

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

3(151)
2009

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Вьюкова Н. И., Галатенко В. А., Самборский С. В. Обобщение задачи программной конвейеризации для гнезд циклов 2
Лаптев В. В., Морозов А. В. Экспертная система оценки качественных характеристик проектной модели программного обеспечения 8
Зуев А. С., Петров Ю. И. Описание модификации строки адреса проводника Windows Explorer 11

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Гаврилов С. В., Каграманян Э. Р., Ходош Л. С. Тенденции развития моделей библиотечных элементов для статического временного анализа цифровых СБИС. . . 20
Шмырев Н. В. Средства моделирования и контроля распределенных систем реального времени. 24

КОМПЬЮТЕРНАЯ ГРАФИКА

- Бимаков В. А. Высокоскоростная обработка n -мерных пиксельных геометрических моделей 29
Архипов О. П., Зыкова З. П. Обеспечение идентичности результатов печати на различных цветных принтерах 37

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Колдаев А. И. Моделирование интеллектуальной системы поддержки принятия решений при управлении технологическим процессом 42
Михеева Т. И., Потапкин А. А., Демьяненко Р. В. Адаптивная кластеризация графа улично-дорожной сети 46

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ

- Куперин Ю. А., Счастливцев Р. Р. Прогнозирование финансовых временных рядов с помощью комитетов искусственных нейронных сетей 51
Серков Л. А. Синергетические аспекты моделирования инновационных процессов 56
Соловецкий А. С. Мобильные информационные технологии в маркетинге и рекламе 62

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Сарафанов А. В., Комаров В. А. Разработка математической модели многопользовательского режима функционирования аппаратно-программных комплексов с удаленным доступом 67
Норенков И. П., Соколов Н. К. Синтез индивидуальных маршрутов обучения в онтологических обучающих системах. 74
Рыбаков А. Е. Моделирование структуры учебных курсов и подсистемы навигации для систем управления обучением, поддерживающих стандарт SCORM 77

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Сердобинцев С. П., Коломейко Ф. В. Применение информационных технологий в прогнозировании распределения объектов рыболовства 82

ИНФОРМАЦИЯ

- Contents 86
Приложение. Ледокол. Страницы биографии Анатолия Ивановича Китова.

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

АВДОШИН С. М.
АНТОНОВ Б. И.
БАТИЩЕВ Д. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ГОРБАТОВ В. А.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б. М.
НАРИНЬЯНИ А. С.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

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

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

УДК 004.416.2

Н. И. Вьюкова, ст. науч. сотр.;
В. А. Галатенко, д-р физ.-мат. наук, ст. науч. сотр.;
С. В. Самборский, ст. науч. сотр.,
НИИ системных исследований РАН,
e-mail: niva@niisi.msk.ru

Обобщение задачи программной конвейеризации для гнезд циклов

Рассмотрен подход к решению задачи программной конвейеризации циклов для случая гнезд вложенных циклов. Подход основан на развертке одного или нескольких из объемлющих циклов с последующей их перестановкой до уровня внутреннего цикла. Предложено также основанное на методах целочисленного линейного программирования совмещение развертки и конвейеризации, применимое, в частности, для случая, когда развертываемый цикл непрерывночен с какими-либо из вложенных в него циклов.

Ключевые слова: оптимизация кода, программная конвейеризация циклов, планирование по модулю, гнездо циклов, развертка цикла, целочисленное линейное программирование (ЦЛП).

Введение

В современных приложениях для встроенных систем решаются сложные вычислительные задачи из области обработки сигналов и изображений. Для ускорения решения этих задач могут выделяться специализированные сопроцессоры, однако это не всегда бывает оправдано. Таким образом, при разработке компилятора для универсального процессора, используемого в составе бортовых систем, остается актуальной задача оптимизации исполняемого кода. При этом всегда уделяется особое внимание трансляции циклов и особенно гнезд вложенных циклов в связи с тем, что именно исполнение таких участков занимает большую часть времени выполнения любой программы.

В современных процессорах исполнение команд, как правило, конвейеризовано, тем самым процессор способен исполнять несколько команд параллельно. Кроме того, команды разных типов исполняются разными функциональными устройствами и требуют разного времени для своего выполнения.

Для эффективного использования параллелизма процессора при трансляции циклов широко

применяется *программная конвейеризация* циклов [1], основанная на совмещении в одной итерации оптимизированного цикла команд, относящихся к разным итерациям исходного цикла. Выигрыш достигается за счет того, что операции из одной итерации исходного цикла могут запускаться в разных итерациях конвейеризованного цикла. При этом длительные операции запускаются заранее, с тем чтобы их результаты были готовы к моменту, когда они понадобятся. Тем самым исключаются простои и лучше используются ресурсы процессора.

В работе [2] представлен обзор алгоритмов программной конвейеризации методом планирования по модулю, которые широко используются в современных промышленных компиляторах. В [3] сформулированы проблемы планирования по модулю в виде задачи целочисленного линейного программирования. В данной работе рассматриваются ограничения метода программной конвейеризации и механизмы их преодоления, если оптимизируется не один цикл, а гнездо вложенных циклов.

Программная конвейеризация простого цикла

Цель программной конвейеризации — получить тело цикла, которое исполняется за возможно меньшее число тактов. Эта длительность обозначается II (*Initiation Interval* — интервал запуска). Название отражает тот факт, что через II тактов может быть запущена новая итерация конвейеризованного цикла. В отечественной литературе II часто называется *основным интервалом*.

Так как необходимо минимизировать II , полезно знать, чем он ограничен снизу. Есть две важные нижние оценки, а именно: оценка по ресурсам — $ResII$ и оценка по зависимостям — $RecII$.

Оценка $ResII$ связана с количеством ресурсов процессора. Под ресурсами понимаются функциональные устройства, пропускная способность памяти, число команд, которые можно одновременно подать на исполнение, поля в командном слове для VLIW-архитектуры. Нетрудно показать (см. [1]), что верна оценка $II \geq ResII = \max_i (n_i/r_i)$, где r_i — число единиц ресурса i -го типа, а n_i — потребность в ресурсе i -го типа на одну итерацию цикла.

Оценка по зависимостям возникает из структуры зависимостей между командами тела цикла. Зависимости между командами описываются графом потоков данных (DFG — *Data Flow Graph*),

в дальнейшем называемом *графом зависимостей*. Вершинами этого ориентированного графа служат команды исходного цикла. Его ребра соответствуют истинным зависимостям команд, т. е. для команд c_1 и c_2 ребро (c_1, c_2) присутствует в DFG тогда и только тогда, когда команда c_2 использует данные, производимые командой c_1 . Ребра DFG снабжены двумя неотрицательными целочисленными пометками: *латентностью* (lat) и *дистанцией* ($dist$). Латентность определяет, через сколько тактов самое раннее может быть запущена команда c_2 после запуска команды c_1 , с тем чтобы результаты c_1 были готовы к тому моменту, когда они потребуются команде c_2 . Дистанция указывает на разницу в итерациях исходного цикла, к которым относятся команды c_1 и c_2 . Конкретнее, дистанция равна относительному номеру итерации, в которой команда c_2 использует результат c_1 . Если результат c_1 используется в той же итерации исходного цикла, то дистанция нулевая, если в следующей — единичная и т. д. Здесь и далее предполагается, что дистанции однородны, т. е. одинаковы для всех итераций, и являются константами, известными на стадии компиляции.

Циклы в графе зависимостей далее будем называть контурами, чтобы отличать их от циклов в компилируемой программе. Можно показать (см., например, [1]), что $RecII \geq ResII = \max_{w \in cycles(DFG)} \frac{\sum_{e \in w} lat_e}{\sum_{e \in w} dist_e}$, где $cycles(DFG)$ — множество контуров графа зависимостей; $lat_e, dist_e$ — соответственно латентность и дистанция зависимости e .

Если в некотором контуре графа зависимостей присутствуют ребра с большой латентностью (длительные операции), то может оказаться, что $RecII > ResII$. В этом случае ограничения по зависимостям не позволят эффективно использовать ресурсы процессора. Простейший пример:

```
DO J = 2, N
  A(J) = SQRT(A(J - 1))
```

Аргумент операции извлечения квадратного корня зависит от ее результата на предыдущей итерации, следовательно, имеем контур в графе зависимостей с суммарной дистанцией 1 и суммарной латентностью не менее латентности операции $SQRT$ ¹. Если она велика (что обычно имеет место), то эффективное выполнение данного цикла невозможно, поскольку длительность II не может быть меньше этой латентности. Можно показать, что развертка цикла не позволяет исправить ситуацию. Однако, если мы имеем дело

¹ Будем считать, что извлечение квадратного корня представляет собой элементарную операцию.

с гнездом циклов, то эффективность вычислений может быть повышена за счет параллельного выполнения итераций объемлющих циклов.

Зависимости в гнезде вложенных циклов

Усложним последний пример, добавив внешний цикл:

```
DO I = 2, N
  DO J = 2, N
    A(I, J) = SQRT(A(I, J - 1))
```

В графе зависимостей внутреннего цикла есть контур, содержащий длительную операцию $SQRT$, и II не может быть меньше латентности этой операции. Однако теперь в примере есть независимые операции.

Определим *пространство итераций* как набор точек, соответствующих итерациям исходного цикла. Размерность пространства равна числу вложенных циклов, а координаты точек определяются индексами циклов, для которых выполняется эта итерация. Если верхние и нижние границы циклов не зависят от индексов объемлющих циклов и шаг цикла всегда равен 1, то пространство итераций есть множество целочисленных точек, попавших внутрь или на границу параллелепипеда с ребрами, параллельными координатным осям. Эти условия не обязательны, однако для простоты далее рассматриваются только такие пространства итераций.

Для команд, принадлежащих гнезду циклов, дистанция зависимости определяется как вектор смещения в пространстве итераций между итерациями, соответствующими этим командам. Порядок на координатах соответствует вложенности циклов, самым младшим является индекс внутреннего цикла. У вектора дистанции могут присутствовать отрицательные координаты, но в предположении, что шаги всех циклов положительны, этот вектор должен быть *лексикографически неотрицателен*, т. е. его старшая ненулевая координата положительна. Этот факт вызван тем, что исходное гнездо циклов задает лексикографический порядок обхода пространства итераций, а любая зависимость направлена вперед в этой последовательности. При корректных преобразованиях гнезда циклов дистанции зависимостей должны оставаться лексикографически неотрицательными. В противном случае возникнет зависимость от еще не выполненных вычислений.

Суммарная дистанция по контуру в графе зависимостей также становится векторной. При вычислении $RecII$ для конвейеризации внутреннего цикла существенны только те контуры в графе зависимостей, у которых дистанция нулевая по всем

координатам, кроме соответствующей внутренней циклу.

В примере (1) имеем один контур в DFG с дистанцией (0, 1) по I и J соответственно. Переставив заголовки циклов местами, что корректно для примера (1), получим следующий код:

```
DO J = 2, N
  DO I = 2, N
    A(I, J) = SQRT(A(I, J - 1))
```

(2)

Суммарная дистанция единственного контура в DFG стала равной (1, 0), и этот контур больше не учитывается при вычислении *RecII*, следовательно, *RecII* = 0, т. е. конвейеризация ограничивается только количеством доступных ресурсов.

Тем не менее, подобный подход нельзя считать ни эффективным, ни универсальным. Во-первых, в предположении, что элементы массивов хранятся в памяти по строкам, при перестановке циклов теряется локальность доступа к элементам массива. Если считать, что строка кэша вмещает четыре элемента и кэш невелик в сравнении с размерами массива, то в примере (1) имеем один промах на четыре итерации, а в примере (2) — в среднем пять промахов на четыре итерации, т. е. в 5 раз хуже.

Во-вторых, перестановка циклов не всегда позволяет снять зависимости. Рассмотрим пример, в котором индексы I, J используются симметрично:

```
DO I = 2, N
  DO J = 2, N
    A(I, J) = SQRT(A(I, J - 1) *
      * A(I - 1, J))
```

(3)

В DFG имеем два контура с суммарными дистанциями (0, 1) и (1, 0). Перестановка заголовков превращает их в (1, 0) и (0, 1) соответственно, поэтому она абсолютно бесполезна.

Для того чтобы использовать параллелизм, присутствующий в гнезде циклов, в общем случае применяется метод гиперплоскостей, впервые предложенный Лампортом в [4] и усовершенствованный в работах [5–7]. Суть метода заключается в том, что выбирается такая гиперплоскость, проходящая через начало координат в пространстве итераций, что все векторы дистанций зависимостей находятся в одном полупространстве. Пространство итераций можно разбить гиперплоскостями, параллельными выбранной, на слои одинаковой толщины. Толщина слоев выбирается как можно больше, однако таким образом, чтобы две итерации, попавшие в один слой, были заведомо независимы. Вычисления по слоям выполняются последовательно, но внутри слоя итерации могут выполняться параллельно. Перестановка циклов — частный случай метода гиперплоскостей.

Для примера (3) метод гиперплоскостей заключается в обходе пространства итераций по

диагоналям: итерации, лежащие на одной диагонали, можно выполнять в произвольном порядке и конвейеризовать без задержек, вызванных зависимостями. Тем не менее, вопрос с эффективностью использования кэш-памяти остается: при любом порядке выполнения итераций, лежащих на одной диагонали, в среднем произойдет не меньше одного промаха на итерацию. Неэффективность использования кэша объясняется тем обстоятельством, что применение метода гиперплоскостей (и, в частности, перестановка циклов) меняет "глобально" порядок обхода пространства итераций. Однако порядок обхода в исходной программе мог быть выбран сознательно, с учетом устройства кэша.

Развертка внешнего цикла

По изложенным выше и ряду других причин в последнее время стали популярны методы, основанные на развертке какого-либо объемлющего цикла в гнезде вложенных циклов. Оригинальный и интересный подход предложен в работах [8, 9]. Данная работа основана на более традиционном подходе к развертке, изложенном, например, в [10].

Посмотрим, как этот подход работает на примере (3). Сначала внешний цикл заменяется парой вложенных циклов, где внутренний выполняется небольшое фиксированное число раз (кратность развертки):

```
DO K = 2, N, 4
  DO I = K, K + 3
    DO J = 2, N
      A(I, J) = SQRT(A(I, J - 1) *
        * A(I - 1, J))
```

(4)

Выбираем кратность развертки, равную четырем, и для простоты считаем, что N - 1 кратно четырем. Далее меняем местами новый короткий цикл и прежний внутренний цикл:

```
DO K = 2, N, 4
  DO J = 2, N
    DO I = K, K + 3
      A(I, J) = SQRT(A(I, J - 1) *
        * A(I - 1, J))
```

Если попытаться выписать новые дистанции зависимостей, то столкнемся с тем фактом, что одна зависимость стала неоднородна: зависимость от вычисления A(I - 1, J) имеет разный вектор дистанции для первой итерации цикла по I и для последующих. Однако это обстоятельство ничем далее не мешает, так как следующим шагом внутренний цикл будет полностью развернут, после чего однородность восстановится (рис. 1).

В новом графе зависимостей можно обнаружить пять контуров. Среди них один имеет сум-

```

DO K = 2, N, 4
  DO J = 2, N
    ! DO I = K, K+3
    A((K+0), J) = &
      SQRT(A((K+0), J-1) * A((K+0) - 1, J))
    A((K+1), J) = &
      SQRT(A((K+1), J-1) * A((K+1) - 1, J))
    A((K+2), J) = &
      SQRT(A((K+2), J-1) * A((K+2) - 1, J))
    A((K+3), J) = &
      SQRT(A((K+3), J-1) * A((K+3) - 1, J))
  
```

Рис. 1. Зависимости в примере (4) после развертки внешнего цикла

марную дистанцию (1, 0) и не требует учета. Остальные связаны с зависимостью от вычисления $A(I, J - 1)$ и имеют те же значения латентности и дистанции, как и в первоначальном цикле. Отсюда получаем, что $RecII$ не изменился. Вместе с тем, операций в новом цикле стало в 4 раза больше. Таким образом, если предположить, что раньше значение $RecII$ превышало $ResII$ не более чем в 4 раза, то в новом цикле оно уже не ограничивает конвейеризацию, что и требовалось.

В первоначальном цикле требовалось зачитать две строчки кэша на четыре итерации. В развернутом цикле требуется пять строчек кэша на четыре итерации, равные 16 первоначальным итерациям. Таким образом, получили выигрыш около 37 %.

Сложности с перестановкой циклов

Попробуем, действуя аналогичным образом, развернуть внешний цикл в примере

```

DO I = 2, N
  DO J = 2, N
    A(I, J) = SQRT(A(I, J - 1) *
      * A(I - 1, J + 1))
  
```

Мы столкнемся с тем, что цикл, который должен быть развернут, нельзя переставить с циклом по J . Причина в том, что "столбик" из четырех итераций зависит от обоих соседних столбиков. Формально затруднение возникает потому, что дистанция зависимости по $A(I - 1, J + 1)$ из (0, 1, -1) превращается в (0, -1, 1), а этот вектор — лексикографически отрицателен. Как следствие, преобразование некорректно.

Метод гиперплоскостей подсказывает естественное решение — обходить по диагоналям полосу в пространстве итераций, соответствующую одной итерации цикла по K . При этом полоска разбивается на три области, соответствующие трем последовательным циклам.

Треугольники можно обходить в любом корректном порядке, а параллелограмм — по диагональным слоям (рис. 2). Цикл обхода слоя оказывается внутренним и может быть развернут. Получаем цикл, готовый к конвейеризации.

При реализации этого подхода возникают некоторые сложности, а именно — непонятно, из

каких соображений следует выбирать слой (рис. 3). Минимальное значение II при конвейеризации простого цикла, как известно, не меньше $ResII$ и $RecII$. Однако оптимальный II не обязательно равен максимуму этих оценок. Длительность II может зависеть от структуры зависимостей между операциями, даже если $RecII < ResII$.

Чем меньше зависимостей (и чем больше их дистанции), тем больше свобода в расположении команд и, следовательно, больше возможностей оптимально использовать ресурсы процессора. Вместе с тем, излишнее увеличение дистанции ведет либо к использованию большего числа регистров, либо к удлинению пролога и эпилога цикла — оба варианта нежелательны.

Однако, и это следует отметить, оказывается, что эта задача может быть решена самим методом конвейеризации. Выполним (формально некорректную) операцию перестановки циклов с последующей разверткой, сохраняя при этом старую структуру зависимостей. В DFG векторы дистанций после перехода к новым координатам могут стать лексикографически отрицательными (рис. 4, 5).

Затем запустим процедуру конвейеризации. Скалярные дистанции зависимостей для конвейеризации — младшие координаты векторных дистанций с нулевыми остальными координатами. Тот факт, что некоторые из них отрицательны, заставит конвейеризатор правильно выбрать итерации исходного цикла, из которых возможно исполнение команд в итерации конвейеризованного, т. е. самому построить оптимальный слой.

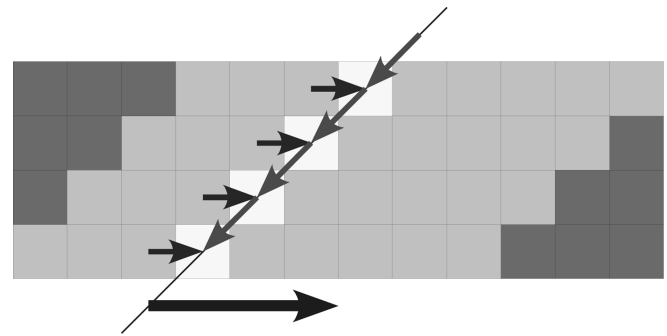


Рис. 2. Правильный обход полоски в пространстве итераций

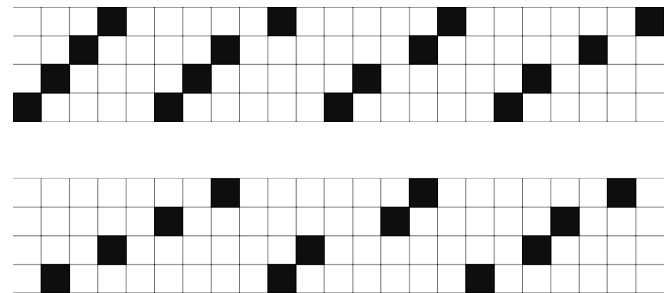


Рис. 3. Варианты слоев

```

DO K = 2, N, 4
  DO J = 2, N
    ! DO I = K, K+3
    A((K+0), J) = &
      Sqrt(A((K+0), J-1) * A((K+0) - 1, J+1))
    A((K+1), J) = &
      Sqrt(A((K+1), J-1) * A((K+1) - 1, J+1))
    A((K+2), J) = &
      Sqrt(A((K+2), J-1) * A((K+2) - 1, J+1))
    A((K+3), J) = &
      Sqrt(A((K+3), J-1) * A((K+3) - 1, J+1))

```

Рис. 4. Зависимости в развернутом цикле для примера (5)

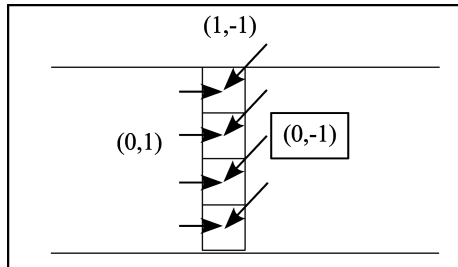


Рис. 5. Дистанция зависимостей в развернутом цикле для примера (5)

При этом треугольники, которые возникали при разбиении полоски на три области, — не что иное, как пролог и эпилог конвейеризованного цикла. Это соответствие приблизительно, так как конвейеризатор оперирует командами, а не итерациями исходного цикла.

Таким образом, алгоритм программной конвейеризации удачно дополняет развертку внешнего цикла в гнезде циклов при условии, что этот алгоритм способен обрабатывать DFG с отрицательными дистанциями.

До сих пор мы неявно предполагали, что гнездо циклов состоит из двух циклов — внутреннего (который в конечном итоге конвейеризируется) и внешнего, который можно развернуть. Вместе с тем, при наличии трех или более вложенных циклов возникает ряд дополнительных вопросов. Какой цикл лучше развертывать, если в гнезде циклов более одного объемлющего цикла? Имеет ли смысл развертывать более одного объемлющего цикла? На последний вопрос ответ, по-видимому, утвердительный. Например, в гнезде циклов

```

DO K = 2, N
  DO I = 2, N
    DO J = 2, N
      A(K, I, J) = Sqrt(A(K, I - 2,
        J + 1) * A(K - 2, I, J + 1))

```

имеем зависимости с дистанциями (0, 2, -1) и (2, 0, -1). При развертке любого из циклов по K или по I в 4 раза возникнет описанная выше проблема с перестановкой этого цикла и цикла по J (хотя решение этой проблемы известно, но оно может иметь отрицательные последствия, например, в виде использования большего числа регистров).

Если же развернуть оба цикла по K и по I в 2 раза (в целом в те же 4 раза), то, как можно убедиться, зависимости не мешают переставлять циклы.

Еще важнее то, что если объемлющих циклов несколько, они могут не переставляться между собой. Рассмотрим следующий пример:

```

DO K = 2, N
  DO I = 2, N
    DO J = 2, N
      A(K, I, J) = Sqrt(A(K - 1,
        I + 1, J))

```

Здесь всего одна зависимость с дистанцией (1, -1, 0). Очевидно, что она не позволяет переставить местами циклы по K и I. Таким образом, если по тем или иным причинам желательно развернуть цикл по K, то, заменив цикл по K на два цикла, мы не сможем сделать новый цикл внутренним, так как он не переставляется с циклом по I.

Если развертываемый цикл не переставляем хотя бы с одним из промежуточных циклов (не с внутренним, но с вложенным в развертываемый), можно поступать так же, как предлагалось выше — использовать метод, родственному методу гиперплоскостей внутри соответствующей полоски. Однако снова возникает задача выбора слоя. Опять попробуем не решать эту задачу самим, а переложить ее решение на алгоритм конвейеризации. Для этого задачу потребуется немного обобщить.

Будем считать, что развертываются все объемлющие циклы (для неразвертываемых циклов кратность развертки единичная). Выделяем в пространстве итераций параллелепипед со сторонами, равными кратности развертки соответствующего цикла, — "ячейку развертки". Мы не можем во внутреннем цикле просто исполнять все итерации исходного цикла, соответствующие одной ячейке. Нам помешают те зависимости, которые не позволяли переставлять циклы. Однако можно снабдить все итерации одной ячейки векторами сдвига, определяющими, в какой итерации развернутого цикла относительно некоторой текущей должна выполняться данная итерация исходного цикла. Заметим, что слой, т. е. множество итераций, которые можно исполнять в теле внутреннего цикла, соответствует ячейке по модулю кратности развертки. Это означает, что если слой "намотать" на ячейку, она будет покрыта вся ровно по одному разу.

Далее строим общий DFG-граф для всех команд всех итераций, входящих в одну ячейку. Дистанции между командами будут векторными и зависящими от неопределенных сдвигов между итерациями. Теперь мы уже не можем использовать любой из удобных алгоритмов конвейеризации, так как они не способны работать с неопределенными и, к тому же, векторными дистанция-

ми. Однако среди методов конвейеризации есть подход, основанный на формулировке ее в виде задачи целочисленного линейного программирования (ЦЛП) [3].

ЦЛП-подход можно обобщить на случай векторных дистанций. Важно, что добавленные неопределенные дистанции оставят все неравенства в ЦЛП-формулировке линейными, так как дистанции в них всегда входят как свободные члены, а не в коэффициенты при переменных.

