриваться как формализация знаний по управлению параллельными процессами в целях выбора оптимальной по производительности конфигурации для заданной задачи в рамках интеллектуального сервиса управления вычислительными модулями (см. рис. 2, «А»).

Выводы

Существующие стандарты по проектированию ПО (IEEE 1016–1998 и IEEE 1471–2000) носят лишь рекомендательный характер в области определения понятий, а не их содержания. Потому в данной работе на примере высокопроизводительного комплекса моделирования экстремальных гидрометеорологических явлений рассматриваются основные аспекты проектирования наукоемкого ПО для моделирования сложных систем. К ним относятся:

- использование сервисно-ориентированной архитектуры, отображающей взаимодействие управляющих и вычислительных компонентов, соответствующих моделям в различных диапазонах изменчивости;
- оптимизация механизмов оперирования большими объемами данных за счет применения технологии REST и комбинированного способа хранения расчетной информации;
- применение иерархической схемы отображения архитектуры программной системы на параллельную архитектуру вычислительной системы на аппаратном, системном, программном и логическом уровнях;
- использование аппарата параметрических моделей производительности, настроенных на основе прототипов, для оценки степени соответствия архитектуры программного комплекса проектным требованиям и оптимизации производительности программного комплекса за счет динамического выбора схемы отображения.

Предлагаемые решения не ограничены только данной предметной областью. В частности, авторы считают целесообразным использовать их для моделирования биологической сложной системы — глобальной популяции вируса иммунодефицита человека [29, 30].

Литература

1. **Соммервилл И.** Инженерия программного обеспечения. 6-е изд.: Пер. с англ. М.: Издательский дом «Вильямс», 2002. 624 с.
2. **Вирт Н.** Алгоритмы + структуры данных = программы. М.: Мир, 1985. 406 с.
3. **Буч Г.** Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд.: Пер. с англ. М.: «Бином»; СПб.: «Невский диалект», 1999. 560 с.
4. **Szyperski C.** Component Software: Beyond Object-Oriented Programming. Addison-Wesley Professional, 1998. 411 p.
5. **Lublinsky B.** Defining SOA as an architectural style. 9 January 2007. <http://www.ibm.com/developerworks/architecture/library/ar-soastyle/>
6. **Fielding R. T.** Architectural Styles and the Design of Network-based Software Architectures: Doctoral thesis. University of California, Irvine, 2000. 162 p.
7. **Navasa A., Pérez M. A., Murillo J. M.** Aspect Modeling at Architecture Design // Lecture Notes in Computer Science. 2005. Vol. 3527. P. 41–58.
8. **Якобсон А., Буч Г., Рамбо Дж.** Унифицированный процесс разработки программного обеспечения. СПб.: Питер, 2002. 496 с.
9. **Carver J. C., Kendall R. P., Squires S. E. et al.** Software Development Environments for Scientific and Engineering Software: A Series of Case Studies: Proc. of the 29th International Conf. on Software Engineering. Washington, DC, USA: IEEE Computer Society Press, 2007. P. 550–559.
10. **Sessions R.** COM and DCOM: Microsoft's Vision for Distributed Objects. N. Y., USA: John Wiley & Sons, 1998. 492 p.
11. Object Management Group. OMG's CORBA website. <http://www.corba.org>
12. **Matena V., Stearns B.** Applying Enterprise JavaBeans: Component-Based Development for the J2EE Platform. Prentice Hall, 2000. 464 p.

13. Bernholdt D. E., Elwasif W. R., Kohl J. A., Epperly T. G. W. A Component Architecture for High-Performance Computing: Proc. of the Workshop on Performance Optimization via High-Level Languages and Libraries. N. Y., USA, 22 June 2002.
14. Sloot P. M. A., Frenkel D., Vorst H. A. Van der et al. Computational e-Science: Studying complex systems in silico. A National Coordinated Initiative. White Paper, February 2007. <http://www.science.uva.nl/research/scs/papers/archive/Sloot2007a.pdf>
15. Voccara N. Modeling Complex Systems. N. Y.: Springer, 2004. 397 p.
16. Справочные данные по режиму ветра и волнения Балтийского, Северного, Черного, Азовского и Средиземного морей / Под ред. Л. И. Лопатухина, А. В. Бухановского, В. А. Рожкова и др. СПб.: Российский Морской Регистр Судоходства, 2006. 450 с.
17. Бухановский А. В., Лопатухин Л. И., Иванов С. В. Подходы, опыт и некоторые результаты исследования волнового климата океанов и морей. I. Постановка задачи и входные данные // Вестник СПбГУ. Сер. 7. 2005. Вып. 3. С. 62–74.
18. Бухановский А. В., Зильберштейн О. И., Иванов С. В. и др. Моделирование экстремальных явлений в атмосфере и океане как задача высокопроизводительных вычислений // Параллельные вычислительные технологии (ПаВТ 2008): Тр. Междунар. науч. конф. Санкт-Петербург, 28 января — 1 февраля 2008. Челябинск: ЮУрГУ, 2008. С. 57–68.
19. Compiler Optimization for Scalable PC / Pandle S., Agrawal D. P. (Eds.) // Lecture Notes in Computer Science. 2001. Vol. 1808.
20. Foster I. Designing and Building Parallel Programs: Concepts and Tools for Software Engineering. Reading, Massachusetts: Addison-Wesley, 1995. 381 p.
21. Cazes J. E., Campbell T., Rogers E. OpenMP Parallel Implementation of SWAN // Navo MSRC Navigator. Fall 2001. P. 13.
22. Grama A., Gupta A., Karypis G., Kumar G. Introduction to Parallel Computing. Second Edition. Addison-Wesley, 2003. 856 p.
23. Lopatoukhin L. J., Rozhkov V. A., Ryabinin V. E. et al. Estimation of extreme wind wave heights // Reports of World Meteorological Organization. 2000. WMO/TD-No. 1041. 71 p.
24. Cohen S. Ontology and Taxonomy of Services in a Service-Oriented Architecture // The Architecture Journal. 2007. Vol. 11. P. 30–35.
25. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2007. 366 с.
26. Vinoski S. REST Eye for the SOA Guy // IEEE Internet Computing. 2007. Vol. 11. N 1. P. 82–84.
27. Obasanjo D. XML Serialization in the .NET Framework. In Microsoft MSDN Library. 23 January 2003. <http://msdn2.microsoft.com/en-us/library/ms950721.aspx>
28. Boukhanovsky A. V., Ivanov S. V. Stochastic simulation of inhomogeneous metocean fields. Part III: High-performance parallel algorithms // Lecture Notes in Computer Science. 2003. Vol. 2658. P. 234–243.
29. Иванов С. В., Колыхматов И. И., Бухановский А. В. Параллельные алгоритмы моделирования комплексных сетей // Параллельные вычислительные технологии (ПаВТ 2008): Тр. Междунар. науч. конф. Санкт-Петербург, 28 января — 1 февраля 2008. Челябинск: ЮУрГУ, 2008. С. 395–402.
30. Sloot P. M. A., Boukhanovsky A. V., Keulen W. et al. A GRID-based HIV expert system // Journal of Clinical Monitoring and Computing. 2005. Vol. 19. P. 263–278.

УДК 519.876.5

ХАРАКТЕРИСТИКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРОЕКТИРОВАНИЯ ДИФРАКЦИОННО- ОГРАНИЧЕННЫХ ОПТИКО-ЭЛЕКТРОННЫХ СИСТЕМ

И. Л. Лившиц,

канд. техн. наук, старший научный сотрудник

М. А. Пашковский,

программист

Санкт-Петербургский государственный университет информационных технологий,
механики и оптики

И. Г. Бронштейн,

директор

ЗАО «КБ Юпитер»

М. Б. Сергеев,

доктор техн. наук, профессор

Санкт-Петербургский государственный университет аэрокосмического приборостроения

Унчун Чо,

доктор техн. наук, профессор

Корейский политехнический университет

Анализируется программное обеспечение, наиболее подходящее для проектирования дифракционно-ограниченных систем с высоким качеством изображения. В качестве примера рассматривается проектирование широкоугольного объектива для миниатюрной ПЗС-камеры.

Введение

Инженеры-оптики, пожалуй, одними из первых осознали перспективы, которые открываются при использовании компьютеров в процессе проектирования оптических систем (ОС). Изначально компьютеры применялись для решения задачи по расчету хода лучей в ОС, что давало проектировщику большую свободу в выборе схемы системы — позволяло проверять несколько вариантов схем с большой (относительно ручного расчета) скоростью. Это открыло возможности по проектированию более сложных ОС, включающих в себя широкий спектр элементов, в том числе элементы с асферическими поверхностями и элементы, произвольно расположенные в пространстве. Помимо этого, при использовании компьютера разработчик может определить качество изображения в терминах аббераций, точечных диаграмм и частотно-контрастных характеристик. Позже появилась возможность оптимизации разработанных ОС в соответствии с функцией оценки, заданной проектировщиком. ОС, разработанные при помощи программного обес-

печения (ПО) подобного рода, внесли серьезный вклад в создание приборов для научных исследований и др.

Существующее ПО для проектирования оптико-электронных систем

Сегодня на рынке представлено около двух десятков пакетов программ для проектирования оптико-электронных систем. Чтобы сделать правильный выбор, следует учитывать спектр задач, которые планируется решать с помощью этого инструмента, а также доступные ресурсы. Наряду с тем, что все пакеты обладают сравнительно одинаковым базовым функционалом (таким как расчет хода лучей через ОС и аббераций Зейделя), существуют и серьезные различия. В основном они сводятся к используемым методам оптимизации, логике создания схемы и удобству интерфейса. Серьезного внимания достойны следующие продукты.

CODE-V компании **Optical Research Associates (США)**. Самый популярный (и самый дорогостоящий) на данный момент пакет программ, удовлет-

воряющий практически все потребности оптика-проектировщика.

SYNOPSIS компании **OSD** и **ASAP** компании **BRO** (США). Программные продукты, решающие большинство задач по проектированию ОС. В них учитываются поглощение, механические и термические свойства материалов; в ОС предусмотрено использование призм, голографических и дифракционных элементов, асферических поверхностей, градианов, а также сдвиг элементов, их вращение и децентровка. Возможна тонкая настройка многовариантной оптимизации и функция оценки. Отличительными особенностями пакетов являются попытка использовать механизм искусственного интеллекта (ИИ) для выбора стартовой точки ОС и удобный графический интерфейс, наглядно представляющий, как изменение того или иного параметра ОС влияет на качество изображения. Программа SYNOPSIS [1] оптимальна по стоимости, в России находится дистрибьютор фирмы.

Zemax, **Focus Software Inc.** (США). Популярный пакет программ, одной из ключевых особенностей которого является возможность работы с поляризованным излучением. В России имеется дистрибьютор фирмы.

Следует отметить также:

- ПО OSLO компании Super-OSLO, Sinclair Optics, США;
- продукты компании Kidger Optics Ltd., Великобритания;
- продукты компании ACCOS V, США;
- продукты компании Genesee Optics Software, США;
- ОПАЛ, разработанный в СПбГУ ИТМО.

Процесс разработки ОС

С появлением первых программных продуктов для оптиков и связанной с этим эйфорией многие начали пророчить полную автоматизацию процесса разработки ОС, но, как показало время, это оказалось утопией. Данному факту существует простое объяснение. Залогом успеха разработки ОС являются два критерия: хорошая стартовая точка для начала расчета ОС и грамотно составленная оценочная функция для ее оптимизации. Как было сказано выше, в пакете SYNOPSIS сделана попытка автоматизировать выбор стартовой точки с использованием ИИ, но даже при этом в процессе оптимизации этой ОС программа может и не реализовать весь потенциал выбранной стартовой точки. Для объяснения этого явления стоит рассмотреть, что такое процесс оптимизации. Оптимизация — это, фактически, поиск глобального минимума в n -мерной плоскости, где размерность n задается количеством варьируемых параметров ОС.

Основная причина, по которой может быть не раскрыт потенциал стартовой точки, — это определение программой оптимизации локального минимума и невозможность для нее, программы, выхода из него. Это является проблемой лишь для про-

граммы, а не для оптика, вкладывающего в процесс разработки весь свой опыт и интуицию. Действительно, синтез вариантов стартовой точки — это вопрос автоматизации, но выбор наиболее подходящей — вопрос творческого потенциала и опыта разработчика.

В данном контексте стоит упомянуть разработки, выполненные в СПбГУ ИТМО — ЗАО «КБ Юпитер», — программы структурного синтеза ОС типа «фотообъектив» (класс ОС, где предмет расположен на бесконечности, а изображение — на конечном расстоянии — фотопленке или ПЗС-матрице). Программы GEOS [2] и Struct [3] на основании технического задания предлагают разработчику несколько вариантов стартовых точек с пояснениями сильных и слабых сторон того или иного варианта.

Таким образом, оптимальным процессом разработки ОС является следующая схема:

1) синтез вариантов стартовых точек программой GEOS или Struct и последующий отбор разработчиком наиболее удачной из схем;

2) определение расстояний между поверхностями и показателей преломления материалов, заполняющих эти промежутки оптиком-проектировщиком;

3) составление оптиком оценочной функции;

4) оптимизация полученной системы в пакете SYNOPSIS.

Проиллюстрируем процесс создания ОС на примере объектива для ПЗС-камеры. Разрабатываемый объектив должен удовлетворять следующим техническим требованиям:

— относительное отверстие — меньше 1:2,8;

— поле обзора — 130°;

— фокальное расстояние — меньше 30°;

— спектральный интервал — ± 250 мкм;

— δ — меньше $5\delta_{\text{Эри}}$;

— задний фокальный отрезок — меньше 0,5 f ;

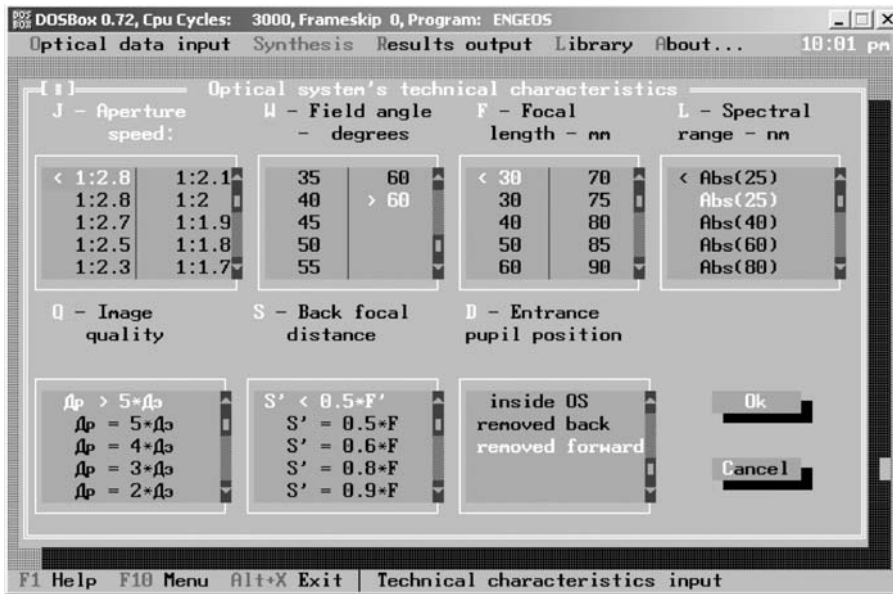
— диафрагма установлена перед системой, где δ — диаметр кружка рассеяния; $\delta_{\text{Эри}}$ — диаметр диска Эри для идеальной оптической системы; f — заднее фокусное расстояние.

Для синтеза стартовой точки эти данные были введены в программу GEOS (рис. 1), которая «предложила» пять вариантов (рис. 2), из которых был выбран следующий:

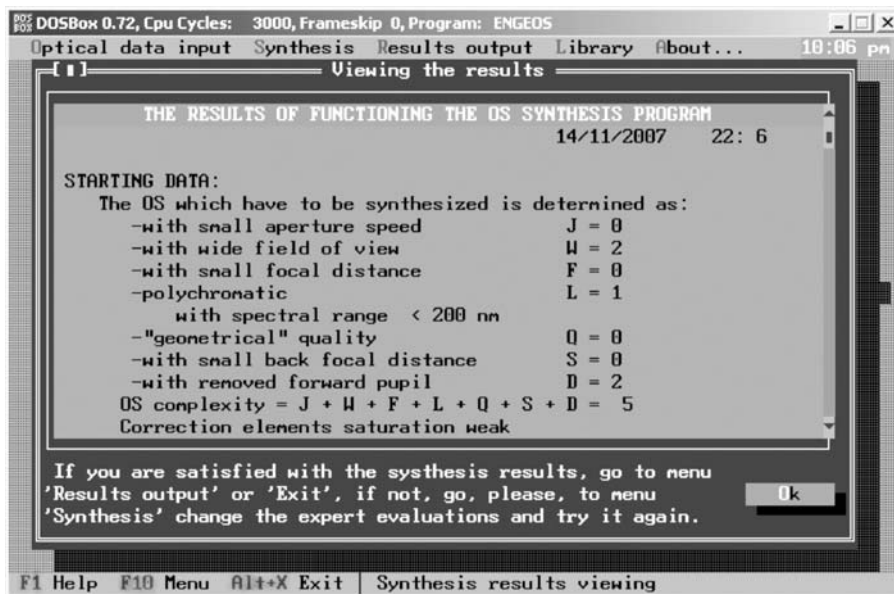
$$C(k, k) + B(a, k) + C(a, k) + L(a, k),$$

где C — коррекционные, B — базовые, L — светосильные элементы; символами k обозначены поверхности элемента, концентричные центру входного зрачка, и a — апланатические.

На основе теории расчета ОС были заданы типы материалов и толщины для каждого элемента и произведена оптимизация в пакете SYNOPSIS. В процессе оптимизации первый и последний элементы были преобразованы в склейки, также была добавлена плоскопараллельная пластина, являющаяся элементом датчика, толщина которой



■ Рис. 1. Установка параметров ОС



■ Рис. 2. Результат выполнения программы — оценка сложности ОС и стартовая точка

должна учитываться при расчете. Таким образом, был разработан широкоугольный объектив для миниатюрной ПЗС-камеры, представленный на рис. 3 и 4 (см. 3-ю стор. обложки) в двумерном и трехмерном виде соответственно.

Перспективы в области автоматизации проектирования ОС

Процесс проектирования, описанный выше, позволяет утверждать, что перспективным направлением в области автоматизации является

разработка систем параметрического синтеза и их интеграция с уже существующими пакетами.

Это гарантирует появление ПО нового класса для полного синтеза ОС, позволяющего на основании технического задания проводить полный синтез ОС:

структурный синтез — предлагать проектировщику возможные принципиальные варианты (стартовые точки) ОС;

параметрический синтез — расчет параметров поверхностей для выбранного варианта.

Иллюстрации к статье
И. А. Лившиц, И. Г. Бронштейн, М. А. Пашковский, М. Б. Сергеев,
Унчун Чо. «Характеристики программного обеспечения для
проектирования дифракционно-ограниченных оптико-электронных
систем», стр 21.

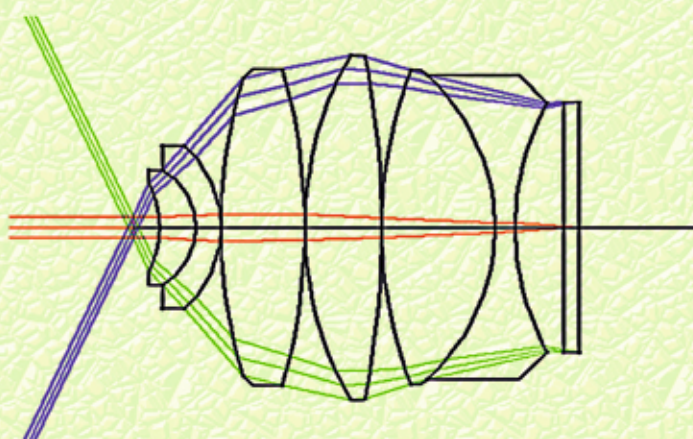


Рисунок 3. Оптическая схема широкоугольного объектива

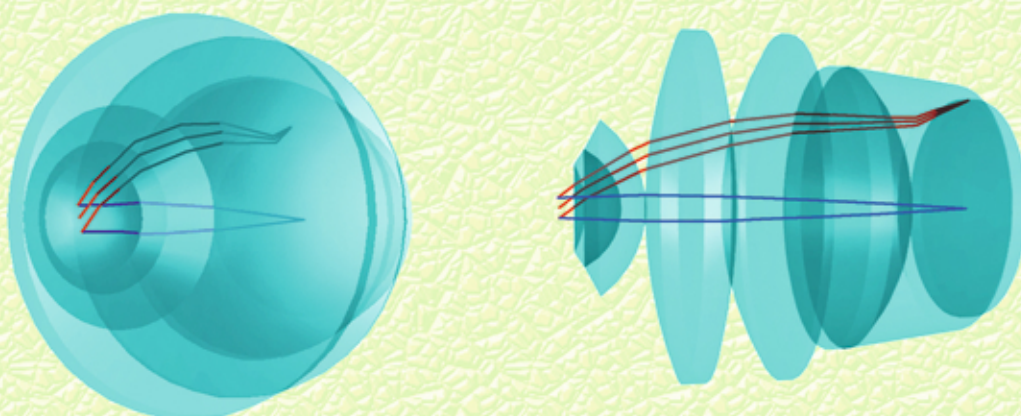


Рисунок 4. Трехмерная визуализация широкоугольного объектива

Главное преимущество такого ПО — высокая скорость разработки ОС.

В последние два года в компьютерной индустрии произошел сдвиг парадигмы от персональных рабочих станций с определенным набором установленного ПО к сетевым сервисам. Этот сдвиг обозначил тенденцию к переносу данных и программ пользователя в сеть в целях повышения мобильности последнего и доступности первого.

Сегодня существует потребность в ПО для научных разработок, которое учитывает данную специфику, поэтому стоит говорить еще об одном направлении развития для ПО полного синтеза ОС — разработке web-сервиса. Это позволит решить проблемы, связанные с обновлением ПО, баз данных оптических материалов и элементной базы, что в свою очередь является серьезной проблемой для конечного пользователя. Дополнительным плюсом такого ПО может являться удобный мультимедийный интерфейс пользователя, включающий в себя не только текстовую информацию и изображения, но и интерактивное представление ОС в трехмерном виде.

Заключение

Практика показала, что сегодня оптимальным пакетом программ для проектирования дифракционно-ограниченных оптико-электронных систем можно считать SYNOPSIS по нескольким критериям:

1) наличие ИИ;

2) оптимальное соотношение цена-функциональность;

3) удобный интерфейс.

Помимо этого можно утверждать, что рынок программных продуктов такого рода на данный момент продолжает расти, и основным направлением роста является совершенствование программ выбора стартовой точки и алгоритмов оптимизации ОС.

Работа выполнена в рамках международного контракта между Санкт-Петербургским государственным университетом информационных технологий, механики и оптики и Корейским политехническим университетом при поддержке гранта Корейского исследовательского фонда, основанного корейским правительством (MOEHRD) (KRF-2006-613-C00002).

Литература

1. SYNOPSIS. V12.054, USA, OSD, Inc., 2007.
2. Anitropova I. L. Formalizing the heuristic synthesis procedure in lens design: OSA Proc. Of the International Optical Design Conf. Rochester, USA, June 1994.
3. Livshits I., Salnikov A. CAD based on developed algorithm and expert rules in proposed in automate lens // 4th International Conf. on Optics-Photonics Design & Fabrication, ODF'04: Proc. Makuhari, Chiba, Japan, July 2004.

УДК 519.725

ПРОСТОЙ АЛГОРИТМ ДЕКОДИРОВАНИЯ АЛГЕБРАИЧЕСКИХ КОДОВ

С. В. Федоренко,

канд. техн. наук, доцент

Санкт-Петербургский государственный университет аэрокосмического приборостроения

Рассматривается алгоритм декодирования, который представляется самым простым и естественным в классе алгоритмов декодирования алгебраических кодов до их корректирующей способности. Приводятся вывод, описание, анализ и доказательство корректности алгоритма. Асимптотическая сложность алгоритма совпадает со сложностью лучших алгоритмов декодирования, а описание является самым простым из описаний известных алгоритмов.

Введение

Корректирующие коды получили широкое применение в задачах передачи, хранения и защиты информации. Для кодов, имеющих алгебраическую структуру, существуют алгебраические алгоритмы декодирования. Под алгебраическим декодированием понимают методы декодирования, основанные на алгебраических свойствах кодов и состоящие в решении уравнений и/или систем уравнений с полиномиальной сложностью. Классические алгебраические алгоритмы декодирования обеспечивают исправление ошибок до конструктивной корректирующей способности кода, но имеют довольно сложное и громоздкое описание.