Теперь решение ЦЛП-задачи автоматически даст не только оптимальный конвейеризованный цикл, но и множество итераций исходного цикла, которые должны исполняться вместе, т. е. искомым слоем, наилучшим образом подходящий для конвейеризации. Разумеется, следует озаботиться тем, чтобы этот слой получился возможно "плотнее", чтобы уменьшить прологи и эпилоги циклов. Достичь этого можно как явными ограничениями на сдвиги (возможно, жертвуя оптимальностью), так и заданием правильно подобранной целевой функции.

Данный подход представляется очень интересным, однако пока он недостаточно исследован. Преждевременно утверждать, что этот подход окажется эффективен. Настороженность вызывает тот факт, что кроме увеличения числа переменных в ЦЛП-задаче в нее добавляются большие коэффициенты, соответствующие интервалам запуска (II) объемлющих циклов (они берутся произвольными, но достаточно большими). Последнее обстоятельство плохо сказывается на работе алгоритмов, используемых в целочисленном линейном программировании.

Важный аспект, требующий дополнительного изучения, заключается в том, что можно снабдить неопределенными векторными сдвигами не итерации исходного цикла, входящие в одну ячейку развертки, а все отдельные команды. Это, с одной стороны, усложнит задачу, увеличив число переменных, а с другой стороны, возможно, позволит найти более удачный конвейеризованный цикл. В этом варианте данный подход можно с полным основанием назвать "многомерной конвейеризацией".

Заключение

Осталось ответить на несколько практических вопросов. Во сколько раз следует развертывать внешний цикл? Понятно, что общая кратность развертки должна быть не меньше чем $RecII/ResII$, чтобы в развернутом теле цикла $RecII$ был не больше $ResII$. Однако вопрос, "насколько не меньше?" — остается.

Далее, если объемлющих циклов более одного, как именно следует разложить общую кратность развертки в произведение кратностей разверток

отдельных циклов? Чтобы ответить на эти вопросы, надо оценить все последствия использованной развертки (или разверток) для последующей конвейеризации внутреннего цикла. В целом, это пока представляется непростой задачей. Напрашивается экспериментальный подход: перебрать все разумные варианты, выполняя конвейеризацию и оценивая результат.

Здесь возможен адаптивный подход [11], когда компилятор сгенерирует несколько вариантов кода для цикла, а во время выполнения вариант выбирается случайным образом и измеряется время его работы. Впоследствии выбирается оптимальный вариант или варианты.

Более традиционный путь заключается в статической оценке полученного кода в компиляторе. Очевидно, что чем меньше отношение II к кратности развертки, тем лучше (меньше тактов требуется на итерацию исходного цикла). Вместе с тем, необходимо учитывать эффективность работы кэш-памяти. Как оценивать вариант трансляции сразу по обоим критериям, выраженным в разных терминах и единицах?

Удачное решение предложено в работе [12]. Выше, при анализе преобразований циклов, вручную оценивались обращения в память с точки зрения попадания или непопадания в кэш данных. Этот процесс несложно формализовать, и тогда компилятор сможет для разных обращений в память учитывать разную латентность в зависимости от того, будет здесь промах или нет, или как часто он здесь будет. Оптимальный конвейеризованный вариант строится уже с учетом этих разных латентностей. Вычислив его II , мы уже учли оба критерия. Разумеется, кроме оценки мы получим и лучший код по сравнению с тем, при построении которого считали латентности всех обращений в память одинаковыми.

Осталось заметить, что для обращений в память существенна не только латентность промахов в кэш, но и их общее число. Даже в том случае, когда за счет удачной конвейеризации удалось запланировать все обращения в память, вызывающие промахи, таким образом, чтобы они не должны были бы задерживать вычисления, — механизм работы с памятью может не справиться с таким числом обращений. Выход видится в том, что пропускную способность памяти следует учитывать как ресурс.

Список литературы

1. Allan V., Jones R., Lee R., Allan S. Software pipelining // ACM Computing Surveys. September, 1995. Vol. 27. N 3. P. 367—432.
2. Вьюкова Н. И., Галатенко В. А., Самборский С. В. Программная конвейеризация циклов методом планирования по модулю // Программирование. 2007. № 6. С. 14—25.
3. Самборский С. В. Формулировка задачи планирования линейных и циклических участков кода // Программные продукты и системы. 2007. № 3. С. 12—16.

4. **Lamport L.** The Parallel execution of DO loops. // Communications of ACM. 1974. Vol. 17. N 2. P. 83—93.

5. **Ramanujam J.** Optimal software pipelining of nested loops. // Proceedings of the International Parallel Processing Symposium. 1994. P. 335—342.

6. **Darte A., Robert Y.** Constructive Methods for Scheduling Uniform Loop Nests // IEEE Trans. on Parallel Distributed Systems. 1994. Vol. 8. N 5. P. 814—822.

7. **Darte A., Robert Y.** Affine-by-statement scheduling of uniform and affine loop nests over parametric domains // Journal of Parallel and Distributed Computing. 1995. Vol. 29. Issue 1. P. 43—59.

8. **Douillet A., Rong H., Gao G. R.** Multi-Dimensional Kernel Generation for Loop Nest Software Pipelining // Proceedings of Euro-Par 2006 Parallel Processing. 2006. P. 311—322.

9. **Rong H., Tang Z., Govindarajan R., Douillet A., Gao G. R.** Single-Dimension Software Pipelining for Multidimensional Loops // ACM Transaction on Architecture and Code Optimization. March 2007. Vol. 4. N 1. Article 7.

10. **Carr S., Ding C., Sweany P.** Improving Software Pipelining with Unroll-and-Jam // Proceedings of the 29th Hawaii International Conference on System Sciences (HICSS'96). Vol. 1: Software Technology and Architecture HICSS'96. January 1996. P. 183—192.

11. **Chuang P., Chen H. C. et al.** Dynamic Profile Driven Code Version Selection // Phoenix, Arizona, USA. Proceedings of the 11th Workshop on Interaction between Compilers and Computer Architecture, February, 2007. P. 74—84.

12. **Ding C., Carr S., Sweany P.** Modulo Scheduling with Cache Reuse Information // Proceedings of the Third International Euro-Par Conference on Parallel Processing Euro-Par'97. 1997. P. 1079—1083.

УДК 004.412+032.26

В. В. Лаптев, канд. техн. наук, доц.,
e-mail: Laptev@astu.org,

А. В. Морозов, начальник отдела АСУ,
e-mail: morozov@astu.org,

Астраханский государственный технический
университет

Экспертная система оценки качественных характеристик проектной модели программного обеспечения

Рассматривается подход к автоматизации задачи повышения качества программного обеспечения. Выделен ряд качественных характеристик программного продукта, значительно влияющих на общее качество. Предлагается автоматизировать их оценку на основе вычисления метрик проектной модели предметной области. Задача определения характеристик по метрикам сведена к задаче классификации. Оценка характеристик на стадии проектирования позволяет своевременно уточнять и улучшать проектную модель, что в конечном итоге существенно снижает затраты на разработку программного продукта.

Ключевые слова: программное обеспечение, проектная модель, качественные характеристики, метрики, нечеткая классификация, нейронная сеть.

Обеспечение высокого качества при разработке программных продуктов — одна из сложнейших проблем. Качественные характеристики программного продукта можно разбить на два множества: внутренние и внешние. Внешние характеристики определяют потребительское качество системы и видимы пользователю. Внутренние характеристики более важны для разработчиков, так как оказывают существенное влияние на потре-

бительское качество и экономические затраты при разработке.

К внешним характеристикам относятся в первую очередь безопасность, функциональность, производительность, эргономичность, надежность. Внутренними характеристиками являются переносимость, модифицируемость, модульность, подверженность ошибкам, понятность, удобство сопровождения и т. д.

Задача определения качественных характеристик может ставиться на разных стадиях разработки. Наиболее просто она может быть решена на завершающих этапах, когда в наличии имеется уже готовый программный продукт или большая его часть. Внешнее качество продукта оценивается путем тестирования, при котором осуществляются непосредственные наблюдения и измерения внешних характеристик. Оценка внутренних качественных характеристик является значительно более сложной задачей, поскольку для этого нужно выполнять инспекцию исходного кода. Эту работу обычно выполняет группа высококвалифицированных экспертов.

Однако оценка качественных характеристик на завершающих этапах разработки, по сути, является просто констатацией достигнутого уровня качества. Значительно более важной задачей является определение, а точнее, прогнозирование качественных характеристик продукта на более ранних этапах разработки, в частности, на этапе проектирования. Актуальность этой задачи очевидна, поскольку оценка и своевременное уточнение и улучшение характеристик будущего продукта могут существенно снизить затраты на производство программных продуктов.

Внутренние характеристики разрабатываемого продукта можно частично оценить на основе разработанной проектной модели. На текущий момент в индустрии информационных технологий данная задача решается только высококвалифицированными экспертами — проектировщиками

и учеными, которые, основываясь на собственном опыте и представлениях, делают эмпирические выводы о характеристиках проекта. Далеко не в каждую команду разработчиков может войти такой эксперт, поэтому чаще всего компании покупают консалтинговые услуги в этой области.

Автоматизация оценки качества проектной модели позволила бы сократить расходы на подобный консалтинг, а также помогла бы осуществлять его удаленно.

Для решения задачи оценки качественных характеристик проектной модели предлагается ограничить множество характеристик следующими:

- **сложность реализации** (*complexity*, CPLX(X)) — отражает объем затрат материальных и человеческих ресурсов, необходимых для первоначальной реализации проектной модели;
- **сложность модернизации** (*volatily*, VLTL(X)) — отражает объем затрат материальных и человеческих ресурсов, которые потребуются в случае развития функциональности программного продукта;
- **сложность сопровождения** (*maintenance*, MNTN(X)) — отражает объем затрат материальных и человеческих ресурсов, необходимых для поддержания в рабочем состоянии программного продукта без изменения его функциональности;
- **подверженность ошибкам** (*weakness*, WCK(M)) — отражает число ошибок, которые могут быть допущены разработчиками при реализации программного продукта, основывающегося на анализируемой проектной модели;
- **возможность повторного использования** (*recycling*, RCCL(M)) — отражает степень готовности проектной модели к повторному использованию в случае реализации другого программного продукта в той же предметной области.

Данные качественные характеристики выделены в результате литературного обзора источников [1, 5, 6, 7] как наиболее актуальные на этапе создания проектной модели. Каждая из перечисленных характеристик не вычисляется количественно, а определяется некоторой лингвистической величиной (хорошо, допустимо, плохо и т. д.) Поэтому в рамках предлагаемого подхода характеристики предлагается определять как лингвистические переменные с фиксированным набором термов.

Предлагаемые к рассмотрению характеристики проектной модели программного обеспечения оказывают значительное влияние друг на друга. Так, излишняя сложность реализации зачастую является причиной высокой подверженности ошибкам, низкой сопровождаемости и модифицируемости. Модифицируемость в свою очередь влияет на факторы возможности повторного использования кода и сопровождаемости. Диаграм-

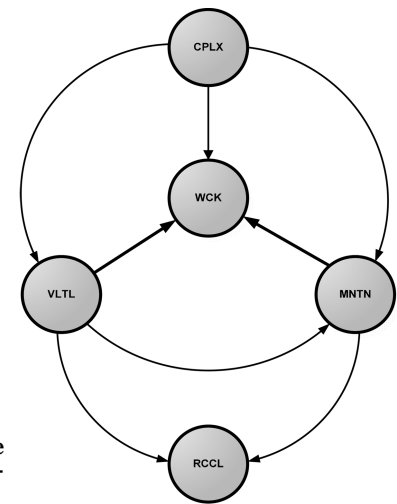


Рис. 1. Взаимное влияние качественных характеристик проектной модели

ма взаимного влияния характеристик показана на рис. 1. Однако необходимо отметить, что связь между характеристиками не является чисто причинно-следственной.

Как уже упоминалось выше, ни одна из предлагаемых к рассмотрению характеристик не может быть оценена формальными методами непосредственно из проектной модели. В данной работе предлагается автоматизированный подход к их оценке на основе метрик проектной модели. Метрики проектной модели — это некоторые статистические показатели, рассчитываемые непосредственно на основе знаний о ее структуре. На сегодняшний день разработано достаточное число наборов метрик для программных проектов и по многим из них уже собрана достаточная историческая и аналитическая информация для различных типов проектов. Примерами широко используемых наборов метрик являются QMOOD (*Quality Model for Object-Oriented Design*), MOOSE (*Metrics for Object-Oriented Software Engineering*), MOOD (*Metrics for Object-Oriented Design*) и MOOD 2 [1, 8, 9].

Задачу определения характеристик на основе вычисленных метрик проектной модели предметной области можно свести к задаче классификации, которая, однако, имеет следующие особенности.

1. Объект классификации представлен вектором четких величин — значений метрик проектной модели программного обеспечения.

2. Результат решения задачи — лингвистическая величина — значение характеристики.

3. Не определены однозначные зависимости между значениями метрик проектной модели и ее характеристиками.

4. Существует ограниченное количество доступных в открытом виде данных о метриках проектов и оценках их успешности.

Для решения поставленной задачи предлагается применить подход, основанный на концепции нечетких нейронных продукционных сетей. В качестве ос-

новой выбрана структура сети, предложенная Вангом и Менделем (*L.-X. Wang, J. M. Mendel*) [2—4].

Нечеткая продукционная сеть Ванга—Менделя реализует нечеткую продукционную модель, основанную на правилах с MISO-структурой:

*Если x_1 есть A_{i1} И ... И x_j есть A_{ij} И ...
И x_m есть A_{im} ТО y есть B_i , $i = 1, \dots, n$.*

Кроме того, выполняемый данной нечеткой нейронной продукционной сетью алгоритм нечеткого вывода базируется на следующих положениях.

1. Входные переменные являются четкими.
2. Функции принадлежности всех нечетких множеств представляются функцией Гаусса [2].
3. Нечеткая импликация — нечеткое произведение.
4. Т-норма — нечеткое произведение.
5. Аккумулятивное активизированных заключений правил не проводится.
6. Метод дефаззификации — средний центр.

Структура применяемой нечеткой нейронной сети представлена четырьмя слоями (рис. 2).

Слой 1 выполняет фаззификацию входных переменных x_j ($j = 1, \dots, n$). Элементы этого слоя вычисляют значения функций принадлежности $\mu_{A_{ij}}(x_j')$, заданных гауссовыми функциями принадлежности с параметрами a_{ij} и b_{ij} .

В слое 2, число элементов которого равно числу правил в базе, осуществляется агрегирование степеней истинности предпосылок соответствующих правил.

В слое 3 первый элемент служит для активизации заключений правил c_i в соответствии со значениями агрегированных на предыдущем слое степеней истинности предпосылок правил. Второй элемент

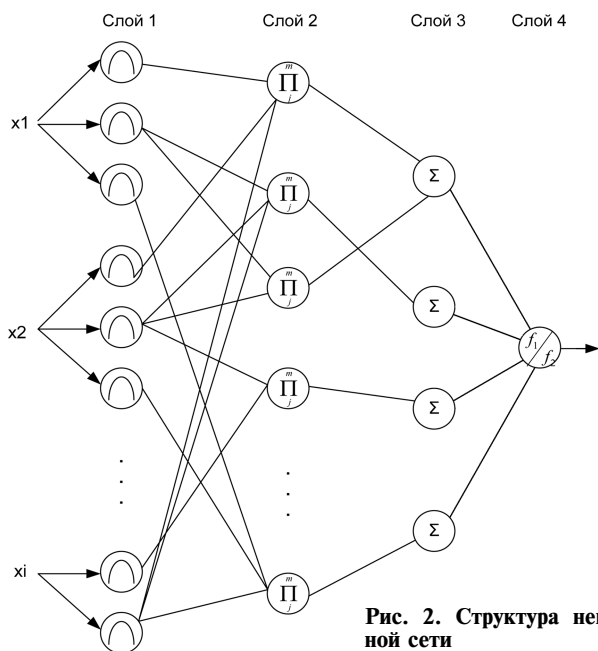


Рис. 2. Структура нейронной сети

слоя проводит вспомогательные вычисления для последующей дефаззификации результата.

Слой 4, состоящий из одного элемента, выполняет дефаззификацию выходной переменной.

Параметрическими слоями сети являются слои 1 и 3, а настраиваемыми параметрами служат соответственно параметры термов посылок и заключений a_{ij} , b_{ij} и c_i .

Таким образом, нечеткую нейронную модель и механизм нечеткого вывода для данной нечеткой нейронной продукционной сети можно представить следующим образом:

$$\begin{aligned} \mu_{B_i'}(y) &= \sup_{x \in X} \{ \mu_{A_i'}(x) T \mu_{A_i \rightarrow B_i}(y) \} = \\ &= \sup_{x \in X} \{ \mu_{A_i'}(x) \mu_{A_i \rightarrow B_i}(y) \} = \\ &= \sup_{x \in X} \{ \mu_{A_i'}(x) \mu_{A_{ij}}(x) \mu_{B_i}(y) \} = \\ &= \sup_{x \in X} \left\{ \mu_{B_i}(y) \prod_{j=1}^m (\mu_{A_{ij}'}(x_j) \mu_{A_{ij}}(x_j)) \right\}. \end{aligned}$$

В данном случае входные переменные x_1, x_2, \dots, x_m являются четкими, поэтому выражение принимает следующий вид:

$$\mu_{B_i'}(y) = \mu_{B_i}(y) \prod_{j=1}^m \mu_{A_{ij}'}(x_j').$$

Так как аккумулятивное активизированных заключений правил не проводится, а методом дефаззификации является метод среднего центра, то дефаззифицированное значение выходной переменной определяется по формуле

$$\begin{aligned} y' &= \frac{\sum_{i=1}^n (\arg \max_y \mu_{B_i'}(y) \mu_{B_i}(y))}{\sum_{i=1}^n \mu_{B_i'}(y)} = \\ &= \frac{\sum_{i=1}^n \left(\arg \max_y \mu_{B_i'}(y) \mu_{B_i}(y) \prod_{j=1}^m \mu_{A_{ij}'}(x_j') \right)}{\sum_{i=1}^n \left(\mu_{B_i}(y) \prod_{j=1}^m \mu_{A_{ij}'}(x_j') \right)}. \end{aligned}$$

С учетом того, что максимальное значение, которое $\mu_{B_i'}(y)$ может принять в точке $\arg \max_y \mu_{B_i}(y)$, равно единице, выражение приводится к следующему виду:

$$y' = \frac{\sum_{i=1}^n \left(\arg \max_y \mu_{B_i}(y) \prod_{j=1}^m \mu_{A_{ij}'}(x_j') \right)}{\sum_{i=1}^n \prod_{j=1}^m \mu_{A_{ij}'}(x_j')}.$$

В нашем случае функции принадлежности всех нечетких множеств представляются функцией Гаусса, поэтому выражение примет вид:

$$y' = \frac{\sum_{i=1}^n \left(\arg \max_y \left(e^{-\frac{(y-c_i)^2}{d_i}} \right) \prod_{j=1}^m e^{-\frac{(x'_j - a_{ij})^2}{b_{ij}}} \right)}{\sum_{i=1}^n \prod_{j=1}^m e^{-\frac{(x'_j - a_{ij})^2}{b_{ij}}}} = \frac{\sum_{i=1}^n \left(c_i \prod_{j=1}^m e^{-\frac{(x'_j - a_{ij})^2}{b_{ij}}} \right)}{\sum_{i=1}^n \prod_{j=1}^m e^{-\frac{(x'_j - a_{ij})^2}{b_{ij}}}}.$$

Здесь c_i , d_i — соответственно центры и ширина гауссовых функций, представляющих функции принадлежности нечетких множеств B_i заключенных правил; a_{ij} , b_{ij} — соответственно центры и ширина гауссовых функций, представляющих функции принадлежности нечетких множеств A_{ij} предпосылок правил.

Обучение нейронной сети осуществляется методом обратного распространения ошибки, путем корректировки параметров первого и третьего слоев.

Предложенный подход позволит осуществить автоматизированную оценку характеристик программного продукта на ранних этапах разработки, что, в свою очередь, упрощает и позволяет повысить эффективность принимаемых при разработке программного обеспечения решений, а также существенно уменьшить стоимость разрабатываемого программного продукта.

Список литературы

1. Орлов С. А. Современные технологии конструирования сложных программных систем. — Рига: ИТС, 1999. — 289 с.
2. Круглов В. В., Дли М. И., Голунов Р. Ю. Нечеткая логика и искусственные нейронные сети. М.: ФИЗМАТЛИТ, 2001. 224 с.
3. Борисов В. В., Круглов В. В., Федулов А. С. Нечеткие модели и сети. — М.: Горячая линия—Телеком, 2007. — 284 с.
4. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. — М.: Горячая линия—Телеком, 2007. — 452 с.
5. Макконелл С. Совершенный код. Мастер-класс / Пер. с англ. — М.: Русская редакция; СПб.: Питер, 2005.
6. Басс Л., Клеменс П., Кацман Р. Архитектура программного обеспечения на практике. — СПб.: Питер, 2006.
7. Константайн Л., Локвуд Л. Разработка программного обеспечения. — СПб.: Питер, 2004.
8. Capers J. Strength and weakness of software metrics — <http://www.compaid.com/caiinternet/ezone/capers-StrngWk.pdf>.
9. Baroni A. Formal Definition of Object-Oriented Design Metrics. Master Thesis. Vrije University, Brussel, Belgium, 2002.

УДК 002.55

А. С. Зуев, канд. техн. наук, ст. преподаватель; Ю. И. Петров, студент,
Московский государственный университет приборостроения и информатики,
e-mail: zuev_andrey@mail.ru

Описание модификации строки адреса проводника Windows Explorer*

Представлено описание модели, реализующей модификацию строки адреса проводника операционной системы Windows и позволяющей упростить навигацию по логическим дискам и папкам. Приведены результаты сравнения модели с аналогичными элементами интерфейса других приложений, обоснована ее конкурентоспособность по сравнению с аналогами.

Ключевые слова: человек-компьютерное взаимодействие, графический пользовательский интерфейс, эргономика программного обеспечения, проектирование графических интерфейсов, оптимизация графических интерфейсов, модификация адресной строки, контекстная интерактивная подсказка, навигация по логическим дискам и каталогам, навигация в сети Интернет, окно операционной системы Windows.

Введение

Одной из основных тенденций развития программного обеспечения является повышение эргономичности и удобства работы пользователей. Примером могут являться результаты деятельности

корпорации Microsoft, доказывающие, что качество и эффективность графических пользовательских интерфейсов являются важными конкурентными преимуществами программных продуктов. Описываемая в статье разработка LineExplorer является моделью, в которой реализованы функциональные возможности, повышающие эргономичность работы пользователя с окнами операционной системы (ОС), сокращающие трудоемкость и за-

*Работа выполнена при финансовой поддержке грантов Президента РФ молодым российским ученым-кандидатам наук (грант № МК-948.2008.9)

траты времени на доступ к папкам, а также снижающие необходимость работы с несколькими окнами. Представляемая модель не является дополнением к ОС Windows, но позволяет оценить удобство работы с ней и сделать выводы о целесообразности реализации ее функциональных возможностей в окнах операционной системы.

Основными дополнительными функциональными возможностями модели LineExplorer являются:

- интерактивная контекстная подсказка, способствующая эффективному вводу адреса папки и позволяющая выполнять автоматическую сортировку имен папок в адресе по вводимым с клавиатуры символам;
- поиск файлов и папок непосредственно из строки адреса в местоположении, определяемом введенным адресом;
- система закладок, позволяющая настроить на быстрый переход к любым папкам без непосредственного ввода их адресов;
- возможность навигации в сети Интернет.

Описание функциональных возможностей модели

Интерфейс модели LineExplorer представлен на рис. 1, ключевым элементом, реализующим основную часть ее функциональных возможностей, является *Адресная панель*. Основные настройки модели представлены в разделе *Инструменты* главного меню и в дополнительном окне, вызываемом нажатием правой клавиши мыши на метке *Адрес*.

Заголовок окна, панель навигации, файловый браузер и строка состояния соответствуют типовым элементам интерфейса окон ОС Windows. Адресная панель содержит опции, являющиеся новшеством, преимуществом и отличительной особенностью модели LineExplorer. В таблице приведено краткое описание назначения элементов интерфейса адресной панели.

Одной из основных особенностей модели LineExplorer является реализация в процессе ввода адреса контекстной интерактивной подсказки, представленной последовательными контекстными списками имен папок с автоматической сортировкой их содержания по вводимым с клавиатуры символам. В результате в контекстной подсказке отображаются каталоги, доступные относительно произвольного положения курсора в адресе папки. Проиллюстрируем возможности, реализованные в контекстной подсказке.

1. Визуальное отображение содержимого папки (имен вложенных папок), адрес которой введен в строку адреса, в виде контекстного списка (рис. 2).

2. Автоматическая сортировка имен папок в контекстных списках по вводимым с клавиатуры символам (рис. 3).

3. Отображение контекстной подсказки в произвольном положении курсора в адресе (рис. 4).

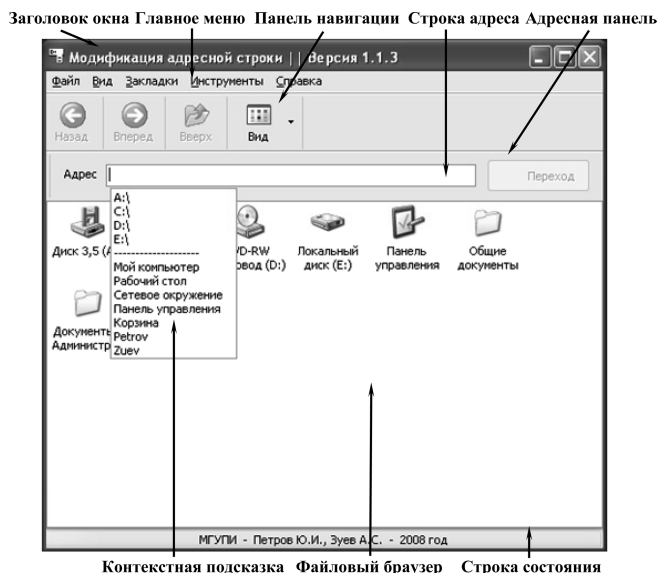


Рис. 1. Интерфейс модели LineExplorer

Элементы интерфейса адресной панели

Элемент	Описание
	Метка на адресной панели — нажатие правой клавиши мыши вызывает дополнительное окно с настройками программы
	Строка адреса папки — реализует навигацию по файловой системе и сети Интернет, а также поиск папок и файлов
	Кнопка запуска проводника ОС Windows

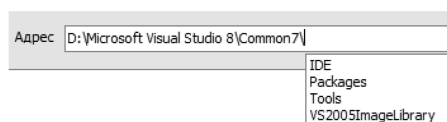


Рис. 2. Контекстный список имен вложенных папок

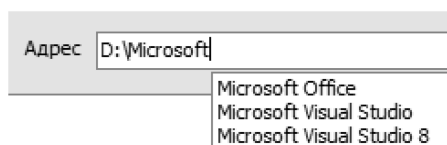


Рис. 3. Пример сортировки содержания контекстного списка

Работа с моделью LineExplorer может заключаться в вводе адреса папки, навигации в сети Интернет, выполнении действий с закладками, организации поиска файлов и папок, а также в изменении настроек. Функциональность модели ориентирована на повышение удобства формирования адреса папки в файловой системе, при этом возможно использование как клавиатуры, так и мыши, в том числе сочетание данных средств манипулирования.

Для ввода адреса папки с помощью клавиатуры требуется:

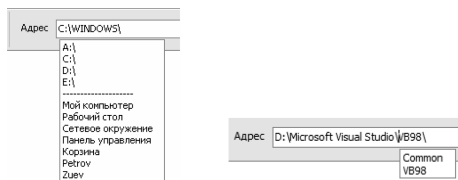


Рис. 4. Примеры контекстной подсказки

- установить курсор в поле строки адреса с помощью клавиши *Tab*;
- перейти к контекстной подсказке нажатием клавиш-стрелок *Вниз* или *Вправо* и выбрать требуемый элемент из открывшегося контекстного списка с помощью клавиш-стрелок и клавиши *Enter*;
- продолжить ввод адреса — выбирать элементы из последовательно открывающихся контекстных списков с помощью нажатия клавиш-стрелок и клавиши *Enter* либо воспользоваться возможностью автоматической сортировки их содержания по вводимым с клавиатуры символам.

Для ввода адреса папки с помощью мыши требуется:

- установить указатель курсора мыши в поле строки адреса;
- в открывшемся контекстном списке выбрать требуемый элемент с помощью указателя курсора мыши нажатием ее левой клавиши;
- продолжить ввод адреса — выбирать элементы из последовательно открывающихся контекстных списков с помощью указателя курсора мыши.

Реализованная в модели LineExplorer система закладок позволяет настроить возможность быстрого перехода к наиболее востребованным папкам без непосредственного ввода их адресов. По умолчанию предусмотрено пять закладок, соответствующих системным папкам ОС Windows: *Мой компьютер*, *Рабочий стол*, *Сетевое окружение*, *Панель управления* и *Корзина*. Данные закладки отображаются в первом контекстном списке строки адреса после логических дисков (см. рис. 4), для них не доступны опции переименования, удаления и изменения порядка перечисления. Список закладок отображается также в разделе главного меню *Закладки* (рис. 5).

Для управления закладками предназначено окно *Менеджер закладок* (рис. 6), вызываемое при выборе пункта главного меню *Закладки* → *Управление закладками...*, или нажатием горячей клавиши *F9*.

Окно *Менеджер закладок* содержит список их имен и адресов, предоставляет возможности добавления, редактирования и удаления, а также изменения порядка отображения закладок в контекстной подсказке с помощью соответствующих кнопок в нижней части окна. Максимальное число создаваемых закладок — 10, запрещено добавлять закладки с одинаковыми именами или адре-

сами, длина имени закладки ограничена 30 символами.

Для добавления закладки требуется выполнить следующие действия:

- нажатием кнопки *Добавить* вызвать окно *Добавление закладки* (рис. 7);
- ввести имя закладки, адрес папки и нажать *OK*. Нажав кнопку "...", можно задать адрес с помощью стандартного диалогового окна.

Для добавления закладки, адрес которой указан в строке адреса, требуется выполнить следующие действия:

- нажать горячую клавишу *F2* либо вызвать контекстное меню нажатием правой клавиши мыши на поле строки адреса и выбрать пункт *Добавить закладку* (рис. 8);

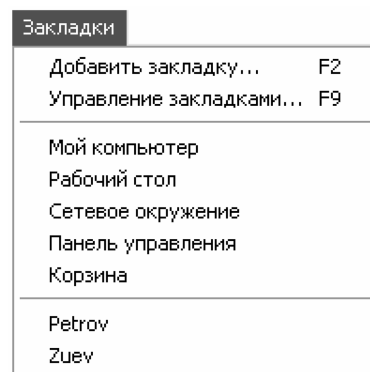


Рис. 5. Раздел главного меню *Закладки*

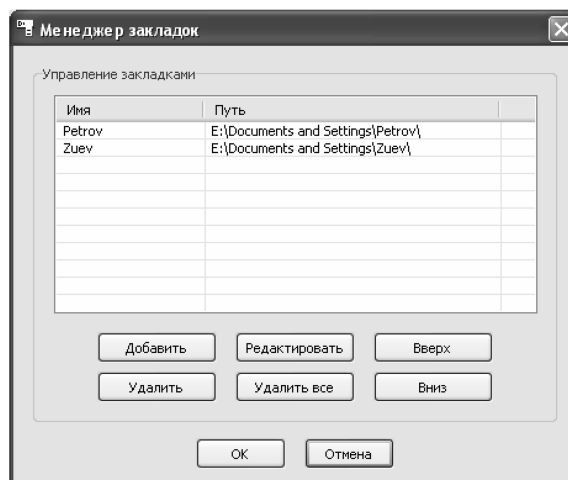


Рис. 6. Окно *Менеджер закладок*

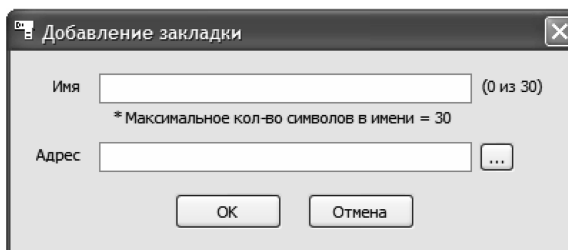


Рис. 7. Окно *Добавление закладки*

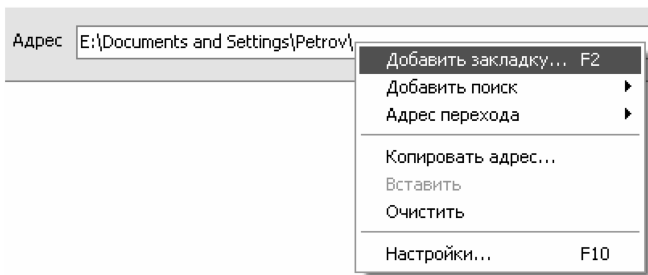


Рис. 8. Контекстное меню строки адреса

- в появившемся окне *Добавление закладки* (см. рис. 7) ввести имя закладки и нажать *ОК* (поле *Адрес* будет заполнено автоматически).

В модели LineExplorer реализована специальная функция *Адрес перехода*, позволяющая в текущем сеансе работы хранить в оперативной памяти один адрес для быстрого доступа к соответствующей папке посредством комбинации горячих клавиш или использования контекстного меню строки адреса (рис. 8).

Для сохранения в оперативной памяти адреса перехода, содержащегося в строке адреса, требуется выполнить одно из следующих действий:

- нажать комбинацию клавиш *Ctrl+Alt+C*;
- в контекстном меню строки адреса выбрать *Адрес перехода* → *Сохранить*;
- в главном меню выбрать *Инструмент* → *Адрес перехода* → *Сохранить*.

Для открытия адреса перехода (для перехода к соответствующей папке) требуется выполнить одно из следующих действий:

- нажать комбинацию клавиш *Ctrl+Alt+V*;
- в контекстном меню строки адреса выбрать *Адрес перехода* → *Открыть*;
- в главном меню выбрать *Инструменты* → *Адрес перехода* → *Открыть*.

Модель LineExplorer реализует оригинальную возможность поиска файлов и папок непосредственно из строки адреса в местоположении, определяемом введенным адресом. Результаты поиска отображаются в контекстном списке, а для его реализации требуется включить соответствующий режим и задать шаблон для поиска папок или файлов.

Включение режима поиска осуществляется путем ввода в строке адреса специальной сигнатуры, определяющей место поиска (во вложенных папках, начиная с предшествующей части имеющегося адреса). По умолчанию включение режима поиска происходит при вводе комбинации символов *+/*.

После включения режима поиска открывается контекстный список, содержащий заготовки шаблонов поиска папок (*) и файлов (*.*). После выбора требуемого пункта списка следует дополнить шаблон поиска, который может включать любые символы, разрешенные ОС Windows, а также специальные символы * и ?, обозначающие соответственно любое число и ровно один произвольный символ.

При работе в режиме поиска пользователю доступен просмотр и выбор элементов из контекстного списка его результатов, а также удаление символов сигнатуры и изменение шаблона поиска. Для продолжения работы со строкой адреса требуется выйти из режима поиска с сохранением или без сохранения текущего результата, адрес которого содержится в строке адреса.

Выход из режима поиска осуществляется нажатием клавиши *Esc* или удалением хотя бы одного из символов сигнатуры поиска (рис. 9).

При выходе без сохранения текущего результата в строке адреса сохранится адрес, предшествующий символам сигнатуры поиска. При выходе с сохранением текущего результата соответствующий адрес будет отображен без символов сигнатуры и шаблона поиска.

Поиск папки может быть организован в любом каталоге, адрес которого введен в строку адреса, для этого требуется выполнить следующие действия.

1. Задать каталог путем ввода соответствующего адреса.
2. Включить режим поиска папок одним из следующих способов:

- ввести с клавиатуры символы сигнатуры поиска (по умолчанию *+/*);
- нажать горячую клавишу *F3*;
- вызвать контекстное меню строки адреса и выполнить команду *Добавить поиск* → *папок...* (см. рис. 8).

3. Задать шаблон поиска или дополнить его, если соответствующий режим был включен с помощью контекстного меню или горячей клавиши. В качестве примера в каталоге WINDOWS найдем папки, имена которых содержат любое число символов и начинаются с буквы *a* (рис. 10).

4. Для запуска поиска нажать клавишу *Enter* либо завершить шаблон поиска вводом обратного слеша (**).

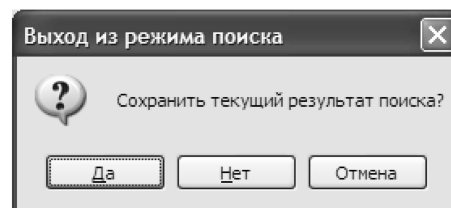


Рис. 9. Выход из режима поиска

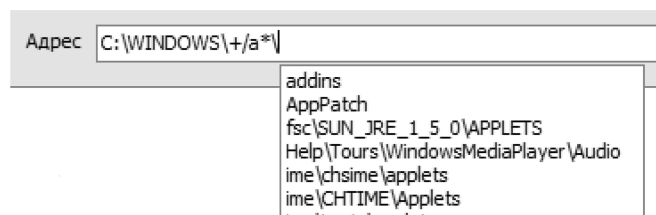


Рис. 10. Пример параметров и результатов поиска папок

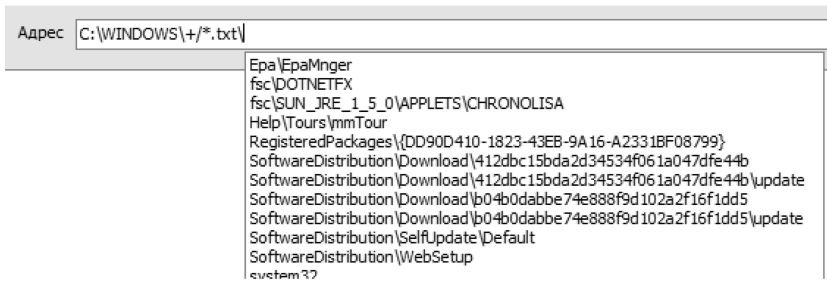


Рис. 11. Пример параметров и результатов поиска файлов

После завершения поиска, при наличии найденных папок, список их адресов будет отображен в контекстной подсказке (рис. 10).

Результатом поиска файлов является список адресов содержащих их папок. Поиск файлов может быть организован в любой папке, адрес которой введен в строку адреса, для этого требуется выполнить следующие действия.

1. Задать папку путем ввода соответствующего адреса.

2. Включить режим поиска файлов одним из следующих способов:

- ввести с клавиатуры символы сигнатуры поиска (по умолчанию +/);
- нажать горячую клавишу *F4*;
- вызвать контекстное меню строки адреса и выполнить команду *Добавить поиск → файлов...* (см. рис. 8).

3. Задать или дополнить шаблон поиска, если соответствующий режим был включен с помощью контекстного меню или горячей клавиши. Например, в папке *WINDOWS* найдем файлы с любым именем и расширением *txt* (рис. 11).

4. Для запуска поиска нажать клавишу *Enter* либо завершить шаблон поиска вводом обратного слеша (**).

После завершения поиска, при наличии найденных файлов, список адресов содержащих их папок отображается в контекстной подсказке (рис. 11).

Настройки программы

Настройки программы представляют собой опции управления ее функциональными возможностями, повышающие удобство эксплуатации.

Вызвать окно настроек можно одним из следующих способов:

- выполнением в главном меню команды *Инструмент → Настройки...*;
- нажатием правой клавиши мыши на метке *Адрес* адресной панели;
- нажатием горячей клавиши *F10*;
- выбором пункта *Настройки...* в контекстном меню строки адреса (см. рис. 8).

Настройки программы подразделяются на общие, связанные с контекстной подсказкой, закладками, поиском и работой в сети Интернет.

Общие настройки

1. *Число позиций истории браузера* (от 3 до 31) определяет число возможных переходов к предыдущим местоположениям в файловой системе в текущем сеансе работы программы.

2. *Шрифт* надписей на элементах интерфейса в окне модели *Line-Explorer* может быть оформлен различными стилями и кеглем.

Настройки контекстной подсказки

1. *Число видимых элементов списка* (от 1 до 30) определяет число элементов, отображаемых без полосы прокрутки контекстного списка.

2. *Автоматический переход указателя на список* обеспечивает автоматическое позиционирование указателя курсора мыши на первый элемент каждого следующего контекстного списка.

3. *Чувствительность к регистру* устанавливает зависимость автоматической сортировки элементов контекстных списков от регистра вводимых с клавиатуры символов.

Настройки закладок

1. *Отображение в контекстной подсказке* определяет представление закладки в контекстном списке в виде имени или адреса папки.

Настройка поиска

1. *Сигнатура поиска* определяет набор символов, ввод которых включает режим поиска, предусмотрены три варианта: +/ или + > или +|.

2. *Сортировка результатов поиска* обеспечивает их упорядочивание в контекстном списке по алфавиту (по именам папок) или по числу слешей (глубине вложенности) в соответствующих адресах.

Настройки работы в сети Интернет

1. Опция *Разрешить ввод адресов сайтов* обеспечивает отображение в первом контекстном списке пункта *http* и возможность использования контекстной подсказки при вводе адресов Интернет-сайтов.

2. Опция *Разрешить ввод IP-адресов* обеспечивает отображение в первом контекстном списке пункта *ftp* и возможность использования контекстной подсказки при вводе IP-адресов удаленных компьютеров.

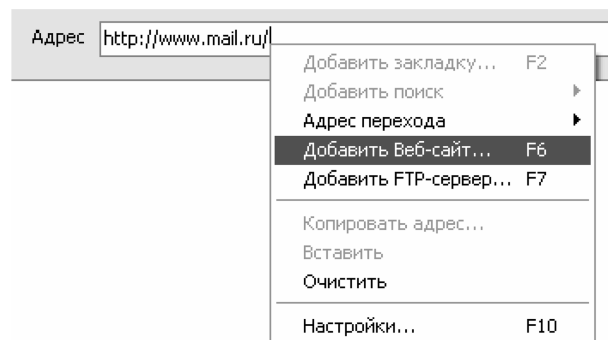


Рис. 12. Дополненное контекстное меню строки адреса

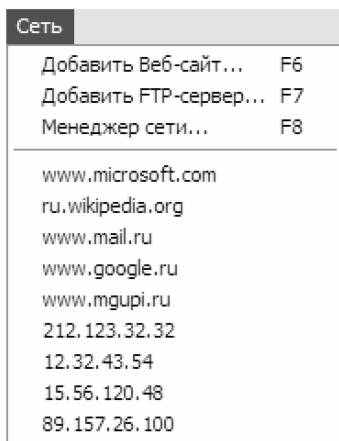


Рис. 13. Раздел главного меню *Сеть*

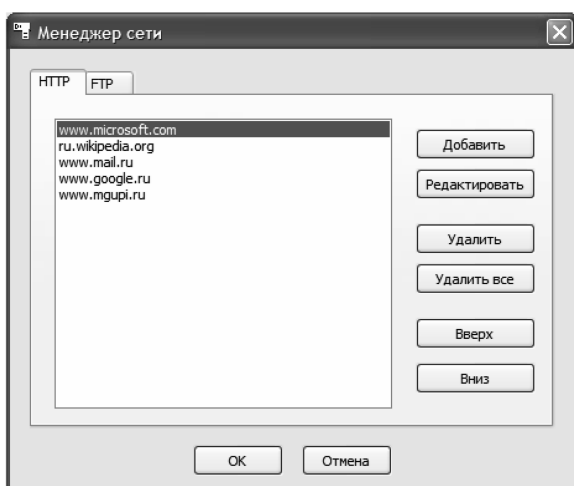


Рис. 14. Окно *Менеджер сети*

Сохранение IP-адресов и адресов Интернет-сайтов реализовано аналогично созданию закладок. При включении перечисленных выше опций контекстное меню строки адреса дополняется соответствующими пунктами (рис. 12).

При включении опций настройки работы в сеть Интернет главное меню дополняется разделом *Сеть* (рис. 13), содержащим опции, аналогичные опциям работы с закладками в разделе *Закладки* (см. рис. 5).

При выборе пункта *Менеджер сети* открывается дополнительное окно, содержащее отображаемые в контекстной подсказке списки сохраненных адресов Интернет-сайтов и IP-адресов удаленных компьютеров (рис. 14).

Универсальность строки адреса при работе с файловой системой и сетью Интернет обеспечивает единство соответствующих информационных пространств и объединяет локальные и удаленные ресурсы пользователя.

Сравнение с аналогами

В данном разделе представлены результаты сравнения функциональных возможностей стро-

ки адреса, реализованной в модели LineExplorer, с аналогичными элементами интерфейса других приложений. Следует заметить, что числовая оценка и сравнение эффективности предложенных технических решений с аналогами затруднены и являются задачами специалистов в такой области знаний, как Human Computer Interaction (человеко-компьютерное взаимодействие). В связи с этим многие параметры сравнения носят качественный характер, а методы, позволяющие оценить затраты времени пользователей на работу с графическими интерфейсами, приведены в следующем разделе без конкретных результатов их применения.

Основными аналогами LineExplorer являются Windows 95-XP Explorer, Windows Vista Explorer и файловый менеджер UltraExplorer. Как показывает представленный далее материал, модель LineExplorer обладает достаточной функциональностью и эффективностью навигации по файловой системе в плане удобства, эргономичности и простоты эксплуатации. Фактически LineExplorer не только предоставляет оптимальное сочетание функциональных возможностей аналогичных элементов интерфейса перечисленных выше приложений, но и обеспечивает оригинальные опции использования закладок, поиска папок и файлов, а также контекстной подсказки и работы в сети Интернет. Вместе с тем модель LineExplorer не претендует на полное превосходство над аналогичными существующими разработками.

Windows 95-XP Explorer. История данной линейки операционных систем Microsoft Windows начинается в августе 1995 г., когда состоялся выпуск ОС Windows 95, и заканчивается выпуском в 2004 г. ОС Windows XP. Основой окон данных операционных систем является проводник Windows Explorer, обеспечивающий графический интерфейс доступа пользователя к папкам и файлам и фактически реализующий графическую оболочку ОС Windows. В окнах представленной линейки ОС реализована строка адреса, функциональные возможности и принцип использования которой не изменялись в течение 10 лет. В связи с этим достаточно провести сравнение модели LineExplorer со строкой адреса любого из соответствующих проводников Windows 95-XP Explorer, представленной на рис. 15.

Рассмотрим особенности строки адреса Windows 95-XP Explorer.

1. Помимо адреса папки в выпадающем списке отображается проводник данных, который содержит дерево папок, но практически не используется при работе с окнами ОС Windows. Данный элемент интерфейса неэффективен в плане навигации по логическим дискам и папкам, так как требует большого числа манипуляций с отображаемым деревом [1, 2] (рис. 16).

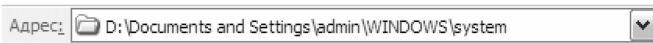


Рис. 15. Строка адреса Windows 95-XP Explorer

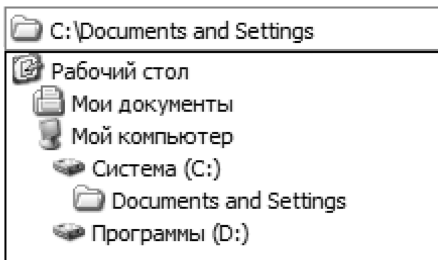


Рис. 16. Проводник данных

2. Контекстная подсказка доступна только при положении курсора в конце адреса и отображает список полных адресов вложенных папок, создавая громоздкие конструкции, рассеивающие внимание пользователя. Список адресов папок не отображается при произвольном положении курсора в адресе, что ограничивает возможность автоматической сортировки его содержания.

3. При использовании контекстной подсказки (при выборе пункта из списка адресов папок) курсор перемещается в начало строки адреса, что затрудняет работу пользователя, не соответствует фокусировке его внимания, а также ограничивает применение мыши и клавиатуры.

4. В строке адреса отсутствуют опции поиска папок и файлов, а также использования закладок и адреса перехода.

5. Системные папки операционной системы представлены в проводнике данных (рис. 16), что затрудняет доступ пользователя к их содержанию.

6. Следствием сложности навигации по файловой системе является работа пользователя с несколькими окнами (в многооконном режиме).

7. При вводе в строку адреса адресов Интернет-сайтов вместо окна ОС Windows отображается окно приложения Internet Explorer.

Перечисленные недостатки строки адреса проводника Windows 95-XP Explorer приводят к ограничению ее функциональности и обуславливают ее использование в основном как справочного элемента интерфейса.

Результаты сравнения с моделью Line Explorer:

1. Возможности проводника данных заменены контекстными списками, что сокращает число требующихся манипуляций с мышью и клавиатурой, а следовательно, сложность и затраты времени на доступ к папкам.

2. В контекстной подсказке отображаются списки имен вложенных папок, что не рассеивает внимание пользователя, а также позволяет легко вводить и изменять адрес с помощью клавиатуры и мыши.

3. В процессе ввода адреса курсор позиционируется относительно его окончания, что позволя-

ет легко продолжать навигацию по файловой системе и соответствует фокусировке внимания пользователя.

4. В строке адреса реализованы опции поиска папок и файлов, а также использования закладок и адреса перехода.

5. Использование строки адреса модели Line Explorer сокращает потребность пользователя в работе с несколькими окнами за счет удобства навигации по файловой системе.

6. Работа с адресами Интернет-сайтов реализована без использования дополнительного программного обеспечения.

Windows Vista Explorer. Вышедшая в ноябре 2006 г. ОС Microsoft Windows Vista внесла множество положительных моментов в развитие проводника Windows Explorer, многие недостатки которого, свойственные предыдущей версии 95-XP, были устранены.

Рассмотрим особенности строки адреса Windows Vista Explorer.

1. Предусмотрено два типа интерфейса строки адреса: текстовый, аналогичный версии Windows 95-XP Explorer, и графический, реализованный последовательными выпадающими списками (рис. 17). Заголовок каждого списка состоит из двух элементов:

- имя соответствующей папки в адресе — нажатие левой клавиши мыши реализует переход к содержанию данной папки и ее адресу;
- кнопка вызова списка — нажатие левой клавиши мыши обеспечивает вызов списка имен вложенных папок.