В работах [7, 8] предложен алгоритм декодирования, который представляется самым простым и естественным в классе алгоритмов декодирования алгебраических кодов до их корректирующей способности. Асимптотическая сложность алгоритма совпадает со сложностью лучших алгоритмов декодирования, а описание является самым простым из описаний известных алгоритмов.

В статье предлагается описание, оригинальный вывод и доказательство корректности алгоритма. Приводится пример алгоритма декодирования.

Основные понятия и определения

Описание алгоритма декодирования приведем для кодов Рида—Соломона, так как для других классов алгебраических кодов принципиальных отличий в описании алгоритма нет.

Рассмотрим код Рида—Соломона (n, k, d) над конечным полем $GF(q)$ с длиной $n = q - 1$, числом информационных символов k и конструктивным расстоянием $d = n - k + 1$, где q — степень простого

числа. Корректирующая способность кода равна

$$\left[\frac{d-1}{2} \right], \text{ где } [a] \text{ — целая часть числа } a.$$

Порождающий многочлен кода Рида—Соломона обозначим через

$$g(x) = \prod_{i=b}^{b+d-2} (x - \alpha^i),$$

где b — произвольное натуральное число; α — примитивный элемент $GF(q)$.

Далее для упрощения изложения будем рассматривать только случай $b = 1$.

Принятый вектор представлен многочленом

$$R(x) = \sum_{i=0}^{n-1} r_i x^i = C(x) + E(x) = \sum_{i=0}^{n-1} c_i x^i + \sum_{i=0}^{n-1} e_i x^i,$$

где $C(x)$ — кодовое слово; $E(x)$ — вектор ошибок.

Пусть $E(x)$ содержит $t \leq \frac{d-1}{2}$ ошибок, которые имеют координаты i_1, i_2, \dots, i_t , причем $0 \leq i_1 < i_2 < \dots < i_t \leq n - 1$, и значения $e_{i_1}, e_{i_2}, \dots, e_{i_t}$. Назовем величины $Z_1 = \alpha^{i_1}, Z_2 = \alpha^{i_2}, \dots, Z_t = \alpha^{i_t}$ локаторами ошибок, а $Y_1 = e_{i_1}, Y_2 = e_{i_2}, \dots, Y_t = e_{i_t}$ — значениями ошибок.

Многочлен локаторов ошибок обозначим через

$$W(x) = \prod_{i=1}^t (x - Z_i),$$

где Z_i — локатор ошибки в векторе ошибок $E(x)$. Положим по определению $W(x) = 1$, если ошибок нет.

Известно несколько методов кодирования для кодов Рида—Соломона. В настоящей работе при-

меняется спектральное кодирование в частотной области для несистематического кодирования. Заметим, что алгоритм декодирования не зависит от метода кодирования. Информационный многочлен кода Рида—Соломона обозначим как

$$M(x) = \sum_{i=0}^{k-1} m_i x^i.$$

При спектральном кодировании компонента c_i кодового слова $C(x)$ вычисляется как

$$c_i = M(\alpha^i), \quad i \in [0, n-1].$$

Вывод алгоритма декодирования

Вывод алгоритма декодирования, основанный на интерпретации работ [1–4], предложен в работах [5, 6].

Если $r_i = c_i$, то $r_i = M(\alpha^i)$. Если $r_i \neq c_i$, то $W(\alpha^i) = 0$. Следовательно:

$$W(\alpha^i)r_i = W(\alpha^i)M(\alpha^i), \quad i \in [0, n-1].$$

Пусть $P(x) = W(x)M(x)$. Тогда ключевое уравнение имеет вид

$$W(\alpha^i)r_i = P(\alpha^i), \quad i \in [0, n-1].$$

Построим такой интерполяционный многочлен $T(x)$, что

$$T(\alpha^i) = r_i, \quad i \in [0, n-1],$$

где $\deg T(x) < n$.

Далее, из

$$W(\alpha^i)T(\alpha^i) = P(\alpha^i), \quad i \in [0, n-1]$$

имеем сравнение

$$\begin{cases} W(x)T(x) \equiv P(x) \pmod{x^n - 1} \\ \deg W(x) \leq \frac{d-1}{2} \\ \text{maximize } \deg W(x) \end{cases}.$$

Учитывая, что $\deg P(x) = \deg M(x) + \deg W(x) \leq (k-1) + \frac{d-1}{2} \leq \frac{n+k}{2} - 1 < \frac{n+k}{2}$, переписываем условие решения сравнения на эквивалентное

$$\begin{cases} W(x)T(x) \equiv P(x) \pmod{x^n - 1} \\ \deg P(x) < \frac{n+k}{2} \\ \text{maximize } \deg P(x) \end{cases}.$$

Решаем сравнение применением расширенного алгоритма Евклида к многочленам $x^n - 1$ и $T(x)$, получая многочлены $P(x)$ и $W(x)$. Информационный многочлен получается делением

$$M(x) = \frac{P(x)}{W(x)}.$$

Алгоритм декодирования

Приведем алгоритм из работы [7]. Заметим, что этот алгоритм ранее был введен в работе [8]. Для упрощения изложения будем рассматривать только классические коды Рида—Соломона с параметрами $n = q - 1$ и $b = 1$.

1. Интерполяция.

Построим такой интерполяционный многочлен $T(x)$, что

$$T(\alpha^i) = r_i, \quad i \in [0, n-1],$$

где $\deg T(x) < n$.

2. Незаконченное вычисление наибольшего общего делителя.

Решаем сравнение

$$\begin{cases} W(x)T(x) \equiv P(x) \pmod{x^n - 1} \\ \deg P(x) < \frac{n+k}{2} \\ \text{maximize } \deg P(x) \end{cases}$$

применением расширенного алгоритма Евклида к многочленам $x^n - 1$ и $T(x)$, получая единственную пару многочленов $P(x)$ и $W(x)$.

3. Деление.

Информационный многочлен есть

$$M(x) = \frac{P(x)}{W(x)}.$$

Асимптотическая сложность алгоритма $O(n(\log n)^2)$ совпадает со сложностью лучших классических алгоритмов декодирования кодов Рида—Соломона [9–11].

Первый шаг алгоритма может быть выполнен любым быстрым алгоритмом вычисления дискретного преобразования Фурье над конечным полем, например [12, 13], со сложностью $O(n(\log n)^2)$ операций. В случае, когда необходимо минимизировать число умножений, лучший алгоритм для малых длин ($n \leq 511$) предложен в работе [14].

Одна из лучших реализаций второго шага есть алгоритм Мёнка [15] со сложностью $O(n(\log n)^2)$ операций, который также воспроизведен в монографиях [16, 17]. Заметим, что при этом второй шаг полностью совпадает с алгоритмом решения ключевого уравнения Сугиямы и других [18].

Деление на третьем шаге алгоритма выполняется за $O(n \log n \log \log n)$ операций.

При приложении алгоритма декодирования к другим классам алгебраических кодов, таких как коды Боуза—Чоудхури—Хоквингема, коды Гоппы или альтернативные коды, необходимо добавить дополнительный шаг, восстанавливающий кодовое слово по информационному многочлену.

Корректность алгоритма декодирования

Для завершения вывода алгоритма декодирования необходимо доказать существование и единственность полученного решения.

Теорема. Алгоритм декодирования приводит к единственному решению при декодировании до корректирующей способности кода Рида—Соломона.

Доказательство. Пусть имеются два решения сравнения

$$\begin{cases} W(x)T(x) \equiv P(x) \pmod{x^n - 1} \\ \deg P(x) < \frac{n+k}{2} \\ \text{maximize } \deg P(x) \end{cases} \quad (*)$$

Первое решение, полученное применением расширенного алгоритма Евклида к многочленам $x^n - 1$ и $T(x)$ с правилом останковки $\deg P(x) < \frac{n+k}{2}$, дает пару многочленов $P(x)$ и $W(x)$. Если $P(x)$ делится на $W(x)$ без остатка, то многочлен $M(x) = \frac{P(x)}{W(x)}$ будет информационным многочленом кода Рида—Соломона.

Заметим, что расширенный алгоритм Евклида надо проводить с нулевого (фиктивного) шага, а не с первого, как обычно. Заканчивать расширенный алгоритм Евклида надо дополнительным (с нулевым остатком) шагом.

Другое решение (истинное) удовлетворяет сравнению

$$\widetilde{W}(x)T(x) \equiv \widetilde{P}(x) \pmod{x^n - 1}$$

при декодировании до корректирующей способности кода Рида—Соломона $\deg \widetilde{W}(x) \leq \frac{d-1}{2}$ и приводит к другому информационному многочлену кода Рида—Соломона

$$\widetilde{M}(x) = \frac{\widetilde{P}(x)}{\widetilde{W}(x)}.$$

Вначале рассмотрим вырожденный случай, когда решение сравнения (*) заканчивается на нулевом шаге расширенного алгоритма Евклида, и докажем, что информационные многочлены, полученные из обоих решений, $M(x) = \widetilde{M}(x)$, совпадают.

Пусть $\widetilde{W}(x) = 1$. Тогда $\widetilde{M}(x) = M(x) = P(x) = T(x)$, так как из нулевого шага расширенного алгоритма Евклида следует, что $\deg T(x) \leq k-1 < \frac{n+k}{2}$.

Далее будем полагать, что $\widetilde{W}(x) \neq 1$.

Заметим, что $\deg W(x) \leq n - \frac{n+k}{2} = \frac{d-1}{2}$ по свойству степени линейных коэффициентов в расширенном алгоритме Евклида.

Выполним деление с остатком

$$P(x) = W(x)M(x) + U(x),$$

причем $\deg U(x) < \deg W(x)$.

Если остаток от деления равен нулю: $U(x) = 0$, то многочлен $M(x)$ будет информационным многочленом кода Рида—Соломона.

Очевидно, что $\deg \widetilde{P}(x) = \deg \widetilde{M}(x) + \deg \widetilde{W}(x) \leq (k-1) + \frac{d-1}{2} \leq \frac{n+k}{2} - 1 < \frac{n+k}{2}$.

Покажем, что информационные многочлены, полученные из обоих решений, совпадают, т. е.

$$\begin{cases} M(x) = \widetilde{M}(x) \\ U(x) = 0 \end{cases}.$$

После преобразований

$$\begin{aligned} P(x)\widetilde{W}(x) &\equiv (W(x)T(x))\widetilde{W}(x) = \\ &= W(x)(T(x)\widetilde{W}(x)) \equiv W(x)\widetilde{P}(x) \pmod{x^n - 1} \end{aligned}$$

получаем сравнение

$$P(x)\widetilde{W}(x) \equiv W(x)\widetilde{P}(x) \pmod{x^n - 1}.$$

Оценим степени произведения многочленов в каждой части сравнения. Из

$$\begin{cases} \deg W(x), \widetilde{W}(x) \leq \frac{d-1}{2} \\ \deg P(x), \widetilde{P}(x) < \frac{n+k}{2} \end{cases}$$

следует

$$\begin{cases} \deg(P(x)\widetilde{W}(x)) < \frac{n+k}{2} + \frac{d-1}{2} = n \\ \deg(W(x)\widetilde{P}(x)) < \frac{d-1}{2} + \frac{n+k}{2} = n \end{cases},$$

т. е.

$$\begin{cases} \deg(P(x)\widetilde{W}(x)) < n \\ \deg(W(x)\widetilde{P}(x)) < n \end{cases}.$$

Видно, что степени произведения многочленов в каждой части сравнения не превышают степень модуля $x^n - 1$, т. е. сравнение справедливо как равенство для многочленов

$$P(x)\widetilde{W}(x) = W(x)\widetilde{P}(x).$$

Отсюда после деления имеем

$$P(x) = W(x) \frac{\widetilde{P}(x)}{\widetilde{W}(x)} = W(x)\widetilde{M}(x).$$

Из последнего равенства видно, что $P(x)$ делится на $W(x)$ без остатка, т. е. $U(x) = 0$ и $M(x) = \frac{P(x)}{W(x)}$.

После еще одного деления имеем

$$\frac{P(x)}{W(x)} = \widetilde{M}(x)$$

и

$$M(x) = \widetilde{M}(x).$$

Показано, что решение сравнения (*), полученное применением расширенного алгоритма Евклида, всегда существует и совпадает с истинным решением. Ч. т. д.

Пример

Рассмотрим процедуры кодирования и декодирования для кода Рида—Соломона (4, 2, 3). Коэффициенты многочлена будем записывать как вектор-строку и как вектор-столбец (без обозначения операции транспонирования).

Конечное поле. Вначале введем конечное поле $GF(5) = \{0, 1, 2, 3, 4\} \pmod{5}$. Или в другом представлении: $GF(5) = \{0, \alpha^0, \alpha^1, \alpha^2, \alpha^3\}$, где $\alpha = 2$ — примитивный элемент:

$$\frac{\alpha^0 \alpha^1 \alpha^2 \alpha^3}{1 \ 2 \ 4 \ 3} \alpha^4 = 1.$$

Код Рида—Соломона. Введем код Рида—Соломона (4, 2, 3) над $GF(5)$. Его порождающий многочлен имеет вид

$$g(x) = (x - \alpha)(x - \alpha^2) = (x - 2)(x - 4) = x^2 + 4x + 3,$$

порождающая матрица

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & 0 \\ 0 & g_0 & g_1 & g_2 \end{bmatrix} = \begin{bmatrix} 3 & 4 & 1 & 0 \\ 0 & 3 & 4 & 1 \end{bmatrix}$$

и проверочная матрица

$$H = \begin{bmatrix} \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \end{bmatrix}.$$

Спектральное кодирование. Кодовое слово вычисляется по формуле

$$C = VM,$$

где матрица Вандермонда есть

$$V = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^2 \\ \alpha^0 & \alpha^3 & \alpha^2 & \alpha^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \\ 1 & 3 & 4 & 2 \end{bmatrix}.$$

Пусть информационный многочлен есть

$$M(x) = [m_0 \ m_1 \ 0 \ 0] = [2 \ 3 \ 0 \ 0] = 2 + 3x.$$

Тогда процедура кодирования имеет вид

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 \\ \alpha^0 & \alpha^2 & \alpha^0 & \alpha^2 \\ \alpha^0 & \alpha^3 & \alpha^2 & \alpha^1 \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ 0 \\ 0 \end{bmatrix};$$

$$\begin{bmatrix} 0 \\ 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 3 \\ 1 & 4 & 1 & 4 \\ 1 & 3 & 4 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \end{bmatrix};$$

$$C(x) = [c_0 \ c_1 \ c_2 \ c_3] = [0 \ 3 \ 4 \ 1] = 3x + 4x^2 + x^3.$$

Обратное преобразование. Запишем матрицу, обратную к матрице Вандермонда:

$$V^{-1} = - \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 4 & 2 \\ 1 & 4 & 1 & 4 \\ 1 & 2 & 4 & 3 \end{bmatrix}.$$

Тогда обратное дискретное преобразование Фурье есть

$$M = V^{-1}C;$$

$$C(x) = [c_0 \ c_1 \ c_2 \ c_3] = [0 \ 3 \ 4 \ 1] = 3x + 4x^2 + x^3;$$

$$\begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \end{bmatrix} = - \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 4 & 2 \\ 1 & 4 & 1 & 4 \\ 1 & 2 & 4 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \\ 4 \\ 1 \end{bmatrix};$$

$$M(x) = [m_0 \ m_1 \ 0 \ 0] = [2 \ 3 \ 0 \ 0] = 2 + 3x.$$

Декодирование. Пусть из канала принят вектор $R(x) = C(x) + E(x)$:

$$\begin{array}{r} C(x) = (0 \ 3 \ 4 \ 1) \\ E(x) = (0 \ 0 \ 2 \ 0) \\ R(x) = (0 \ 3 \ 1 \ 1) \\ \hline Z \quad \alpha^0 \ \alpha^1 \ \alpha^2 \ \alpha^3 \\ Z \quad \quad 1 \ 2 \ 4 \ 3 \end{array}$$

$$R(x) = [r_0 \ r_1 \ r_2 \ r_3] = [0 \ 3 \ 1 \ 1] = 3x + x^2 + x^3.$$

1. Интерполяция:

$$\begin{bmatrix} 0 \\ 0 \\ 3 \\ 2 \end{bmatrix} = - \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 4 & 2 \\ 1 & 4 & 1 & 4 \\ 1 & 2 & 4 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \\ 1 \\ 1 \end{bmatrix};$$

$$\mathbf{T} = V^{-1}\mathbf{R} = [T_0 \ T_1 \ T_2 \ T_3] = \\ = [0 \ 0 \ 3 \ 2] = 3x^2 + 2x^3.$$

2. Незаконченное вычисление наибольшего общего делителя:

$$\begin{cases} W(x)T(x) \equiv P(x) \pmod{x^n - 1} \\ \deg P(x) < \frac{n+k}{2} \\ \text{maximize } \deg P(x) \end{cases};$$

$$\begin{cases} W(x)(3x^2 + 2x^3) \equiv P(x) \pmod{x^4 - 1}; \\ \deg P(x) < 3 \end{cases};$$

$$x^4 - 1 = (3x^2 + 2x^3)(3x + 3) + (x^2 - 1);$$

$$(3x + 3)(3x^2 + 2x^3) = -(x^2 - 1) + x^4 - 1;$$

$$(3x + 3)(3x^2 + 2x^3) \equiv 4x^2 + 1 \pmod{x^4 - 1};$$

$$\begin{cases} W(x) = 3x + 3 = 3(x - 4) = 3(x - \alpha^2) \\ P(x) = 4x^2 + 1 \end{cases}.$$

3. Деление:

$$M(x) = \frac{P(x)}{W(x)} = \frac{4x^2 + 1}{3x + 3} = 3x + 2.$$

Информационный многочлен есть

$$M(x) = [m_0 \ m_1 \ 0 \ 0] = [2 \ 3 \ 0 \ 0] = 2 + 3x.$$

Заключение

Рассмотренный метод декодирования предложен Гао [7] и Шиозаки [8], однако вывод и анализ алгоритма, доказательство его корректности и генетическая связь с алгоритмами [1–4, 9, 10, 18] является оригинальным результатом автора [5, 6].

На текущий момент этот метод декодирования и его модификации представляются автору самыми простыми для кодов с ограниченной длиной при любых реализациях.

Автор выражает признательность фонду имени Александра фон Гумбольдта (Германия) за многолетнюю поддержку научных исследований.

Литература

1. Welch L., Berlekamp E. R. Error correction for algebraic block codes. U.S. Patent 4,633,470. Sep. 27, 1983.
2. Morii M., Kasahara M. Generalized key-equation of remainder decoding algorithm for Reed—Solomon codes // IEEE Transactions on Information Theory. Nov. 1992. Vol. IT-38. N 6. P. 1801–1807.
3. Chambers W. G. Solution of Welch—Berlekamp key equation by Euclidean algorithm // Electronics Letters. 1993. Vol. 29. N 11. P. 1031.
4. Gemmell P., Sudan M. Highly resilient correctors for polynomials // Information Processing Letters. 1992. Vol. 43. N 4. P. 169–174.
5. Fedorenko S. V. A simple algorithm for decoding Reed—Solomon codes and its relation to the Welch—Berlekamp algorithm // IEEE Transactions on Information Theory. Mar. 2005. Vol. IT-51. N 3. P. 1196–1198.
6. Fedorenko S. V. Correction to «A simple algorithm for decoding Reed—Solomon codes and its relation to the Welch—Berlekamp algorithm» // IEEE Transactions on Information Theory. Mar. 2006. Vol. IT-52. N 3. P. 1278.
7. Gao S. A new algorithm for decoding Reed—Solomon codes // Communications, Information and Network Security / V. Bhargava, H. V. Poor, V. Tarokh, and S. Yoon, Eds. Norwell, MA: Kluwer, 2003. Vol. 712. P. 55–68.
8. Shiozaki A. Decoding of redundant residue polynomial codes using Euclid's algorithm // IEEE Transactions on Information Theory. Sep. 1988. Vol. IT-34. N. 5. P. 1351–1354.
9. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов, исправляющих ошибки. М.: Связь, 1979. 744 с.
10. Блейхут Р. Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986. 576 с.
11. Justesen J. On the complexity of decoding Reed—Solomon codes // IEEE Transactions on Information Theory. Mar. 1976. Vol. IT-22. N 2. P. 237–238.
12. Wang Y., Zhu X. A fast algorithm for the Fourier transform over finite fields and its VLSI implementation // IEEE Journal on Selected Areas in Communications. Apr. 1988. Vol. 6. N 3. P. 572–577.
13. Afanasyev V. On complexity of FFT over finite field: Proc. of the Sixth Joint Swedish-Russian International Workshop on Information Theory. Molle, Sweden, August 1993. P. 315–319.
14. Трифонов П. В., Федоренко С. В. Метод быстрого вычисления преобразования Фурье над конечным полем // Проблемы передачи информации. 2003. Т. 39. Вып. 3. С. 3–10.
15. Moenck R. T. Fast computation of GCDs: Proc. of the 5th Annual ACM Symposium on Theory of Computing. Austin, TX, 1973. P. 142–151.
16. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979. 535 с.
17. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. М.: Мир, 1989. 448 с.
18. Sugiyama Y., Kasahara M., Hirasawa S., Namekawa T. A method for solving key equation for decoding Goppa codes // Information and Control. 1975. Vol. 27. P. 87–99.

УДК 004.72

ПОДХОД К ОРГАНИЗАЦИИ АДАПТИВНОГО СОГЛАСУЮЩЕГО ЦЕНТРА КОРПОРАТИВНОЙ СЕТИ

М. О. Колбанев,

доктор техн. наук, профессор

Т. М. Татарникова,

канд. техн. наук, доцент

К. О. Малков,

аспирант

Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича

Приводится один из подходов к организации согласующего центра, выполняющего функции интеграции и согласования неоднородных компьютерных сетей, основанный на динамической настройке структуры центра в зависимости от решаемых задач и внешних условий.

Корпоративная деятельность в распределенных технологических, организационно-административных и технических комплексах осуществляется с поддержкой информационных систем сетевого типа.

В отдельных частях корпоративной сети могут применятся различные телекоммуникационные технологии и уже сложившиеся процедуры обмена и управления данными. Все функции по интеграции и согласованию выполняются в межсетевых устройствах разного уровня и назначения: мостах, коммутаторах, маршрутизаторах и шлюзах, — обеспечивающих сопряжение и необходимое преобразование в терминах как аппаратуры, так и программного обеспечения.

Межсетевые устройства (МУ) реализуются в виде многопроцессорных мультипрограммных систем с определенной конфигурацией. Выполняя основную функцию взаимодействия нескольких неоднородных сетей, МУ обрабатывают большие потоки информации, что приводит к необходимости распараллеливания во времени функций приема, передачи и обработки поступающих на его вход сообщений. Классическим путем создания подобных систем является применение процессоров с типичными архитектурами I386, PowerPC, ARM, RISC и др. Такой подход обладает рядом недостатков: производительность МУ при выполнении фиксированных задач используется не полностью, а лишь на 10–20 %; жесткая привязка архитектуры к функциональности МУ не позволяет гибко управлять ресурсами и распределять нагрузку; моральное устаревание элементной базы при-

водит к необходимости полной замены межсетевого устройства.

Появившиеся на сегодняшний день технологии, инструментальная и элементная базы определили альтернативный подход к построению сетевого оборудования, основывающийся на идеях универсальности, аппаратной конфигурируемости и динамической настройки структуры МУ в зависимости от решаемых задач и внешних условий. Согласно этим идеям, синтез структуры МУ осуществляется в соответствии с модульным принципом построения из базовых блоков (модулей).