2. Текстовый тип интерфейса строки адреса полностью соответствует ее версии в проводнике Windows 95-XP Explorer и обладает всеми перечисленными выше недостатками.

3. Реализация контекстной подсказки посредством последовательных выпадающих списков не предусматривает возможность автоматической сортировки их содержания по вводимым с клавиатуры символам.

4. В обоих типах интерфейса строки адреса отсутствуют опции использования закладок и адреса перехода. Опция поиска представлена текстовым полем, расположенным справа от строки адреса и

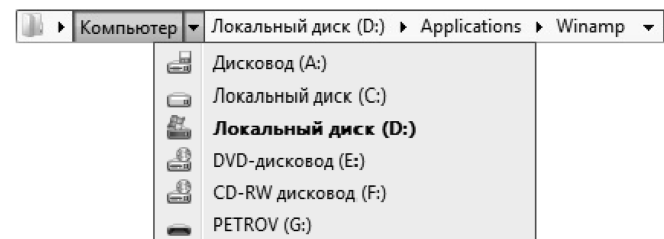


Рис. 17. Строка адреса Windows Vista Explorer

реализующим поиск папок и файлов в местоположении, определяемом введенным адресом.

5. В графическом типе интерфейса строки адреса адрес папки не может быть введен с клавиатуры, скопирован в буфер обмена или вставлен из него.

6. Переход между графическим и текстовым типами интерфейса строки адреса может вводить пользователя в заблуждение.

7. При вводе в строку адреса адресов Интернет-сайтов осуществляется автоматический переход к окну приложения Internet Explorer.

Результаты сравнения с моделью Line Explorer:

1. Использование в строке адреса контекстных списков позволяет реализовать автоматическую сортировку их содержания по вводимым с клавиатуры символам, а также организовать навигацию по файловой системе от любого произвольного положения курсора в адресе.

2. В строке адреса реализованы опции поиска папок и файлов, а также использования закладок и адреса перехода.

3. Строка адреса адаптирована к использованию мыши и клавиатуры, включая комбинирование данных средств манипулирования.

4. Работа в Интернете доступна без использования дополнительных окон.

Ultra Explorer. Программа Ultra Explorer — это профессиональный файловый менеджер, впервые появившийся в декабре 2006 г. Рассмотрим особенности использования реализованной в нем строки адреса.

1. В отдельном поле отображаются имя текущей папки и проводник данных, аналогичный проводнику в строке адреса Windows 95-XP Explorer.

2. Строка адреса реализована системой многоуровневых меню (рис. 18).

3. Многоуровневые меню требуют меньших расстояний перемещения курсора мыши, чем выпадающие списки в проводнике Windows Vista Explorer. При этом с ростом числа пунктов меню и списков данное отличие приобретает все большее значение. На рис. 19 представлены схемы, описывающие перемещения курсора мыши в процессе ввода адреса.

Из схем, представленных на рис. 19, видно, что вертикальные перемещения курсора эквивалентны, а горизонтальным соответствует меньшее расстояние, чем диагональным (меню предпочтительнее выпадающих списков).

4. Работа с многоуровневыми меню сложна для пользователей, так как задействует большое пространство окна, а также требует высоких концентрации внимания и точности манипулирования мышью и клавиатурой.

5. В строке адреса отсутствуют опции поиска папок и файлов, а также использования закладок и адреса перехода. Адрес папки не может быть

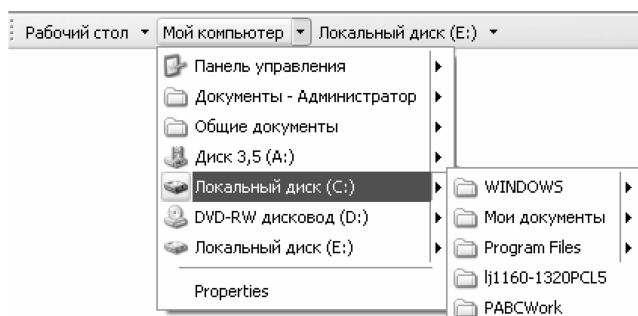


Рис. 18. Строка адреса проводника Ultra Explorer

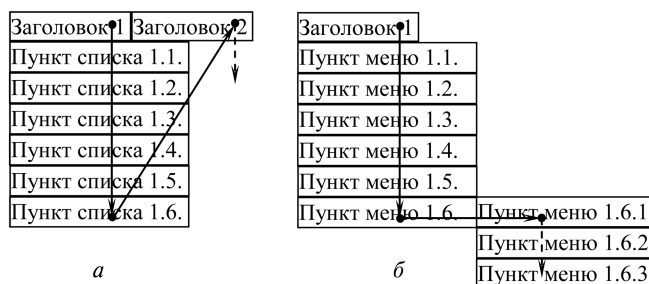


Рис. 19. Схемы перемещений курсора мыши: а — по выпадающим спискам; б — по меню

введен с клавиатуры, скопирован в буфер обмена или вставлен из него.

6. Наблюдаются затруднения, связанные с фокусировкой внимания пользователя при работе с адресами, содержащими большое число имен папок.

7. Отсутствуют опции навигации в сети Интернет.

Результаты сравнения с моделью Line Explorer:

1. Использование контекстных списков позволяет реализовать автоматическую сортировку их содержания по вводимым с клавиатуры символам, а также организовать навигацию по файловой системе от любого произвольного положения курсора в адресе.

2. При работе с последовательными контекстными списками отсутствуют затруднения с фокусировкой внимания пользователя.

3. Опция автоматического позиционирования курсора мыши на первый пункт каждого следующего списка исключает необходимость выполнения как диагональных, так и горизонтальных перемещений курсора (рис. 19).

4. Последовательный выбор элементов контекстных списков требует меньшей точности манипулирования мышью и клавиатурой и поэтому более предпочтителен, чем множественный выбор пунктов многоуровневых меню.

5. В строке адреса реализованы опции поиска папок и файлов, а также использования закладок и адреса перехода.

6. Реализованы опции навигации в сети Интернет.

Определение затрат времени пользователя на работу с графическими интерфейсами

Формализованное описание процесса работы пользователя с графическими интерфейсами позволяет получить применение метода GOMS [2] (от английского Goals, Operators, Methods and Selection Rules — цели, операторы, методы и правила выбора). Данный метод подразумевает разделение каждой задачи пользователя на ряд когнитивных и моторных шагов (операторов), требующихся для ее решения с помощью интерфейса программы. Предполагается, что для достижения каждой цели возможен ряд методов, использующих наборы различных операторов. Реализуемый метод определяется с помощью правил выбора, составленных на основании контекста решаемой задачи. Время выполнения задачи определяется затратами времени на выполнение всех задействованных в ней операторов. Заметим, что выкладки результатов использования метода GOMS являются весьма объемными, а результаты его применения зависят от набора тестовых задач, на котором выполняются оценка и сравнение эффективности различных интерфейсов.

Как было показано в предыдущем разделе, модель LineExplorer требует от пользователя меньше перемещений курсора мыши (меньше операторов), чем аналогичные элементы интерфейса других приложений. В результате, даже не изучая подробно декомпозиции задач пользователя, можно утверждать, что предлагаемая модель более эффективна в плане затрат времени на ввод адреса.

При определении затрат времени пользователя на выполнение каждого оператора целесообразно использовать следующий когнитивный закон: чем дальше от текущей позиции курсора находится элемент интерфейса или чем меньше его размеры, тем больше времени потребуется пользователю для выполнения воздействия на него [3].

Длина прямой линии, соединяющей начальную позицию курсора и ближайшую точку элемента интерфейса, определяется как дистанция D . Если размер элемента вдоль проходящей через него линии D равен S , то время выбора t_v определяется по формуле

$$t_v = a + b \log_2 \left(\frac{D}{S} + 1 \right) \text{ или } t_v = a + b \left(\frac{D}{S} \right)^{\frac{1}{2}},$$

где a и b — константы, зависящие от навыков пользователя и технических характеристик средства манипулирования.

Значение отношения D к S является безразмерной величиной, поэтому если пользователь перемещает средство манипулирования на расстояние, отличное от перемещения курсора по экрану, данная формула имеет силу при условии, что за-

висимость между двумя этими перемещениями является линейной. Рассмотренный закон описывает взаимоотношение между скоростью воздействия и его точностью, характеризующейся значением S . Скорость движения может быть высокой, если не требуется достижения высокой точности. В то же время повышение точности приводит к снижению скорости.

Применение метода GOMS и рассмотренного когнитивного закона позволяет провести оценку и сравнение затрат времени пользователя на выполнение некоторого набора задач с помощью Windows 95-XP Explorer, Windows Vista Explorer, UltraExplorer и LineExplorer. Очевидно, что получаемые результаты будут зависеть от заданного набора задач, поэтому каждому исследователю целесообразно проводить независимое определение эффективности перечисленных приложений в соответствии с особенностями предполагаемой области их внедрения и задач пользователя в ней.

Заключение

Модель LineExplorer является конкурентоспособной оригинальной разработкой, соответствующей передовому уровню организации человеко-компьютерного взаимодействия. Данная модель максимально адаптирована к работе с мышью и клавиатурой, что является отдельным преимуществом и позволяет внедрять ее также в различные диалоговые системы, требующие последовательного ввода иерархически структурированной информации. Модель LineExplorer эффективно расширяет доступное пользователю информационное пространство за счет предоставления возможности навигации в сети Интернет без использования дополнительных окон и дополнительного программного обеспечения. Некоторые разработки, положенные в основу функциональных возможностей модели LineExplorer, опубликованы в [1].

Представленная модель является прототипом, позволяющим оценить предоставляемые функциональные возможности и реализуемые технические решения, в том числе с точки зрения оценки целесообразности их реализации в различном прикладном программном обеспечении.

Список литературы

1. Зуев А. С. Подход к разработке и модернизации структур интерфейсов компьютерных программ // Информационные технологии. 2007. № 1. С. 55–62.
2. Зуев А. С. Математическое и программное обеспечение средств проектирования и совершенствования интерактивных графических человеко-машинных интерфейсов: Дис. ... канд. техн. наук / А. С. Зуев. Московский гос. ун-т приборостроения и информатики. Москва, 2006. 125 с.
3. Раскин Д. Интерфейс: новые направления в проектировании компьютерных систем: Пер. с англ. СПб.: Символ-Плюс, 2005. 272 с.

УДК 621.3.049.771.14

С. В. Гаврилов, д-р техн. наук, зав. сектором,
Институт проблем проектирования

в микроэлектронике РАН,

Э. Р. Каграманян, аспирант,

Московский государственный институт
электронной техники (технический университет),

Л. С. Ходош, канд. техн. наук,

ст. науч. сотр., гл. спец.,

Институт проблем проектирования

в микроэлектронике РАН

Тенденции развития моделей библиотечных элементов для статического временного анализа цифровых СБИС

Проанализированы тенденции развития моделей временных параметров стандартных цифровых элементов для статического временного анализа КМОП СБИС. Рассмотрены преимущества и недостатки различных типов моделей, проанализированы ограничения по их применению при переходе на глубоко-субмикронные и нанометровые нормы проектирования в процессе масштабирования КМОП-технологии.

Ключевые слова: временной анализ СБИС, характеристика стандартных цифровых элементов, нелинейная модель задержки, NLDМ, полиномиальная модель задержки, SPDM, токовая модель задержки, ECSM, CCS.

Введение

Процесс эволюции КМОП-технологии сопровождается миниатюризацией размеров схемных элементов, увеличением числа элементов на кристалле СБИС, а также появлением новых и усилением воздействия уже известных эффектов, влияющих на рабочие характеристики, надежность и срок службы СБИС. При переходе на техпроцессы с глубоко-субмикронными и нанометровыми проектными нормами на рабочие характеристики цифровых СБИС начинают существенно влиять такие эффекты, как электрическое сопротивление межсоединений, эффект Миллера, эффект короткого канала, вариации технологических и внешних параметров и т. д. Адекватный учет этой разнородной информации в статических временных моделях стандартных цифро-

вых элементов на стадии проектирования СБИС необходим в целях повышения эффективности методов верификации проектов и, следовательно, обеспечения приемлемого параметрического выхода годных микросхем [1].

Одним из основных этапов логико-временной верификации проектов цифровых СБИС является статический временной анализ (СВА) (*Static timing Analysis — STA*) [2], который при заданном описании схемы на вентиляльном уровне (*gate-level netlist — *.vg*) и на основе библиотеки временных параметров стандартных цифровых элементов (формат *Liberty — *.lib, *.db*) позволяет делать оценку таких временных характеристик цифровой схемы, как минимальная и максимальная задержки, минимальная ширина синхроимпульса и т. д. (рис. 1). Если результаты СВА показывают, что схема удовлетворяет всем временным ограничениям, то снижается вероятность функционально-временных сбоев в процессе функционирования разработанной СБИС.

В качестве входных данных для статического временного анализа используется информация о временных параметрах стандартных цифровых элементов и межсоединений, полученная посредством их характеристики — электрического моделирования для отобранных комбинаций входных воздействий с учетом разнородных вариаций технологических (длина/ширина канала МОП-транзисторов, толщина оксида и т. д.) и внешних (напряжение питания, температура) параметров. Таким образом, в процессе СВА достигается адекватный учет всех прогнозируемых временных нарушений. При этом время прогона СВА, как правило, имеет линейную зависимость от размеров схемы [2]. Два указанных преимущества СВА делают его неотъемлемым этапом временной верификации современных цифровых СБИС, а динамический временной анализ часто используется лишь в качестве

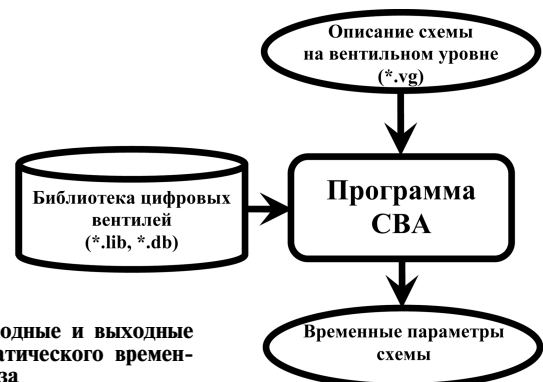


Рис. 1. Входные и выходные данные статического временного анализа

ве дополнения к СВА для более точного расчета отдельных временных характеристик.

Современные модели временных параметров стандартных цифровых элементов

В наши дни в индустрии САПР ИС постоянно развивается механизм моделирования временных параметров и характеристик цифровых элементов как на транзисторном, так и на вентильном уровне. Например, если в начале эры логико-временного моделирования каждому цифровому вентилю в соответствие ставилась статическая (так называемая собственная) задержка, то дальнейшее развитие программ моделирования сопровождалось постоянным усложнением моделей для достижения приемлемой точности результатов анализа. Можно отметить следующие основные причины, приводящие к периодической необходимости создания новых моделей временных параметров:

- ухудшение точности (относительно SPICE-подобных программ электрического моделирования) предыдущих моделей при оценке временных параметров;
- учет новых эффектов деградации временных параметров схемы, влияние которых усилилось вследствие масштабирования размеров транзисторов, повышения степени их интеграции на кристалле, увеличения рабочих частот СБИС и т. д.;
- увеличение требуемого объема памяти для хранения библиотеки временных параметров вентиляей;
- повышение требований к скорости расчета модели в программе СВА.

Первые попытки моделирования задержек цифровых элементов

Первым этапом усложнения модели стало представление задержки как функции от емкостной нагрузки вентиля [3]. Была предложена формула $D = I + R \cdot C_o$, где I (*Intrinsic*) — собственная задержка; R — эквивалентное сопротивление драйвера; C_o — емкость нагрузки. Следующим шагом стало представление задержки как линейной функции от емкости нагрузки C_o и длительности фронта входного сигнала S_i , которая является промежутком времени, за который входное напряжение успевает возрасти, например, от уровня 10 % до уровня 90 % амплитуды сигнала. В этом случае задержка рассчитывалась как $D = I + K \cdot S_i + R \cdot C_o$, где K — чувствительность задержки к входному фронту сигнала. Однако, когда для ряда КМОП-технологий точность этой формулы стала недостаточной, ее заменили на выражение $D = I + K \cdot S_i + (R_j \cdot C_o + Y_j)$, где Y_j — поправка для собственной задержки для каждого j -го диапазона выходной емкости [3]. Несложно заметить, что все приведенные выше формулы

представляют собой разновидности линейной зависимости задержки от пары параметров S_i/C_o . Однако реальные временные характеристики цифровых элементов существенно нелинейны, следовательно, существует предел, при достижении которого дальнейшее использование линейной аппроксимации нелинейных характеристик может привести к неприемлемым погрешностям. Основными причинами отказа от использования линейных моделей послужили повышение степени интеграции и увеличение быстродействия цифровых СБИС, что привело к уменьшению абсолютных значений временных параметров, таких как период синхросигнала, время установки/удержания и т. д. Таким образом, для разработчиков САПР СБИС стало "жизненной" необходимостью создание качественно новой нелинейной модели временных параметров цифровых элементов.

Нелинейная модель задержки

До недавнего времени "де-факто" стандартом модели цифровой ячейки являлась так называемая нелинейная модель задержки (*Non-Linear Delay Model — NLDM*), которая на сегодняшний день поддерживается большинством современных программ СВА (например *Synopsys PrimeTime*, *Cadence Encounter*). В *NLDM*-модели каждой комбинации "входной фронт/емкостная нагрузка" (S_i/C_o) соответствуют определенные значения задержки вентиля и фронта сигнала на его выходе [4] (рис. 2).

Табличные значения задержек (D) и длительностей выходных фронтов (S_o) (рис. 2) рассчитываются посредством электрического (*SPICE*) моделирования для каждого стандартного цифрового элемента из библиотеки, сохраняются в формате *Liberty* и используются затем в качестве входных данных *NLDM*-модели в процессе СВА. Подробнее рассмотрим механизм расчета временных параметров на основе *NLDM*-модели. Допустим, из *Liberty*-файла известны четыре значения задержки ($D_{11}(S_{i1}, C_{o1})$, $D_{12}(S_{i1}, C_{o2})$, $D_{21}(S_{i2}, C_{o1})$, $D_{22}(S_{i2}, C_{o2})$) определенного элемента библиотеки и в процессе СВА требуется рассчитать значение задержки этого элемента D_{XY} при произвольных значениях S_{iX} и C_{oY} , удовлетворяющих усло-

Параметр	Емкостная нагрузка (фф)						
	2	4	8	16	64	128	384
Длительность входного фронта (пс)							
25							
50							
100							
200							

Рис. 2. Иллюстрация представления входных данных для *NLDM*-модели в формате *Liberty*

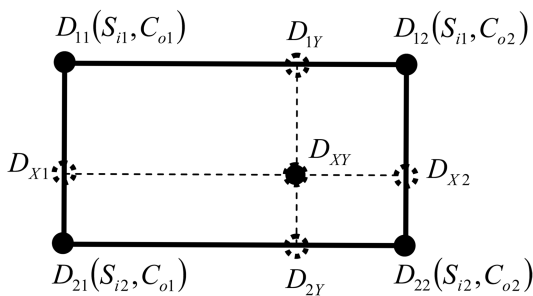


Рис. 3. Определение приблизительного значения задержки $D_{XY}(S_{iX}, C_{oY})$ посредством билинейной интерполяции — совокупности двух последовательных линейных интерполяций по осям абсцисс и ординат

виям $S_{i1} \leq S_{iX} \leq S_{i2}$ и $C_{o1} \leq C_{oY} \leq C_{o2}$ (рис. 3). *NLDM*-модель позволяет определить приблизительное значение задержки D_{XY} с помощью билинейной интерполяции на основе известных значений D_{11} , D_{12} , D_{21} и D_{22} . Процедура заключается в следующем: сначала с помощью линейной интерполяции определяются промежуточные пары значений D_{1Y} , D_{2Y} или D_{X1} , D_{X2} , а затем на их основе линейной интерполяцией определяется неизвестное значение D_{XY} .

Несмотря на то, что *NLDM*-модель иногда используется для СВА схем даже с проектной нормой 90 нм, как правило, при переходе на глубоко-субмикронные и нанометровые технологии точность такой модели оказывается недостаточной [5].

Отметим основные недостатки *NLDM*-модели.

Для каждого прямоугольника (рис. 3) задержка является линейной функцией от емкости нагрузки и длительности входного фронта. Такая аппроксимация может привести к заниженной оценке (*underestimation*). Погрешность может составить порядка 10 % [6].

Очевидно, что точность *NLDM*-модели зависит от числа и расположения табличных значений, используемых в качестве входных данных. Точность модели может быть улучшена при увеличении числа табличных значений, однако при этом растет размер библиотеки в *Liberty*-формате. Практически все таблицы во всех реальных библиотеках имеют неравномерное распределение точек по осям S_i и C_o . Неравномерность распределения точек может привести к ухудшению точности аппроксимации, так как очевидно, что чем больше расстояние между двумя соседними точками, тем больше будет ошибка при интерполяции. Следовательно, очевидна заинтересованность фирм-разработчиков таких моделей в нахождении оптимального набора точек выборки [6]. Несмотря на то, что в современных маршрутах входные данные в *Liberty*-формате, как правило, представляют собой таблицы размером 7×7 элементов, в прошлом некоторые фирмы-разработ-

чики библиотек стандартных элементов делали попытки использования таблиц размерностью 64×128 [3], что сегодня представляется неприемлемым с точки зрения объемов памяти. Необходимый компромисс между объемом занимаемой памяти, скоростью моделирования и точностью расчетов приводит к вопросу о выборе структуры таблицы, т. е. к определению точек в рабочем диапазоне значений параметра, которые необходимы и достаточны для включения в таблицу.

Дополнительные трудности с увеличением объема библиотеки возникают в случаях, когда требуется провести характеризацию для нескольких значений рабочей температуры и напряжения питания. Если учесть, что в процессе анализа цифровых СБИС расчеты выполняются для нескольких режимов работы (типовой/наилучший/наихудший — *Typical/Best/Worst*), то объем такой библиотеки как минимум утраивается.

В начале XXI века лидерами рынка САПР СБИС были предложены два новых подхода для более точного моделирования временных параметров цифровых ячеек, которые позволяют существенно повысить точность СВА. Первый подход — масштабируемая полиномиальная модель задержки вентиля. Второй подход — использование токовых моделей задержки.

Полиномиальная модель задержки

Масштабируемая полиномиальная модель задержки (*Scalable Polynomial Delay Model — SPDM*) представляет собой универсальный формат моделирования временных параметров стандартных цифровых элементов на основе полиномиальной функции от емкости нагрузки, длительности входного фронта, температуры и напряжения питания [6—8]. Компания *Synopsys* анонсировала *SPDM* в 2000 г. в качестве модели, позволяющей преодолеть ряд недостатков *NLDM*-модели. Формат *SPDM* поддерживался пакетом *Library Compiler*, а также большинством других компонентов САПР *Synopsys* для синтеза, временного анализа и физического проектирования.

Основная идея *SPDM*-модели состоит в следующем. В качестве альтернативы табличным значениям предлагается использовать масштабируемые полиномы с набором коэффициентов. Термин "масштабируемый" означает, что форма и степень полинома определяется (масштабируется) в зависимости от значений параметров. Одновременно, в случае невозможности такого преобразования, *NLDM*-таблицы также сохраняются в составе библиотеки [6].

Как известно, полиномы играют ключевую роль в численных расчетах. Все сложные функции можно разложить на полиномы или их комбина-

ции. Приближение полиномами младшего порядка обеспечивает приемлемую точность для таких функций, как временные характеристики цифровых вентилях [8].

Основными проблемами при характеристике *SPDM*-моделей являются относительно небольшая точность, ошибки аппроксимации и большое время работы алгоритма характеристики. Фирмой *Synopsys* со временем был разработан новый *ATG*-генератор *SPDM*-моделей, который позволял создавать модели с приемлемой точностью аппроксимации нелинейных характеристик в большом диапазоне рабочих режимов [9]. Преимущества *ATG*-генератора — это жесткий контроль точности, надежный метод аппроксимации, высокая скорость работы программы для решения уравнений при аппроксимации, а также усовершенствованные методы дискретизации. Именно перечисленные доработки позволили резко повысить точность результатов. Созданная новым генератором полиномиальная модель удовлетворяла требованиям точности и компактности. Типовое значение максимально допустимой относительной погрешности задержки составляло 1 %, а абсолютной погрешности — 5 пс. Новый генератор *SPDM*-моделей был включен в *Synopsys Master ToolBox (MTB)* в июне 2005 г. для предоставления заказчикам возможности характеристики в этом формате. Таким образом, *SPDM* имеет большую точность в сравнении с более ранней и широко распространенной нелинейной моделью задержки (*NLDM*). Существенными недостатками данной модели являются небольшая точность при аппроксимации реальной нелинейной характеристики схемы полиномами с конечным числом членов при переходе на нанометровые нормы проектирования, а также необходимость создания довольно сложных алгоритмов для программ-генераторов таких моделей.

В работе [3] в качестве входных рассмотрены полиномиальные функции с двумя переменными, однако предложенный метод легко можно применить и для большего числа переменных. Для пары переменных x и y , а также таблицы результатов $D(x, y)$ предлагается рассматривать два класса полиномов. Первый класс может быть представлен как $D(x, y) = P_m(x)Q_n(y)$, где P_m и Q_n — полиномы соответственно m -го и n -го порядка. P и Q должны быть рядами Тейлора m -го и n -го порядка соответственно для переменных x и y . Такой выбор степеней полиномов обусловлен тем, что временной параметр в общем случае может зависеть от m -й степени аргумента x и n -й степени аргумента y . Для заданных значений m и n существуют $(m + 1)(n + 1)$ коэффициентов. Например,

$$P_1 Q_2 = (a_0 + a_1 x)(b_0 + b_1 y + b_2 y^2) = A_{00} + A_{01} y + A_{02} y^2 + A_{10} x + A_{11} xy + A_{12} xy^2.$$

Второй класс — это зависимость $D(x, y) = T_n(x, y)$, которая представляет собой разложение в ряд Тейлора n -го порядка. В этом случае временной параметр зависит от n -х степеней двух переменных. Для данного n получаются $(n + 1)(n + 2)/2$ коэффициентов. Например,

$$T_3 = (B_{01}) + (B_{11}x + B_{12}y) + (B_{21}x^2 + B_{22}xy + B_{23}y^2) + (B_{31}x^3 + B_{32}x^2y + B_{33}xy^2 + B_{34}y^3).$$

В дополнение к таблице с выборочными значениями такая модель также требует наличия их стандартного отклонения σ , которое также называется ошибкой *SPICE*-моделирования или, в общем случае, статистической ошибкой выборок. Значение σ играет ключевую роль в определении приемлемости формулы. Такая значимость σ естественна, так как временные параметры носят статистический характер. Чем меньше σ , тем жестче критерий приемлемости, а случай $\sigma = 0$ практически исключает аппроксимацию формулы. Стоит отметить, что стандартное отклонение также влияет на коэффициенты конечных формул. Таким образом, естественно, имея физический смысл, в данном случае девиация рассматривается как параметр, контролирующий точность.

Для нахождения коэффициентов для заданной формы полинома, как правило, используется линейный метод наименьших квадратов (*LSE*). Приемлемость полученной формулы проверяется с помощью анализа χ -квадрат [3]. Алгоритм *LSE* находит коэффициенты, которые минимизируют эту величину:

$$\chi^2 = \sum_{i=1}^N \left(\frac{Y_i - y_i}{\sigma_i} \right)^2,$$

где N — число точек выборки; Y_i — табличное значение в точке i ; y_i — расчетное значение в точке i ; σ_i — стандартное отклонение в точке i .

Генератор *SPDM*-моделей создает полиномы, которыми можно с приемлемой точностью моделировать нелинейные характеристики в большом диапазоне рабочих режимов.

Токовые модели задержки

Еще одной перспективной альтернативой *NLDM*-модели выступает модель цифровой ячейки на основе источника тока. Две разновидности такой модели — *ECSM*-модель фирмы *Cadence* и *CCS*-модель фирмы *Synopsys* — позволяют достичь выигрыша по времени в сравнении со *SPICE*-моделированием при сравнительно небольшой потере в точности (около 2 %), а также дают возможность учитывать влияние *RC*-параметров межсоединений, влияние сопротивления которых стало доминировать при определении за-

держек в схемах с нанометровыми размерами элементов [10, 11].

Модели *ECSM* и *CCS* представляют собой трехмерные таблицы типа $V_o(t, S_i, C_o)$ и $I_o(t, S_i, C_o)$ соответственно, где V_o — выходное напряжение; I_o — выходной ток цифрового элемента; t — моменты выборки значений соответственно напряжения или тока. Очевидно, что необходимость хранения трехмерных таблиц резко увеличивает объемы памяти для библиотек. В среднем, объем *lib*-файла библиотеки в *CCS*-формате примерно в 10 раз больше объема аналогичного файла в *NLDM*-формате [12]. С этой точки зрения практический интерес представляет поиск возможных путей упрощения данных моделей при сохранении приемлемой точности СВА.

Как первый существенный недостаток предыдущих моделей задержки можно отметить потерю точности при замене реальной входной емкости приемника (*receiver*) на эквивалентную заземленную пассивную емкость. Реальная входная емкость приемника сигнала изменяется в процессе переключения. Кроме того, она зависит от напряжения питания, емкости нагрузки, длительности входного фронта и предыдущего состояния выхода. Следовательно, замена такой емкости статичным пассивным эквивалентом является грубой аппроксимацией. В токовых моделях (*ECSM*, *CCS*) емкость приемника представлена двумя значениями на интервале переключения выходного сигнала, что позволяет с большей точно-

стью аппроксимировать реальные задержки и фронты сигналов и является несомненным преимуществом токовых моделей задержки.

Список литературы

1. Keller I., Tam K. H., Kariat V. Challenges in Gate Level Modeling for Delay and SI at 65 nm and Below // Proc. of ACM/IEEE Design Automation Conference, 2008. — P. 468—473.
2. Blaauw D., Chopra K., Srivastava A., Scheffer L. Statistical Timing Analysis: From Basic Principles to State of the Art // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2008. V. 27. P. 589—607.
3. Wang F., Chang S.-S. Scalable Polynomial Delay Model for Logic and Physical Synthesis // Proc. of IEEE International Conference on Chip Design Automation, 2000. — P. 101—106.
4. Croix J. F., Wong D. F. A Fast And Accurate Technique To Optimize Characterization // Proc. of Design Automation Conference, 1997. — P. 337—340.
5. Trihu R. Addressing Library Creation Challenges from Recent Liberty Extensions // Proc. of Design Automation Conference, 1997. — P. 474—479.
6. Cirit M. A. Libraries with Scalable Polynomial Delay Model Improve Modeling Accuracy // ISD Magazine. V. 7. 2001. — P. 56—59.
7. Girishankar G., Shitanshu T. Generating Scalable Polynomial Models: Key to Low Power High Performance Designs // Proc. of the 19th International Conference on VLSI Design held jointly with 5th International Conference on Embedded Systems Design, 2006. — P. 625—630.
8. Shiue W. T., Wanalerlak W. An Advanced Cell Polynomial-Base Modeling for Logic Synthesis // Proc. of IEEE International SOC Conference, 2003. — P. 393—396.
9. Wang X. et al. Scalable Polynomial Model, A Technical White Paper on a New Delay Model for DSM Technology // www.synopsys.com
10. CCS Timing: Technical White Paper // www.synopsys.com
11. Korshak A., Lee J.-C. An Effective Current Source Cell Model for VDSM Delay Calculation // Proc. of the 2nd International Symposium on Quality Electronic Design, 2001. — P. 296—300.
12. Электронный источник: <http://www.snug-universal.org/>

УДК 004.416.2

Н. В. Шмырев, мл. науч. сотр.,

Научно-исследовательский институт системных исследований РАН, г. Москва

Средства моделирования и контроля распределенных систем реального времени

Рассмотрены варианты использования формальных моделей для описания поведения распределенных приложений, функционирующих в распределенной сетевой среде. Особое место среди них занимают приложения реального времени, корректность которых зависит от времени, затраченного на выполнение определенных операций, например на обработку текущих координат самолета и принятие решения об изменении его курса. К числу таких относятся приложения, реализующие управление бортовыми автономными объектами или большими, сложно организованными системами, работоспособность которых критически важна. Вопросы контроля за состоянием подобных систем, в том числе с использованием формальных моделей, являются крайне актуальными.

Ключевые слова: распределенные системы реального времени, моделирование, контролируемое выполнение.

Введение

На основе формальной модели приложения можно не только контролировать его состояние — набор параметров в фиксированный момент времени, но и доказывать некоторые свойства, такие

как корректность указателей или отсутствие блокировок. Описание поведения распределенного приложения на языке высокого уровня позволяет разработчику наглядно представить его свойства, эффективно организовать управление, локализо-

вать возникающие ошибки. Именно эти возможности наиболее востребованы для полной реализации концепции контролируемого выполнения распределенных систем, суть которой подробно рассмотрена в работе [9]. Наиболее трудный вопрос, который возникает на этом направлении, заключается в выборе формализма, используемого для описания приложения. В настоящее время предложены различные способы и инструментальные средства описания состояния распределенных приложений. В их числе конечные автоматы, сети Петри, исчисления предикатов со временем и целый ряд других.

Существуют относительно быстрые алгоритмы, на основе которых можно создавать эффективные средства верификации свойств модели приложения, построенной с помощью современных инструментальных средств. Однако выразительности таких моделей, как правило, не хватает для описания режимов работы даже простейших распределенных приложений. Для приложений, в которых нужно учитывать требования по потребляемым ими ресурсам, этот вопрос становится неразрешимым. Заметим, что зачастую во внимание приходится принимать не только время выполнения отдельных участков кода приложения. На приложения обычно накладываются ограничения по объему используемой ими памяти или пропускной способности сетевой среды.

Важно не только создать модель и верифицировать ее свойства. Необходимо установить степень адекватности свойств модели приложению, которое она описывает. Часто для этого используется валидация модели — экспериментальное подтверждение того, что результаты моделирования соответствуют поведению приложения. В этой статье рассмотрен иной подход, основанный на комбинации статического анализа модели, исходного кода приложения и средств самоконтроля. Такой подход позволяет не только удостовериться в применимости модели к работающему приложению, но и расширить класс доказанных свойств. Некоторые свойства модели могут проверяться в процессе выполнения, обеспечивая таким образом эффективную проверку всего набора утверждений о модели. Использование методов статического анализа позволяет автоматизировать применение модели, упростить ее создание и значительно оптимизировать проверку корректности модели. Описанные далее методы абстракции при статическом анализе исходного кода совместно с использованием средств самоконтроля приложения позволяют разрабатывать сложные верифицируемые системы.

В настоящей статье критически анализируются некоторые, наиболее востребованные инструментальные средства моделирования приложений, которые представлены в настоящее время на ми-

ровом IT-рынке. Отмечается, что ни одно из них не удовлетворяет перечисленным выше требованиям полностью. Анализируются изменения, которые необходимы для создания инструментального средства моделирования, связанного с исходным кодом распределенных приложений и позволяющего адекватно описывать их состояние и поведение.

Моделирование приложений

В качестве примера математического аппарата, широко применяемого в моделировании приложений реального времени, в том числе — систем управления физическими объектами, можно рассмотреть гибридные автоматы, как правило, описываемые с помощью пакета *Hytech* [7].

Гибридный автомат состоит из набора состояний с условиями на некоторые переменные. В качестве таких переменных обычно рассматриваются физические величины — время, масса, скорость. В каждом из состояний определены скорости изменения переменных и границы, при которых происходит переход из одного состояния в другое. Такое описание позволяет не только доказывать утверждения о том, что исследуемая характеристика не превышает некоторого значения, но и получать значение параметров, при которых работа автомата будет корректной.

В качестве иллюстрирующего примера рассмотрим описание процесса автоматизированной поддержки должного уровня воды в цистерне с помощью гибридного автомата:

```
var
    x: clock;
    y: analog;
automaton water
synclabs;;
initially on_0 & y = 1;
loc on_0: while y<= 10 wait {dy = 1}
    when y = 10 do {x' = 0} goto on_1;
loc on_1: while x<= 2 wait {dy = 1}
    when x = 2 goto off_0;
loc off_0: while y>= 5 wait {dy = -2}
    when y = 5 do {x' = 0} goto off_1;
loc off_1: while x<= 2 wait {dy = -2}
    when x = 2 goto on_0;
end
```

Переменная x в данном случае описывает время, а переменная y — уровень воды в цистерне. В зависимости от уровня воды открываются и закрываются заслонки, скорость наполнения изменяется. Используя описание, можно доказать, что уровень воды не превысит критического значения.

Несмотря на то, что гибридные автоматы позволяют удобно описывать класс систем управления, применение их на практике вызывает значительные сложности [3]. Накопление вычислитель-

ной ошибки затрудняет применение доказанных утверждений. Создание исходного кода, соответствующего автомату, также представляет собой достаточно сложную задачу.

Другой подход к решению исходной задачи можно продемонстрировать на примере программного средства верификации моделей *Blast* [1]. Используя механизмы отложенной абстракции [8] для доказательства утверждений о достижимости заданной метки в исходном коде на языке Си, инструментарий *Blast* позволяет доказывать утверждения о корректности приложения. Он предоставляет возможность отслеживать принадлежность переменных заранее заданному интервалу или проверять наличие блокировок в коде. Суть механизма отложенной абстракции заключается в следующем. На первом шаге доказательства утверждения строится самая простейшая модель приложения — граф вызовов функции. Далее осуществляется попытка доказать необходимое утверждение. В случае неудачи модель уточняется на основе элементов построенного примера, опровергающего принятое ранее утверждение, и эвристик. Например, если в первом приближении считалось, что вызов функции будет осуществлен при любых условиях, во втором приближении уточняются значения локальных переменных, при которых будет вызвана функция. Модель уточняется до тех пор, пока не удастся получить требуемое утверждение. Так как при доказательстве используются только необходимые свойства, появляется возможность доказывать утверждения о приложениях из сотен тысяч строк исходного кода.

Уточняя модель для доказательства достижимости метки в исходном коде, *Blast* позволяет получить схему вызовов, которые привели к ошибочной ситуации, и может использоваться для создания автоматических тестов. Например, для функции

```
int foo(int x, int y) {
    if (x > y) {
        x = y - x;
        assert (x > 0);
    }
}
```

Программный комплекс *Blast* позволяет не только доказать невыполнимость утверждения $x > 0$, но и определить последовательность вызовов функций, приводящую к нештатной ситуации:

```
Size: 5
4 :: 4: Pred (x@foo > y@foo) :: 5
5 :: 5: Block(x@foo = y@foo - x@foo;) :: 6
6 :: 6: Pred(Not (x@foo > 0)) :: 6
6 :: 6: FunctionCall (_assert ("foo.c", 6, "x > 0")) :: -1
```

К сожалению, не всякое утверждение может быть легко выведено из исходного кода. В представленном далее примере утверждение $(x == y)$ приходится проверять во время выполнения приложения. Такие дополнительные утверждения,

проверяемые динамически, позволяют, как уже отмечалось ранее, значительно расширить класс доказанных утверждений.

```
void main ()
{
    int x = 0, y = 0;
    while (foo()) {
        x++; y++;
    }
    runtime_assert (x == y);
    while (x > 0)
        if (x != y) {
            x--; y--;
        }
    assert (y == 0);
}
```

В данном случае утверждение $(x == y)$ проверяется на стадии выполнения приложения и вызывает исключительную ситуацию. Используется оно при доказательстве другого утверждения $(y == 0)$, проверяемого с помощью *Blast* статически.

Подобный метод, представленный в работе [2], использует средства самоконтроля и применяется для проверки корректности указателей. С помощью инструментария код приложения изменяется. Каждое использование указателя сопровождается проверкой на его корректность. Далее необходимость каждого изменения проверяется с помощью *Blast*. Если доказано, что указатель корректен при любых условиях, проверка отбрасывается. Таким образом, из всего набора проверок остается только небольшое число действительно необходимых. Такие проверки включаются в исходный код выполняемого приложения. Вследствие небольшого числа внесенных изменений производительность приложения падает незначительно, а доказанные свойства приложения гарантируют корректность использования указателей.

Вместе с тем, и это необходимо отметить, *Blast* имеет серьезный недостаток. Он заключается в том, что современные алгоритмы отложенной абстракции неэффективны при верификации моделей распределенных приложений. С помощью механизмов доказательства достижимости метки сложно описать взаимодействия процессов. Более того, построенные модели распределенных приложений имеют очень большое число состояний и проверяются неэффективно.

Наилучшим, по мнению авторов, подходом, комбинирующим механизмы для описания моделей и их связи с исходным кодом, является подход, предложенный создателями инструментального комплекса MAGIC [6]. В нем используется модель на основе системы с отмеченными переходами (*Labelled Transition System, LTS*), позволяющая описывать конечные автоматы. Важно, что созданная таким образом модель не только может проверяться на наличие некоторых свойств (достижимости меток), но и на соответствие ис-

ходному коду приложения. Например, приведенный далее фрагмент кода описывает вычисление с условиями в терминах LTS:

```
int my_proc(int x) {
    int y;
    if (x == 0) {
        y = foo();
        if (y == 0) return 10;
        else return 20;
    }
}

cprog my_prog = my_proc {
    abstract abs_5, {(S1 == 0)}, S1;
}

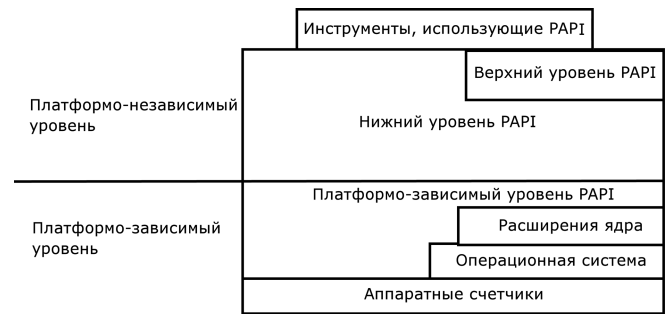
cproc my_proc {
    abstract {ans_1, (S1 == 0), S1 };
}

S1 = (call_foo -> S2),
S2 = (return {S0 == 10} -> STOP |
return {S0 == 20} -> STOP).
```

Программное средство MAGIC позволяет описывать поведение распределенных приложений. Оно применяется для проверки корректности таких, например, приложений, как совместное взаимодействие клиента и сервера по протоколу SSL [5]. Применение MAGIC для описания использования приложением ресурсов целевой среды (памяти, процессорного времени, сетевых интерфейсов) требует дополнительных модификаций. Важным компонентом подобного инструментального средства должен стать высокоточный механизм измерения времени и производительности приложения.

РАPI — интерфейс средств профилирования

Современные процессоры семейств x86, IA-64, Power, MIPS и другие поддерживают профилирование приложений посредством использования аппаратных счетчиков — специальных регистров, фиксирующих аппаратные события определенного типа. В качестве примера аппаратных событий можно привести общее число процессорных тактов, общее число выполненных команд, число промахов при обращении к виртуальной и кэш-памяти, а также ряд других. Использование аппаратных средств измерения производительности приложений позволяет наиболее эффективно проводить верификацию моделей на предмет использования ресурсов целевой среды. Для доступа к аппаратным счетчикам на разных платформах консорциумом *Parallel Tools Consortium* (<http://www.ptools.org>) был предложен интерфейс РАPI [4] (*Performance Application Programming Interface*). В настоящее время РАPI фактически является стандартом для разработчиков программного обеспечения, осуществляющего профилирование кода. На рисунке представлена архитектура интерфейса РАPI.



Архитектура РАPI

В архитектуре РАPI выделяется два основных уровня: платформо-независимый и платформо-зависимый, скрывающий от пользователя детали реализации доступа к аппаратным счетчикам конкретного процессора. Для этого, в частности, функции РАPI на платформо-зависимом уровне могут использовать расширения ядра, функции целевой операционной системы или непосредственно язык ассемблера.

Пользователям РАPI предоставляются интерфейсы верхнего и нижнего уровней, различающиеся сложностью настройки и использования. В настоящее время существуют реализации интерфейсов для языков высокого уровня Си и Фортран.

В реализации интерфейса РАPI впервые предложено стандартизованное, переносимое решение для профилирования кода посредством управления аппаратными счетчиками событий. В настоящее время существуют реализации РАPI в виде библиотек для многих современных платформ. Следует отметить также хорошую степень документирования проекта и простоту использования предлагаемых в его рамках интерфейсов.

Таким образом, применение интерфейса РАPI для организации профилирования распределенных приложений позволяет минимизировать воздействие на целевую среду за счет использования аппаратных средств профилирования — аппаратных счетчиков.

В настоящее время на основе библиотеки РАPI в рамках инструментального комплекса СОМ (Система отладки и мониторинга), разработанного в НИИСИ РАН, создана библиотека профилирования (БП), представляющая собой программную библиотеку. Функции этой библиотеки ориентированы на получение информации о времени выполнения отдельных участков кода прикладных программ программируемых процессоров сигналов (ППС), а также потоков операционной системы реального времени ос2000, выполняющейся на MIPS-совместимых процессорах [10]. Программируемый процессор сигналов представляет собой многопроцессорную ЭВМ, состоящую из отдельных плат — модулей обработки сигналов (МОС). На каждом модуле обработки сигналов располагаются четыре или восемь процессоров цифровой обработки сигналов (ЦПОС), имею-

ших как локальную память, так и память, разделяемую с остальными процессорами модуля. Собранная информация может обрабатываться независимыми программами-профилировщиками.

Расширение инструментального комплекса MAGIC для описания использования ресурсов в приложении

Эффективным способом создания надежной системы разработки и поддержки приложений является расширение инструментального комплекса MAGIC для работы с приложениями, неоптимально использующими ресурсы целевой среды. Этот комплекс позволяет не только планировать с его помощью процесс разработки приложения, но и предоставляет возможности использования полученных моделей для проверки корректности его выполнения. Большая часть компонентов, необходимых для реализации комплекса, описана выше. С помощью библиотеки PAPI можно точно измерить время, затраченное на выполнение переходов. Описание стандартных вызовов функций выделения памяти позволяет контролировать расход динамической памяти в приложении.

Простое расширение синтаксиса MAGIC позволяет учитывать потребление приложением некоторого ресурса в целевой среде:

```
S1 = (call_foo - > S2, +10),  
S2 = (return {S0 == 10} - > STOP, -20 |  
return {S0 == 20} - > STOP, +30).
```

Для каждого состояния можно задать не только возможные условия переходов, но и затраты ресурса на выполнение перехода, например, расход памяти после обработки запроса, поступившего к приложению. Используя полученное описание, можно вывести статические свойства приложения, например, максимальное используемое значение ресурса. Комплекс MAGIC позволяет связать подобное описание с исходным кодом приложения, используя описания только небольшого числа стандартных функций, таких, как функция выделения памяти. В тех случаях, когда подобное связывание вызывает уже отмеченные ранее затруднения, проверки на стадии выполнения с использованием функций библиотеки PAPI и дополнительных библиотек позволяют упростить задачу связывания, обеспечив соответствие приложения заданной модели.

К сожалению, не все примитивы, используемые в приложениях реального времени, просто описываются в терминах LTS. Как показывает анализ, легко покрывается только класс приложений, взаимодействие между компонентами которых осуществляется с помощью синхронных сообщений. Тем не менее, этот класс достаточно широк для описания большого спектра распре-

ленных приложений. Однако следует учитывать, что значительное усложнение модели может сделать недостижимым доказательство утверждений о некоторых ее свойствах.

Заключение

Поддержка эффективных режимов работы приложений, выполняющихся под управлением систем реального времени, является актуальной и сложной задачей. Для верификации поведения распределенных приложений в настоящее время существует представительный набор средств их моделирования. Различные варианты использования высокоуровневых моделей при контролируемом выполнении программ, рассмотренные выше, позволяют с разной степенью точности и достоверности представлять приложение. Полученное представление активно применяется на различных этапах разработки.

К числу первоочередных в перечне исследований и разработок на обсуждаемом направлении следует отнести использование построенных моделей в различных компонентах инструментального отладочного комплекса. Кроме традиционных проверок на корректность указателей и генерации тестов, перспективными направлениями исследований являются визуализация производительности приложения с использованием модели, использование сигнатур ошибок, взаимодействие со средствами мониторинга.

Список литературы

1. Beyer D., Henzinger T. A., Jhala R., Majumdar R. The Software Model Checker BLAST: Applications to Software Engineering // Journal on Software Tools for Technology Transfer. 2007. N 9 (5–6). P. 505–525.
2. Beyer D., Henzinger T. A., Jhala R., Majumdar R. Checking Memory Safety with BLAST // Proc. FASE'05. 2005. P. 2–18.
3. Anand M., Kim J., Lee I. Code Generation from Hybrid Systems Models for Distributed Embedded Systems // Proc. ISORC'05. 2005. P. 28–36.
4. PAPI User's Guide // <http://icl.cs.utk.edu/projects/papi/>
5. Chaki S., Clarke E., Groce A., Ouaknine J., Strichman O., Yorav K. Efficient Verification of Sequential and Concurrent C Programs // Formal Methods in System Design (FMDS). 2004. V. 25. Is. 2–3. P. 129–166.
6. Chaki S., Clarke E., Groce A., Jha S., Veith H. Modular Verification of Software Components in C, Transactions on Software Engineering (TSE). 2004. V. 30. N 6. P. 388–402.
7. Henzinger T. A., Ho P., Wong-Toi H. HyTech: the next generation // Proc. TACAS'95, LNCS 1019. 1995. P. 41–71.
8. Henzinger T. A., Jhalla R., Majumdar R., Sutre G. Lazy abstraction. Principles of Programming Languages // ACM. 2002. P. 58–70.
9. Вьюкова Н. И., Галатенко В. А., Костюхин К. А., Шмырев Н. В. Организация отладочного комплекса для целевых систем со сложной архитектурой // Информационная безопасность. Микропроцессоры. Отладка сложных систем: Сб. статей под ред. В. Б. Бетелина. М.: Изд. НИИСИ РАН. 2004. С. 120–160.
10. Безруков В. Л., Годунов А. Н., Назаров П. Е., Солдатов В. А., Хоменков И. И. Введение в ос2000 // Вопросы кибернетики: Сб. статей под ред. В. Б. Бетелина. — М.: Изд. НИИСИ РАН. 1999. С. 76–106.