Базисный набор состоит из следующих структурных единиц (рисунок):

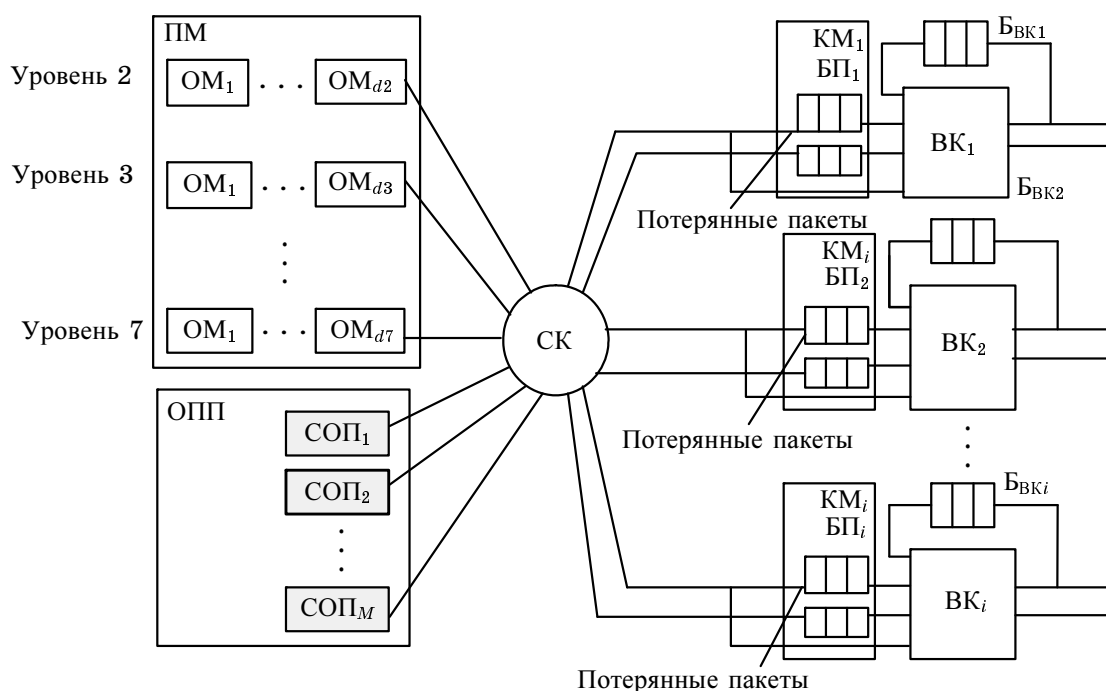
— канальных модулей KM , предназначенных для выполнения процессов приема и передачи кадров, управления перегрузкой с помощью системы квитирования и ограниченного объема буферной памяти BP в KM ; обозначим как множество

$M1 = \{KM_{i=1, S}\}$, где S — число сопрягаемых неоднородных сетей;

— процессорных модулей PM , предназначенных для выполнения уровней протоколов МУ. PM состоит из набора обрабатывающих модулей OM ;

PM обозначим как множество $M2 = \{OM_{i=1, N; j=2,7}\}$, где N — число OM , реализующих протоколы j -го уровня эталонной модели взаимодействия открытых систем (ЭМВОС);

— модуля памяти — общего поля памяти $ОПП$, предназначенного для хранения общих информа-



■ Структура универсального МУ

ционных ресурсов, необходимых при выполнении уровневого протокола и хранения передаваемых пакетов; обозначим как множество $MЗ$;

— схемы комплексирования $СК$, предназначенной для объединения всех модулей в рамках одного центра сопряжения;

— виртуального канала $ВК$ для отображения входа/выхода $КМ$ и поступающей на него нагрузки: как только интенсивность потока сообщений, принадлежащих $ВК$ увеличивается, большее число сообщений скапливается в буфере $БВК$ и для внешних сообщений соответственно увеличивается время задержки, что свидетельствует о возникновении перегрузки; обозначим как множество $M4$.

Структурно МУ выполнен в виде стойки с отсеками для вставки модулей. Для каждого конкретного случая взаимодействия сетей подбирается достаточная (не избыточная по производительности и минимальная по стоимости) комплектация МУ. Похожая технология, основанная на модульном принципе, давно применяется в центрах коммутации телефонных сетей общего пользования и мобильных телефонных сетях.

Предложенная модель универсального МУ позволяет расширять возможности МУ и структурируется из перечисленных готовых функциональных модулей в зависимости от нагрузки и степени неоднородности сопрягаемых сетей. С увеличением нагрузки возможно увеличение производительности МУ расширением процессорного модуля по горизонтали путем добавления одного или нескольких $ОМ$ определенного уровня. При подключении новой сети активизируется новый $КМ$ и до-

бавляются новые $ОМ$ по вертикали и секции общей памяти ($СОП$), число которых определяется заранее для соблюдения необходимых показателей качества обслуживания. Переключением связей между модулями задается схема комплексирования.

Канальный модуль — это интерфейсный модуль, выполняющий прием и передачу кадров по протоколам и функциям канального уровня ЭМВОС; передачу пакета через интерфейс в модуль памяти; фильтрацию пакетов по алгоритму моста, разгружающую процессорный модуль от обработки пакетов собственной сети.

Процессорный модуль выполняет функции обработки пакетов, т. е. принимает решения по их фильтрации/продвижению; модифицирует заголовки пакетов, если это необходимо; передает пакеты непосредственно присоединенному $КМ$ или другому процессору обработки пакетов; обновляет базу данных адресной и топологической информации; поддерживает SNMP-управление и выполняет другую административную работу. $ПМ$ состоит из набора $ОМ$. Каждый $ОМ$ поддерживает все основные протоколы определенного уровня ($ОМ_3$ — протоколы 3-го уровня и т. д. до 7-го уровня ЭМВОС) и имеет свои копии программного обеспечения, т. е. возможна параллельная обработка на разных $ОМ$ как одного и того же уровня, так и разных уровней.

Общее поле памяти разбито на секции для параллельного обслуживания нескольких канальных и обрабатывающих модулей $MЗ = \{ОПП_{i=1, M}\}$,

где M — число секций общего поля памяти. Память хранит различные общие информационные ресурсы, например маршрутные и адресные таблицы.

Организация вычислительного процесса в МУ также зависит от способа соединения функциональных блоков системы. По топологии межмодульных связей выделяются следующие схемы комплексирования: СК₁ — матричного типа; СК₂ — на базе общей шины; СК₃ — на базе разделяемой памяти.

Виртуальный канал в явном виде не является структурным элементом МУ, но его можно представить как некий функциональный модуль, задающий нагрузку на порты МУ.

Таким образом, модель некоего обобщенного МУ запишется в виде следующего кортежа элементов, характеристик и связей:

$$\text{МУ} = \langle M1(q_1, q_4, q_5, S, \text{БП}_{i=\overline{1, S}}), M2(q_1, q_5, N), \\ M3(q_2, q_3, M, RS), M4\bigcap_{i=1}^5 \text{СК}_i, G \rangle.$$

Частные варианты межсетевых устройств описываются следующими кортежами:

— мост:

$$\langle M1(q_1, q_4, q_5, S=2, \text{БП}_{i=\overline{1, 2}}), M4, \text{СК}_2, G \rangle;$$

— коммутатор на базе матрицы:

$$\langle M1(q_1, q_4, q_5, S, \text{БП}_{i=\overline{1, S}}), M4, \text{СК}_1, G \rangle;$$

— коммутатор на базе общей шины:

$$\langle M1(q_1, q_4, q_5, S, \text{БП}_{i=\overline{1, S}}), M4, \text{СК}_2, G \rangle;$$

— коммутатор на базе общей памяти:

$$\langle M1(q_1, q_4, q_5, S, \text{БП}_{i=\overline{1, S}}), \\ M3(q_2, q_3, M, RS), M4, \text{СК}_3, G \rangle;$$

— маршрутизатор:

$$\langle M1(q_1, q_4, q_5, S, \text{БП}_{i=\overline{1, S}}), \\ M2(q_1, q_5, N), M3(q_2, q_3, M, RS), \\ M4, (\text{СК}_1 \vee \text{СК}_3) \wedge (\text{СК}_2 \vee \text{СК}_3), G \rangle;$$

— шлюз:

$$\langle M1(q_1, q_4, q_5, S, \text{БП}_{i=\overline{1, S}}), \\ M2(q_1, q_5, N), M3(q_2, q_3, M, RS), \\ M4, (\text{СК}_1 \vee \text{СК}_3) \wedge (\text{СК}_2 \vee \text{СК}_3), G \rangle.$$

Структура МУ характеризуется открытой модульной архитектурой, приспособленной к автоматической реконфигурации, и состоит из набора функциональных модулей, каждый из которых выполняет функцию определенного протокола, на-

пример прием или передачу кадра, буферизацию, обработку поступившего пакета по протоколу и др. Каждый модуль является аппаратно-программируемым устройством. Такая технология позволяет, объединяя как функционально одинаковые модули, так и модули различного назначения, масштабировать структуру МУ. Возможность гибкого наращивания функций и протоколов делает структуру МУ универсальной, т. е. не зависящей от степени неоднородности взаимодействующих сетей.

В связи с этим можно ввести новый тип межсетевого устройства — адаптивный согласующий центр (АСЦ) корпоративной сети. При этом термин «адаптивный» принимает двойственное значение: во-первых, при работе данного согласующего центра на одном уровне ЭМВОС его производительность можно увеличивать или уменьшать, в зависимости от задачи, путем добавления или удаления соответствующих модулей, например модулей памяти; во-вторых, данный согласующий центр может адаптироваться для работы на разных уровнях ЭМВОС, опять же путем добавления или удаления модулей, например обрабатывающих модулей.

Поскольку современные корпоративные сети связи являются изменяющимися, развивающимися системами, где наряду с увеличением производительности МУ иногда необходимо и ее уменьшение, когда при этом высвобождаются производительные ресурсы, которые могут быть использованы в другой части сети, данные обстоятельства выгодно отличают АСЦ от простых межсетевых устройств.

В результате рассматриваемой идеи адаптации согласующего центра возникает задача оптимизации такого устройства с точки зрения производительности и стоимости. В связи с этим введем в рассмотрение множество параметров, определяющих МУ с точки зрения пользователя: показатели качества; параметры, задающие технические характеристики структурно-функциональных модулей МУ; параметры, характеризующие определенное решение, принятое для синтеза структуры межсетевого устройства.

Определены следующие показатели качества межсетевого устройства:

- пропускная способность МУ — g_1 ;
- задержка передачи кадра — g_2 ;
- вероятность блокировки принимаемых кадров для входящего канала i (процент потерянных кадров) — g_3 .

Ко множеству параметров Q , задающих технические характеристики структурно-функциональных модулей МУ, отнесены:

- номинальное быстродействие процессоров — q_1 ;
- цикл обращения к памяти — q_2 ;
- объем памяти — q_3 ;
- длина машинного слова для обмена процессоров с памятью — q_4 ;
- частота обращения процессора к памяти — q_5 .

В качестве оптимизируемых параметров H в зависимости от типа МУ выбраны следующие:

- число процессоров обработки или обрабатывающих модулей — N ;
- число СОП — M ;
- число буферов для хранения пакетов, закрепленных за каждым входящим каналом — $\text{БП}_{i=1, \overline{S}}$;
- матрица распределения ресурсов по секциям памяти — $RS_{i=1, \overline{R}; j=1, \overline{M}}$, где R — число общих ресурсов.

Таким образом, задача оптимизации сформулирована как смешанная многопараметрическая и многокритериальная экстремальная задача с ограничениями: при заданных значениях элементов множества параметров Q найти оптимальную структуру МУ, такую, чтобы выполнялись условия:

$$0 < g_i(Q, H) \leq g_i^{\text{доп}}, \quad g_i \in G; \quad (1)$$

$$\phi \rightarrow \min f(Q, H), \quad (2)$$

где ϕ — аддитивная функция стоимости МУ.

Решение сформулированной задачи распадается на два этапа. На первом этапе на соответствующих моделях находятся зависимости показателей качества. Для многопроцессорных вычислительных комплексов, каким является АСЦ, применим аппарат экспоненциальных сетей массового обслуживания [1]. Данный аппарат позволяет получить оценивание производительности и времен задержки в аналитическом виде.

На втором этапе решается оптимизационная задача, определяемая выражениями (1), (2). На сегодняшний день предпочтителен интеллектуальный подход в методах оптимизации [2], предполагающий использование методов выбора параметров МУ, которые гарантировали бы обеспечение требуемого качества обслуживания пользователей при минимальных затратах. Одним из таких методов является генетический алгоритм поиска оптимальной структуры устройства, который можно применить и в АСЦ [3].

Таким образом, использование адаптивных согласующих устройств в корпоративных сетях приводит к более эффективному использованию и администрированию сетей, обеспечивая при этом требуемое качество обслуживания пользователей.

Литература

1. **Ивницкий В. А.** Теория сетей массового обслуживания. М.: Физматлит, 2004. 772 с.
2. **Реклейтис Г., Рейвиндран А., Рэгсдел К.** Оптимизация в технике. М.: Мир, 1986. 346 с.
3. **Колбанев М. О., Кузьма Э. А., Яковлев С. А.** Внедрение генетических алгоритмов поиска в управлении инфокоммуникациями // Проблемы подготовки кадров в сфере инфокоммуникационных технологий: Санкт-Петербургская науч.-практ. конф. / СПОИСУ. СПб., 2005. С. 81–84.

УДК 007:004.3

МЕТОД МНОГОКРИТЕРИАЛЬНОГО РАНЖИРОВАНИЯ СЛОЖНЫХ СИСТЕМ ПРИ РАЗЛИЧНЫХ ВИДАХ НЕОПРЕДЕЛЕННОСТИ ИСХОДНЫХ ДАННЫХ

Ю. В. Ведерников,

канд. техн. наук, доцент

Михайловская военная артиллерийская академия

В. В. Сафронов,

доктор техн. наук, профессор

ОАО «КБ Электроприбор»

Рассматривается важная прикладная задача принятия решения для случая, когда критерии могут быть одновременно заданы: в формализованном, количественном виде; в неопределенном, лингвистическом виде; в частично формализованном виде. Задача сводится к построению упорядоченного множества эффективных вариантов (кортежа Парето) сложных технических систем для случая, когда критерии оптимальности неоднородны. Приведен численный пример.

Введение

Выбор оптимального варианта структуры сложных технических систем (СТС) из множества возможных допустимых вариантов является важной задачей, решаемой на ранних этапах их проектирования. В настоящее время накоплен большой опыт построения оптимальных структур СТС [1–7]. Разработаны методы решения, основанные на достижениях в области системного анализа, исследования операций, теории принятия решений, графов, дискретного программирования и т. д.

Однако при практическом использовании математических моделей возникают трудности, связанные с обеспечением полноты, точности и достоверности исходных данных, а также многокритериальным характером выбора варианта структуры. Для реальных СТС характерно наличие разнородной информации [8]: о точечных замерах и значениях параметров; о допустимых интервалах их изменения; о статистических законах распределения отдельных величин; о лингвистических оценках и ограничениях, полученных от специалистов-экспертов, и т. д.

Исследователь поставлен перед необходимостью округлять или огрублять имеющиеся у него знания об объекте исследования, математическая модель при этом оказывается недостаточно адекватной реальности, что может привести к неоптимальному выбору структуры синтезируемой системы.

Проведенный анализ [3, 4, 8, 9] показал, что задача выбора оптимальной структуры СТС в ус-

ловиях наличия неоднородных (формализованных, частично формализованных, неформализованных) критериев оптимальности относится к задачам многокритериального ранжирования. Однако разработка методов, которые могли бы решать задачи в условиях комплексного использования неоднородных исходных данных, на сегодняшний день еще полностью не завершена. Это создает определенную трудность при выборе оптимальных структур СТС, которая связана с тем, что достаточно сложно соизмерять улучшение значения одного критерия с ухудшением значения другого критерия, когда критерии представлены в различных измерительных шкалах; невозможно адекватно определить расстояние между разными классами эквивалентности.

В статье предлагается метод, который бы позволил решать указанный класс задач. При этом используется математический аппарат теории нечетких множеств [9–13] как основной аппарат формирования и нормализации (т. е. приведения к единой величине и виду, удобному для сравнения) исходных данных при решении задачи многокритериального ранжирования СТС.

Заметим, что еще в 1981 г. в предисловии к монографии С. А. Орловского [12] академик Н. Н. Моисеев акцентировал внимание на то, что «...Центральная процедура общего подхода к проблеме выбора альтернатив опирается на различные принципы отбраковки. И в этом контексте принципы Парето, гарантированного результата, формализм Заде занимают свое определенное ме-

сто. ... Вопрос о будущем всего этого направления решается не столько математическим совершенством развиваемой теории, сколько удобством, которое обеспечивается оперирующей стороне при анализе и выборе альтернатив».

Математическая постановка задачи

Для решения задач векторной оптимизации, в которых необходимо проводить учет различных видов неопределенности исходных данных, предлагается в качестве критериев использовать функции принадлежности, являющиеся средством числовой нормализации разнородной информации о качествах исследуемых СТС.

Рассмотрим математическую постановку задачи и с этой целью введем необходимые в дальнейшем обозначения [7, 12, 14–16].

1. $S = \{S_\alpha, \alpha = \overline{1, n}\}$ — множество возможных вариантов СТС.

2. $K(S_\alpha) = \{K_1(S_\alpha), K_2(S_\alpha), \dots, K_j(S_\alpha), \dots, K_r(S_\alpha)\}$ — векторный критерий, характеризующий систему S_α .

3. $K_j(S_\alpha), j = \overline{1, r}$ — частный критерий качества, характеризующий систему S_α .

4. $A = \{a_j, j = \overline{1, r}\}$ — множество коэффициентов важности критериев, где a_j — коэффициент важности j -го критерия, причем $\sum_{j=1}^r a_j = 1$.

5. $PK_j(S_k, S_l) = [(S_k, S_l); \mu K_j(S_k, S_l)]$ — нечеткое отношение предпочтения (НОП) по j -му частному критерию качества, $j = \overline{1, r} \forall k = \overline{1, n}, l = \overline{1, n}, k \neq l$, где (S_k, S_l) — множество упорядоченных пар систем; $\mu K_j(S_k, S_l)$ — функция принадлежности НОП.

6. $\mu_D K_j(S_k, S_l)$ — функция принадлежности нечеткого отношения строгого предпочтения, характеризующая интенсивность доминирования системы S_k над системой S_l по j -му частному критерию.

7. $\mu_{ND} K_j(S_k, S_l)$ — функция принадлежности отношения недоминирования, характеризующая степень, с которой система S_k недоминируется системой S_l по j -му частному критерию.

8. $\mu_D^* K_j(S_k)$ — функция принадлежности нечеткого множества недоминируемых систем, характеризующая степень «недоминируемости» системы S_k ни одной другой системой по j -му частному критерию.

9. $\mu_D^* K(S_k)$ — функция принадлежности нечеткого множества недоминируемых систем, характеризующая степень «недоминируемости» системы S_k ни одной другой системой по векторному критерию.

С учетом введенных обозначений сформулируем задачу. Даны множества S, A , выражения для вычисления $\mu K_j(S_k, S_l)$, решающие правила [3, 15].

Требуется найти множество эффективных упорядоченных систем (кортеж Парето) $S^P \subset S$, для элементов которого $S_\alpha^* \in S^P$ справедливо

$$\mu_D^* K(S_\alpha^*) = \max_{S_\alpha \in S} \mu_D^* K(S_\alpha). \quad (1)$$

Основная трудность при решении подобных задач заключается в том, что определение функций принадлежности является достаточно сложной процедурой, а отождествление функций принадлежности с критериальными функциями представляет собой трудоемкий процесс.

Метод нормализации неоднородных критериальных оценок

Для преодоления перечисленных трудностей и решения представленных задач предлагается не изменять содержательного смысла критериев, а ввести в рассмотрение расплывчатость шкал, в которых эти критерии фиксируются (т. е. искусственно «размыть» критерии). Сущность метода заключается в следующем [9, 12, 13, 15].

Определяем НОП $PK_j(S_k, S_l)$ по j -му частному критерию качества для пары решений (S_k, S_l) функцией принадлежности [9, 13]:

$$\mu K_j(S_k, S_l) = \begin{cases} \frac{K_j(S_k) - K_j(S_l)}{m_j}, & \text{если } K_j(S_k) > K_j(S_l); \\ 0, & \text{если } K_j(S_k) \leq K_j(S_l), \end{cases} \quad (2)$$

где m_j — ширина интервала оценок по j -му критерию; $K_j(S_k)$ и $K_j(S_l)$ — значения j -го критерия для систем S_k и S_l .

Важным моментом в данном случае является назначение величины m_j . В рассматриваемой постановке ширина интервала оценок будет равна максимальному значению каждого рассматриваемого

критерия, т. е. $\left[0; \max_{k=1, n} K_j(S_k)\right]$. При необходимости можно использовать в качестве m_j значения:

- критериев эталонной системы;
- критериев, которые хотелось бы достигнуть в ходе решения задачи оптимизации;
- выделяемых ресурсов по каждому из критериев.

В задачах контроля m_j эталонными значениями могут выступать предельно допустимые значения контролируемых параметров.

Нечеткое отношение строгого предпочтения системы S_k над системой S_l определим функцией принадлежности $\mu_D K_j(S_k, S_l)$, характеризующей интенсивность доминирования S_k над S_l по j -му частному критерию [12, 15]:

$$\mu_D K_j(S_k, S_l) = \begin{cases} \mu K_j(S_k, S_l) - \mu K_j(S_l, S_k), & \text{если } \mu K_j(S_k, S_l) > \mu K_j(S_l, S_k); \\ 0, & \text{если } \mu K_j(S_k, S_l) \leq \mu K_j(S_l, S_k). \end{cases} \quad (3)$$

■ Таблица 1

Критерий	Система			
	<i>Mx8000Dual</i> (S_1)	<i>HiSpeed NX/i</i> (S_2)	<i>Somatom Plus 4</i> (S_3)	<i>Asteion Multi</i> (S_4)
Возможность реконструкции системы в ходе измерения $K_1(S_\alpha)$, балл	1	3	4	5
Максимальный диаметр объекта контроля $K_2(S_\alpha)$, см ²	18,7	37,4	15,6	31,2
Максимальная длина поля сканирования $K_3(S_\alpha)$, см	171,7	367,2	149,49	304,44
Скорость движения опоры под объектом $K_4(S_\alpha)$, мм/с	800	1200	1150	1100

Отношение недоминирования системы S_k системой S_l определим функцией принадлежности $\mu_{ND}K_j(S_k, S_l)$ как дополнение к $\mu_DK_j(S_k, S_l)$ в виде [12, 15]

$$\mu_{ND}K_j(S_k, S_l) = 1 - \mu_DK_j(S_k, S_l). \quad (4)$$

Степень «недоминируемости» системы S_k ни одним другим вариантом по j -му частному критерию характеризуется функцией принадлежности нечеткому множеству недоминируемых вариантов $\mu_D^*K_j(S_k)$ и показывает степень полезности варианта системы по рассматриваемому критерию:

$$\mu_D^*K_j(S_k) = \min_{j=1,r} \mu_{ND}K_j(S_k, S_l). \quad (5)$$

Проиллюстрируем порядок использования формульных зависимостей (2)–(5).

Пример. Выбор оптимального варианта измерительной системы типа GANTRY.

Выбор будем осуществлять из четырех вариантов систем. Варианты систем и значения критериев оптимальности представлены в табл. 1.

Характеристика критериев:

— $K_1(S_\alpha)$ — критерий неформализованный, лингвистический (1 — «очень плохо», 3 — «удовлетворительно», 4 — «хорошо», 5 — «отлично»), интервал оценок $m_1 = [0; 5]$;

— $K_2(S_\alpha)$ — критерий формализованный, количественный интервал оценок $m_2 = [0; 37,4]$;

— $K_3(S_\alpha)$ — критерий формализованный, количественный интервал оценок $m_3 = [0; 367,2]$;

— $K_4(S_\alpha)$ — критерий формализованный, количественный интервал оценок $m_4 = [0; 1200]$.

Как видно из табл. 1, измеряемые критериальные величины неоднородны. Кроме того, хотя критерии $K_2(S_\alpha)$, $K_3(S_\alpha)$, $K_4(S_\alpha)$ и являются формализованными и выражаются количественно, однако они представлены в различных единицах измерения.

Проведем нормализацию критериальных оценок на основе аппарата теории нечетких множеств (т. е. искусственно «размоем» критерии).

Решение задачи.

1. С использованием (2) определяем значения функции принадлежности НОП $\mu_{K_1}(S_k, S_l)$ для всех k и l :

$$\mu_{K_1}(S_1, S_2) = \frac{1-3}{5} = 0; \mu_{K_1}(S_1, S_3) = \frac{1-4}{5} = 0;$$

$$\mu_{K_1}(S_1, S_4) = \frac{1-5}{5} = 0; \mu_{K_1}(S_2, S_1) = \frac{3-1}{5} = 0,4;$$

$$\mu_{K_1}(S_2, S_3) = \frac{3-4}{5} = 0; \mu_{K_1}(S_2, S_4) = \frac{3-5}{5} = 0;$$

$$\mu_{K_1}(S_3, S_1) = \frac{4-1}{5} = 0,6; \mu_{K_1}(S_3, S_2) = \frac{4-3}{5} = 0,2;$$

$$\mu_{K_1}(S_3, S_4) = \frac{4-5}{5} = 0; \mu_{K_1}(S_4, S_1) = \frac{5-1}{5} = 0,8;$$

$$\mu_{K_1}(S_4, S_2) = \frac{5-3}{5} = 0,4; \mu_{K_1}(S_4, S_3) = \frac{5-4}{5} = 0,2.$$

Аналогично вышепредставленным вычислениям определяем значения $\mu_{K_2}(S_k, S_l)$, $\mu_{K_3}(S_k, S_l)$, $\mu_{K_4}(S_k, S_l)$ и полученные данные сведем в табл. 2.