УДК 004.925.84

В. А. Бимаков, директор,
Общество с ограниченной ответственностью
"Робинт", г. Ижевск,
e-mail: robint@mail.ru

Высокоскоростная обработка n -мерных пиксельных геометрических моделей

Излагается подход к построению высокопроизводительных устройств, предназначенных для обработки n -мерных изображений и n -мерных сцен. Подход базируется на преимуществах пиксельного (точечного или дискретного) метода геометрического моделирования по глубокому распараллеливанию операций и процедур обработки сцен, являющихся наиболее "проблемными" с точки зрения затрат времени на их выполнение.

Ключевые слова: высокоскоростной метод обработки изображений и сцен, n -мерные пиксельные геометрические модели, моделирование геометрических преобразований, архитектура специализированной ОКМД-системы.

Введение

Повышение уровня автоматизации невозможно без дальнейшего совершенствования средств вычислительной техники. Имеющийся опыт в разработке систем автоматизированного проектирования, сложных робототехнических устройств убеждает в недостаточной эффективности (а часто и невозможности) использования существующих вычислительных средств в качестве аппаратной поддержки систем, призванных заменить человека в самых сложных областях его деятельности. Ярким примером тому служат задачи, связанные с необходимостью организации скоростной обработки плоских изображений и пространственных сцен. К числу таких задач относятся различные задачи размещения, распознавания образов, машинной графики, задачи управления мобильными и автономными роботами.

В настоящее время наибольшее применение для целей скоростной автоматической обработки сцен нашли два метода геометрического моделирования: 1) метод моделирования многогранниками (многоугольниками); 2) пиксельный (точеч-

ный или дискретный) метод моделирования. При синтезе и анализе геометрических моделей почти всегда применяется метод моделирования многогранниками, поскольку он наиболее приспособлен для числовой (арифметической) обработки, под нее собственно и создавалась традиционная цифровая техника. Второй (пиксельный) метод моделирования чаще всего применяется только при вводе видеоизображений, а также на этапе вывода изображений на экран растрового дисплея.

Использование двух методов геометрического моделирования является крайне нерациональным. Во-первых, смена метода моделирования сопряжена с дополнительными затратами машинного времени и вычислительных ресурсов. Во-вторых, и это главное, метод моделирования многогранниками не может обеспечить предельной степени распараллеливания основных процедур обработки сцен. К тому же, именно применением этого метода моделирования объясняется чрезвычайно сильная зависимость времени обработки сцены от ее геометрической сложности.

Основные процедуры обработки сцен

Алгоритмы решения задач обработки сцен содержат в себе типовые укрупненные процедуры, использующие геометрические модели объектов в качестве своих операндов. К таким операциям и процедурам относятся:

- теоретико-множественные операции над моделями объектов;
- анализ геометрических моделей на пересекательность;
- геометрические преобразования (поворот, перенос, сжатие);
- операции вычисления объема (площади) модели;
- определение положения модели в пространстве.

Доля каждого из этих видов процедур в алгоритмах решения задач не постоянна, зависит от типа решаемой задачи и выбранного метода решения. Однако все эти процедуры объединяет то, что затраты времени на их выполнение с помощью существующей техники чрезвычайно сильно зависят от сложности геометрической формы обрабатываемых объектов.

Сокращение времени выполнения перечисленных процедур может быть достигнуто за счет глубокого распараллеливания процесса их выполнения. С этой точки зрения пиксельный метод

моделирования не имеет себе равных и позволяет реализовать все перечисленные выше "проблемные" операции и процедуры с предельной степенью параллелизации (на битовом уровне описания геометрических моделей). Применение пиксельных моделей полностью снимает зависимость между временем выполнения анализа сцены и сложностью геометрической формы ее объектов. Появляется возможность повысить скорость и качество анализа сцен за счет организации их поэтапной обработки с использованием сначала пиксельных моделей низкого разрешения, а затем пиксельных моделей большего разрешения. Однако все эти преимущества пиксельного моделирования могут быть реализованы только при наличии специальной "пиксельно-ориентированной" вычислительной техники. Данная статья посвящена изложению основ построения таких систем [1–3], существенно отличающихся от традиционной вычислительной техники (однопроцессорных, конвейерных или нейронных систем).

Используемые термины

Определение 1. *n*-мерной сценой *S* формата $F^{(n)}$ назовем семейство $S = \{E_m\}_{m \in I}$ (где $l = 1, 2, 3, \dots, m_s$) точечных множеств E_m евклидового пространства R^n , удовлетворяющее выражению $(\forall E_m \in S)[(E_m \subseteq F^{(n)}) \wedge (\mu^{(n)}(E_m) \neq 0)]$, где $\mu^{(n)}(E_m)$ — *n*-мерная мера *n*-мерного множества E_m (понятие *n*-мерной меры $\mu^{(n)}(E^{(n)})$ является естественным обобщением следующих понятий: длины $\mu^{(1)}(E^{(1)})$ отрезка $E^{(1)}$ — для R^1 ; площади $\mu^{(2)}(E^{(2)})$ плоской фигуры $E^{(2)}$ — для R^2 ; объема $\mu^{(3)}(E^{(3)})$ трехмерной фигуры $E^{(3)}$ — для R^3 [4]).

Определение 2. Множество $E_m \in S$ назовем *m*-м объектом сцены *S*.

Определение 3. Сцену $S = \{E_m\}_{m \in I}$ формата $F^{(n)}$, для которой одновременно справедливы выражения

$$F^{(n)} = \bigcup_{m=1}^{m_s} E_m \text{ и } (\forall i, j \in I)[(i \neq j) \Rightarrow (E_i \cap E_j = \emptyset)],$$

назовем *m_s*-нарным (или *m_s*-уровневым) *n*-мерным изображением формата $F^{(n)}$. Каждое *m_s*-нарное изображение формата $F^{(n)}$ может быть представлено *m_s*-значной функцией Φ , принимающей в каждой точке $a \in F^{(n)}$ значение $\Phi(a)$, удовлетворяющее высказыванию

$$(\forall E_m \in S)(\forall a \in E_m)(\forall b \in F^{(n)})[(\Phi(a) = \Phi(b)) \Rightarrow (b \in E_m)] \wedge ((\Phi(a) \neq \Phi(b)) \Rightarrow (b \notin E_m)).$$

Разобьем пространство R^n на классы точечных множеств $p_l^{(n)} \subset R^n$ (где $l \in L, L = 1, 2, \dots, \infty$), для которых справедливы выражения

$$(\forall l \in L)(\forall a, b \in p_l^{(n)})[(\rho(a, b) \leq \xi_p)],$$

где $\rho(a, b)$ — расстояние между точками a и b множества $p_l^{(n)}$, ξ_p — постоянная;

$$(\forall l \in L)[\mu^{(n)}(p_l^{(n)}) \neq 0]; (\forall i, j \in L)[(i \neq j) \Rightarrow (p_i^{(n)} \cap p_j^{(n)} = \emptyset)]; \bigcup_{l \in L} p_l^{(n)} = R^n.$$

Определение 4. Множества $p_l^{(n)} \subset R^n$, одновременно удовлетворяющие всем перечисленным требованиям, назовем *n*-мерными пикселями $p^{(n)}$ -пиксельного разбиения R^n , а ξ_p — постоянной этого разбиения.

Определение 5. Точечное множество $\bigcup_{l \in L_k} p_l^{(n)}$, составленное из конечного множества пикселей $p_l^{(n)}$ ($L_k \in L$), назовем $p^{(n)}$ -дискретным объектом.

Определение 6. $p^{(n)}$ -дискретной (пиксельной) моделью множества $\dot{E} \subset R^n$ назовем $p^{(n)}$ -дискретный объект $E = \bigcup_{l \in L_E} p_l^{(n)}$, составленный из всех пикселей $p_l^{(n)}$ (где $l \in L_E, L_E \subset L$), для которых выполняется условие $(\forall l \in L_E)[\mu^{(n)}(\dot{E} \cap p_l^{(n)})/\mu^{(n)}(p_l^{(n)}) \geq \delta_l]$, где δ_l — постоянная ($0 < \delta_l \leq 1$).

Определение 7. Вычислительную процедуру f^p по формированию $p^{(n)}$ -дискретной модели $f^p(E) \subset R^n$ множества $f(E)$ из множества E , где $f(E)$ — образ множества $E \subset R^n$ при отображении $f: E \rightarrow f(E)$, назовем $p^{(n)}$ -дискретизирующим отображением f^p , моделирующим отображением $f: E \rightarrow f(E)$ (или просто дискретизирующим отображением, рис. 1, а).

Определение 8. $F^{(n)}$ -форматное геометрическое преобразование f , при котором $f(F^{(n)}) = F^{(n)}$, назовем собственным $F^{(n)}$ -форматным геометрическим преобразованием.

Определение 9. *n*-мерное изображение, формат которого состоит из двух или более взаимно-непересекающихся фрагментов, конгруэнтных $F^{(n)}$, назовем $F^{(n)}$ -составным *n*-мерным изображением.

Способы распараллеливания основных процедур обработки сцен

Глубина распараллеливания определяется числом одновременно обрабатываемых элементов описания модели. В цифровом устройстве самыми низшими и многочисленными элементами описания модели являются биты. Поэтому наивысшая степень распараллеливания процедуры обработки

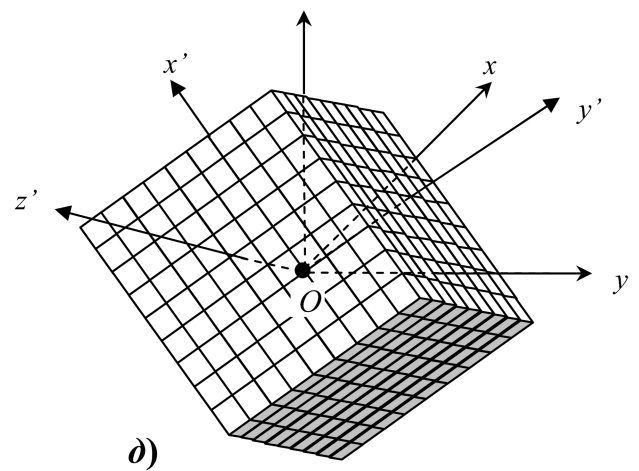
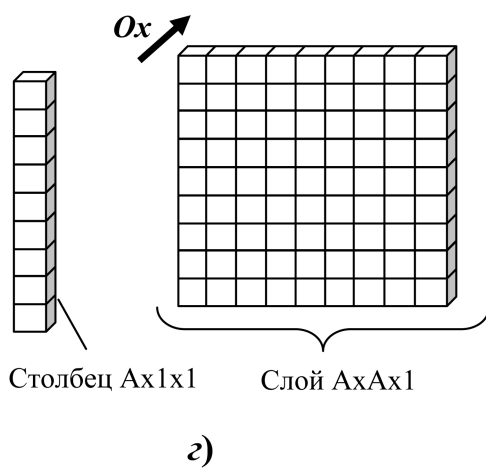
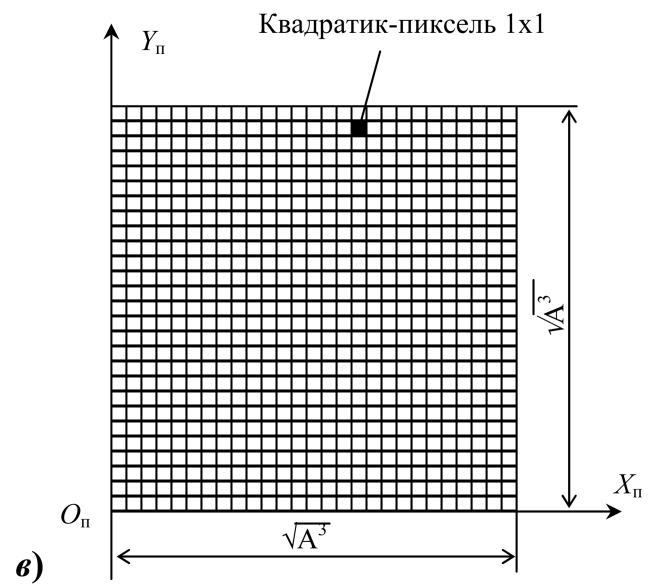
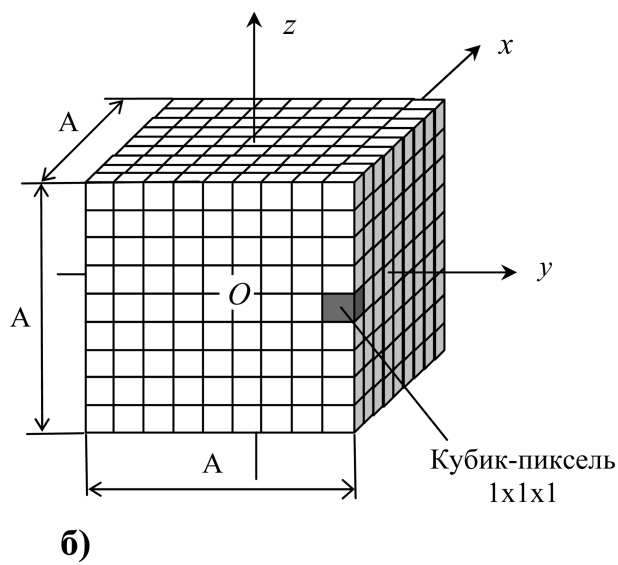
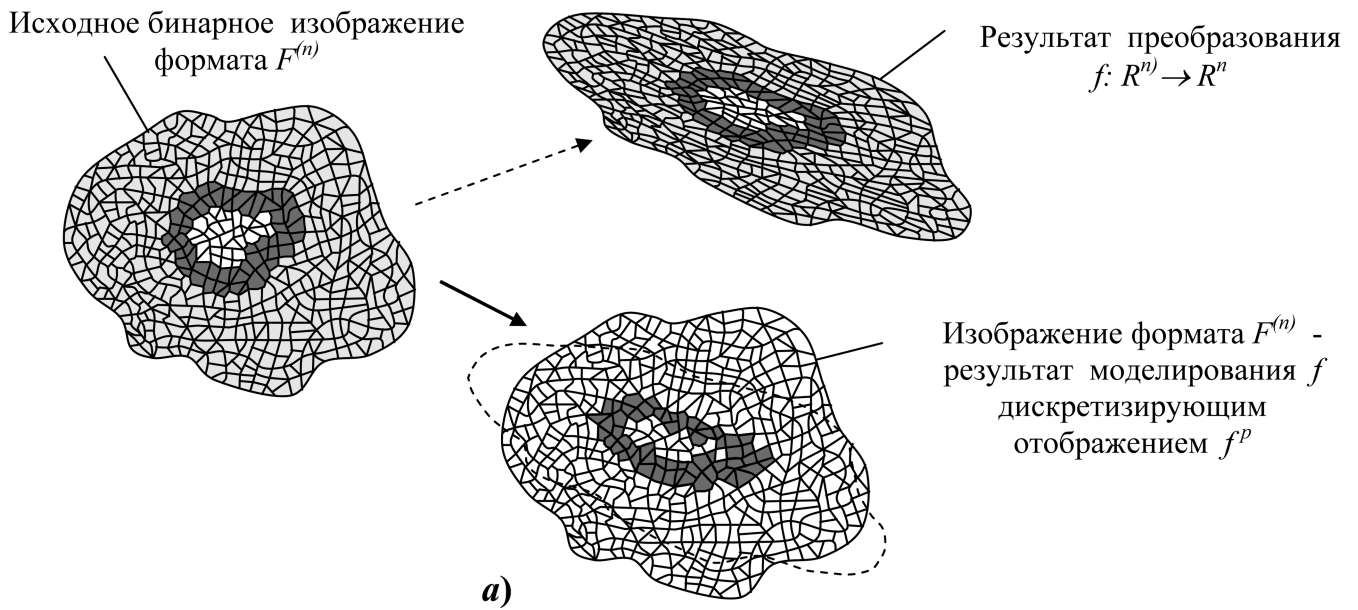


Рис. 1

может быть достигнута только на уровне битовых элементов описания моделей.

Способы высокопараллельной реализации теоретико-множественных операций над пиксельными моделями хорошо известны — это однородные процессорные матрицы ОКМД-систем [5]. Каждый процессор такой матрицы соответствует одному пикселю модели и содержит двоичные запоминающие элементы (для хранения значений параметров пикселей-операндов) и арифметико-логический узел, связанный с этими запоминающими элементами индивидуальными линиями связи. Способ высокопараллельного подсчета числа единичных битов в бинарном изображении (т. е. определения "площади" или "объема" пиксельной модели) также хорошо известен — это различные типы параллельных комбинационных сумматоров.

Глубокое распараллеливание анализа пиксельных моделей на пересекаемость и процедур определения пространственного положения пиксельной модели может быть осуществлено путем применения специального кода заполнения бинарного изображения (КЗБИ). Для формирования КЗБИ формат $F^{(n)}$ бинарного изображения разбивается на ν пиксельных областей, каждой области ставится в соответствие свой бит КЗБИ. Бит равен 1, если в указанной области расположен хотя бы один единичный пиксель, и равен 0 — в противном случае. Аппаратный способ формирования КЗБИ с высокой степенью распараллеливания заключается в использовании многоходовых схем ИЛИ, каждый вход i -й схемы ИЛИ соединен с выходом j -го запоминающего элемента, в котором хранится значение j -го пикселя i -й пиксельной области бинарного изображения. Перед выполнением анализа на пересекаемость двух бинарных изображений S_1 и S_2 формата $F^{(n)}$ сначала выполняется теоретико-множественная операция $S = S_1 \cap S_2$. Затем описанным выше способом формируется КЗБИ результирующего изображения S . Если КЗБИ $\neq 0$, единичное содержимое S_1 и единичное содержимое S_2 пересекаются; если КЗБИ = 0, — не пересекаются.

В рассматриваемом типе устройств выполнение геометрического преобразования над пиксельной моделью заменено высокопараллельным моделированием этого преобразования дискретизирующим отображением (ДО). Несмотря на очевидный недостаток ДО (низкая точность моделирования ими геометрических преобразований), дискретизирующие отображения имеют три важнейших достоинства в случае машинной реализации ДО:

- возможность предельного распараллеливания процесса выполнения ДО;

- отсутствие зависимости между временем выполнения ДО и сложностью геометрической формы преобразуемого объекта;
- возможность замены бесконечного спектра моделируемых геометрических преобразований конечным спектром ДО, уменьшающимся вместе с уменьшением разрешения преобразуемой пиксельной модели.

Аппаратная реализация дискретизирующих отображений

Общий принцип построения $p^{(n)}$ -дискретной модели $f^p(E)$ множества $f(E)$ из объекта E (где $E = \bigcup_{l \in L_E} p_l^{(n)}$ — некоторый произвольный

$p^{(n)}$ -дискретный объект; а f — биективное отображение $f: R^n \rightarrow R^n$) состоит в следующем. Исходный объект E задан бинарным n -мерным изображением S_1 некоторого формата $F_1^{(n)} = \bigcup_{l \in L_1} p_l^{(n)}$, где

$E \subseteq F_1^{(n)}$. Результатом построения должно стать бинарное n -мерное изображение S_2 некоторого формата $F_2^{(n)} = \bigcup_{l \in L_2} p_l^{(n)}$, содержащее внутри себя

объект $f^p(E)$. Пусть $F_1^{(n)} = F_2^{(n)} = F^{(n)}$, $L_1 = L_2 = L_F$. Поскольку S_1 и S_2 являются бинарными, они могут быть представлены следующими двухзначными функциями для $l \in L_F$:

$$\Phi_1(p_l^{(n)}) = \begin{cases} 1, & \text{если } p_l^{(n)} \subseteq E, \\ 0, & \text{если } p_l^{(n)} \subset \bar{E}, \text{ где } \bar{E} = F^{(n)} \setminus E; \end{cases}$$

$$\Phi_2(p_l^{(n)}) = \begin{cases} 1, & \text{если } p_l^{(n)} \subseteq f^p(E), \\ 0, & \text{если } p_l^{(n)} \subset \overline{f^p(E)}, \\ \text{где } \overline{f^p(E)} = F^{(n)} \setminus f^p(E). \end{cases}$$

Построение $p^{(n)}$ -дискретной модели образа $f(E)$ предполагает определение меры $\mu^{(n)}(p_l^{(n)} \cap f(E))$ для каждого $p_l^{(n)} \subset F^{(n)}$. Из равенства $E = \bigcup_{l \in L_E} p_l^{(n)}$

следует $f(E) = \bigcup_{l \in L_E} f(p_l^{(n)})$, отсюда $p_l^{(n)} \cap f(E) = \bigcup_{l \in L_E} (p_l^{(n)} \cap f(p_l^{(n)}))$.

Так как $(\forall i, j \in L_F)[(i \neq j) \Rightarrow (p_i^{(n)} \cap p_j^{(n)} = \emptyset)]$, а f является биективным отображением $f: R^n \rightarrow R^n$, то $f(p_i^{(n)}) \cap f(p_j^{(n)}) = \emptyset$. Отсюда согласно свойству

аддитивности меры [4] вытекает: $\mu^{(n)}(p_i^{(n)} \cap f(E)) = \sum_{i \in L_E} [\mu^{(n)}(p_i^{(n)} \cap f(p_i^{(n)}))]$. Используя введенную функцию $\Phi_1(p_i^{(n)})$, для $i \in L_l \subset L_F$, удовлетворяющих условию $(\exists i \notin L_l)[p_i^{(n)} \cap f(p_i^{(n)}) \neq \emptyset]$, можно записать

$$\begin{aligned} \mu^{(n)}(p_i^{(n)} \cap f(E)) &= \\ &= \sum_{i \in L_l} [\mu^{(n)}(p_i^{(n)} \cap f(p_i^{(n)})) \Phi_1(p_i^{(n)})]. \end{aligned} \quad (1)$$

Из определения 6 следует:

$$\Phi_2(p_i^{(n)}) = \begin{cases} 1, & \text{если } \mu^{(n)}(p_i^{(n)} \cap \\ & \cap f(E))/\mu^{(n)}(p_i^{(n)}) \geq \delta_l, \\ 0, & \text{если } \mu^{(n)}(p_i^{(n)} \cap \\ & \cap f(E))/\mu^{(n)}(p_i^{(n)}) < \delta_l. \end{cases} \quad (2)$$

Выражения (1) и (2) определяют зависимость значения $\Phi_2(p_i^{(n)})$ от значений $\Phi_1(p_i^{(n)})$. Простейший способ распараллеливания f^p , непосредственно вытекающий из выражений (1) и (2), состоит в применении пары "многовходовый аналоговый сумматор — пороговый элемент" (вход l -го порогового элемента соединен с выходом l -го аналогового сумматора) для каждого пикселя $p_i^{(n)} \subset F^{(n)}$. Каждый l -й сумматор реализует выражение (1), каждый l -й пороговый элемент — выражение (2).

Варьируя значением δ_l , можно упростить данный способ аппаратной реализации ДО путем замены аналоговых элементов на простейшие логические элементы. Действительно, если для выбранного значения δ_l высказывание $(\forall i \in L_l) \times [\mu^{(n)}(p_i^{(n)} \cap f(p_i^{(n)}))/\mu^{(n)}(p_i^{(n)}) \geq \delta_l]$ — истинно, нет необходимости вычислять (1), а достаточно реализовать булеву функцию "дизъюнкция":

$$\Phi_2(p_i^{(n)}) = \bigvee_{i \in L_l} \Phi_1(p_i^{(n)}), \quad (3)$$

где $\Phi_1(p_i^{(n)})$ и $\Phi_2(p_i^{(n)})$ выполняют уже роль булевых переменных (1 — истинно, 0 — ложно).

Еще более упрощается аппаратная реализация ДО в случае моделирования преобразований $t: R^n \rightarrow R^n$, называемых перемещениями (движениями или наложениями). К их числу относятся: параллельный перенос, поворот, различные виды симметрии. Для этих преобразований справедливо равенство $\mu^{(n)}(E) = \mu^{(n)}(t(E))$, в связи с чем при регулярном пиксельном разбиении R^n должно соблюдаться $|L_E| \approx |L_{t(E)}|$ (число пикселей, образующих E , должно совпадать (в идеале) с чис-

лом пикселей, образующих $t^p(E)$). А потому само $p^{(n)}$ -дискретизирующее отображение t^p , моделирующее перемещение t , должно быть преимущественно взаимнооднозначным (один пиксель-прообраз $p_i^{(n)}$ должен отображаться, как правило, только в один пиксель-образ $p_j^{(n)}$).

Пусть S_1 — преобразуемое бинарное изображение, представленное n -мерным множеством значений $\Phi_1(p_i^{(n)})|_{i \in L_F}$, хранящихся в одноразрядных двоичных аккумуляторах a_i ; S_2 — бинарное изображение, получаемое в результате моделирования перемещения t над единичным содержимым S_1 и представленное n -мерным множеством значений $\Phi_2(p_i^{(n)})|_{i \in L_F}$, хранящихся в одноразрядных двоичных аккумуляторах b_l . Аппаратная реализация процедуры формирования S_2 из S_1 с максимальной степенью параллелизации состоит в параллельной передаче значений $\Phi_1(p_i^{(n)})$ из аккумуляторов a_i в аккумуляторы b_l ($\Phi_2(p_i^{(n)}) = \Phi_1(p_i^{(n)})$) по индивидуальным линиям связи. При этом взаимнооднозначное соответствие между пикселем-образом $p_i^{(n)}$ (входом аккумулятора b_l) и пикселем-прообразом $p_j^{(n)}$ (выходом аккумулятора a_i) устанавливается индивидуально для каждого пикселя $p_i^{(n)}$ принудительным образом согласно следующих правил:

- пиксель-прообраз для $p_i^{(n)}$ ищется только из числа $p_i^{(n)}$, для которых $i \in L_l$;
- наибольший приоритет при выборе соответствия с $p_i^{(n)}$ имеет пиксель $p_i^{(n)}|_{i \in L_l}$, обладающий наибольшим значением меры $\mu^{(n)}(p_i^{(n)} \cap f(p_i^{(n)}))$;
- пиксель $p_i^{(n)}$, для которого $L_l \neq \emptyset$, обязательно должен иметь пиксель-прообраз $p_i^{(n)}$ ($L_l = \emptyset$, если все $p_i^{(n)}|_{i \in L_l}$ пикселя $p_i^{(n)}$ расположены за пределами формата $F^{(n)}$);
- если для всех $p_i^{(n)}|_{i \in L_l}$ соответствие уже установлено, а для $p_i^{(n)}$ — еще нет, пикселю $p_i^{(n)}$ дается в соответствие один из пикселей $p_i^{(n)}|_{i \in L_l}$, для которого в этом случае принцип взаимной однозначности нарушается.

Простейший способ аппаратной реализации ДО, моделирующего сжатие $s: R^n \rightarrow R^n$ (изменение масштаба), основывается на аппаратной реализации выражения (3) и состоит в применении $|L_l|$ -входовых логических схем ИЛИ для всех пикселей $p_i^{(n)}$, для которых $L_l \neq \emptyset$. Выход каждой $|L_l|$ -входовой схемы ИЛИ соединяется с входом аккумулятора b_l , а вход соединяется с выходом од-

ного из аккумуляторов a_i , для которого $L_i \neq \emptyset$. Кроме того, в однородных процессорных матрицах моделирование сжатия (согласно (3)) может быть выполнено с высокой параллелизацией и программным способом.

О выборе формата $F^{(n)}$ и базиса аппаратно-реализуемых ДО

Формат $F^{(n)}$ — это формат n -мерного бинарного изображения, обработка которого осуществляется в рассматриваемом типе устройств с абсолютной степенью распараллеливания (по всем N пикселям формата $F^{(n)}$ одновременно). Для повышения степени распараллеливания целесообразно увеличивать N . Однако неограниченный рост N приведет к неограниченному росту аппаратуры и снижению тактовой частоты управления. Другая проблема связана с тем, что, несмотря на конечность спектра ДО, их общее число (даже в рамках одного вида моделируемых геометрических преобразований) весьма велико, и все они не могут быть реализованы аппаратно. В этой связи возникают проблема выбора оптимального формата $F^{(n)}$ и проблема выбора оптимального базиса аппаратно-реализуемых ДО (базиса АРДО).

Обе проблемы тесно взаимосвязаны. Действительно, поскольку отдельное ДО моделирует преобразование в общем случае с погрешностью, последовательное выполнение нескольких ДО приведет к наложению погрешностей, вносимых каждым ДО. Погрешность моделирования преобразования обусловлена двумя причинами: погрешностью дискретизации и выходом содержимого преобразованного изображения за пределы формата $F^{(n)}$. Учитывая это, главными претендентами на аппаратную реализацию должны стать собственные $F^{(n)}$ -форматные геометрические преобразования (моделируемые без каких-либо погрешностей) и дискретно-инвариантные геометрические преобразования (моделируемые без погрешности дискретизации). Вопросы, связанные с выбором оптимального формата $F^{(n)}$ и оптимального базиса АРДО, здесь не рассматриваются.

Вариант устройства для обработки 2D и 3D пиксельных моделей

Пусть максимальный формат $F^{(3)}$ устройства представляет собой куб $A \times A \times A$ (где $A = n^2$, $n \geq 2$ — число целое), состоящий из пикселей-кубиков $1 \times 1 \times 1$ (рис. 1, б). Собственными $F^{(3)}$ -форматными преобразованиями будут:

1) повороты на углы, кратные 90° относительно трех взаимно-ортогональных осей Ox , Oy , Oz , проходящих через центр куба $A \times A \times A$ и параллельных ребрам данного куба;

2) преобразования симметрии относительно плоскостей Oxy , Oyz , Oxz . Куб $A \times A \times A$ состоит из трехмерных фрагментов, часть из которых (рис. 1, з) может быть отнесена к двумерным фрагментам (слои $A \times A \times 1$), а часть — к одномерным фрагментам (строки и столбцы $A \times 1 \times 1$).

Пусть максимальный формат $F^{(2)}$ того же устройства представляет собой квадрат размером $\sqrt{A^3} \times \sqrt{A^3}$, состоящий из пикселей-квадратиков 1×1 (рис. 1, в). Тогда собственными $F^{(2)}$ -форматными преобразованиями будут:

- повороты на углы, кратные 90° , относительно центра O_{Π} квадрата $\sqrt{A^3} \times \sqrt{A^3}$;
- преобразования симметрии относительно взаимно-ортогональных осей $O_{\Pi}X_{\Pi}$ и $O_{\Pi}Y_{\Pi}$, направленных параллельно сторонам квадрата $\sqrt{A^3} \times \sqrt{A^3}$.

Поскольку число пикселей $1 \times 1 \times 1$ в $F^{(3)}$ и число пикселей 1×1 в $F^{(2)}$ совпадают, процессорную матрицу $A \times A \times A$, A^3 — входовой сумматор и A^3 -входовой формирователь КЗБИ можно использовать как для параллельной обработки изображений формата $F^{(3)}$, так и для параллельной обработки изображений формата $F^{(2)}$. Вариант функционально-полного базиса АРДО для рассматриваемого варианта устройства включает:

- группу АРДО, моделирующих параллельный поворот изображения $A \times A \times A$ (на 90° вокруг Ox ; на 90° вокруг Oy ; на углы, не кратные 90° , вокруг O);
- группу АРДО, моделирующих параллельный перенос изображения $A \times A \times A$ (на 1 и на $\sqrt{A^3}$ шагов вдоль вектора Ox);
- АРДО, моделирующее преобразование симметрии содержимого изображения $A \times A \times A$ относительно плоскости Oyz ;
- группу АРДО, моделирующих параллельный поворот содержимого слоя $A \times A \times 1$ на углы, не кратные 90° , не вошедшие в 1-ю группу АРДО;
- группу АРДО, моделирующих центральное сжатие содержимого слоя $A \times A \times 1$ относительно Ox ;
- АРДО, моделирующее поворот изображения $\sqrt{A^3} \times \sqrt{A^3}$ на 90° вокруг O_{Π} .

Принцип выбора АРДО, моделирующих повороты на углы, не кратные 90° , состоит в следующем. С кубом $A \times A \times A$, произвольно вращающимся вокруг центра O , свяжем систему координат $Ox'y'z'$ (рис. 1, д), оси которой параллельны ребрам этого куба и проходят через центр O . Пусть i' — единичный вектор оси Ox' . Тогда при выполнении пространственных поворотов куба $A \times A \times A$ вокруг центра O , характеризующихся произвольными значениями углов α и β , конец вектора i' , восстановленного из центра O , будет перемещать-

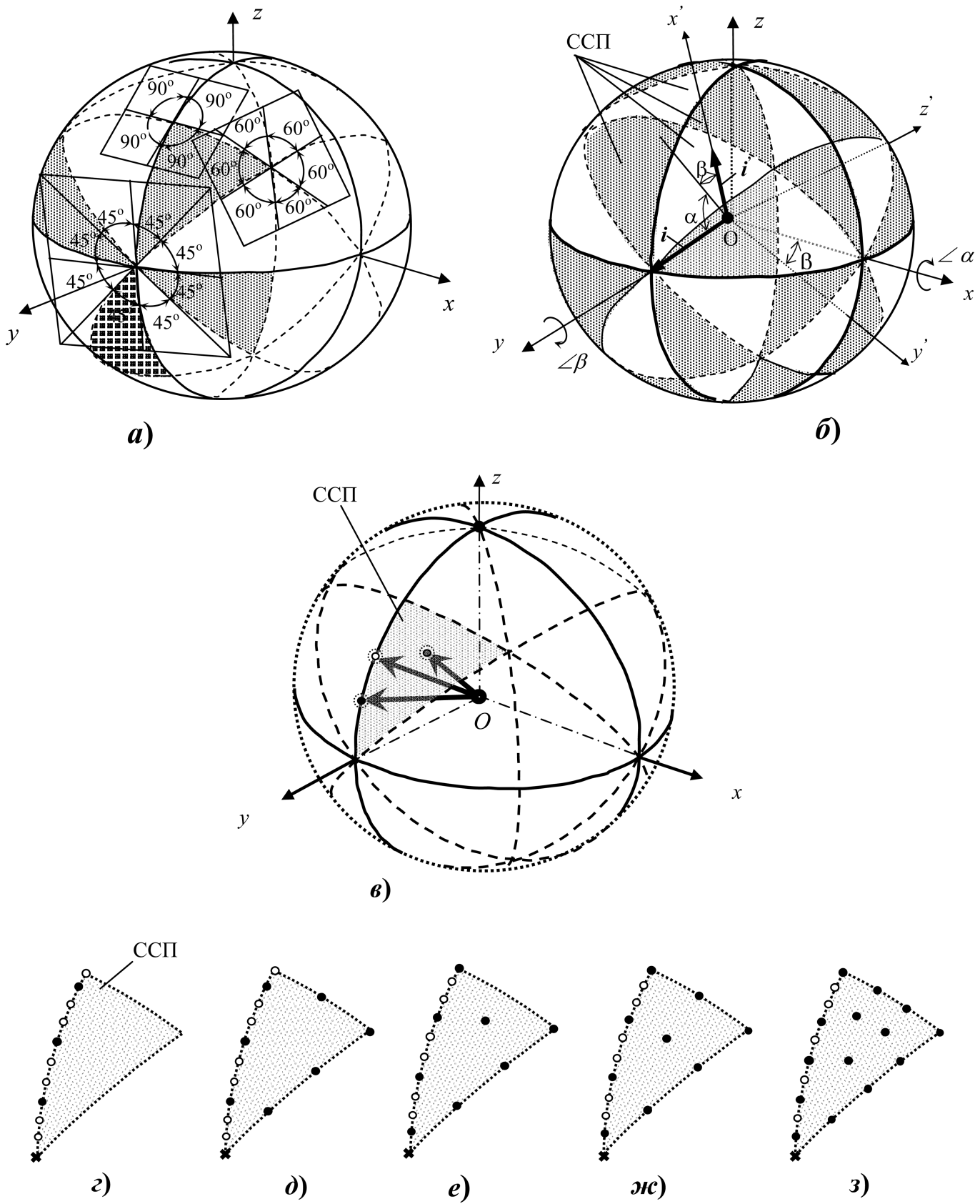


Рис. 2

ся по поверхности сферы единичного радиуса (рис. 2, б). Разобьем поверхность указанной сферы на 48 однотипных сегментов сферической поверхности (ССП). Для этого каждый участок поверхности сферы, отсекаемый октантом неподвижной системы координат $Oxuz$, дополнительно рассечем еще на шесть однотипных СПП (рис. 2, а) тремя плоскостями, проходящими соответственно через оси Ox , Oy , Oz и расположенными под углом 45° к плоскостям Oxz , Oxy , Oxz соответственно. Полученные СПП имеют одинаковую площадь и переходят друг в друга при поворотах на углы, кратные 90° ; вокруг осей Ox , Oy , Oz и преобразованиях симметрии относительно плоскостей Oxz , Oxy , Oxz .

Выберем один из 48 указанных СПП. Множество векторов i' , чьи концы полностью покрывают данный СПП, соответствуют некоторому бесконечному множеству M' преобразований поворота. Это бесконечное множество M' поворотов моделируется в устройстве конечным множеством M (конечным спектром) ДО. Для моделирования поворотов на углы, не кратные 90° , аппаратно реализуются только ДО, входящие в указанное множество M .

В рассматриваемом варианте устройства только часть ДО из M реализуется аппаратно. При этом с помощью АРДО, относящихся к 1-й группе, моделируются только наиболее часто используемые повороты в алгоритмах обработки, а с помощью АРДО, относящихся к 4-й группе, — все остальные повороты, входящие в M . На рис. 2, в—з АРДО 1-й группы представлены черными кружками на поверхности СПП, а АРДО 4-й группы — светлыми кружками. Возможные варианты базисов АРДО поворота (на углы, не кратные 90°) показаны на рис. 2, г—з. Здесь АРДО, относящиеся к 1-й группе и выполняемые с максимальной параллелизацией, выбраны из условия равномерности распределения концов соответствующих векторов i' (черные кружки) на поверхности указанной выше сферы единичного радиуса.

Устройство представляет собой многопроцессорную SIMD-систему. Основным вычислительным ядром устройства является система взаимо-

связанных матриц различного назначения и разрешения со сверхбольшим числом проводных связей, функционально и конструктивно объединенная в блок обработки 2-мерных и 3-мерных изображений (БОИ). Процессоры объединены в матрицу $A \times A \times A$ и соединены между собой связями, аппаратно реализующими ДО над содержимым бинарных изображений, хранящихся в памяти матрицы $A \times A \times A$. Здесь же располагаются распределенный параллельный A^3 -входовый сумматор и распределенный параллельный A^3 -входовый формирователь КЗБИ. БОИ управляется программно посредством блока управления (БУ), представляющего собой обычный компьютер. Система команд состоит из двух групп команд. Первую группу образует система команд самого БУ. Вторая группа команд используется для организации высокопараллельной обработки 2D- и 3D-изображений внутри БОИ. Именно наличие второй группы команд позволяет организовать высокопараллельную обработку 2D- и 3D-изображений, предложить новые, более эффективные алгоритмы такой обработки. Эти алгоритмы компактны, содержат небольшое число команд, время их выполнения не зависит от геометрической сложности обрабатываемых объектов.

Список литературы

1. А. с. 1456965 СССР, МКИ G 06F 15/16. Устройство для параллельной обработки трехмерных сцен / В. А. Бимаков, опубл. 07.02.89. Бюл. 5.
2. А. с. 1612307 СССР, МКИ G 06F 15/16, 15/62. Устройство для параллельной обработки трехмерных сцен / В. А. Бимаков, опубл. 07.12.90. Бюл. 45.
3. А. с. 1817108 СССР, МКИ G 06F 15/62. Устройство обработки изображений / В. А. Бимаков, опубл. 23.05.93. Бюл. 19.
4. А. с. 1817109 СССР, МКИ G 06F 15/62. Устройство для параллельной обработки трехмерных сцен / В. А. Бимаков, опубл. 23.05.93. Бюл. 19.
5. Патент 2289161 Российская Федерация, МПК G06T 1/00, G06F 15/173. Устройство обработки двумерных и трехмерных изображений / В. А. Бимаков, Е. В. Бимаков, опубл. 10.12.2006. Бюл. 34.
6. Бимаков В. А., Бимаков Е. В. Устройство управления автономным роботом // Мехатроника, автоматизация, управление: Материалы конф. Таганрог: Изд-во ТТИ ЮФУ, 2007. С. 340.
7. Колмогоров А. Н., Фомин С. В. Элементы теории функций и функционального анализа. М.: Наука, 1981.
8. Хокни Р., Джессхоуп К. Параллельные ЭВМ. М.: Радио и связь, 1986.

О. П. Архипов, канд. техн. наук, директор,
З. П. Зыкова, канд. физ.-мат. наук, зав. лаб.,
 Орловский филиал ИПИ РАН,
 e-mail: ofran@orel.ru

Обеспечение идентичности результатов печати на различных цветных принтерах

Рассматривается задача обеспечения максимальной идентичности отпечатков цветных изображений, полученных при печати на различных принтерах, обладающих стабильной цветопередачей. Для ее решения предлагается применять новый метод, основанный на тестировании цветопередачи, описании и согласовании цветовых пространств принтера. Предлагаемый метод может быть использован не только производителями, но и обычными пользователями принтеров. Производители могут применить его при написании драйверов принтеров в части определения функции цветodelения. Пользователи принтеров могут воспользоваться методом при наличии специального программного обеспечения (ПО). Специальное ПО, переопределяя функцию цветodelения, позволяет преобразовывать изменяющиеся RGB-изображения в такие, отпечатки которых на разных принтерах являются максимально идентичными.

Ключевые слова: цветная печать, сканирование, отпечатки цветных изображений, цветовые пространства, цветопередача.

Введение

Известно, что при печати одного и того же RGB-изображения на разных принтерах (даже одной фирмы-производителя) отпечатки могут значительно отличаться друг от друга, а в разных областях применения (репродуцирование на принтерах по имеющемуся образцу, допечатная подготовка в полиграфии и т. д.) требуется идентичность отпечатков, полученных на разных принтерах. В связи с этим компания Hewlett Packard представила интегрированную систему, которая, в частности, обеспечивает идентичную цветопередачу на разных устройствах цифровой печати [1–3]. Пока вследствие дороговизны, обусловленной применением спектрофотометрических датчиков, система доступна только для графической индустрии и профессиональных фотографов.

Для обеспечения идентичности отпечатков на разных принтерах мы предлагаем систему, использующую вместо спектрофотометрических датчиков обыкновенные цветные офисные сканеры. Это даст возможность доступа к достижению идентичности всем пользователям компьютерных систем с цвет-

ной периферией, обеспечивающих стабильность цветопередачи (печать + сканирование).

Возможность замены спектрофотометра сканером обеспечивается тем, что в данной области применения не требуется аппаратно-независимых количественных измерений, и, как показывает наш опыт, можно ограничиться сравнением значений цветовой характеристики пикселей в аппаратно-зависимых цветовых пространствах.

Известно, что на современных технически совершенных принтерах стабильность печати достигается, по крайней мере, до исчерпания ресурса применяемых расходных материалов. В связи с этим предлагается осуществлять тестирование цветопередачи каждый раз, как только происходит замена/возобновление красителей или бумаги.

Постановка задачи

Пусть имеется принтер, обладающий стабильной печатью. Это означает, что при разных цветопередачах из RGB-пространства в Lab-пространство образы одних и тех же пикселей принадлежат одной и той же зоне толерантности, т. е. если

$$(R, G, B) \xrightarrow[k\text{-я цветопередача}]{} (L_k, a_k, b_k)$$

и

$$(R, G, B) \xrightarrow[n\text{-я цветопередача}]{} (L_n, a_n, b_n),$$

то при любых k и n пиксели (L_k, a_k, b_k) и (L_n, a_n, b_n) удовлетворяют условию

$$\rho((L_k, a_k, b_k), (L_n, a_n, b_n)) \leq \Delta E, \quad (1)$$

где ρ — расстояние между пикселями в Lab-пространстве:

$$\begin{aligned} \rho((L_k, a_k, b_k), (L_n, a_n, b_n)) &= \\ &= \sqrt{(L_k - L_n)^2 + (a_k - a_n)^2 + (b_k - b_n)^2}, \end{aligned}$$

где ΔE — коэффициент цветоразличия.

Пусть на принтерах P' и P'' , обладающих стабильной печатью, напечатано произвольное изображение Prototype, состоящее из пикселей RGB-пространства:

$$\text{Prototype} = \{(R_{ij}, G_{ij}, B_{ij})\}, 0 \leq R_{ij}, G_{ij}, B_{ij} \leq 255, 1 \leq i \leq I, 1 \leq j \leq J,$$

и известны образы пикселей изображения в Lab-пространстве

$$(R_{ij}, G_{ij}, B_{ij}) \xrightarrow[P']{\text{цветопередача}} (L'_{ij}, a'_{ij}, b'_{ij});$$

$$(R_{ij}, G_{ij}, B_{ij}) \xrightarrow[P'']{\text{цветопередача}} (L''_{ij}, a''_{ij}, b''_{ij}).$$

Если при любых i и j выполнены условия

$$\rho((L'_{ij}, a'_{ij}, b'_{ij}), (L''_{ij}, a''_{ij}, b''_{ij})) \leq \delta \Delta E, \quad (2)$$

где

$$2 \leq \delta \leq 6, \quad (3)$$

то в соответствии с [4, 5] печать на принтерах P' и P'' признается идентичной. В соответствии с европейским стандартом допуск при печати равен трем:

$$\delta \Delta E = 3,$$

что позволяет контролировать идентичность печати на разных печатающих устройствах, проверяя выполнение критерия (3).

Поскольку контроль идентичности печати в рамках данной работы предлагается осуществлять с помощью сканера, необходимо переформулировать определения (1)–(3) в терминах RGB-цветового пространства в соответствии с подходом, выработанным в [6–8] при исследовании цветопередачи в офисных компьютерных системах с цветной периферией.

Чтобы при цветопередачах из произвольного RGB-пространства в RGB-пространство применяемого сканера

$$(R, G, B) \xrightarrow{k\text{-я цветопередача}} (r_k, g_k, b_k);$$

$$(R, G, B) \xrightarrow{n\text{-я цветопередача}} (r_n, g_n, b_n)$$

пиксели (r_k, g_k, b_k) и (r_n, g_n, b_n) принадлежали одной и той же зоне толерантности, они должны быть цветонеразличаемыми. В соответствии с [9] различие в 1–4 младших битах RGB-координат не приводит к цветовому различию пикселей, что позволяет в самом общем случае оценить сверху величину цветоразличия и сформулировать аналог критерия стабильности печати (1) в терминах RGB-пространства:

$$\rho((r_k, g_k, b_k), (r_n, g_n, b_n)) \leq \Delta E' \leq 16, \quad (4)$$

где ρ — округленное до ближайшего целого числа расстояние между пикселями в RGB-пространстве:

$$\rho((r_k, g_k, b_k), (r_n, g_n, b_n)) = \left[\sqrt{\frac{(r_k - r_n)^2 + (g_k - g_n)^2 + (b_k - b_n)^2}{3}} \right],$$

где $\Delta E'$ — коэффициент цветоразличия в RGB-пространстве сканера.

Пусть на принтерах P' и P'' , обладающих свойством стабильности, напечатано изображение Prototype. В случае, когда отпечаток сканируется, размеры получаемых изображений зависят от разрешения изображения Prototype, разрешения принтера и сканера. Рассмотрим изображения, размеры которых при цветопередаче остаются неизменными:

$$(R_{ij}, G_{ij}, B_{ij}) \xrightarrow[\text{цветопередача}]{P'} (r'_{ij}, g'_{ij}, b'_{ij});$$

$$(R_{ij}, G_{ij}, B_{ij}) \xrightarrow[\text{цветопередача}]{P''} (r''_{ij}, g''_{ij}, b''_{ij}).$$

Если при любых i и j выполнены условия

$$\rho((r'_{ij}, g'_{ij}, b'_{ij}), (r''_{ij}, g''_{ij}, b''_{ij})) \leq 2\Delta E',$$

то печать на принтерах P' и P'' является идентичной.

Априори значение коэффициента $\Delta E'$ неизвестно. В качестве его приближенного значения предлагается использовать следующую величину $\Delta E''$, в дальнейшем называемую *коэффициентом нестабильности*:

$$\begin{aligned} \Delta E' &\approx \Delta E'' = \\ &= \max_{i,j,k,n} \rho((r_{ijk}, g_{ijk}, b_{ijk}), (r_{ijn}, g_{ijn}, b_{ijn})). \end{aligned}$$

Если

$$\Delta E'' > 16,$$

то следует признать, что печать является нестабильной, а принтер не пригоден для решения задачи идентичной печати. В противном случае, вычислив величину $\Delta E''$ на предварительном тестировании цветопередачи, можно контролировать стабильность печати, проверяя выполнение условия

$$\rho((r_k, g_k, b_k), (r_n, g_n, b_n)) \leq \Delta E''. \quad (5)$$

Определим критерий идентичности печати аналогично (2):

$$\rho((r'_{ij}, g'_{ij}, b'_{ij}), (r''_{ij}, g''_{ij}, b''_{ij})) \leq \Delta E'_1 + \Delta E'_2, \quad (6)$$

где $\Delta E'_1$ — коэффициент нестабильности первого принтера, а $\Delta E'_2$ — второго.

Замечание. Если $\Delta E'_1 = \Delta E'_2$, то правая часть (6) соответствует (2) при выборе минимального значения δ в диапазоне (3).

Задача. Пусть у пользователя имеется сканер и два принтера (P' и P''), которые обеспечивают стабильную цветопередачу в смысле (5). Пусть изображение Prototype напечатано на принтере P' и получен отпечаток Print'. Требуется на принтере P'' получить отпечаток Print'' изображения Prototype, идентичный отпечатку Print' по критерию (6).

Замечание. Если коэффициенты нестабильности применяемых принтеров различаются значительно, то на принтере с худшей стабильностью априори нельзя добиться стабильности другого принтера, поэтому задачу будем решать только для случая, когда значение коэффициента нестабильности $\Delta E'_2$ принтера P'' меньше, равно или незначительно превышает значение коэффициента $\Delta E'_1$.