2. С использованием (4) находим значения $\mu_{ND}K_1(S_k, S_l)$, $\mu_{ND}K_2(S_k, S_l)$, $\mu_{ND}K_3(S_k, S_l)$ и $\mu_{ND}K_4(S_k, S_l)$. Полученные данные представим в табл. 3.

3. С помощью (5) вычисляем интенсивность доминирования каждого варианта системы $\mu_D^*K_j(S_k)$ по критериям K_1, K_2, K_3, K_4 . Выбор значений осуществляем из l -х столбцов оценочных матриц (табл. 3, выделенные значения). Данные сведены в табл. 4.

В результате преобразований (2)–(5) значение критерия $K_j(S_k)$ для системы S_k принадлежит нечеткому множеству. Функция принадлежности $\mu_D^*K_j(S_k)$ показывает степень полезности варианта S_k по рассматриваемому критерию. Эту величину будем рассматривать уже как новый нормализо-

■ Таблица 2

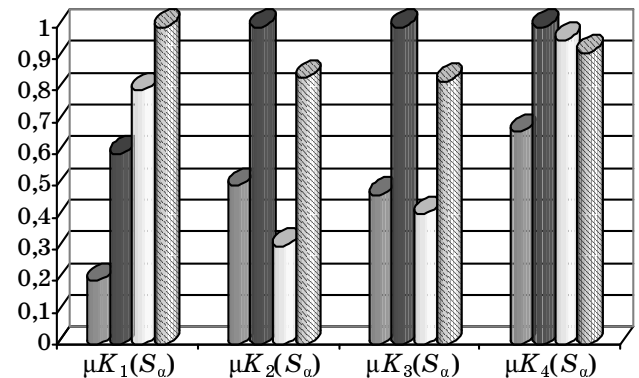
Система S_k	Система S_l			
	S_1	S_2	S_3	S_4
$\mu_{K_1}(S_k, S_l)$				
S_1	0	0	0	0
S_2	0,4	0	0	0
S_3	0,6	0,2	0	0
S_4	0,8	0,4	0,2	0
$\mu_{K_2}(S_k, S_l)$				
S_1	0	0	0,08	0
S_2	0,5	0	0,69	0,16
S_3	0	0	0	0
S_4	0,33	0	0,41	0
$\mu_{K_3}(S_k, S_l)$				
S_1	0	0	0,06	0
S_2	0,53	0	0,59	0,17
S_3	0	0	0	0
S_4	0,36	0	0,42	0
$\mu_{K_4}(S_k, S_l)$				
S_1	0	0	0	0
S_2	0,33	0	0,04	0,08
S_3	0,29	0	0	0,04
S_4	0,25	0	0	0

■ Таблица 3

Система S_k	Система S_l			
	S_1	S_2	S_3	S_4
$\mu_{ND}K_1(S_k, S_l)$				
S_1	1	1	1	1
S_2	0,6	1	1	1
S_3	0,4	0,8	1	1
S_4	0,2	0,6	0,8	1
$\mu_{ND}K_2(S_k, S_l)$				
S_1	1	1	0,92	1
S_2	0,5	1	0,31	0,84
S_3	1	1	1	1
S_4	0,67	1	0,59	1
$\mu_{ND}K_3(S_k, S_l)$				
S_1	1	1	0,94	1
S_2	0,47	1	0,41	0,83
S_3	1	1	1	1
S_4	0,64	1	0,58	1
$\mu_{ND}K_4(S_k, S_l)$				
S_1	1	1	1	1
S_2	0,67	1	0,96	0,92
S_3	0,71	1	1	0,96
S_4	0,75	1	1	1

■ Таблица 4

Система S_k	$\mu_D^*K_1(S_k)$	$\mu_D^*K_2(S_k)$	$\mu_D^*K_3(S_k)$	$\mu_D^*K_4(S_k)$
S_1	0,2	0,5	0,47	0,67
S_2	0,6	1	1	1
S_3	0,8	0,31	0,41	0,96
S_4	1	0,84	0,83	0,92

■ Нормализованные разнородные критериальные оценки: ■ — S_1 ; ■ — S_2 ; □ — S_3 ; ▨ — S_4

ванный критерий. В итоге все критериальные оценки приведены к общему виду, удобному для сравнения при решении задачи оптимизации СТС.

Графически результаты нормализации разнородных критериальных оценок вариантов измерительных систем представлены на рисунке.

При использовании данного метода существенным является факт определения нижней границы, т. е. определение минимально допустимого значения $\mu_D^*K_j(S_k)$ путем решения опытно-экспериментальных задач.

Метод ранжирования структур, характеризующихся множеством неоднородных критериев

Рассмотренный метод нормализации неоднородных критериальных оценок позволяет решать задачу выбора лучшего варианта S_k^* по векторному нормализованному критерию оптимальности (1).

Ответственным этапом при решении подобного рода задач является формирование коэффициентов важности критериев, значения которых для рассматриваемого примера приведены в табл. 5.

■ Таблица 5

$\mu_D^*K_j(S_k)$	Коэффициент важности a_i
$\mu_D^*K_1(S_k)$	$a_1 = 0,15$
$\mu_D^*K_2(S_k)$	$a_2 = 0,5$
$\mu_D^*K_3(S_k)$	$a_3 = 0,3$
$\mu_D^*K_4(S_k)$	$a_4 = 0,05$

Проведенный анализ показал, что для решения задачи (1) наиболее эффективным представляется метод «жесткого» ранжирования [7, 16]. В ходе решения задачи будем анализировать множество упорядоченных пар $S_k \in S, S_l \in S (k=\overline{1, n}; l=\overline{1, n}; k \neq l)$, а результат анализа заносить в специальную оценочную матрицу $\|C_{kl}\|$.

Сущность метода заключается в следующем.

1. На основе попарного сравнения вариантов

$S_k, S_l (k=\overline{1, n}; l=\overline{1, n}; k \neq l)$ определяем элементы C_{kl} оценочной матрицы $\|C_{kl}\|$. Значения элементов C_{kl} подбирают таким образом, чтобы отсеять неэффективные варианты. У эквивалентных вариантов S_k, S_l все соответствующие критерии равны. Полагаем $C_{kl} = 1$. К числу неэффективных отнесем варианты, у которых:

а) все значения критериев k -го варианта хуже, чем l -го варианта, тогда полагаем $C_{kl} = N_2 \gg 1$;

б) значения $m (m < r)$ критериев k -го варианта хуже соответствующих значений критериев l -го варианта при равных соответствующих значениях остальных критериев этих систем. Тогда полагаем $C_{kl} = N_3, 1 \ll N_3 < N_2$.

Если же для систем k, l имеем лучшие, худшие и, возможно, равные критерии, то значение C_{kl} определим по методу, изложенному в работе [17].

Запишем выражения для элементов оценочной матрицы. Обозначим $N_{kl}^+, N_{kl}^-, N_{kl}^-$ соответственно подмножества номеров лучших, худших и равных критериев для каждой пары вариантов $S_k, S_l (k=\overline{1, n}; l=\overline{1, n}; k \neq l)$. Будем осуществлять попарное сравнение систем S_k, S_l на основе анализа критериев $K_j(S_k), K_j(S_l), j=\overline{1, r}$. Для возможных значений подмножеств номеров $N_{kl}^+, N_{kl}^-, N_{kl}^-$ критериев введем следующие значения элементов оценочной матрицы $\|C_{kl}\|$:

$$\text{если } N_{kl}^+ = \emptyset, N_{kl}^- = \emptyset, N_{kl}^- = \{\overline{1, r}\}, \quad (6)$$

$$\text{то } C_{kl} = 1, C_{lk} = 1; \quad (7)$$

$$\text{если } N_{kl}^+ = \{\overline{1, r}\}, N_{kl}^- = \emptyset, N_{kl}^- = \emptyset, \quad (8)$$

$$\text{то } C_{kl} = N_2, C_{lk} = 0, N_2 \gg 1; \quad (9)$$

$$\text{если } N_{kl}^+ = \emptyset, N_{kl}^- = \{\overline{1, r}\}, N_{kl}^- = \emptyset, \quad (10)$$

$$\text{то } C_{kl} = 0, C_{lk} = N_2; \quad (11)$$

$$\text{если } N_{kl}^+ \neq \emptyset, N_{kl}^- = \emptyset, N_{kl}^- \neq \emptyset, \quad (12)$$

$$\text{то } C_{kl} = N_3, C_{lk} = 0, 1 \ll N_3 < N_2; \quad (13)$$

$$\text{если } N_{kl}^+ = \emptyset, N_{kl}^- \neq \emptyset, N_{kl}^- \neq \emptyset, \quad (14)$$

$$\text{то } C_{kl} = 0, C_{lk} = N_3; \quad (15)$$

$$\text{если } N_{kl}^+ \neq \emptyset, N_{kl}^- \neq \emptyset, |N_{kl}^-| \geq 0, \quad (16)$$

то C_{kl} определим в виде [17]

$$C_{kl} = \left(\sum_{j \in N_{kl}^+} a_j \right) \left(\sum_{j \in N_{kl}^-} a_j \right)^{-1}, C_{lk} = C_{kl}^{-1}. \quad (17)$$

2. Для формулировки решающих правил введем числа: H_l — количество элементов в l -м столбце оценочной матрицы, значение которых больше единицы; M_l — количество элементов в l -м столбце той же матрицы, значение которых меньше единицы; $C_{kl\max}$ — максимальное значение элемента в l -м столбце матрицы $\|C_{kl}\|$.

Физический смысл чисел: H_l показывает, сколько вариантов из рассматриваемого множества превышают l -й; M_l — в скольких вариантах доминирует l -я система; $C_{kl\max}$ определяет, во сколько раз l -й вариант «превышается» k -м ($k \in \{\overline{1, n}\}, k \neq l$).

3. Для реализации «жесткого» ранжирования перейдем от одношагового процесса поиска приоритетного расположения альтернатив к многошаговому процессу [2]. На каждом шаге $t (t = 1, 2, \dots, n - 1)$ выбираем j -ю альтернативу, лучшую с точки зрения предлагаемого ниже решающего правила. Затем ее номер включаем в кортеж Парето P и в последующем рассмотрении j -я альтернатива больше не участвует (в матрице $\|C_{kl}\|$ вычеркиваем j -ю строку и j -й столбец). Это позволяет исключить влияние варианта S_j на выбор лучшей альтернативы, проводимой уже на шаге $(t + 1)$.

При формулировке решающих правил вновь используем, но теперь на каждом шаге t , числа $H_l^{(t)}, M_l^{(t)}, C_{kl\max}^{(t)}$, которые имеют оговоренный выше физический смысл.

Решающие правила «жесткого» ранжирования.

- Ранжирование необходимо проводить среди эффективных альтернатив по шагам. Число шагов $t \leq (n - 1)$.

- На каждом шаге $t (t = 1, 2, \dots, n - 1)$:

- найти числа $H_l^{(t)}, M_l^{(t)}, C_{kl\max}^{(t)}$ и определить лучшую альтернативу S_j с минимальным значением $H_j^{(t)}$ и $C_{ij} \geq 1 \forall l \in \{\overline{1, n}\}, l \neq j$;

- номер j занести в множество P ;

- исключить из оценочной матрицы j -ю строку и j -й столбец.

Если альтернативы с номерами $l_j \in L_{k(t)} = \{l_1, l_2, \dots, l_j, \dots, l_{k(t)}\}$ имеют одинаковые минимальные значения $H_{l_j}^{(t)}$, то лучшей является альтернатива

S_{l_j} с максимальным значением $M_{l_j}^{(t)} = \max_{l_j \in L_{k(t)}} M_{l_j}^{(t)}$.

- Если варианты с номерами $l_j \in L_{k(t)} = \{l_1, l_2, \dots, l_j, \dots, l_{k(t)}\}$ имеют соответственно одинаково-

вые значения $H_{ij}^{(t)}, M_{ij}^{(t)}$, то лучшей является альтернатива S_{ij} с минимальным значением $C_{klj}^{(t)}$.

• Если лучшие системы имеют соответственно равные значения $H_{ij}^{(t)}, M_{ij}^{(t)}, C_{klj}^{(t)}$, то такие системы считают эквивалентными.

В ходе построения решающих правил при выполнении условий (16) были использованы аналитические выражения (17) Б. Руа [17]. При несомненных достоинствах такой подход не лишен и определенных недостатков: не учитывается, на какую величину значения соответствующих критериев больше (меньше) друг друга; не определяется, насколько далека система от идеальной. Для устранения этих недостатков при определении решающих правил в работе [7] предлагается применять не только схему Руа [17], но и методы «идеальной» точки в пространстве критериев и равномерной оптимальности [3]. Например, при использовании метода равномерной оптимальности выражение для вычисления элементов C_{kl} оценочной матрицы при $N_{kl}^+ \neq \emptyset, N_{kl}^- \neq \emptyset, |N_{kl}^-| \geq 0, (k=1, n; l=1, n; k \neq l)$ имеет вид [7]

$$C_{kl} = \left(\sum_{j \in N_{kl}^+} a_j \Delta_j \right) \left(\sum_{j \in N_{kl}^-} a_j \Delta_j \right)^{-1}, C_{lk} = C_{kl}^{-1}, \quad (18)$$

где $\Delta_j = |K_j(S_l) - K_j(S_k)|$.

В настоящей статье, с учетом особенностей рассматриваемой задачи, формулу (17) преобразуем к виду

$$C_{kl} = \left(\sum_{j \in N_{kl}^+} a_j \mu_D^* K_j(S_k) \right) \left(\sum_{j \in N_{kl}^-} a_j \mu_D^* K_j(S_l) \right)^{-1},$$

$$C_{lk} = C_{kl}^{-1}. \quad (19)$$

При этом физический смысл чисел $H_l, M_l, C_{kl\max}$, вводимых для формирования решающих правил и реализации метода «жесткого ранжирования», остается прежним.

Рассмотрим формирование числа C_{23} классическим методом «жесткого» ранжирования:

$$C_{23} = \frac{a_2 + a_3 + a_4}{a_1} = \frac{0,5 + 0,3 + 0,05}{0,15} = 5,67,$$

и предлагаемой его модификацией:

$$C_{23} = \frac{a_2 \mu_D^* K_2(S_2) + a_3 \mu_D^* K_3(S_2) + a_4 \mu_D^* K_3(S_2)}{a_1 \mu_D^* K_1(S_3)} =$$

$$= \frac{0,5 \cdot 1 + 0,3 \cdot 1 + 0,05 \cdot 1}{0,15 \cdot 0,8} = 7,08.$$

В результате использования метода «жесткого» ранжирования с учетом формулы (19) первая оценочная матрица для рассматриваемого примера будет иметь вид, представленный в табл. 6.

4. Анализ оценочной матрицы позволяет получить характерные числа $H_l^{(1)}, M_l^{(1)}, C_{kl\max}^{(1)}$, которые приведены в табл. 7.

5. Анализ табл. 7 показывает, что система S_1 является неэффективной (см. теорему 1 [7]). Исключаем S_1 из рассмотрения (в оценочной матрице вычеркиваем первую строку и первый столбец).

6. Ход дальнейших решений представлен в табл. 8 и 9.

Лучшей является система S_2 . Включаем ее в кортеж Парето P . В табл. 8 удаляем первую строку и первый столбец. Получаем оценочную матрицу (табл. 10) и характерные числа (табл. 11).

7. Лучшей на шаге 3, в соответствии с принятыми правилами, является система S_4 . В ре-

■ Таблица 6

Система S_k	Система S_l			
	S_1	S_2	S_3	S_4
S_1	–	0	2,56	0
S_2	N_2	–	7,08	5,67
S_3	0,39	0,105	–	0,06
S_4	N_2	0,176	17,06	–

■ Таблица 7

Числа	Система S_l			
	S_1	S_2	S_3	S_4
$H_l^{(1)}$	2	0	3	1
$M_l^{(1)}$	1	2	0	1
$C_{kl\max}^{(1)}$	N_2	0,176	17,06	5,67

■ Таблица 8

Система S_k	Система S_l		
	S_2	S_3	S_4
S_2	–	7,08	5,67
S_3	0,105	–	0,06
S_4	0,176	17,06	–

■ Таблица 9

Числа	Система S_l		
	S_2	S_3	S_4
$H_l^{(2)}$	0	3	1
$M_l^{(2)}$	2	0	1
$C_{kl\max}^{(2)}$	0,176	17,06	5,67

■ Таблица 10

Система S_k	Система S_l	
	S_3	S_4
S_3	–	0,06
S_4	17,06	–

■ Таблица 11

Числа	Система S_l	
	S_3	S_4
$H_l^{(3)}$	1	0
$M_l^{(3)}$	0	1
$C_{klmax}^{(3)}$	17,06	0,06

зультате получим следующий кортеж Парето $P = \langle S_2, S_4, S_3 \rangle$, т. е. предпочтение следует отдать второй системе (*HiSpeed NX/i*).

Заключение

Таким образом, поставлена и решена важная в прикладном плане задача многокритериального ранжирования сложной технической системы при разнородных исходных данных, которая сведена к задаче построения упорядоченного множества эффективных вариантов СТС для случая, когда критерии оптимальности предварительно нормализованы.

Предложенный метод позволяет ранжировать системы, характеризующиеся множеством формализованных и неформализованных величин, дает возможность ввести в рассмотрение величины с лингвистическими оценками качеств систем. Ключевым моментом метода является то, что содержательный смысл критериев не изменяется, а вводится в рассмотрение расплывчатость шкал, в которых эти критерии фиксируются, т. е. критерии искусственно «размываются». На наш взгляд, метод может найти применение при решении прикладных задач принятия решений в экономике, социальной сфере, оценке вариантов сложных технических систем.

Литература

1. Алексеев О. Г. Комплексное применение методов дискретной оптимизации. М.: Наука, 1987. 248 с.
2. Белкин А. Р., Левин М. Ш. Принятие решений: комбинаторные модели аппроксимации информации. М.: Наука. 1990. 160 с.
3. Дубов Ю. А., Травкин С. И., Якимец В. Н. Многокритериальные модели формирования и выбора вариантов систем. М.: Наука, 1986. 296 с.
4. Емельянов С. В., Ларичев О. И. Многокритериальные методы принятия решений. М.: Знание, 1985. 32 с.
5. Перегудов Ф. И., Тарасенко Ф. П. Введение в системный анализ: Учеб. пособие для вузов. М.: Высш. шк., 1989. 367 с.
6. Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач. М.: Наука, 1982. 256 с.
7. Сафронов В. В. Основы системного анализа: методы многокритериального ранжирования: Монография / Поволж. кооп. ин-т Российского ун-та кооперации. Энгельс: Ред.-изд. центр ПКИ, 2007. 185 с.
8. Алтунин А. Е., Семухин М. В. Модели и алгоритмы принятия решений в нечетких условиях: Монография. Тюмень: Изд-во Тюменского государственного университета, 2000. 352 с.
9. Жуковин В. Е. Нечеткие многокритериальные модели принятия решений. Тбилиси: Мецниереба, 1988. 71 с.
10. Заде Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 165 с.
11. Заде Л. А. Размытые множества и их применение в распознавании образов и кластер-анализе // Классификация и кластер. М.: Мир, 1980. С. 208–247.
12. Орловский С. А. Проблемы принятия решений при нечеткой исходной информации. М.: Наука, 1981, 203 с.
13. Трахтенгерц Э. А. Взаимодействие агентов в многоагентных системах // АиТ. 1998. № 8. С. 3–52.
14. Ведерников Ю. В. Современные методы системного анализа: Монография / СПБИЭУ. СПб., 2007. 154 с.
15. Сафронов В. В., Ведерников Ю. В., Шахова О. А. Векторная оптимизация сложных технических систем при неопределенности исходных данных // Информационные технологии. 2001. № 2. С. 49–63.
16. Сафронов В. В. Гипервекторное ранжирование сложных систем // Информационные технологии. 2003. № 5. С. 23–26.
17. Руа Б. Проблемы и методы решений в задачах с многими целевыми функциями // Вопросы анализа и процедуры принятия решений. М.: Мир, 1976. С. 20–58.

УДК 004.435

АРХИТЕКТУРА ПРОЦЕССОРА МЕТАДААННЫХ

Д. В. Ковалев,

аспирант

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

В. В. Шаповалов,

доктор техн. наук, профессор

Федеральное государственное научное учреждение «Научно-исследовательский конструкторско-технологический институт биотехнических систем»

Приводятся краткие сведения о методологии метауправления и описывается подход, обеспечивающий независимость реализации метауправления в информационных системах от конкретного языка представления и средства доступа к метаданным.

Введение

На заре развития компьютерных технологий разработка программного обеспечения (ПО) была искусством, которым владела горстка посвященных. Сейчас программирование относится к категории общепризнанных научных технологий [1]. Создание, внедрение и сопровождение программных систем — это набирающий обороты бизнес, в который вовлечены сотни тысяч квалифицированных специалистов.

Свободная конкуренция вынуждает производителей ПО выпускать качественные программы в максимально короткие сроки. Современные реалии таковы, что помимо первичных требований к программным системам, таких как полнота функциональных возможностей, надежность, производительность и дружелюбность интерфейса пользователя, все большую значимость приобретают требования, ранее бывшие вторичными. Речь идет об оперативности создания и модернизации программных продуктов, а также независимости успешного выполнения проекта от конкретных исполнителей. В таких условиях, как отмечает Эдвард Йордон [2], параметры (срок разработки, штат разработчиков и бюджет) большинства современных софтверных проектов отклоняются от нормальных значений на 50 % и больше. Подобные проекты Йордон относит к категории «безнадежных» и утверждает, что «безнадежные проекты являются нормой, а не исключением».

В целом проблема «выживания» в безнадежных проектах является весьма актуальной и только совершенствованием методов управления решена быть не может — необходимо использование прогрессивных технологий, таких как обобщенное и объектно-ориентированное программирование,

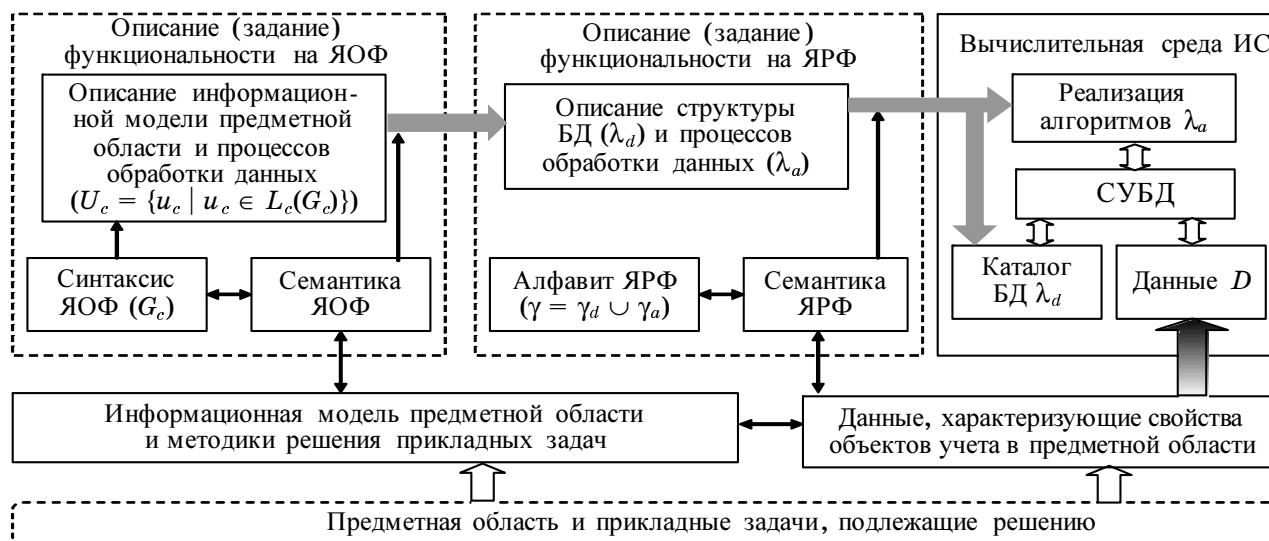
программирование с использованием шаблонов (паттернов), мультипарадигменное программирование, рефакторинг, разработка через тестирование, методология метауправления функциональностью информационных систем и др. Применение этих технологий должно повысить сопровождаемость ПО и, как следствие, оперативность разработки, а также обеспечить читабельность исходного текста, полноту и актуальность документации, что позволит добиться независимости от исполнителей.

Таким образом, исследования, направленные на повышение гибкости и унифицированности прогрессивных технологий, всегда будут своевременными и актуальными. В настоящей статье рассматривается архитектура процессора метаданных MDE (Metadata Engine) как средства повышения гибкости реализации метауправления в информационных системах.

Методология метауправления

Метауправление (МУ) — это методология построения информационных систем (ИС), при которой пользователю предоставляется возможность варьировать структуру данных и содержание вычислительного процесса их обработки при изменениях в соответствующей предметной области [3]. Другими словами, в ИС с МУ реализована возможность метауправления функциональностью во время ее эксплуатации с использованием понятийной модели соответствующей предметной области.

Совокупность структур хранения данных и алгоритмов выполнения типовых операций их обработки можно рассматривать как алфавит ИС, который определим как язык реализации функциональности (ЯРФ). При метауправлении со сторо-



■ Рис. 1. Использование языков описания и реализации функциональности

ны пользователя алфавиту ставится в соответствие метаописание понятийной модели предметной области и типовых операций обработки предметных данных. Используя имеющийся алфавит, пользователь формирует (изменяет) метаописания, далее преобразуемые в описания структур баз данных (БД) и (или) определяющие содержание процессов обработки предметных данных.

Таким образом, суть МУ функциональностью информационной системы заключается в возможности задания метаописаний компонентов ИС, определяющих ее функциональность, на предметно-ориентированном дескриптивном языке. Выполнение метаописаний должно приводить к получению результатов решения прикладных задач, а при необходимости — и к изменению используемых структур хранения данных.

Алфавит γ ЯРФ образуют идентификаторы структур хранения данных в БД и заданных на них отношений γ_d , а также идентификаторы алгоритмов обработки данных γ_a . Синтаксис ЯРФ определяется его алфавитом и правилами ν композиции символов алфавита в логически правильные (синтаксически допустимые) цепочки (предложения), описывающие структуру БД (λ_d) и процессы обработки данных (λ_a). В свою очередь, правила ν — это правила, являющиеся результатом применения синтаксических правил инструментальных лингвистических средств, использованных при создании БД и функциональных приложений ИС к алфавиту γ . Для любой ИС можно определить следующие понятия:

$\gamma = \gamma_d \cup \gamma_a$ — алфавит ЯРФ;

$G_R = (\gamma, \nu)$ — грамматика, описывающая синтаксис ЯРФ;

$L_R(G_R)$ — собственно ЯРФ как множество всех терминальных цепочек, порождаемых грамматикой G_R ;

$\lambda = \lambda_d \cup \lambda_a$ — совокупность предложений ЯРФ ($\lambda \in L_R(G_R)$), реализованных в ИС и определяющих ее функциональность.

Реализация свойства синтаксической вариативности (возможности варьирования алфавита) в ИС предполагает введение нового языка — языка описания функциональности (ЯОФ), предоставляемого пользователю для задания (описания) γ_d , γ_a и λ_d , λ_a (рис. 1). По своей сути это должен быть предметно-ориентированный язык декларативного типа, синтаксис и семантика которого определяются при создании конкретной ИС (на рис. 1 этот язык обозначен L_c , а его грамматика — G_c).

Поскольку, манипулируя предложениями на ЯОФ, пользователь осуществляет управление синтаксисом ЯРФ и (или) описаниями на ЯРФ, а предложения на ЯРФ определяют структуру БД и возможные процессы содержательной обработки данных в интересах решения прикладных задач, в синтаксически вариативной информационной системе реализуется двухуровневая схема управления. Верхний уровень управления по данной схеме является уровнем метауправления.

Совокупность предложений на ЯОФ U_c (см. рис. 1) можно рассматривать как совокупность компонентов метаинформации (МИ), имеющейся в ИС. Каждый компонент метаинформации u_c является одним предложением на ЯОФ L_c ($u_c \in L_c$).

Таким образом, метауправление функциональностью ИС заключается в разнесении предметно-ориентированного описания компонентов, определяющих функциональность ИС, и программной реализации этих компонентов так, что изменение описания приводит к формально вычисляемым изменениям в указанной выше реализации.

Особенности реализации метауправления в информационных системах

Для представления МИ необходима соответствующая модель. В настоящее время во множестве приложений с МУ используется так называемая базовая модель представления метаинформации (БМПМ), организующая знания в виде дерева следующих понятий: сущностей, атрибутов, экземпляров сущностей, значений атрибутов и отношений.

В качестве языка представления метаинформации во многих ИС с МУ используется расширяемый язык разметки XML (Extensible Markup Language), соответствующий БМПМ (таблица).

Однако такое решение ограничивает разработчиков возможностями языка и средств доступа к сформированным на этом языке компонентам МИ. XML как языку представления МИ присущи следующие недостатки:

- 1) отсутствие типизации атрибутов;
- 2) отсутствие поддержки для указания отношений между сущностями, кроме отношения «предок — потомок», что вызвано отсутствием поддержки для идентификации:
 - экземпляров сущностей;
 - xml-документа как категории сущностей;
- 3) большой объем xml-документа.

Что касается средств доступа к компонентам МИ, описанным на XML, то их недостатками являются:

- 1) низкая производительность, что обуславливается:
 - необходимостью загрузки всего xml-документа в оперативную память;
 - возможностью только последовательного доступа к экземплярам сущностей;
- 2) отсутствие приемлемого контроля семантической целостности (действительности) xml-документа.

Кроме того, метаданные в свою очередь могут выступать в роли данных, подлежащих учету в другой ИС. В этом случае метаданные должны храниться в сетевой или локальной БД, а для доступа

должны использоваться средства соответствующей системы управления базами данных (СУБД), которые для разных СУБД принципиально различны.

Таким образом, использование при реализации МУ в информационных системах конкретного языка представления метаинформации и, следовательно, соответствующих средств доступа к метаданным в общем случае представляется неэффективным.

Наиболее предпочтительным выглядит подход, обеспечивающий независимость реализации МУ в ИС от конкретного языка представления, что достигается путем внесения дополнительной прослойки между средствами формирования, перевода и интерпретации компонентов МИ и метаданными. При этом метаданные целесообразно представлять в виде БМПМ. В качестве упомянутой прослойки можно предложить процессор метаданных MDE (Metadata Engine).

Процессор метаданных MDE

MDE — это унифицированный механизм доступа к метаданным (рис. 2).

Абоненты метаданных — это объекты ИС с МУ, осуществляющие формирование, перевод и интерпретацию компонентов МИ. Для доступа к метаданным абоненты используют MDE, а точнее — провайдеры служб через интерфейсы оболочки.

Процессор метаданных MDE предоставляет абонентам доступ к метаданным, выраженным с использованием конкретного средства представления, в виде БМПМ. Таким образом, MDE обеспечивает независимость абонентов от средств представления метаданных. MDE состоит из оболочки доступа к метаданным и провайдеров служб.

Оболочка доступа к метаданным представляет метаданные в виде БМПМ. Оболочка является набором полиморфных интерфейсов семейства IMetaDataEngine. Под интерфейсом понимается абстрактный класс, состоящий только из открытых «чисто» виртуальных функций. Здесь и далее используются понятия объектно-ориентированного программирования (ООП), подробнее о котором можно узнать, например, из работы [4].

Провайдеры служб представляют метаданные, выраженные с использованием конкретных средств представления, в виде БМПМ. Доступ к метаданным осуществляется через провайдеры метаданных. Провайдеры служб являются классами, реализующими интерфейсы семейства IMetaDataEngine. Под реализацией интерфейса понимается наследование и реализация всех его методов. Для каждого средства представления метаданных должна быть своя группа провайдеров служб.

Провайдеры метаданных обеспечивают доступ к метаданным, выраженным с использованием конкретного средства представления.

Метаданные — это выраженные с использованием конкретного средства представления компоненты МИ.

■ Соответствие понятий БМПМ и XML

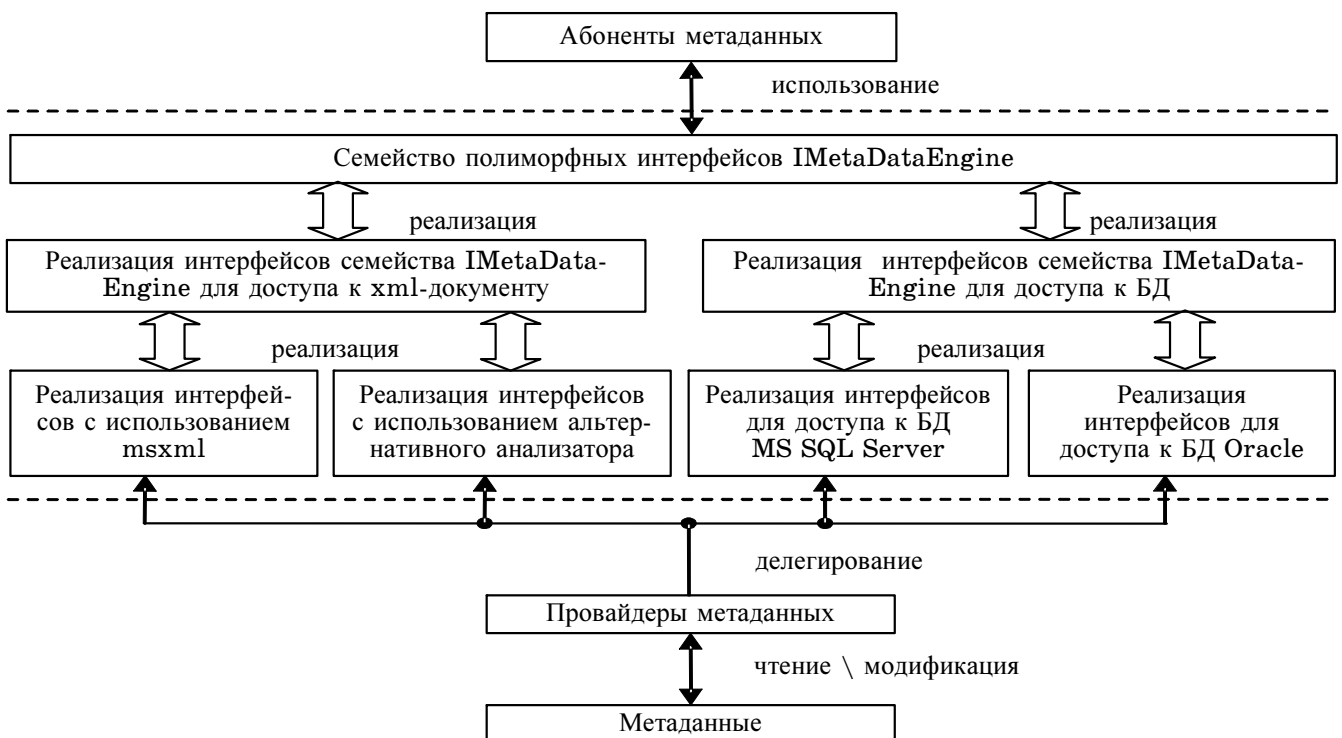
Понятие БМПМ	Понятие XML
Категория сущностей	Документ
Сущность	Описание тега (например, в определении типа документа DTD)
Экземпляр сущности	Тег
Атрибут	Описание атрибута (например, в определении типа документа DTD)
Значение атрибута	Значение атрибута
Отношение	Вложенность тегов



■ Рис. 2. Структурная схема доступа к метаданным посредством MDE

Таким образом, при использовании MDE разработчик абонентов метаданных оперирует не с понятиями языка представления МИ, такими как «xml-тег», «xml-атрибут», «таблица БД», «столбец таблицы БД» и т. п., а с понятиями БМПИМ.

Рассмотрим MDE подробнее. На рис. 3 показана структура гипотетического MDE, обеспечивающего доступ к метаданным, одна часть которых хранится в виде xml-документов, вторая — в БД MS SQL Server, а третья — в БД Oracle. До-



■ Рис. 3. Структура гипотетического процессора метаданных MDE

ступ к xml-документам осуществляется использованием анализатора MSXML (Microsoft XML Core Services) и некоторого альтернативного гипотетического анализатора.

К семейству `IMetaDataEngine` относятся следующие интерфейсы:

- интерфейсы для доступа к структуре МИ:
 - `IMDEntity` (сущность);
 - `IMDAttribute` (атрибут сущности);
 - `IMDRelationship` (отношение);
- интерфейсы для доступа к МИ:
 - `IMDInstance` (экземпляр сущности);
 - `IMDProperty` (значение атрибута сущности).

`IMDEntity` соответствует понятию «сущность» БМПП и представляет ее (сущности) структуру, а именно: наименование, сведения об атрибутах (`IMDAttribute`) и сведения о сущностях, находящихся в отношении с данной сущностью (`IMDRelationship`).

`IMDAttribute` соответствует понятию «атрибут сущности» БМПП и представляет о нем (атрибуте) следующие сведения: наименование, тип данных, ограничения целостности данных.

`IMDRelationship` соответствует понятию «отношение» БМПП и представляет о нем (отношении) следующие сведения: сведения об объектах отношения (`IMDEntity`), вид отношения (зависимость, ассоциация, агрегация, композиция или обобщение). Виды отношений заимствованы из унифицированного языка моделирования UML (Unified Modeling Language) [5].

`IMDInstance` соответствует понятию «экземпляр сущности» БМПП и инкапсулирует следующие данные: ссылку на сведения о структуре сущности (`IMDEntity`), значения атрибутов (`IMDProperty`), а также сведения об экземплярах сущностей, находящихся в отношении с экземпляром данной сущности (`IMDInstance`, `IMDRelationship`).

`IMDProperty` соответствует понятию «значение атрибута сущности» БМПП и инкапсулирует следующие данные: ссылку на сведения об атрибуте (`IMDAttribute`), данные атрибута.

Соответствующие диаграммы классов на языке UML представлены на рис. 4, а—в.

В качестве примера реализации MDE в листинге приведен исходный текст части интерфейса `IMDProperty` на языке C++, детально описанном в работе [6].

Листинг. Интерфейс `IMDProperty`

```
#define interface struct
//-----
interface IMetaDataEngine
{
...
}
//-----
interface IMDAttribute : public IMetaDataEngine
{
// Получить наименование атрибута
virtual string getName () = 0;
...
}
```

```
//-----
interface IMDProperty : public IMetaDataEngine
{
// Получить указатель на соответствующий атрибут
virtual IMDAttribute* getAttr() = 0;
// Чтение/установка данных
virtual Variant getData() = 0;
virtual void setData( const Variant vValue) = 0;
...
}
//-----
class TXMLProperty : public IMDProperty
{
virtual IMDAttribute* getAttr() {...}
...
}
//-----
class TMSXMLProperty : public TXMLProperty
{
private:
IXMLNode* FXMLNode; // Провайдер метаданных

public:
// Чтение/установка данных
virtual Variant getData() = 0;
virtual void setData( const Variant vValue) = 0;
};
//-----
class TTagXMLProperty : public TXMLProperty
{
private:
TAttr* FAttr; // Провайдер метаданных

public:
// Чтение/установка данных
virtual Variant getData() = 0;
virtual void setData( const Variant vValue) = 0;
};
//-----
class TDBProperty : public IMDProperty
{
virtual IMDAttribute* getAttr() {...}
...
}
//-----
class TMSDBProperty : public TDBProperty
{
private:
TMSDataSet* FDataSet; // Провайдер метаданных

public:
// Чтение/установка данных
virtual Variant getData() = 0;
virtual void setData( const Variant vValue) = 0;
};
//-----
class TOracleDBProperty : public TDBProperty
{
private:
TOracleDataSet* FDataSet; // Провайдер метаданных

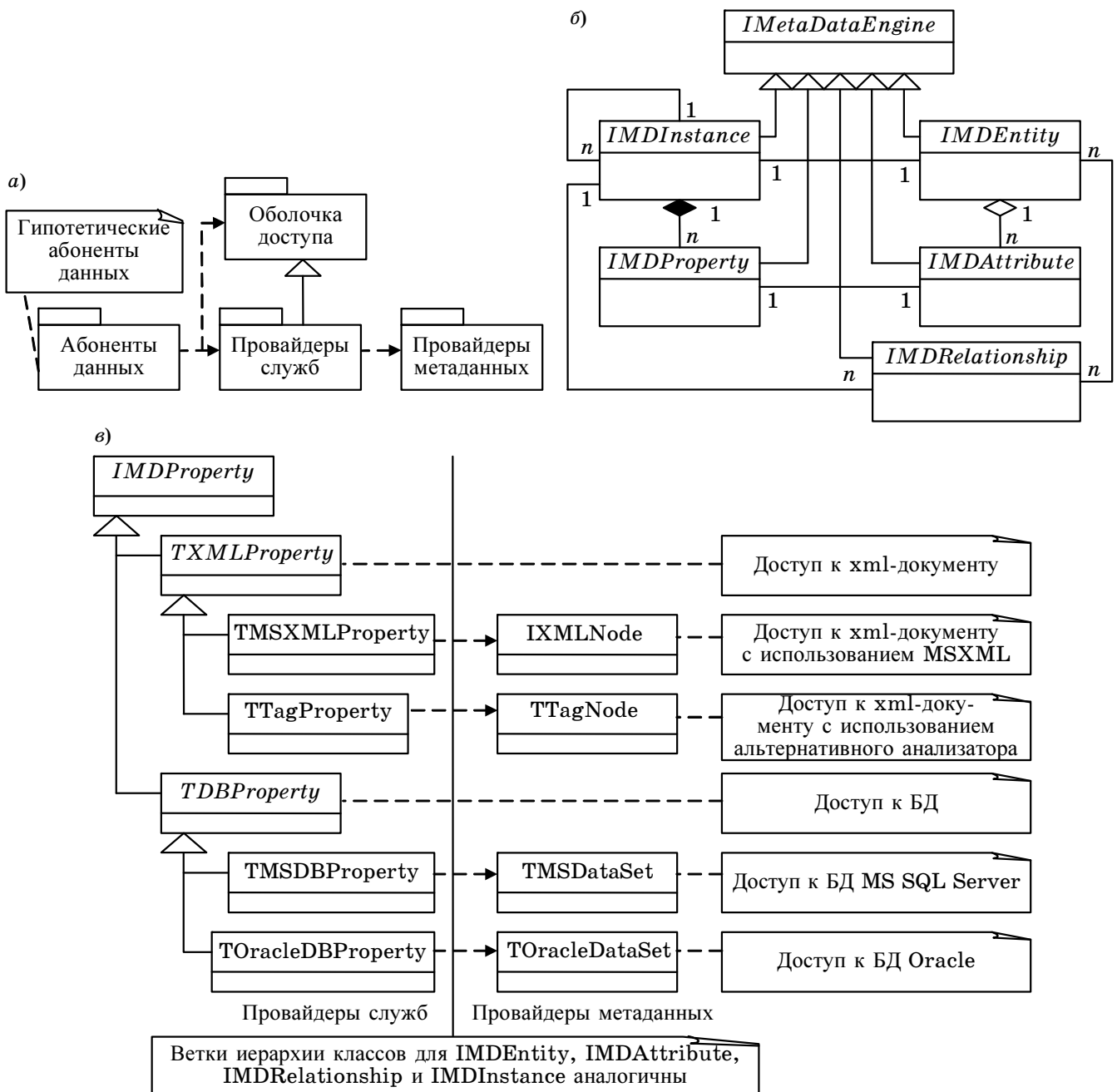
public:
// Чтение/установка данных
virtual Variant getData() = 0;
virtual void setData( const Variant vValue) = 0;
};
//-----
Variant TMSXMLProperty::getData()
{
return FXMLNode->NodeValue;
}
//-----
void TMSXMLProperty::setData( const Variant vValue )
```

```

{
  FXMLNode->NodeValue = vValue;
}
//-----
Variant TTagXMLProperty::getData()
{
  return FAttr->Value;
}
//-----
void TTagXMLProperty::setData( const Variant vValue )
{
  FAttr->Value = vValue;
}
    
```

```

//-----
Variant TMSDBProperty::getData()
{
  return FDataSet->FieldByName(
    getAttr()->getName()
  )->AsVariant;
}
//-----
void TMSDBProperty::setData( const Variant vValue )
{
  FDataSet->FieldByName(
    getAttr()->getName()
  )->AsVariant = vValue;
}
    
```



■ Рис. 4. Диаграмма классов гипотетического MDE: а — уровень 1; б — уровень 2, пакет «Оболочка доступа»; в — уровень 2, пакеты «Провайдеры служб» и «Провайдеры метаданных»

```

}
//-----
Variant TOracleDBProperty::getData()
{
    return FDataSet->FieldByName(
        getAttr()->getName()
    )->AsVariant;
}
//-----
void TOracleDBProperty::setData( const Variant vValue )
{
    FDataSet->FieldByName(
        getAttr()->getName()
    )->AsVariant = vValue;
}
//-----
void SetVals( IMDProperty* Prop )
{
    Prop->SetData(666);
}
//-----
void main()
{
    IMDProperty* Prop1 = new TMSXMLProperty;
    IMDProperty* Prop2 = new TTagXMLProperty;
    IMDProperty* Prop3 = new TMSDBProperty;
    IMDProperty* Prop4 = new TOracleDBProperty;
    SetVals( Prop1 );
    SetVals( Prop2 );
    SetVals( Prop3 );
    SetVals( Prop4 );
    delete Prop1;
    delete Prop2;
    delete Prop3;
    delete Prop4;
}

```

Пример демонстрирует независимость абонента данных (функция SetVals от конкретного представления метаданных).

К достоинствам предложенного подхода следует отнести гибкость и унифицированность, а к недостаткам — необходимость дополнительных накладных расходов на реализацию MDE.

Заключение

В настоящей работе рассмотрен процессор метаданных MDE как средство повышения гибкости реализации метауправления в информационных системах путем обеспечения независимости объектов информационных систем с метауправлением, осуществляющих формирование, перевод и интерпретацию компонентов метаданных, от средств представления метаданных.

Литература

1. Дейкстра Э. Ремесленник или ученый? // Компьютеры. 2002. № 41. <http://offline.computerra.ru/2002/466/21138/>
2. Йордон Э. Путь камикадзе. М.: Лори, 2003. 272 с.
3. Шерстюк Ю. М. Основы метауправления функциональностью в информационных системах. СПб.: СПИИРАН, 2000. 156 с.
4. Буч Г. Объектно-ориентированный анализ и проектирование. 2-е изд. СПб.: Бином, 2001. 560 с.
5. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. СПб.: Питер, 2004. 430 с.
6. Страуструп Б. Язык программирования C++. Специальное издание. СПб.: Бином, 2001. 1099 с.

ПАМЯТКА ДЛЯ АВТОРОВ

Поступающие в редакцию статьи проходят обязательное рецензирование.

При наличии положительной рецензии статья рассматривается редакционной коллегией. Принятая в печать статья направляется автору для согласования редакторских правок. После согласования автор представляет в редакцию окончательный вариант текста статьи.

Процедуры согласования текста статьи могут осуществляться как непосредственно в редакции, так и по e-mail (80x@mail.ru).

При отклонении статьи редакция представляет автору мотивированное заключение и рецензию, при необходимости доработать статью — рецензию. Рукописи не возвращаются.

Редакция журнала напоминает, что ответственность за достоверность и точность рекламных материалов несут рекламодатели.

УДК 621.391.1

ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ И ИХ МОДЕЛИ ДЛЯ ЗДРАВООХРАНЕНИЯ

А. Д. Сотников,

канд. техн. наук, профессор

Санкт-Петербургский государственный университет телекоммуникаций

им. проф. М. А. Бонч-Бруевича

Рассматриваются способы формального описания процессов информационного взаимодействия в сложных инфокоммуникационных системах и их классификация. Определены основные понятия и термины. Предлагается комплекс разноуровневых моделей для прикладных инфокоммуникационных систем телемедицинского назначения.

Введение

Развитие современной телекоммуникационной отрасли характеризуется несколькими ключевыми тенденциями, среди которых:

- Глубокое проникновение информационных и телекоммуникационных технологий внутрь прикладных областей и их превращение в ключевой функциональный элемент отраслевых систем. Информационная и телекоммуникационная экспансия в административно-управленческую и торгово-распределительную отрасли получила «фирменные» названия — «электронное правительство», «электронная коммерция», «телеработа». В социально ориентированных отраслях здравоохранения и образования развиваются междисциплинарные направления «телемедицины» и «телеобучения».

- Соединение в одно целое информационной (содержательно-смысловой) и телекоммуникационной составляющих порождает новую сущность — инфокоммуникационную систему (ИКС), обеспечивающую предоставление инфокоммуникационной услуги (ИКУ). Такое объединение изменяет способы анализа, проектирования и построения ИКС, организации и предоставления ИКУ. Телемедицина (ТМ) и телеобучение выступают наиболее типичными реализациями концепции «инфокоммуникационной услуги».

Телемедицина представляет междисциплинарный синтез здравоохранения, телекоммуникаций и информационных технологий (ИТ), которые традиционно реализуют специфические методы анализа и проектирования автономных систем и служб, создания и предоставления услуг на основе собственных организационно-функциональных принципов и технологий. Телемедицинские системы (ТМС) представляют объединение нескольких разнородных взаимозависимых систем,

что позволяет отнести их к классу сложных систем (СС) [1]. Эффективные методы распределения общесистемных требований к СС на отдельные подсистемы, оптимизация межсистемных взаимодействий остаются неразработанными. Эти проблемы выходят на первый план при проектировании социально ориентированных ИКС (здравоохранения, образования, государственного управления), характеризующихся жесткими ограничениями стоимостных характеристик.

Основная *тенденция* развития ТМ состоит в стабильном увеличении количества ТМС и сетей, наращивании объемов и расширении спектра услуг. Постоянный количественный прирост сопровождается высоким процентом учреждений здравоохранения, прекративших использование ТМС (10–15 % в США и до 25–30 % в РФ) [2, 3], что говорит о неудовлетворенности потребителей ТМС и об определенном несовпадении целей и результатов деятельности создателей ТМС с ожиданиями ТМ-сообщества. Наблюдается четкое структурно-функциональное деление на поставщиков и потребителей ТМ-услуг в составе ТМС. Целеполагание участников заметно различается, но не находит адекватного отражения в моделях ТМС и отрицательно сказывается на их функционировании.

Главная *проблема* состоит в отсутствии методов описания и анализа прикладной области, пригодных для ИТ и телекоммуникационной индустрии, что не позволяет корректно формулировать требования при проектировании ТМС. Целью работы является исследование принципов информационного взаимодействия в ИКС; классификация ИКС по ряду признаков; создание моделей сложных ИКС, ориентированных на количественные оценки параметров прикладной области здравоохранения.

Теоретические основы инфокоммуникаций

Для инфокоммуникаций ключевыми категориями и понятиями, вокруг которых концентрируются смысловые единицы, являются категории «информация», «информационное взаимодействие», «информационный процесс», «информационный объект».

Информация как фундаментальная общенаучная категория не имеет строгого, устраивающего всех специалистов, определения. Информацией называют и «... содержание сигнала, сообщения», и «... бесконечный законо-процесс триединства энергии, движения и массы с различными плотностями кодовых структур бесконечно-беспредельной Вселенной» [4]. Отсутствие единства в определении говорит о недостаточной разработанности категории и дает основания для новых определений и трактовок.

Информационное взаимодействие. Категория взаимодействия отражает действия различных объектов друг на друга, их взаимную обусловленность, изменение состояния.

Информационный процесс как реализацию определенного информационного взаимодействия ряд авторов определяет как «... неэнергетическое взаимодействие объектов, при котором осуществляется передача (генерация и освоение) идеальных категорий (смыслов, значений, образов, эмоций)» или «... взаимодействие объектов, приводящее к изменению знаний хотя бы одного из них» [4].

Основные понятия и определения

Информационная система имеет дело с информационными объектами — представлениями $\langle\langle A \rangle\rangle$, $\langle\langle B \rangle\rangle$, ... сущностей $\{A, B, \dots\}$ [5].

Информация передана, когда изменился сигнал, переносящий образ (представление) из многообразия тезауруса системы-источника A в многообразии тезауруса системы-получателя B :

$$\langle A \rangle^{\xi_A} \xrightarrow{\text{Сигнал}} \langle\langle A \rangle\rangle^{\xi_A} \xrightarrow{\xi_B}$$

Информация воспринята, когда возник новый образ источника в многообразии тезауруса получателя:

$$\langle A \rangle^{\xi_A} \xrightarrow{Q_1} \langle C \rangle^{\xi_C} \xrightarrow{Q_2} \langle\langle A \rangle\rangle^{\xi_A} \xrightarrow{\xi_C} \langle\langle A \rangle\rangle^{\xi_C} \xrightarrow{\xi_B}$$

Информационное воздействие — воздействие «источника» A на состояние «получателя» B , выражающееся в изменении представления $\langle B \rangle$, различного в многообразии элементов тезауруса «источника» $\langle B \rangle^{\xi_A}$. Так как тезаурусы «источника» ξ_A и «получателя» ξ_B различны, то исходный образ в собственном тезаурусе $\langle B \rangle^{\xi_B}$ и образ $\langle B \rangle^{\xi_A}$ в тезаурусе «источника», выступающего наблюдателем, также различаются. Это один из источников ошибок, или, точнее, погрешности представления объекта в тезаурусе «наблюдателя».

Информационный обмен — передача и прием сигналов, приводящих к взаимному изменению образов $\langle A \rangle^{\xi_B}$ и $\langle B \rangle^{\xi_A}$ участников обмена. Это может быть связано с изменением (расширением) тезаурусов ξ_A и ξ_B участников.

Информационное взаимодействие — взаимное изменение образов собственных систем $\langle A \rangle^{\xi_A}$ и $\langle B \rangle^{\xi_B}$, приводящее к изменению образов $\langle A \rangle^{\xi_B}$ и $\langle B \rangle^{\xi_A}$ у других участников.

Информационная система (ИС) — это система, содержащая информацию и обеспечивающая ею пользователя. *Необходимое условие:* неотъемлемыми компонентами ИС являются пользователь и потенциальная информация. *Достаточное условие:* пользователь и потенциальная информация образуют ИС. ИС составляют элементы, которые являются информационными представлениями $\langle A \rangle$ реальных (материальных и нематериальных) сущностей A и имеют «информационную значимость».

Информационная значимость — это свойство представления¹ сущности, которое присутствует, если известен способ описания, содержащий множество базовых «смыслов», имманентно принадлежащих сущности. Формализованное множество «смыслов» является тезаурусом ξ_A . Экземпляр сущности обладает множеством различных состояний, которые воспринимаются наблюдателем как множество образов объекта, каждый из которых имеет свой «смысл». Множество состояний определяет *потенциальную информацию* объекта. При возникновении у наблюдателя образа объекта (в результате восприятия и распознавания) происходит *актуализация потенциальной информации* на основе информационного представления объекта. Потенциальное наличие информации в системе A определяется множеством различных состояний системы и многообразием тезауруса системы ξ_A .

Восприятие переданной информации — возникновение у получателя R нового образа системы A в многообразии тезауруса получателя $\langle\langle A \rangle\rangle^{\xi_A} \xrightarrow{\xi_R}$.

Пользователь U — субъект, объект или процесс, способный воспринимать представления $\langle S \rangle^{\xi_S}$ и обладающий собственным тезаурусом пользователя ξ_U .

Потенциальная информация — множество различных состояний сущности S во множестве представлений $\langle S \rangle^{\xi_S}$ на основе тезауруса ξ_S , обладающих информационной значимостью.

Актуальная информация — это представление (образ) $\langle\langle S \rangle\rangle^{\xi_S} \xrightarrow{\xi_U}$ в тезаурусе ξ_U пользователя U .

Информационная система S : $\{S, \xi_S, \langle S \rangle^{\xi_S}, U, \xi_U, \langle\langle S \rangle\rangle^{\xi_S} \xrightarrow{\xi_U}\}$ — совокупность, включающая сущности системы, *пользователей*, тезаурус системы

¹ «Информационную значимость» не следует путать с «ценностью» и «полезностью» — эти свойства связаны с потребителем и возможностью удовлетворения его информационных потребностей.

и тезаурусы пользователей, множество информационных представлений системы (потенциальная информация) и множество образов системы (представления в тезаурусах пользователей).

Коммуникационная система $C: \{C, \xi_C, \langle C \rangle^{\xi_C}, Q_{A^{\xi_C}C}, Q_{C^{\xi_C}B}\}$ — совокупность последовательных преобразований

$$\xrightarrow{Q_{12}^{\xi_A \xi_C}}, \xrightarrow{Q_{23}^{\xi_C \xi_B}},$$

обеспечивающих представление множества образов источника $\langle A \rangle$ в множество образов потребителя $\langle B \rangle$ через множество состояний (образов) $\langle C \rangle^{\xi_C}$ сигнала C при требуемой точности.

Телекоммуникационная система (ТС) $T: \{C, \xi_C, \langle C \rangle^{\xi_C}, Q_{A^{\xi_C}C}, Q_{C^{\xi_C}B}\}$ — совокупность последовательных преобразований $Q_{A^{\xi_C}C}, Q_{C^{\xi_C}B}$ сигналов и их представлений вне зависимости от пространственного размещения «источника» A и «потребителя» B ($A \in D, B \in D, D \neq A \cap B$).

Инфокоммуникационная система $F: \{S, \xi_S, \langle S \rangle^{\xi_S}, U, \xi_U, \langle \langle S \rangle^{\xi_S} \rangle^{\xi_U}, C, \xi_C, \langle C \rangle^{\xi_C}, Q_{A^{\xi_C}C}, Q_{C^{\xi_C}B}\}$ — совокупность, включающая сущности информационной системы S и телекоммуникационной системы T .

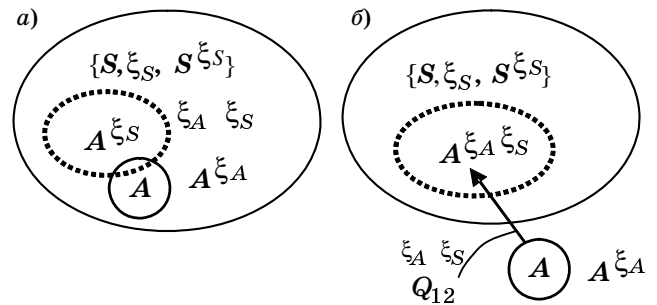
Классификация ИКС

В рамках данных определений возможно классифицировать ИКС по ряду признаков, характеризующих различные стороны взаимоотношения системы с пользователем и информацией, которые важны с точки зрения анализа и последующего проектирования ИКС [6, 7]. В данном контексте пространственная удаленность участников не имеет значения, и понятия ИКС и ИС совпадают.

Размещение информации. Информационные объекты (представления сущностей) — всегда составляющая часть ИКС. Сами сущности (носители потенциальной информации) не всегда являются составляющими ИКС. Наличие или отсутствие сущностей в составе ИКС — первый классификационный признак, в зависимости от которого по-разному реализуется владение сущностями и/или информационными объектами, т. е. потенциальной информацией, и ее адекватное представление, что имеет важные последствия.

1. Сущности, обладающие потенциальной информацией, находятся «внутри» ИС, и их информационные представления располагаются там же. Включение объекта в систему (рис. 1, а) предполагает совпадение подмножества тезауруса объекта с множеством тезауруса системы $\xi_A \in \xi_S$ и, как следствие, — совпадение представлений $\langle A \rangle^{\xi_A} \equiv \langle A \rangle^{\xi_S}$.

2. Сущности находятся «вне системы» (рис. 1, б). В этом случае тезаурусы системы и объекта связаны соотношением $\xi_S \in \xi_A$, возможно только подобие образов $\langle A \rangle^{\xi_A} \approx \langle A \rangle^{\xi_S}$, определяемое близостью тезаурусов ξ_S, ξ_A , т. е. информация представляется системой с искажениями.



■ Рис. 1. Информация «в системе» (а) и «вне системы» (б): A — объект; S — информационная система

Отношение пользователя к ИС. Это второй признак, по которому классифицируются ИКС. Пользователь может быть «в системе» или «вне системы». Термины «в» и «вне» отражают возможности взаимодействия тезаурусов системы и пользователя, а не пространственные или топологические свойства.

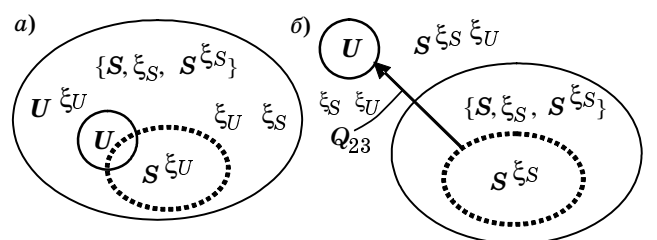
1. Пользователь «в системе» (рис. 2, а) когда:

а) тезаурусы системы и пользователя полностью совпадают: $\xi_S \equiv \xi_U$. Это вырожденный случай, когда пользователь отождествляется с системой и не может рассматриваться как «получатель» информации, поскольку обладает всей полнотой образов, возникающих из многообразия тезауруса системы, а ИС не может служить источником информации, т. е. нарушается определение и утрачивается утилитарность ИС;

б) тезаурус пользователя может быть произвольно изменен (расширен) за счет элементов тезауруса системы по инициативе пользователя и до определенной степени «согласован» с тезаурусом системы: $\xi_S \approx \xi_U$;

в) тезаурус системы может быть произвольно изменен пользователем.

В тех случаях, когда пользователь «в системе», отсутствует операция взаимного преобразования элементов тезаурусов $Q^{\xi_S \xi_U}$, а есть отображение одного подмножества на другое $F: S \rightarrow U$, т. е. возможно адекватное восприятие образа. Степень адекватности определяется степенью совпадения тезаурусов, т. е. степенью общности множеств ξ_S, ξ_U .



■ Рис. 2. Пользователь U «в системе» (а) и «вне системы» (б)

2. Пользователь «вне системы» (рис. 2, б) когда: а) тезаурус пользователя ξ_U и системы ξ_S автономны ($\xi_S \leftarrow \times \rightarrow \xi_U$) и не могут быть согласованы до произвольной степени по инициативе пользователя, а только «по воле системы» в результате предоставления пользователю информации.

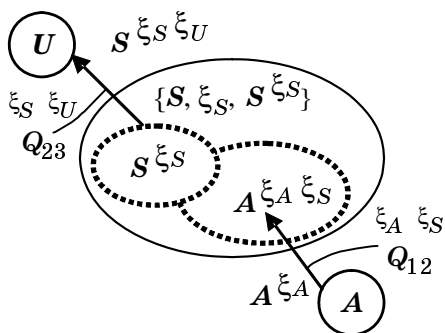
Когда пользователь «вне системы», становится значима операция преобразования тезаурусов ($Q^{\xi_U \xi_S} = \xi_U \cup \xi_S$), где $S' \in S$ или реже $Q^{\xi_S \xi_U}$, что соответствует изменению состава (расширению) тезауруса, обеспечивает расширение пространства образов и большую точность восприятия образа объекта $\langle A \rangle^{\xi_A}$ в тезаурусе пользователя $\langle \langle A \rangle^{\xi_A} \rangle^{\xi_U}$. Для классификации ИС принципиально соотношение тезаурусов пользователя, объектов и системы (ξ_U, ξ_A, ξ_S), потому что образ создается в многообразии тезауруса, т. е. образ произведен от тезауруса: $A \Rightarrow \xi_A \Rightarrow \langle A \rangle^{\xi_A}$. Наиболее сложный случай, когда пользователь и сущности находятся «вне системы» (рис. 3). Именно здесь возникает двусторонний обмен, который в формальной нотации описывается следующим образом. Пусть сущность, ИС и пользователь с тезаурусами и представлениями соответственно: $\{A, \xi_A, \langle A \rangle^{\xi_A}\}, \{S, \xi_S, \langle S \rangle^{\xi_S}\}, \{U, \xi_U, \langle U \rangle^{\xi_U}\}$, тогда возникновение образа сущности в тезаурусе пользователя соответствует одностороннему обмену между сущностью, ИС и пользователем, определяемому записью

$$\langle A \rangle^{\xi_A} \xrightarrow{Q^{\xi_S \xi_A}} \langle \langle A \rangle^{\xi_A} \rangle^{\xi_S} \xrightarrow{Q^{\xi_S \xi_U}} \langle \langle \langle A \rangle^{\xi_A} \rangle^{\xi_S} \rangle^{\xi_U} \xrightarrow{Q^{\xi_S \xi_U}} \langle \langle \langle A \rangle^{\xi_A} \rangle^{\xi_S} \rangle^{\xi_U} \xrightarrow{Q^{\xi_S \xi_U}} \langle \langle \langle A \rangle^{\xi_A} \rangle^{\xi_S} \rangle^{\xi_U} \xrightarrow{Q^{\xi_S \xi_U}} \dots$$

где выражение в скобках описывает «восстановление» образа $\langle A \rangle^{\xi_A}$ в образе $\langle \langle A \rangle^{\xi_A} \rangle^{\xi_U}$, созданном в многообразии пользовательского тезауруса.

На основании первых двух признаков классифицируются наиболее важные характеристики структурной и функциональной организации ИКС, выражаемые в ее типе:

тип А — один пользователь, информация «внутри», пользователь — «вне системы»;



■ Рис. 3. Пользователь и данные «вне системы»

тип А1 — много пользователей, информация «внутри», пользователи — «вне ИКС»;

тип В — один или ограниченное число пользователей, информация и пользователь «внутри ИКС»;

тип В1 — много пользователей, информация и пользователи «внутри системы»;

тип С — два пользователя, информация у пользователей, пользователи «вне ИКС»;

тип D — пользователь «внутри», информация «вне системы»;

тип Е — два (много) пользователей и информация «внутри ИКС»;

тип F — комбинация систем типов А — С. В зависимости от совокупности типов составляющих систем выделяются несколько стандартных подтипов (F1 — F4).

Временное взаимодействие. По этому признаку ИКС разделяются на четыре категории:

— непрерывные — взаимодействие пользователя и ИКС носит постоянный, «непрерывный» во времени характер, рамки которого не определены заранее;

— сеансовые — взаимодействие пользователя и ИКС носит эпизодический характер с ограниченной, как правило, известной или уверенно прогнозируемой длительностью;

— транзакционные — взаимодействие пользователя и ИКС носит кратковременный, повторяющийся характер в не планируемые заранее моменты времени;

— комбинированные — в большей или меньшей степени одновременно присутствуют несколько типов взаимодействия.

Участие пользователя. По характеру участия пользователя в функционировании ИКС выделены следующие категории:

— распределительные — пользователь преимущественно получает данные, не участвуя в формировании содержательного потока данных, и не влияет на алгоритмы функционирования системы, т. е. не участвует в управлении ИКС;

— диалоговые — пользователь участвует в формировании содержательных информационных потоков, но не влияет активным образом на алгоритмы функционирования ИКС;

— интерактивные — пользователь активно влияет на функционирование ИКС и участвует в формировании информационных потоков;

— комбинированные — встречаются в разных комбинациях две и более вышеназванные системы.

Количество пользователей. Классификация по признаку количества пользователей выделяет ИКС с одним пользователем, ограниченным числом пользователей (ИКС коллективного пользования) и ИКС массового обслуживания.

Прикладные ИКС

Особое место занимает класс ИКС, в которых отсутствует возможность влиять на сущности, являющиеся генераторами представлений, и отно-

сящиеся не к информационной отрасли, а к иным областям практической деятельности — здравоохранению, образованию, военному делу, управлению и т. п. Выделим этот класс ИКС и обозначим его как «прикладные ИКС». Он характеризуется следующими свойствами:

1) сущности (реальные объекты) принципиально находятся «вне системы»;

2) тезаурусы системы и сущности связаны соотношением $\xi_S \in \xi_A$;

3) отсутствует возможность изменения тезауруса сущности со стороны ИС, т. е. недопустимость операции $|Q^{\xi_S \xi_A}|$;

4) информация представляется ИКС с искажениями, так как возможно только подобие информационных представлений $\langle A \rangle^{\xi_A} \approx \langle A \rangle^{\xi_S}$, определяемое близостью пар тезаурусов $\xi_S \xi_A$ и $\xi_S \xi_U$.

Внешним признаком класса «прикладных ИКС» является строгое требование размещения информации и пользователя «вне системы» (тип С) или не строгое (типы А, А1, С, D, F), когда информация находится у пользователя. Прикладные ИКС, обладая в большинстве случаев низкой степенью формализации, представляют наибольшие трудности при системном проектировании и нуждаются в разработке механизмов их описания.

Прикладные ИКС в здравоохранении

Телемедицинские системы представляют типичный пример ИКС со слабо структурированными данными, неявными алгоритмами и с разнообразным характером взаимодействия участников. Проектирование ТМ-ИКС при отсутствии методологических подходов опирается на технологические новации и эмпирические знания, не гарантирующие эффективного результата. Разработка ТМ-ИКС идет «от технологии» и ее возможностей, что позволяет манипулировать характеристиками и параметрами телекоммуникационных систем и технологий с фактической целью навязать заказчику варианты реализации ТМ-ИКС, выгодные поставщику. Ситуация усугубляется тем, что потребители (медицинские специалисты), как правило, не могут сформулировать технические требования к ТМС. В среднесрочной перспективе это ведет к неудовлетворенности потребителя и дискредитации систем и технологий, заложенных в ее основу.

Модели инфокоммуникационных систем

В здравоохранении, при множестве решаемых задач, сильной внутрипрофессиональной дифференциации, калейдоскопическом разнообразии школ, методик и технологий, чрезвычайно трудно, если возможно, использовать единую модель. Предлагается совокупность моделей, последовательно приближающих к количественному описанию ТМ-ИКС. Эти модели последовательно:

— описывают ИКС в целом, определяя ее границы;

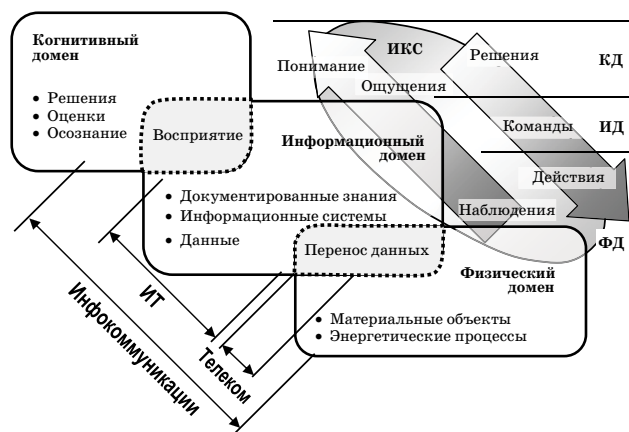
— структурируют предметную область, выделяя группы медицинских задач и применений, обладающих близкими, маловариативными наборами характеристик;

— обеспечивают описание характеристик прикладной ИКС в терминах, пригодных для проектирования или количественной оценки телекоммуникационных систем (услуг) и оптимизации решений.

Доменная модель инфокоммуникаций

Наиболее общей и абстрактной является доменная модель (ДМ) (рис. 4), позволяющая выявить компоненты ИКС, обозначить межкомпонентные интерфейсы и рассмотреть процессы информационного взаимодействия. ДМ выделяет три домена, где концентрируются три относительно самостоятельных, хотя и тесно связанных, вида деятельности. В *физическом домене ФД* протекают преимущественно энергетические процессы и взаимодействуют материальные объекты. Анализ ситуаций и интеллектуальная деятельность, продуктом которой являются оценки и принятие решений, — это продукт ментальной и психической активности, протекающей в *когнитивном домене КД*. *Информационный домен ИД* — это область, в которой циркулируют данные, используемые в КД и представляющие объекты, явления и процессы ФД.

Согласно ДМ, область телекоммуникаций располагается на границе ФД и ИД. Восприятие физической реальности, представленной сущностями ИД, связано с информационным взаимодействием и протекает на границе ИД и КД. Таким образом, инфокоммуникации охватывают все три домена, хотя сегодняшнее состояние характеризуется неглубоким проникновением в КД в силу неразработанности категорий сознания и других сущностей и отношений КД, связанных с генерацией и потреблением содержательно-смыслового ядра информации.



■ Рис. 4. Доменная модель инфокоммуникаций

В общем виде элементарное взаимодействие двух информационных систем в ИД состоит в односторонней передаче/приеме представления объекта ФД, описываемых выражением

$$\langle \langle A_n \rangle^{\xi_{A_n}} \rangle^{\xi_{C^m}} \xrightarrow{Q_{22}^{\xi_{C^m} C^k}} \langle \langle A_n \rangle^{\xi_{A_n}} \rangle^{\xi_{C^k}}.$$

Информационное взаимодействие осуществляется посредством обмена «сообщениями», которые являются подмножествами множества информационных представлений $\langle A_1 \rangle^{\xi_{A_n}}$ объекта A_n , передаваемых между ИС C_m и C_K . Тогда взаимодействие нескольких ИС внутри ИД описывается как

$$\begin{aligned} & \left\langle \left\langle \langle A_n \rangle^{\xi_{A_n}} \right\rangle_{n=1, \dots, N} \right\rangle^{\xi_{C^m}} \xleftarrow{Q_{22}^{\xi_{C^k} \xi_{C^m}}} \\ & \xrightarrow{Q_{22}^{\xi_{C^m} \xi_{C^k}}} \left\langle \left\langle \langle A_n \rangle^{\xi_{A_n}} \right\rangle_{n=1, \dots, N} \right\rangle^{\xi_{C^k}} \Big|_{k=1, \dots, K} \Big|_{m=1, \dots, M}. \end{aligned} \quad (1)$$

Информационный процесс есть совокупность элементарных информационных взаимодействий, происходящих в ИД. Выражение (1) представляет ДМ — описание информационного взаимодействия, затрагивающего сущности трех доменов.

Пусть $\{A_1, \dots, A_6\} \in A$ — множество объектов ФД, $\{C_i^j\} \in C$ — множество объектов ИД, где $i = \{1, \dots, 9\}$, $j = \{1, \dots, 5\}$, при этом подмножества представлений $\{C_m^i\}$ относятся к m автономным ИС, действующим в ИД. Тогда для объектов A_n и потребителей B_k в общем виде

$$\begin{aligned} & \langle A_n \rangle^{\xi_{A_n}} \xrightarrow{Q_{12}^{\xi_{A_n} \xi_{C^m}}} \\ & \rightarrow \langle \langle A_n \rangle^{\xi_{A_n}} \rangle^{\xi_{C_n^m}} \xrightarrow{Q_{23}^{\xi_{C_n^m} \xi_{B_k}}} \langle \langle C_n^m \rangle^{\xi_{C_n^m}} \rangle^{\xi_{B_k}}. \end{aligned} \quad (2)$$

Выражение (2) представляет модель информационного взаимодействия двух субъектов КД, выраженную в терминах представлений систем ИД и объектов ФД. Представление объекта A_1 в тезаурусе потребителя B_1 служит основой для принятия решения $\langle R_1 \rangle^{\xi_{B_1}}$ субъекта (B_1) КД:

$$\left\langle \left\langle \langle A_1 \rangle^{\xi_{A_1}} \right\rangle^{\xi_{C_1^1}} \right\rangle^{B_1} \xrightarrow{\Theta_{B_1}} \langle R_1 \rangle^{\xi_{B_1}},$$

где Θ_{B_1} — оператор формирования решения.

Вся последовательность формирования образа объекта A_1 в ИД, восприятия его субъектом B_1 в КД, принятия решения и передаче его через субъекту B_4 в КД описывается выражением

$$\begin{aligned} & \left\langle \left\langle \langle A_1 \rangle^{\xi_{A_1}} \right\rangle^{\xi_{C_1^1}} \right\rangle^{B_1} \xrightarrow{\Theta_{B_1}} \langle R_1 \rangle^{\xi_{B_1}} \xrightarrow{Q_{32}^{\xi_{B_1} \xi_{C_7^4}}} \\ & \rightarrow \langle \langle R_1 \rangle^{\xi_{B_1}} \rangle^{\xi_{C_7^4}} \xrightarrow{Q_{23}^{\xi_{C_7^4} \xi_{B_4}}} \langle \langle C_7^4 \rangle^{\xi_{C_7^4}} \rangle^{B_4}, \end{aligned}$$

представляющим модель информационного взаимодействия субъектов КД в терминах представлений ИД, объектов ФД и решений субъектов КД.

Модели ИКС в здравоохранении

Характерной чертой медицины является низкий уровень формализации² используемых данных, процедур их получения и анализа по сравнению с техническими и бизнес-приложениями.

Трехмерная модель. Для обоснованного построения ТМС выделены характеристики, описывающие различные стороны применения ИКС, а именно:

- *круг решаемых медицинских задач* — профилактика и предупреждение, диагностика и лечение, мониторинг и сопровождение хронических больных, экстренная помощь и медицина катастроф, медицинское образование, администрирование и управление и др.;

- *медицинские специализации и области применения*, как сложившаяся внутрипрофессиональная классификация, отражают методологические подходы, связанные с информационными моделями объекта деятельности (пациента);

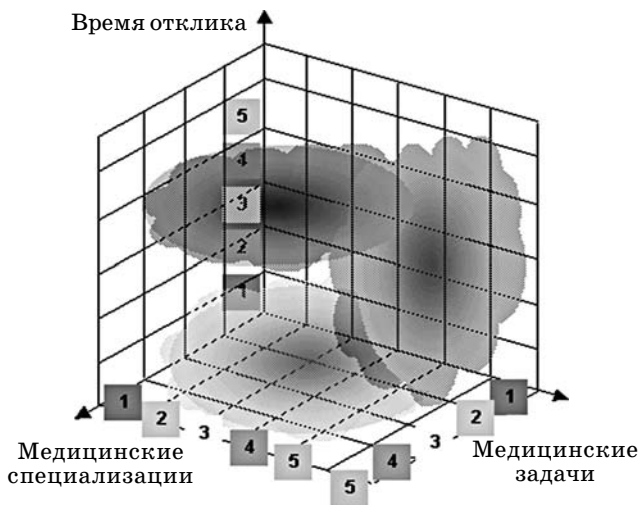
- *временные характеристики системы* (время отклика ТМС на запрос) определяют характер ее взаимодействия с пользователем и задают требуемые параметры ТС.

Названные характеристики положены в основу трехмерной модели ТМ-ИКС (рис. 5), которая классифицирует область применения и позволяет избежать неоправданного расширения на смежные, похожие внешне, но отличающиеся характеристиками использования ИКС. Результатом анализа трехмерной модели является дифференциация областей применения ИКС и отнесение ТМ-ИКС к определенному классу ИКС с выявлением главенствующего типа взаимодействия между участниками.

Двухкомпонентная модель, детализируя параметры каждой из названных характеристик, обеспечивает переход к количественным оценкам.

Комбинация прикладной медицинской «задачи» и «специализации» образует область с мало-

² Стандартизация в здравоохранении заметно улучшает ситуацию, но не меняет характера проблемы. Медицинские стандарты и классификаторы (DICOM, МКБ-10, HL-7 и т. п.), формализуя структурные характеристики данных (представлений сущности), не затрагивают источник всех данных — пациента.

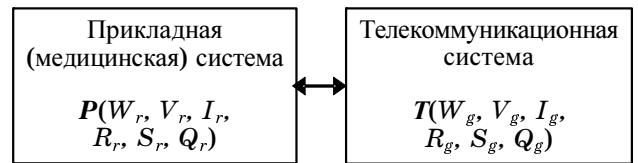


■ Рис. 5. Трехмерная модель ТМ-ИКС

вариативными значениями параметров (рис. 6), называемую «областью применения». «Применение» P описывается набором целочисленных параметров $P(W_r, V_r, I_r, R_r, S_r, Q_r)$, характеризующих «требования», где W_r — объем данных в сеансе; V_r — требуемая скорость передачи; I_r — вид требуемой службы; R_r — требуемое время отклика ИКС на запрос; S_r — симметрия/асимметрия службы; Q_r — необходимость QoS (рис. 7).

Аналогично ТС описывается набором параметров, характеризующих ее свойства обеспечивать требуемые характеристики, и параметров, учитывающих внутренние свойства системы $T(W_g, V_g, I_g, R_g, S_g, Q_g)$, или более широким, включающим стоимостные характеристики (разовые и эксплуатационные затраты для прикладной системы (ПС) и ТС): $Z_{rr}, Z_{er}, Z_{rg}, Z_{eg}$.

Последовательная детализация и количественная оценка приводят к двум ограниченным множествам наборов $P_i \in P$ и $T_j \in T, i \in [1, \dots, N], j \in [1, \dots, K], N \leq K$ параметров (требуемых ПС и гарантируемых ТС), достаточно точно описыва-



■ Рис. 7. Двухкомпонентная модель ИКС

ющих «прикладную» и «телекоммуникационную» компоненты конкретной ТМ-ИКС.

Двухкомпонентная модель (см. рис. 7) позволяет, сопоставляя P_i и T_j , определить параметры, совпадающие или наиболее близкие требуемым. Формулируя различные критерии близости, можно решать следующие задачи проектирования ИКС:

— прямая задача — поиск варианта, который при допустимых разовых затратах Z_{rg} и удовлетворении набора технологических требований ($W_r, V_r, I_r, R_r, S_r, Q_r$) обеспечивает $\min(Z_{eg})$ — минимальную стоимость эксплуатации;

— обратная задача — поиск такого значения Z_{eg} , при котором удовлетворяются все требования ($W_r, V_r, I_r, R_r, S_r, Q_r$) при $\min(Z_{rg})$ или заданном значении $Z < Z_{rg}$;

— задача оценки границ вариации тарифообразующих параметров (Z_{rg} и Z_{eg}) модели, при которых услуга остается выгодной оператору телекоммуникаций. Общая постановка задачи: «Найти такие реализации ИКС (совокупности ПС_{*i*} и ТС_{*j*}), которые удовлетворяют заданному критерию».

Основные задачи, связанные с организацией и предоставлением ИКУ, следующие.

1. На этапе проектирования ТМС — определение набора реализаций подсистем, который обеспечит заданный уровень функциональности $F(V_i) \geq \text{const}$ при минимуме затрат $\min(F(C_i))$.

2. Определение набора реализаций подсистем, обеспечивающих максимум функциональности $\max(F(V_i))$ при фиксированном уровне инвестиций ($F(C_i) \leq \text{const}$).



■ Рис. 6. Применения ТМ-ИКС

3. Определение для выбранной реализации ИКС значений стоимостных параметров $F(C_i)$, при которых удовлетворяются все требования качества ($F(V_i) \geq \text{const}$).

4. На этапе эксплуатации ИКС — определение оптимальной стратегии обновления/поддержания/расширения системы, т. е. оценка возможного диапазона изменения параметров V_i при заданных ограничениях C_i .

Двухкомпонентная модель позволяет перейти к количественным оценкам и сформулировать технические требования к ИКС. Итоговыми измерителями для оценки медицинских применений выступают объемные значения медицинских данных и время отклика системы на запрос услуги определенного вида, а также свойства интерактивности услуги.

Заключение

Основной недостаток традиционных подходов состоит в исключении пользователя и его прикладных процессов из состава ИКС, что затрудняет исследование процессов генерации и утилизации информации в ИС. Формальный анализ информационного взаимодействия компонентов ИКС, их базовых свойств и характеристик позволяет выделить различные типы ИКС. Предложенные принципы классификации основаны на признаках, характеризующих взаимоотношения ИКС с пользователем и информацией. Определен класс прикладных ИКС, характерных для здравоохранения и образования.

Границы ИКС, их свойства и ряд характеристик адекватно описываются комплексом из трех разноуровневых моделей, включающих ДМ инфо-

коммуникаций, трехмерную и двухуровневую модели ТМ-ИКС.

Рассмотренный подход позволяет перейти от субъективных оценок к формулировке задач количественного анализа и синтеза слабо формализованных прикладных ИКС телемедицины.

Литература

1. Luman R. R. Integrating Cost and Performance Models to Determine Requirements Allocation for Complex Systems // Technical Digest / Johns Hopkins University APL. 2000. Vol. 21. N 3. P. 408–425.
2. Основные показатели работы лечебно-профилактических учреждений федерального подчинения: статистические материалы: <http://www.mednet.com/stat/stat00/index.htm>
3. Сотников А. Д. Структурно-функциональная организация услуг телемедицины в прикладных инфокоммуникационных системах. СПб.: Судостроение, 2007. 200 с.
4. Полонников Р. И. Основные концепции общей теории информации. СПб.: Наука, 2006. 203 с.
5. Сотников А. Д. Принципы анализа прикладной области в инфокоммуникационных системах здравоохранения // Тр. учеб. заведений связи. 2004. № 171. С. 174–183.
6. Сотников А. Д. Классификация и модели прикладных инфокоммуникационных систем // Тр. учеб. заведений связи. 2003. № 169. С. 149–162.
7. Сотников А. Д. Оптимизация инфокоммуникационных систем в здравоохранении // Тр. учеб. заведений связи. 2003. № 169. С. 163–174.

УДК 681.883.022: 681.883.65

ФРАКТАЛЬНЫЙ ОБНАРУЖИТЕЛЬ ЭХО-СИГНАЛА

Н. Н. Семенов,

соискатель

Санкт-Петербургский морской технический университет

Предложен новый алгоритм для обнаружения морских подводных целей, основанный на оценке меры случайности сигнала на входе гидролокатора. В качестве меры предлагается фрактальная размерность.

Введение

Классическая задача обнаружения гидролокационных сигналов на фоне случайных помех рассматривается с помощью методов статистической теории решений, которая является разделом математической статистики. Из-за статистической природы анализируемого случайного входного процесса полученный результат имеет ту или иную степень достоверности.

Существующие обнаружители эхо-сигналов сравнивают уровень выходного отклика гидролокационного приемника с заданным порогом для двух возможных гипотез: когда на входе присутствует только помеха или совокупность эхо-сигнала и помехи. С превышением заданного порога принимается решение об обнаружении сигнала. Выбранный уровень порога определяет вероятность правильного обнаружения сигнала при фиксированной вероятности ложной тревоги [1, 2]. Недостатком такого подхода является наличие минимального отношения сигнал/помеха (ОСП), при котором возможно подобное обнаружение. В данной статье рассматривается один из алгоритмов обнаружения эхо-сигнала на фоне помех не по энергетическому критерию, а по косвенному признаку — фрактальной размерности.

Сравнение различных типов существующих обнаружителей

Условимся, что объект локации имеет бликовую структуру [1]. Таким образом, эхо-сигнал от объекта локации представляется состоящим из 2–3 ярких бликовых точек, 20–50 точек с яркостью в 10 раз меньше и 200–600 — с яркостью в 100 раз меньше. Шум имеет равномерную спектральную плотность мощности в полосе приема [1, 3]. Несущая частота реального гидролокатора 20 кГц, полоса 2 кГц, длительность посылки 100 мс. В качестве сигнала был использован фазоманипулированный сигнал, кодированный M-последовательностью с длиной кода 127. Время окна анали-

за 300 мс. ОСП вычислялось через отношение полной энергии сигнала к энергии помехи за время анализа в заданной полосе частот.

1. Некогерентный обнаружитель.

На входе обнаружителя присутствует сумма эхо-сигнала и помехи:

$$s(t) = \sum_N z_i(t) + e(t),$$

где $z_i(t)$ — эхо-сигнал от i -й бликовой точки; $e(t)$ — помеха.

Некогерентный обнаружитель сравнивает амплитуду суммы сигнала и шума с амплитудой шума до и после прихода сигнала, и при превышении заданного относительного порога принимается решение об обнаружении. Плюсом такого обнаружителя является только его простота, минусом — низкая эффективность.

2. Когерентный обнаружитель.

Когерентный обнаружитель характеризуется использованием фильтра, согласованного с сигналом посылки. При этом обеспечивается наилучшее ОСП для эхо-сигналов каждой бликовой точки. Плюсом такого обнаружителя является его лучшее по сравнению с некогерентным приемом ОСП, минусом — использование при приеме только энергии эхо-сигнала каждой бликовой точки по отдельности.

3. Обнаружитель с накоплением энергии.

Поскольку объект локации имеет сложную бликовую структуру, необходимо использовать для обнаружения всю энергию эхо-сигнала, отраженного объектом. Алгоритм работы такого обнаружителя состоит из вычисления суммарной энергии сигнала на выходе согласованного фильтра за заданное время накопления и сравнения полученной энергии с порогом.

4. Автокорреляционный обнаружитель.

Автокорреляция случайного сигнала дает практически нулевые значения при любых сдвигах, кроме 0. Автокорреляция сложного эхо-сигнала

от одной бликующей точки тоже дает очень маленький уровень боковых лепестков, но так как объект локации отражает большим количеством бликующих точек, на выходе автокоррелятора появляются всплески, по которым можно обнаружить сложный эхо-сигнал.

Есть ли способы дальнейшего увеличения ОСП и дистанции обнаружения?

Фрактальная размерность шума и эхо-сигнала

Любой случайный процесс, такой как шум или эхо-сигнал со случайной фазой, можно представить для дальнейшей обработки как фрактальный объект [4], основной характеристикой которого является его размерность [4, 5]. Размерность представлена неотрицательным нецелым числом, отражающим геометрическую сложность объекта, т. е. является мерой случайности процесса. Появление в шумовом сигнале гидролокатора эхо-сигнала от искусственного объекта изменит величину фрактальной размерности сигнала в целом, что позволит использовать величину фрактальной размерности сигнала гидролокатора для обнаружения эхо-сигнала. Наиболее эффективно применять такой обнаружитель для поиска протяженных слабоконтрастных объектов на фоне сильных помех, так как на величину фрактальной размерности сигнала практически не влияют амплитудные характеристики формирующих его сигналов [6].

В связи с тем, что сигналы гидролокатора для когерентной обработки являются узкополосными, наиболее информативной с точки зрения обработки является амплитуда сигнала на выходе согласованного фильтра. Построим логарифмическую зависимость количества срабатываний абстрактного обнаружителя от величины порога для узкополосного шума и сигнала. Для этого выберем текущее окно в сигнале, найдем максимальную ампли-

туду сигнала и будем менять порог от нуля до этого максимума. Сколько отсчетов сигнала получается выше заданного порога для эхо-сигнала, шума и их суммы при ОСП = -5 дБ, показано на рис. 1.

Наклон зависимости на этих графиках сильно отличается и не связан с абсолютной амплитудой сигнала — все величины на рисунке относительные. Чтобы сравнивать графики, достаточно численно описать угол наклона асимптотической прямой. Так вычисляется хаусдорфова или фрактальная размерность [5]:

$$D|_x = \lim_{\Delta x \rightarrow 0} \frac{\log N(\Delta x)}{\Delta x},$$

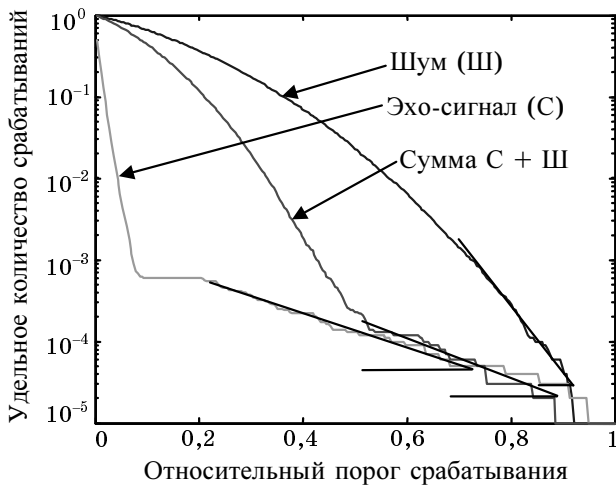
где Δx — интервал порога обнаружения; $N(\Delta x)$ — удельное количество срабатываний обнаружителя в интервале Δx .

Для случайного шума $D_{ш} = 1,03$ (и нет ярко выраженного линейного участка на графике), для чистого эхо-сигнала $D_c = 0,37$, для суммы сигнала и шума с ОСП = -5 дБ $D_{с+ш} = 0,41$.

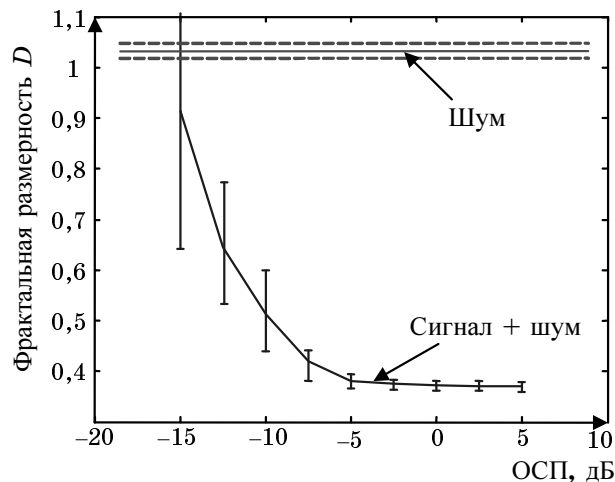
Моделирование работы обнаружителя в условиях шумов

Чтобы проверить работоспособность алгоритма, была построена компьютерная модель обнаружителя, позволяющая вычислить рабочую харак-

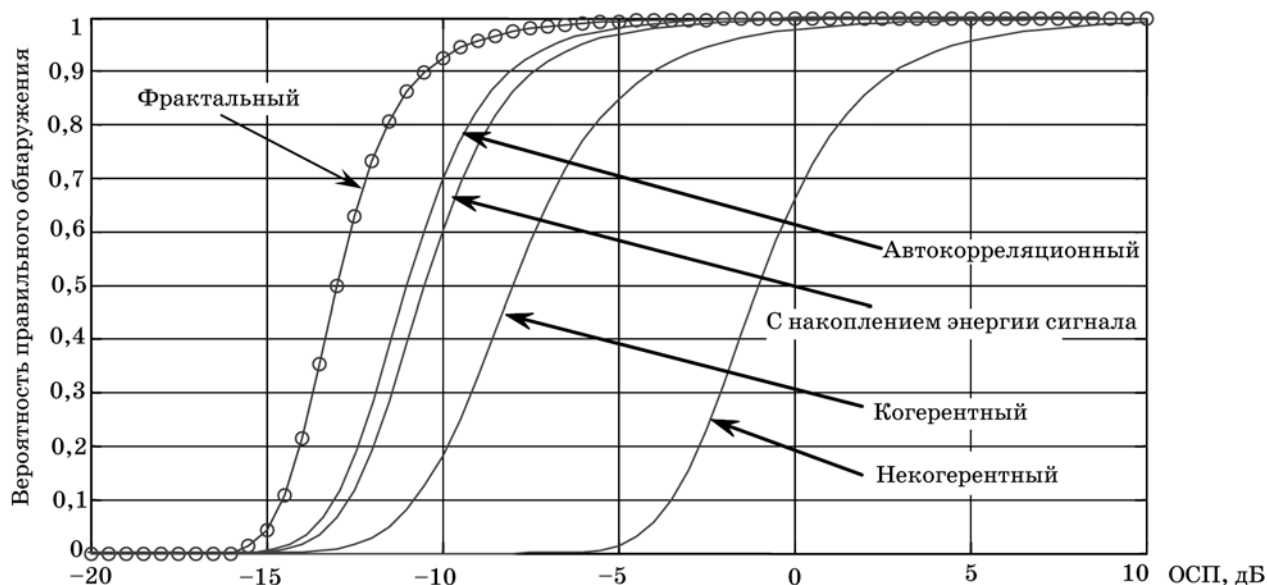
ОСП, дБ	Размерность D	Дисперсия размерности	Вероятность обнаружения
+10	0,374	0,015	0,999
0	0,380	0,064	0,999
-5	0,411	0,094	0,99
-10	0,527	0,141	0,92
-12	0,638	0,274	0,73
-15	0,825 (линейный участок выражен неявно)	4,190	0,06



■ Рис. 1. Поиск фрактальной размерности для суммы шума и сигнала



■ Рис. 2. Зависимость фрактальной размерности от ОСП



■ Рис. 3. Сравнение РХО различных обнаружителей

теристика обнаружения (РХО), т. е. зависимость вероятности правильного обнаружения от ОСП (при фиксированной вероятности ложной тревоги). Результаты работы приведены в таблице и на рис. 2.

Из таблицы видно, что предложенный алгоритм позволяет обнаруживать эхо-сигнал при ОСП = -12 дБ с вероятностью 0,73. Для сравнения с существующими алгоритмами представим графики РХО для различных методов обнаружения при вероятности ложной тревоги $P_{л.т} = 0,01$ (рис. 3).

Выводы

Предложенный алгоритм обнаружения позволяет улучшить качество работы обнаружителя при низких значениях ОСП после согласованного фильтра на 3–5 дБ, что подтверждено численным моделированием при узкополосной помехе и когерентном приеме сложного сигнала.

Направлениями дальнейшего развития данной темы являются исследование и оптимизация работы классификатора при таких значениях ОСП,

при которых может успешно работать обнаружитель, а также влияние нестационарных помех и девиации амплитуды эхо-сигнала на качество функционирования обнаружителя.

Литература

1. Евтютов А. П., Митько В. Б. Инженерные расчеты в гидроакустике. Л.: Судостроение, 1988. 287 с.
2. Бартон Д., Вард Г. Справочник по радиолокационным измерениям: Пер. с англ. М.: Сов. радио, 1976. 562 с.
3. Казаринов Ю. М. Радиотехнические системы. М.: Высш. шк., 1990. 210 с.
4. Пайтген Х.-О., Рихтер П. Х. Красота фракталов. Образы комплексных динамических систем: Пер. с англ. М.: Мир, 1993. 530 с.
5. Mandelbrot B. V. The Fractal Geometry of Nature. N. Y.: Freeman, 1982. 382 с.
6. Кроновер Р. М. Фракталы и хаос в динамических системах: Пер. с англ. М.: Постмаркет, 2000. 470 с.

**БРОНШТЕЙН
Игорь
Григорьевич**



Директор Центра оптико-информационных технологий и систем Санкт-Петербургского государственного университета информационных технологий, механики и оптики.
В 1973 году окончил Ленинградский институт точной механики и оптики.
Является автором 60 научных публикаций.
Область научных интересов — оптико-информационные системы.

**БУХАНОВСКИЙ
Александр
Валерьевич**



Доктор технических наук, профессор кафедры информационных систем Санкт-Петербургского государственного университета информационных технологий, механики и оптики.
В 1995 году окончил Санкт-Петербургский государственный морской технический университет по специальности «Прикладная математика».
В 2005 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более 150 научных публикаций.
Область научных интересов — высокопроизводительные вычисления, статистический анализ и обработка данных, моделирование сложных систем.

**ВЕДЕРНИКОВ
Юрий
Вадимович**



Доцент, старший преподаватель кафедры систем управления ракет Михайловской военной артиллерийской академии.
В 1992 году окончил Саратовское высшее военное командно-инженерное училище ракетных войск им. А. И. Лизюкова.
В 2001 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором 56 научных публикаций.
Область научных интересов — системный анализ, теория принятия решений, исследование операций, инновационные технологии управления, векторная оптимизация сложных технических систем, принятие решений в условиях неопределенности и риска, интервальный анализ.

**ГОРДЕЕВ
Александр
Владимирович**



Профессор, заведующий кафедрой компьютерных систем проектирования Санкт-Петербургского государственного университета аэрокосмического приборостроения.
В 1975 году окончил Ленинградский институт авиационного приборостроения.
В 1998 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более 80 научных публикаций.
Область научных интересов — моделирование параллельных вычислительных систем, распределенные вычисления, вычислительные сети.

**ИВАНОВ
Сергей
Владимирович**



Младший научный сотрудник НИИ наукоемких компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики.
В 2004 году окончил Санкт-Петербургский государственный морской технический университет по специальности «Программное обеспечение вычислительной техники и автоматизированных систем».
Является автором десяти научных публикаций.
Область научных интересов — высокопроизводительные вычисления, моделирование сложных систем.

**КОБЗАРЕВ
Николай
Анатольевич**



Заместитель начальника отдела разработки программного обеспечения ОАО «РИВЦ-Пулково», аспирант.
В 2005 году окончил Санкт-Петербургский государственный университет аэрокосмического приборостроения по специальности «Вычислительные машины, системы, комплексы и сети».
Является автором шести научных публикаций.
Область научных интересов — проектирование систем распределенного доступа к данным и их практическое применение для решения задач обеспечения полетов и наземного обслуживания средств авиакомпаний.

**КОВАЛЕВ
Даниил
Викторович**



Аспирант кафедры биомедицинской электроники и охраны среды Санкт-Петербургского государственного электротехнического университета «ЛЭТИ». В 2005 году окончил Санкт-Петербургский государственный университет информационных технологий, механики и оптики по специальности «Вычислительные машины, комплексы, системы и сети». Является автором 16 научных публикаций, в том числе соавтор одной монографии. Область научных интересов — методология объектно-ориентированного анализа и проектирования, ее развитие и применение в интересах создания программного обеспечения.

**КОВАЛЬЧУК
Сергей
Валерьевич**



Аспирант, младший научный сотрудник НИИ наукоемких компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики. В 2006 году с отличием окончил Оренбургский государственный университет по специальности «Программное обеспечение вычислительной техники и автоматизированных систем». Является автором семи научных публикаций. Область научных интересов — архитектура высокопроизводительных программных комплексов, распределенные вычислительные системы, высокопроизводительная обработка сверхбольших объемов данных на кластерных системах, параллельные алгоритмы (разработка, адаптация, оптимизация).

**КОЛБАНЕВ
Михаил
Олегович**



Профессор, заведующий кафедрой информационных управляющих систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. Мастер связи. В 1977 году окончил Ленинградский электротехнический институт связи им. проф. М. А. Бонч-Бруевича по специальности «Автоматическая электросвязь». В 2004 году защитил диссертацию на соискание ученой степени доктора технических наук. Является автором более 100 научных публикаций. Область научных интересов — инфокоммуникации, системы реального времени.

**КОЛЫХМАТОВ
Илья
Игоревич**



Стажер-исследователь НИИ наукоемких компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики. В 2007 году окончил Санкт-Петербургский государственный университет информационных технологий, механики и оптики по специальности «Прикладная математика и информатика». Является автором пяти научных публикаций. Область научных интересов — разработка параллельных алгоритмов.

**ЛИВШИЦ
Ирина
Леонидовна**



Старший научный сотрудник, заведующая лабораторией «Специальные оптические и ТВ-системы» Санкт-Петербургского государственного университета информационных технологий, механики и оптики. В 1974 году окончила Ленинградский институт точной механики и оптики. В 1980 году защитила диссертацию на соискание ученой степени кандидата технических наук. Является автором 122 научных публикаций. Область научных интересов — оптические и оптико-информационные системы.

**МАЛКОВ
Константин
Олегович**



Аспирант Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. В 2004 году окончил Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича по специальности «Информационные системы и технологии». Является автором семи научных публикаций. Область научных интересов — инфокоммуникации, управление сетями связи.

**МАЛЬЦЕВ
Георгий
Николаевич**



Профессор, начальник кафедры космических радиотехнических систем Военно-космической академии им. А. Ф. Можайского, заслуженный деятель науки Российской Федерации, действительный член Академии космонавтики им. К. Э. Циолковского. В 1980 году окончил Ленинградский инженерный Краснознаменный институт им. А. Ф. Можайского. В 1994 году защитил диссертацию на соискание ученой степени доктора технических наук. Является автором более 200 научных публикаций. Область научных интересов — обработка сигналов в радиотехнических и оптико-электронных информационных системах, космические радиотехнические комплексы управления, сбора и передачи информации.

**ПАШКОВСКИЙ
Матвей
Александрович**



Программист межкафедрального компьютерного класса естественно-научного факультета Санкт-Петербургского государственного университета информационных технологий, механики и оптики. В 2005 году окончил Санкт-Петербургский государственный университет информационных технологий, механики и оптики. Является автором пяти научных публикаций. Область научных интересов — параметрический синтез оптических систем, моделирование и визуализация физических процессов.

**САФРОНОВ
Валерий
Васильевич**



Профессор, главный научный сотрудник ОАО «КБ Электроприбор» (г. Саратов), действительный член Академии военных наук, заместитель председателя Поволжского отделения Академии военных наук по научной работе. Награжден знаком «Изобретатель СССР». В 1997 году защитил диссертацию на соискание ученой степени доктора технических наук. Является автором более 150 научных и учебно-методических публикаций. Область научных интересов — методы целочисленного программирования, методы теории принятия решений.

**СЕМЕНОВ
Николай
Николаевич**



Соискатель, инженер-разработчик ИТЦ «Протей», преподаватель кафедры морских информационно-управляющих и измерительных систем Санкт-Петербургского морского технического университета. В 1998 году окончил Санкт-Петербургский морской технический университет по специальности «Роботы и робототехнические системы». Является автором шести научных публикаций. Область научных интересов — теория распознавания образов, оптимальный прием сигналов и их обнаружение, обработка, сжатие, распознавание и синтез речи, гидроакустика, цифровая обработка сигналов.

**СЕРГЕЕВ
Михаил
Борисович**



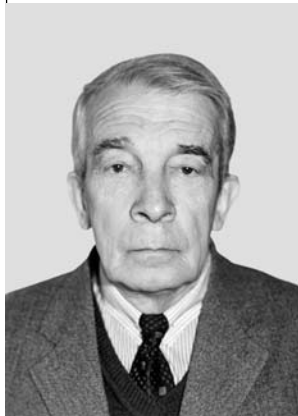
Профессор, заведующий кафедрой вычислительных систем и сетей Санкт-Петербургского государственного университета аэрокосмического приборостроения, почетный работник высшего профессионального образования Российской Федерации. В 1980 году окончил Ленинградский электротехнический институт им. В. И. Ульянова (Ленина) по специальности «Электронные вычислительные машины». В 2001 году защитил диссертацию на соискание ученой степени доктора технических наук. Является автором более 100 научных публикаций, 13 авторских свидетельств на изобретения. Область научных интересов — теория разрядных вычислений, методы проектирования спецпроцессоров для систем контроля и управления, оптико-информационные системы.

**СОТНИКОВ
Александр
Дмитриевич**



Профессор кафедры информационных технологий в экономике Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. В 1975 году окончил Ленинградский электротехнический институт связи им. проф. М. А. Бонч-Бруевича по специальности «Автоматическая электрическая связь». В 1983 году защитил диссертацию на соискание ученой степени кандидата технических наук. Является автором 85 научных публикаций, 10 авторских свидетельств на изобретения. Область научных интересов — теория информационных процессов и методы информационного взаимодействия, анализ и проектирование информационных систем для здравоохранения.

**СТОГОВ
Генрих
Владимирович**



Профессор кафедры космических радиотехнических систем Военно-космической академии им. А. Ф. Можайского. В 1957 году окончил Ленинградскую Краснознаменную военно-воздушную инженерную академию им. А. Ф. Можайского. В 1972 году защитил диссертацию на соискание ученой степени доктора технических наук. Является автором более 300 научных и учебно-методических публикаций. Область научных интересов — обработка сигналов и сложных систем.

**ТАТАРНИКОВА
Татьяна
Михайловна**



Доцент кафедры информационных управляющих систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. В 1993 году окончила Восточно-Сибирский технологический институт по специальности «Электронно-вычислительные машины, комплексы, системы и сети». В 1997 году защитила диссертацию на соискание ученой степени кандидата технических наук. Является автором более 50 научных публикаций. Область научных интересов — инфокоммуникации, взаимодействие неоднородных сетей.

УНЧУН ЧО



Профессор факультета инженерной механики Корейского политехнического университета. В 1987 году окончил университет Ёнсей (Республика Корея) со степенью бакалавра, в 1988 году получил степень магистра в университете Карнеги Меллон (США), в 1997 году — степень доктора в Политехническом институте Ренсселер (США). Является автором 37 научных публикаций. Область научных интересов — техническая оптика, лазеры, нанотехнологии, микрокомпоненты и материалы.

**ФЕДОРЕНКО
Сергей
Валентинович**



Доцент, докторант кафедры безопасности информационных систем Санкт-Петербургского государственного университета аэрокосмического приборостроения. В 1990 году окончил Ленинградский институт авиационного приборостроения. В 1994 году защитил диссертацию на соискание ученой степени кандидата технических наук. Является автором 50 научных публикаций. Область научных интересов — алгебраическая и комбинаторная теория кодирования, преобразование Фурье над конечным полем.

**ШАПОВАЛОВ
Валентин
Викторович**



Директор Федерального государственного научного учреждения «Научно-исследовательский конструкторско-технологический институт биотехнических систем». В 1969 году окончил Ленинградский государственный университет по специальности «Физика». В 2000 году защитил диссертацию на соискание ученой степени доктора технических наук. Является автором 94 научных публикаций. Область научных интересов — медицинское приборостроение, теория и синтез биотехнических систем медицинского назначения.

УДК 621.391

Двумерное преобразование Хаара и особенности его вычисления при обработке оптических изображений

Мальцев Г. Н., Стогов Г. В. Информационно-управляющие системы, 2008. № 3. С. 2–4.

Рассмотрены особенности вычисления преобразования Хаара при обработке оптических изображений, которые описываются дискретными и непрерывными двумерными функциями. Приводятся обобщенная матрица преобразования Хаара и ее модифицированная форма. Показано, что у групп функций Хаара, соответствующих одинаковой эквивалентной секвенте, дисперсии коэффициентов разложения равны между собой.

Список лит.: 6 назв.

УДК 004.421.4

Метод получения оценок времени выполнения на ранних этапах проектирования сложных распределенных приложений

Гордеев А. В., Кобзарев Н. А. Информационно-управляющие системы, 2008. № 3. С. 5–9.

Рассмотрены основные средства построения распределенных приложений, возможности средств моделирования. Предлагается новый подход к анализу моделей приложений, позволяющий определять потенциальные ошибки проектирования до выполнения стадии программирования и тестирования, не усложняя работу системного архитектора. Данный метод заключается в преобразовании исходных UML-диаграмм проекта в диаграммы Ганта. Необходимые данные для построения диаграмм Ганта предлагается получать на основе метаданных исходных UML-диаграмм.

Список лит.: 6 назв.

УДК 681.3.069, 681.324

Особенности проектирования высокопроизводительных программных комплексов для моделирования сложных систем

Ковальчук С. В., Иванов С. В., Колыхматов И. И., Бухановский А. В. Информационно-управляющие системы, 2008. № 3. С. 10–18.

Рассматриваются особенности проектирования высокопроизводительных программных комплексов для моделирования сложных систем на примере задачи моделирования экстремальных явлений в атмосфере и океане. Обсуждается подход к формализации структуры комплекса, сочетающий в себе специфику концепций SOA и REST, в соответствии с задачей отображения компонентов сложной системы на параллельную вычислительную архитектуру. Предложен способ оценки архитектуры на основе анализа параметрических моделей параллельной производительности.

Список лит.: 30 назв.

УДК 621.391

Two dimensional Haar transform and specifics of its calculation while processing optical images

Maltsev G. N., Stogov G. V. IUS, 2008. N 3. P. 2–4.

We discuss some specifics of calculating Haar transform applied to processing optical images that are described using discrete and continuous two-dimensional functions. A generalized matrix of Haar transform, as well as its modified form, is presented. It is shown that with Haar function groups corresponding to the same equivalent sequent, factorization dispersion coefficients are equal.

Refs: 6 titles.

УДК 004.421.4

A method of acquiring estimated application execution time during early phases of complex distributed applications design.

Gordeev A. V., Kobzarev N. A. IUS, 2008. N 3. P. 5–9.

Basic tools and concepts of distributed application design are reviewed. The whole new concept of application diagram analysis is proposed by authors giving an ability for software architects to detect and eliminate possible architect issues before the implementation of project's programming stage and project's test stage, without any additional complication of architect's work. This method is based on source UML diagrams converted into Gant diagrams. The required data for conversion is based on the source UML diagram's metadata.

Refs: 6 titles.

УДК 681.3.069, 681.324

Special characteristics of high performance software for complex systems simulations

Kovalchuk S. V., Ivanov S. V., Kolykhmatov I. I., Bukhanovskii A. V. IUS, 2008. N 3. P. 10–18.

The characteristics of high performance software for complex systems simulations are presented in this paper using extreme events in atmosphere and ocean modeling as an example. An approach to formalization of the structure of scientific computing software is discussed, which combines SOA and REST concepts according to mapping of software components to parallel computer architecture. Software architecture evaluation criteria based on analysis of parametric performance models is suggested.

Refs: 30 titles.

УДК 519.876.5

Характеристики программного обеспечения для проектирования дифракционно-ограниченных оптико-электронных систем

Лившиц И. Л., Паиковский М. А., Бронштейн И. Г., Сергеев М. Б., Унчун Чо. Информационно-управляющие системы, 2008. № 3. С. 19–22.

Анализируется программное обеспечение, наиболее подходящее для проектирования дифракционно-ограниченных систем с высоким качеством изображения. В качестве примера рассматривается проектирование широкоугольного объектива для миниатюрной ПЗС-камеры.

Список лит.: 3 назв.

УДК 519.725

Простой алгоритм декодирования алгебраических кодов

Федоренко С. В. Информационно-управляющие системы, 2008. № 3. С. 23–27.

Рассматривается алгоритм декодирования, который представляется самым простым и естественным в классе алгоритмов декодирования алгебраических кодов до их корректирующей способности. Приводятся вывод, описание, анализ и доказательство корректности алгоритма. Асимптотическая сложность алгоритма совпадает со сложностью лучших алгоритмов декодирования, а описание является самым простым из описаний известных алгоритмов.

Список лит.: 18 назв.

УДК 004.72

Подход к организации адаптивного согласующего центра корпоративной сети

Колбанев М. О., Татарникова Т. М., Малков К. О. Информационно-управляющие системы, 2008. № 3. С. 28–31.

Приводится один из подходов к организации согласующего центра, выполняющего функции интеграции и согласования неоднородных компьютерных сетей, основанный на динамической настройке структуры центра в зависимости от решаемых задач и внешних условий.

Список лит.: 3 назв.

УДК 519.876.5

Characteristic properties of optics-electronics design software for diffraction-limited systems

Livshits I. L., Pashkovskii M. A., Bronshtein I. G., Sergeev M. B., Un Chung Cho. IUS, 2008. N 3. P. 19–22.

Software most suitable for high quality diffraction-limited systems design is discussed. Design of a wide angle lens for miniature CCD camera is given as an example.

Refs: 3 titles.

УДК 519.725

A simple algorithm for decoding algebraic codes

Fedorenko S. V. IUS, 2008. N 3. P. 23–27.

A simple and natural algorithm for decoding algebraic codes in the class of algorithms decoding up to the designed error correcting capability is described. The design, description, analysis and correctness proof of this algorithm are given. The asymptotic complexity of the algorithm coincides with the complexity of the best decoding algorithms, and the description is the simplest for the known algorithms descriptions.

Refs: 18 titles.

УДК 004.72

An approach to set up a corporate network matching center

Kolbanev M. O., Tatarnikova T. M., Malkov K. O. IUS, 2008. N 3. P. 28–31.

We suggest another approach to set up a corporate network matching center which performs functions of integration and coordination of non-uniform computer networks, based on dynamic adjustment of center structure depending on soluble tasks and external conditions.

Refs: 3 titles.

УДК 007:004.3

Метод многокритериального ранжирования сложных систем при различных видах неопределенности исходных данных

Ведерников Ю. В., Сафронов В. В. Информационно-управляющие системы, 2008. № 3. С. 32–38.

Рассматривается важная прикладная задача принятия решения для случая, когда критерии могут быть одновременно заданы: в формализованном, количественном виде; в неопределенном, лингвистическом виде; в частично формализованном виде. Задача сводится к построению упорядоченного множества эффективных вариантов (кортежа Парето) сложных технических систем для случая, когда критерии оптимальности неоднородны. Приведен численный пример.

Список лит.: 17 назв.

УДК 004.435

Архитектура процессора метаданных

Ковалев Д. В., Шаповалов В. В. Информационно-управляющие системы, 2008. № 3. С. 39–45.

Приводятся краткие сведения о методологии метауправления и описывается подход, обеспечивающий независимость реализации метауправления в информационных системах от конкретного языка представления и средства доступа к метаданным.

Список лит.: 6 назв.

УДК 621.391.1

Инфокоммуникационные системы и их модели для здравоохранения

Сотников А. Д. Информационно-управляющие системы, 2008. № 3. С. 46–53.

Рассматриваются способы формального описания процессов информационного взаимодействия в сложных инфокоммуникационных системах и их классификация. Определены основные понятия и термины. Предлагается комплекс разноуровневых моделей для прикладных инфокоммуникационных систем телемедицинского назначения.

Список лит.: 7 назв.

УДК 681.883.022: 681.883.65

Фрактальный обнаружитель эхо-сигнала

Семенов Н. Н. Информационно-управляющие системы, 2008. № 3. С. 54–56.

Предложен новый алгоритм для обнаружения морских подводных целей, основанный на оценке меры случайности сигнала на входе гидролокатора. В качестве меры предлагается фрактальная размерность.

Список лит.: 6 назв.

УДК 007:004.3

A method of multi—criteria ranking of complex systems with different types of data uncertainty

Vedernikov Yu. V., Safronov V. V. IUS, 2008. N 3. P. 32–38.

We discuss a task of decision making in the case when the criteria could be formalized quantitatively, given in an uncertain linguistic form, and in a partially formalized form at the same time. The task is reduced to building an ordered set of effective variants (Pareto tuple) of complex technical systems when the optimization criteria are not uniform. A numerical example is given.

Refs: 17 titles.

УДК 004.435

Metadata processor architecture

Kovalyov D. V., Shapovalov V. V. IUS, 2008. N 3. P. 39–45.

A brief overview of metamanagement methodology is discussed. A method of providing a metamanagement implementation independent from metadata language representation and metadata access in informational systems is offered.

Refs: 6 titles.

УДК 621.391.1

Info-communication systems and their respective models for healthcare providers

Sotnikov A. D. IUS, 2008. N 3. P. 46–53.

Methods for formal description of informational interaction in complex info-communicational systems and its models are investigated. Main notions and terms are defined. A complex of multilevel models for applied info-communicational systems for telemedicine is proposed.

Refs: 7 titles.

УДК 681.883.022: 681.883.65

Fractal sonar signal detector

Semenov N. N. IUS, 2008. N 3. P. 54–56.

A new algorithm of the sea underwater targets detection, based on the evaluation of the signal «randomness» measure at the input of sonar is proposed. The measure proposed is fractal dimension.

Refs: 6 titles.



УралПриборЭкспо Электротехника Автоматика-2008

МЕЖРЕГИОНАЛЬНАЯ СПЕЦИАЛИЗИРОВАННАЯ ВЫСТАВКА
21–23 октября 2008 г. Екатеринбург

Организаторы:

- Компания «ЭкспоГрад»
- Комитет промышленной политики и развития предпринимательства Администрации г. Екатеринбурга
- Технопарк «Приборостроение»

При поддержке:

- Администрации г. Екатеринбурга
- Министерства промышленности и науки Свердловской области
- Уральского отделения РАН

Цели выставки:

Содействие развитию межотраслевых и межрегиональных связей в целях расширения производственной и сбытовой деятельности; демонстрация современных технологий и продукции для обеспечения промышленных предприятий, обмен передовым опытом.

Тематические разделы выставки:

Приборы:

контрольно-измерительные, испытательные, диагностические, аналитические, лабораторные;

Электроника:

электронные устройства, компоненты, элементная база, печатные платы, датчики и микросистемы, телекоммуникационные устройства, радиостанции, полупроводниковые устройства;

Электротехника:

высоковольтная аппаратура, генераторы, кабельно-проводниковая продукция, низковольтная аппаратура, трансформаторы, электродвигатели, электроизоляторы, электростанции, электрощитовое оборудование, источники питания, промышленное освещение.

Оборудование, инструмент для производства и монтажа электротехнической продукции. Электрооборудование для промышленности и ЖКХ. Бытовые электротехнические изделия;

Энергетика:

энергосберегающие технологии, оборудование и материалы;

Автоматика:

автоматизированные системы и технические средства управления производством, предприятием и технологическими процессами;

Метрология. Весоизмерительное оборудование.

Передовые технологии производства.

Нанотехнологии и наноматериалы в приборостроении.

Защита интеллектуальной собственности.

Спец-проект: «Научные разработки — производству».

В программе выставки:

Научно-техническая конференция, семинары, презентации, консультации специалистов, конкурс по номинациям среди экспонентов.

Участники выставки:

Промышленные предприятия Урала и других регионов; научные учреждения, разработчики новых технологий и материалов; предприятия малого и среднего бизнеса; периодические издания.

Посетители выставки:

Руководители, специалисты предприятий сферы машиностроения, энергетического комплекса, транспорта, жилищно-коммунального хозяйства.

Постоянным участникам и товаропроизводителям предоставляются скидки.

Место проведения:

Г Р В Ц г. Екатеринбург, ул. Громова, 145
проезд: автобус: № 76, 46
маршрутное такси: № 067, 46, 76
троллейбус: № 11 остановка «Громова»

Выставочный оператор: ООО «ЭкспоГрад»
ул. Первомайская, 104, оф. 511
Тел/факс: (343) 379-04-28, 379-04-47, 379-04-46
http: www.expograd.ru

Режим работы:

Регистрация и оформление экспозиции:
20 октября с 12.00–18.00
Работа выставки: 21–22 октября с 10.00–17.00
23 октября с 10.00–15.00
Демонтаж: 23 октября с 15.00

Руководитель проекта: **Попкова Татьяна Николаевна**
Тел/факс: (343) 379-04-47. E-mail: prom@expograd.ru
Старший менеджер: **Пасынкова Ирина Николаевна**
Тел/факс: (343) 379-04-28. E-mail: socium@expograd.ru
http: www.expograd.ru