Решение задачи

Поскольку изображение Prototype состоит из произвольных пикселей цветового RGB-куба, то для решения поставленной задачи достаточно вы-

числить значения коэффициентов нестабильности $\Delta E_1''$ и $\Delta E_2''$ для каждого из принтеров и добиться выполнения условия (6) для всех пикселей с координатами

$$\{(i, j, k)\}, \quad i, j, k = 0, 1, \dots, 255. \quad (7)$$

Пусть для пикселей из (7) получены значения цветовой характеристики при цветопередаче и с помощью принтера P' :

$$\{(r_i', g_j', b_k')\}, \quad i, j, k = 0, 1, \dots, 255$$

и с помощью принтера P'' :

$$\{(r_i'', g_j'', b_k'')\}, \quad i, j, k = 0, 1, \dots, 255.$$

Если при любых значениях индексов выполнено соотношение

$$\rho((r_i', g_j', b_k'), (r_i'', g_j'', b_k'')) \leq \Delta E_1'', \quad (8)$$

$$0 \leq i, j, k \leq 255,$$

то очевидно, что отпечатки пикселей на первом и втором принтере являются цветонеразличаемыми. Следовательно, тем более выполнено определение идентичной печати (6) и никаких дополнительных действий для обеспечения идентичности печати не требуется.

Предположим, что соотношение (8) выполнено не для всех значений индексов, а именно, пусть для пикселя $(R_{i'}, G_{j'}, B_{k'})$ с цветовыми характеристиками (r_i', g_j', b_k') и (r_i'', g_j'', b_k'') справедливо такое соотношение:

$$\rho((r_i', g_j', b_k'), (r_i'', g_j'', b_k'')) > \Delta E_1''.$$

Для решения задачи необходимо организовать поиск в целях отыскания такого пикселя $(R_{i''}, G_{j''}, B_{k''})$, который при печати на втором принтере имеет цветовую характеристику (r_i'', g_j'', b_k'') , более близкую к цветовой характеристике (r_i', g_j', b_k') пикселя $(R_{i'}, G_{j'}, B_{k'})$ при печати на первом принтере, а именно:

$$\rho((r_i', g_j', b_k'), (r_i'', g_j'', b_k'')) = \min_{0 \leq i, j, k \leq 255} \rho((r_i', g_j', b_k'), (r_i'', g_j'', b_k'')).$$

Отсюда

$$\rho((r_i', g_j', b_k'), (r_i'', g_j'', b_k'')) \leq \rho((r_i', g_j', b_k'), (r_i'', g_j'', b_k'')). \quad (9)$$

Конечно, выполнение соотношения (9) не гарантирует выполнение неравенства (8), однако, во всяком случае в текущих условиях, отпечаток пикселя $(R_{i''}, G_{j''}, B_{k''})$, полученный на втором принтере, является максимально идентичным отпечатку пикселя $(R_{i'}, G_{j'}, B_{k'})$, полученному на первом принтере.

Если же условие (8) выполнено, т. е.

$$\rho((r_i', g_j', b_k'), (r_i'', g_j'', b_k'')) \leq \Delta E_1'', \quad (10)$$

то для значения цветовой характеристики $(\underline{r}_i'', \underline{g}_j'', \underline{b}_k'')$ пикселя $(R_{i''}, G_{j''}, B_{k''})$, полученной при любой следующей процедуре печати на втором принтере, имеем

$$\begin{aligned} & \rho((r_i', g_j', b_k'), (\underline{r}_i'', \underline{g}_j'', \underline{b}_k'')) \leq \\ & \leq \rho((r_i', g_j', b_k'), (r_i'', g_j'', b_k'')) + \\ & + \rho((r_i'', g_j'', b_k''), (\underline{r}_i'', \underline{g}_j'', \underline{b}_k'')) \leq \\ & \leq \Delta E_1'' + \rho((r_i'', g_j'', b_k''), (\underline{r}_i'', \underline{g}_j'', \underline{b}_k'')). \end{aligned}$$

Поскольку печать является стабильной, а коэффициент нестабильности на принтере P'' равен $\Delta E_2''$, то

$$\rho((r_i'', g_j'', b_k''), (\underline{r}_i'', \underline{g}_j'', \underline{b}_k'')) \leq \Delta E_2'',$$

и окончательно имеем

$$\rho((r_i', g_j', b_k'), (\underline{r}_i'', \underline{g}_j'', \underline{b}_k'')) \leq \Delta E_1'' + \Delta E_2'',$$

что и требовалось доказать для подтверждения идентичности печати.

Замечание. Если цветовые пространства принтеров различаются значительно, то выполнения условия (10) строго для всех рассматриваемых пикселей добиться не удастся. Однако ясно, что в процессе решения будет обеспечена максимально возможная идентичность печати. Требуется экспертная оценка в этом случае, чтобы принять решение о допустимости полученных результатов для практического применения.

Пример 1

Для примера были рассмотрены цветопередачи, которые реализуются с помощью специального ПО (в дальнейшем, программа CtrI), разработанного нами для демонстрации возможностей предлагаемых методов, на программно-технической базе из двух принтеров (P' — HP Color Laser Jet 1550L и P'' — HP Color Laser Jet 4700n) и сканера Fujitsu fi-60F.

Поскольку общее число пикселей в RGB-кубе превышает 16,5 млн цветов, то получение для всех них значений цветовой характеристики представляет серьезную техническую проблему. В связи с этим при рассмотрении примера мы ограничились рассмотрением меньшего числа пикселей:

$$\{(R_i, G_j, B_k)\} = \{(i \cdot 17, j \cdot 17, k \cdot 17)\}, \quad i, j, k = 0, 1, \dots, 15. \quad (11)$$

Выбранные пиксели равномерно распределены в цветовом RGB-кубе.

Вместо изображения Prototype, пиксели которого являются произвольными компонентами RGB-куба, необходимо рассмотреть его модификацию — изображение с палитрой (11) из 4096 цветов. Заметим, что алгоритмы оптимизации и псевдосмещения цветов [10] в этом случае делают разницу между оригиналом и модификацией незаметной.

Было проведено предварительное тестирование для определения значений коэффициентов нестабильности при печати на выбранных принтерах. При проведении тестирования использовалось изображение Image_{Test}, имеющее размер 64×64 пикселя и состоящее из пикселей выбранной палитры (11).

Изображение Image_{Test} по пять раз было напечатано на каждом из принтеров. Полученные отпечатки были отсканированы, обработаны специальным образом так, чтобы выделить из сканов полезную часть, для каждой растровой точки полезной части были вычислены значения цветовой характеристики — координаты пикселей, из которых были составлены изображения Scan_{*i*'}, Scan_{*i*''}, *i* = 1, 2, ..., 5.

При попиксельном сравнении изображений Scan_{*i*'} и Scan_{*j*''}, *i, j* = 1, 2, ..., 5, *i* ≠ *j*, соответствующих каждой паре отпечатков, полученных на принтере *P'*, был определен диапазон изменения значений коэффициента нестабильности Δ*E*_{1''}:

$$12 \leq \Delta E_1'' \leq 13.$$

Аналогично при попиксельном сравнении изображений Scan_{*i*'} и Scan_{*j*''}, *i, j* = 1, 2, ..., 5, *i* ≠ *j*, соответствующих каждой паре отпечатков, полученных на принтере *P''*, был определен диапазон изменения значений коэффициента нестабильности Δ*E*_{2''}:

$$8 \leq \Delta E_2'' \leq 9.$$

Для дальнейших вычислений за значения коэффициентов нестабильности были приняты максимальные значения из вычисленных диапазонов:

$$\Delta E_1'' = 13 \text{ и } \Delta E_2'' = 9.$$

Соответственно, в правой части (6) было использовано значение

$$\Delta E_1'' + \Delta E_2'' = 13 + 9 = 22.$$

Заметим, что в соответствии с полученными результатами печать на принтере *P'* является менее стабильной по сравнению с печатью на принтере *P''*. Это связано с несовершенством применяемого механизма и технологии печати на принтере *P'*, что проявляется в периодическом появлении более светлых и темных горизонтальных полос при печати однотонных цветовых фрагментов.

Поскольку принтеры имеют коэффициенты нестабильности, значительно отличающиеся друг от друга, то на *P'* печать, идентичная печати на *P''*, априори не достижима. Решалась задача: на принтере *P''* достичь максимально идентичной печати по сравнению с печатью на принтере *P'*. Были построены таблицы перехода цветов:

- *T*₁ — при попиксельном сравнении изображений Scan_{1'} и Scan_{1''};
- *T*₂ — при попиксельном сравнении изображений Scan_{1'} и Scan_{2''};
- *T*₃ — при попиксельном сравнении изображений Scan_{2'} и Scan_{1''};
- *T*₄ — при попиксельном сравнении изображений Scan_{2'} и Scan_{2''}.

При построении таблиц вычислялись значения следующих величин:

- *k*₁ — число пикселей, для которых цветоразличие больше Δ*E*_{1''};
- *m*₁ — исходное максимальное значение цветоразличия;
- *k*₂ — число пикселей, для которых цветоразличие после поиска уменьшилось и не превышает Δ*E*_{1''};
- *k*₃ — число пикселей, для которых цветоразличие до Δ*E*_{1''} уменьшить не удалось;
- *m*₂ — максимальное значение цветоразличия после поиска.

Полученные результаты приведены в табл. 1.

Приведенные результаты близки и удовлетворительны. В каждом случае максимальное цветоразличие было уменьшено, что позволяет надеяться, что применение построенных таблиц перехода цветов приведет к получению на принтере *P''* печати, идентичной печати на принтере *P'*.

Было проведено преобразование изображения Image_{Test} в соответствии с каждой из таблиц перехода цветов *T*_{*i*}, *i* = 1, 2, 3, 4. Полученным при

Таблица 1

Результаты, полученные при построении таблиц перехода цветов

№ таблицы	<i>k</i> ₁	<i>m</i> ₁	<i>k</i> ₂	<i>k</i> ₃	<i>m</i> ₂
1	3424	65	3423	1	14
2	3337	64	3337	0	—
3	3249	67	3249	0	—
4	3148	66	3148	0	—

Таблица 2

Результаты, полученные при проверке идентичности печати

Название изображения	Название изображения	Расстояние между изображениями
Scan _{1'}	Scan ₁	19
Scan _{1'}	Scan ₂	19
Scan _{2'}	Scan ₃	22
Scan _{2'}	Scan ₄	22

этом изображениям $Image_i$, $i = 1, 2, 3, 4$, отпечатанным на принтере P'' , были после сканирования и обработки сканов отпечатков (замены сканов отпечатков растровых точек на значения цветовых характеристик) сопоставлены сканы $Scan_i$, $i = 1, 2, 3, 4$.

Для практического подтверждения того, что предлагаемый метод позволяет добиться большей идентичности печати по образцу на принтере P'' по сравнению с печатью на принтере P' , было проведено попиксельное сравнение изображений $Scan'_1$ и $Scan_i$, $i = 1, 2$, а также $Scan'_2$ и $Scan_i$, $i = 3, 4$. А именно, вычислялось расстояние между изображениями, равное максимуму расстояний между соответствующими парами пикселей. Полученные результаты приведены в табл. 2.

Расстояние между изображениями удовлетворяет критерию (6), следовательно, в рассмотренном примере поставленная задача решена с помощью предлагаемого метода.

Пример 2

На принтере HP Color Laser Jet 4700n эмулировалось два разных принтера P' и P'' путем печати из разных программ:

- Adobe Photoshop Elements;
- специального ПО ClrI.

В соответствии с результатами, полученными ранее, в качестве значения коэффициента неустойчивости используется значение $\Delta E''_2 = 9$, а допуски при печати составляют 18 единиц.

Изображение $Image_{Test}$ было напечатано из Adobe Photoshop Elements 2.0 в стандарте по умолчанию — sRGB, а полученный отпечаток отсканирован на применяемом сканере. Затем скан был обработан в целях получения значений цветовой характеристики, из которых было составлено изображение $Scan'_1$. В качестве образца печати из программы ClrI было использовано ранее полученное изображение $Scan''_1$.

Поскольку вычисленное расстояние между изображениями $Scan'_1$ и $Scan''_1$

$$\rho(Scan'_1, Scan''_1) = 12$$

не превосходит 16, то печать из двух применяемых программ следует признать идентичной. В процессе вычислений были также получены результаты, приведенные в табл. 3.

Из табл. 3 видно, что цветоразличие подавляющего числа (больше 99 %) пар пикселей имеют значения характеристики цветоразличия, не превосходящее значения коэффициента неустойчивости.

Итак, в рассматриваемом случае имеет место идентичная печать. Иначе обстоит дело при печати Adobe Photoshop Elements 2,0 в стандарте по умолчанию Adobe RGB (1998).

Изображение $Image_{Test}$ было напечатано из Adobe Photoshop Elements 2.0 в стандарте Adobe RGB (1998), а полученный отпечаток отсканирован на применяемом сканере. Затем скан был обработан в целях получения значений цветовой характеристики, из которых было составлено изображение $Scan'_2$. В качестве образца печати из программы ClrI было по-прежнему использовано ранее полученное изображение $Scan''_1$.

Вычисленное расстояние между изображениями $Scan'_2$ и $Scan''_1$

$$\rho(Scan'_2, Scan''_1) = 57$$

значительно превосходит 16, поэтому печать в этом случае нельзя признать идентичной.

В процессе вычислений были также получены результаты, приведенные в табл. 4.

Из табл. 4 видно, что цветоразличие более 10 % пар пикселей недопустимо велико. Чтобы добиться идентичной печати на основании изображений $Scan'_2$ и $Scan''_1$ была построена таблица перехода цветов T_5 . В соответствии с ней было модифицировано изображение $Image_{Test}$. Полученному при этом изображению $Image_5$, отпечатанному из программы ClrI, после сканирования и обработки скана отпечатка был сопоставлен скан $Scan_5$.

Для практического подтверждения того, что предлагаемый метод позволил добиться идентичности печати, было проведено попиксельное сравнение изображений $Scan'_2$ и $Scan_5$. А именно, было вычислено расстояние между изображениями

$$\rho(Scan'_2, Scan_5) = 17.$$

Таблица 3

Цветоразличие пикселей изображений $Scan'_1$ и $Scan''_1$

Диапазон цветоразличия	Число пикселей	Число пикселей в % от общего числа
[0,8]	4006	97,80
[9,9]	58	1,42
[10, 10]	26	0,64
[11, 11]	5	0,12
[12, 12]	1	0,02

Таблица 4

Цветоразличие пикселей изображений $Scan'_2$ и $Scan''_1$

Идентификатор цветоразличия	Число пикселей	Число пикселей в % от общего числа
[0,8]	1998	48,77
[9,9]	317	7,73
[10, 10]	264	6,44
[11, 11]	244	5,95
[12, 12]	221	5,40
[13, 13]	194	4,73
[14, 14]	169	4,15
[15, 15]	126	3,08
[16, 16]	121	2,95
[17, 17]	442	10,80

Таблица 5
Цветоразличие пикселей изображений Scan₂ и Scan₅

Идентификатор цветоразличия	Число пикселей	Число пикселей в % от общего числа
[0,8]	3468	84,68
[9,9]	258	6,30
[10, 10]	130	3,17
[11, 11]	94	2,29
[12, 12]	74	1,82
[13, 13]	34	0,83
[14, 14]	23	0,56
[15, 15]	12	0,29
[16, 16]	2	0,04
[17, 17]	1	0,02

Поскольку критерий идентичности печати (6) выполнен, следует признать, что поставленная задача с помощью предлагаемого метода решена. Полученные при этом результаты приведены в табл. 5.

Из табл. 5 видно, что цветоразличие подавляющего числа (больше 91 %) пар пикселей имеют значение характеристики цветоразличия, не превосходящее значения коэффициента нестабильности, и всего у одной пары пикселей цветоразличие на единицу превосходит число 16.

Заключение

Таким образом, предложен новый метод обеспечения максимальной идентичности отпечатков цветных изображений, полученных при печати на различных принтерах, обладающих стабильной цветопередачей. Предлагаемый метод может быть

использован не только производителями, но и обычными пользователями принтеров. Производители могут применить его при написании драйверов принтеров в части определения функции цветоделения. Пользователи принтеров могут воспользоваться методом при наличии специального ПО. Специальное ПО, переопределяя функцию цветоделения, позволяет преобразовывать имеющиеся RGB-изображения в такие, отпечатки которых на разных принтерах являются максимально идентичными.

Список литературы

1. **Единый цвет** // Курсив. 2007. № 2. С. 25.
2. **НР представила** технологию DreamColor // Компьютерпресс. 2007. № 4. С. 154.
3. **Асмаков С.** Технология DreamColor // Компьютерпресс. 2007. № 7. С. 190.
4. **Дегтярь Е.** Каждый желает знать... // Полиграфия. 2000. № 3. С. 69–70.
5. **X-Rite.** Руководство по работе с цветом // КомпьюАрт. 1999. № 12. С. 62–65.
6. **Архипов О. П.** Преобразование форм представления цветных графических изображений в офисной компьютерной системе // Научные технологии. 2004. Т. 5. № 7. С. 36–42.
7. **Архипов О. П.** Алгоритмы идентификации визуально не различаемых по цвету пикселей // Научные технологии. 2005. Т. 6. № 6. С. 70–75.
8. **Архипов О. П., Архипов П. О., Захаров В. Н., Зыкова З. П.** Аппроксимация цветных изображений для получения машиночитаемых объектов // Информационные технологии. 2007. № 7. С. 53–57.
9. **Городецкий В. И., Самойлов В. В.** Стеганография на основе цифровых изображений // Информационные технологии и вычислительные системы. 2001. № 2/3. С. 51–64.
10. **Просис Дж.** Методы цветовой оптимизации и псевдосмещения цветов // PC MAGAZINE. 1995. № 7. С. 131–135.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

УДК 004.414.23

А. И. Колдаев, аспирант,
Невинномысский технологический институт
(филиал Северо-Кавказского государственного технического университета),
e-mail: ventilator83@mail.ru

Моделирование интеллектуальной системы поддержки принятия решений при управлении технологическим процессом

Предложена интеллектуальная система нейронечеткого вывода для решения задачи поддержки принятия решений при управлении технологическим процессом реакторной установки гидрирования бутанола.

Ключевые слова: нечеткая логика, нейронечеткая сеть.

Введение

Для подавляющего большинства как искусственных, так и естественных объектов управления, которыми необходимо управлять, построение точных математических моделей практически невозможно

ввиду их плохой формализуемости. В случае плохой формализуемости объекта управления особый интерес вызывают системы, построенные на интеллектуальных принципах (нечеткая логика, экспертные системы, искусственные нейронные сети) [1, 2].

В данной статье рассмотрена система поддержки принятия решений при управлении технологическим процессом реакторной установки гидрирования бутанола, основанная на применении гибридной системы, где выводы делаются на основе аппарата нечеткой логики, но соответствующие функции принадлежности подстраиваются с использованием алгоритмов обучения нейронных сетей [1].

Нейронные сети и нечеткая логика — принципиально различные конструкции — являются универсальными аппроксиматорами сложных (нелинейных) функциональных зависимостей во многих интеллектуальных задачах [1, 2].

Главной особенностью нейронных сетей является их способность к обучению. Достоинством нечеткой логики является возможность использования экспертных знаний о структуре объекта в виде лингвистических высказываний: если "входы", то "выход" [1].

Объединение нечеткой логики с нейронными сетями дает принципиально новое качество. Получаемая в результате такого объединения нейрон-нечеткая сеть обладает двумя важнейшими интеллектуальными свойствами:

- лингвистичностью, т. е. использованием знаний на естественном языке;
- обучаемостью в реальном масштабе времени.

Кроме того, нечеткие контроллеры могут работать с не полностью описанными системами с неизвестной динамикой, так как для них не требуется априорная математическая модель объекта управления.

На рис. 1 представлена структурная схема исследуемой установки гидрирования бутанола. Бутанол с помощью насоса через подогреватель подается в верхнюю часть реактора. Водород также через подогреватель подается в реактор, где происходит процесс гидрирования бутанола. Из нижней части реактора бутанол поступает в сепаратор, где происходит его отделение от водорода, не вступившего в реакцию.

Особенностью исследуемой реакторной установки, как и большинства реакторов, является наличие в ней большого числа нелинейных связей, и "традиционные" методы управления к ней неприменимы. В таком случае возникает необходимость использовать методы искусственного интеллекта.

На рис. 2 приведена схема управления исследуемым объектом. Информация о текущем состоянии управляемого процесса поступает на вход нейрон-нечеткой сети, выходное значение которой помогает оператору установки принять решение в управлении объектом.

В качестве исходных данных об объекте воспользуемся информацией, содержащейся в технологическом регламенте производства бутанола,

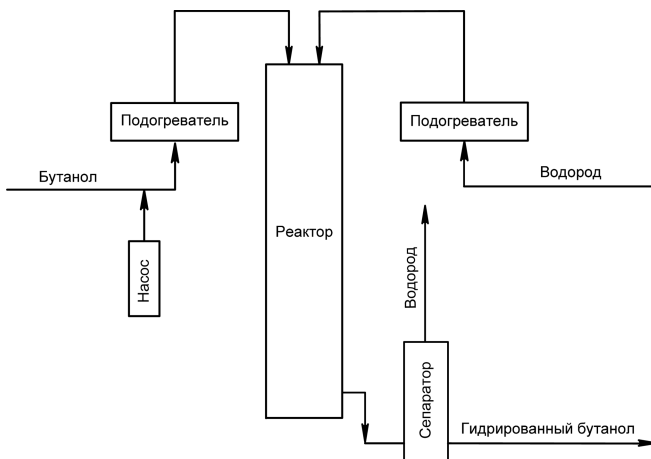


Рис. 1. Структурная схема установки гидрирования бутанола



Рис. 2. Схема управления процессом

в котором приведены нормы технологического процесса. Нормы процесса приведены в табл. 1.

Любое отклонение какого-либо параметра от нормы приводит к нарушению технологического процесса. При этом от оператора установки требуется принять решение, способное исправить сложившуюся ситуацию. Исследуемую реакторную установку можно рассмотреть как нелинейный объект вида

$$y = f(x_1, x_2, x_3), \quad (1)$$

для которого связь "входы x_i — выход y " можно представить в виде экспертной матрицы знаний (табл. 2). Этой матрице соответствует нечеткая база знаний

$$\begin{aligned} &\text{ЕСЛИ } [(x_1 = a_1^{j1}) \text{ И } (x_2 = a_2^{j1}) \\ &\text{И } (x_3 = a_3^{j1})] \text{ (с весом } w_{j1}) \end{aligned}$$

...

Таблица 1

Нормы технологического процесса

Контролируемый параметр	Нормы и технические показатели
Температура в реакторе	120—150 °С
Давление водорода в реакторе	Не менее 1,9 МПа
Перепад давления "верх—низ" в реакторе	Не более 0,1 МПа

Таблица 2

Экспертная матрица знаний

Номер правила	ЕСЛИ "входы"			ТО "выход"	Вес правила
	x_1 (температура в реакторе, °С)	x_2 (давление водорода, МПа)	x_3 (перепад давления "верх-низ" в реакторе, МПа)		
1	150	1,9	0,095	0	1
...
6	120	2,5	0,01	0	1
7	151	1,9	0,095	1	1
...
13	160	1,9	0,095	1	1
14	150	1,89	0,095	2	1
...
20	120	1,5	0,01	2	1
21	150	1,9	0,105	3	1
...
27	120	2,5	0,105	3	1

ИЛИ $[(x_1 = a_1^{jk_j})$ И $(x_2 = a_2^{jk_j})$

И $(x_3 = a_3^{jk_j})]$ (с весом w_{jk})

ТО $y = d_j; j = 1, m; i = 1, n,$

где a_i^p — лингвистический терм, оценивающий переменную x_i в строке $p = k_j$; k_j — число строк-конъюнкций, соответствующих классу d_j выходной переменной y (в данной задаче использовалось $k_j = 27$); w_{jp} — число в диапазоне $[0, 1]$, характеризующее субъективную меру уверенности эксперта относительно высказывания с номером $p = k_j$ [1]; n — число входных параметров (в нашем случае $n = 3$); m — число уровней квантования диапазона выходной переменной (в нашем случае $m = 4$, так как выходная переменная характеризуется четырьмя значениями $\{0, 1, 2, 3\}$ (см. графу «ТО "выход"» в табл. 2)).

Матрица знаний (табл. 2) представляет собой векторы входных значений основных контролируемых параметров технологического процесса. В табл. 2 значение графы «ТО "выход"»: "0" означает нормальное течение технологического процесса; "1" — температура в реакторе превышает 150 °С; "2" — прекращение подачи водорода или снижение его давления менее 1,9 МПа; "3" — увеличивается перепад давления в реакторе более 0,1 МПа.

Определение структуры нейронечеткой сети

Лингвистическую переменную, характеризующуюся набором параметров <"входы" — "выход">, можно представить в виде специальной нечеткой нейронной сети [1]. Нечеткой нейронной сетью обычно называют нейронную сеть, которая построена на основе многослойной архитектуры с

использованием нечеткого нейрона "И" и нечеткого нейрона "ИЛИ" [3, 4]. Нечетким нейроном "И" называется нейрон, в котором умножение веса w на вход x моделируется конормой, представляющей собой двуместную функцию $S(w, x)$, а сложение — нормой, представляющей собой двуместную функцию $T(w, x)$ [4].

В нечетком нейроне "И" сигналы x_i и веса w_i объединяются с помощью t -конормы при треугольной функции активации, а выход образуется с применением t -нормы (рис. 3, а).

В нечетком нейроне "ИЛИ" сигналы x_i и веса w_i объединяются с помощью t -нормы, а выход образуется с применением t -конормы (рис. 3, б).

При создании гибридной технологии кроме объединения систем по данным можно использовать нейрокомпьютинг для решения частной подзадачи настройки параметров функций принадлежности. В качестве функций принадлежности выбирается параметризованная функция формы, параметры которой настраиваются с помощью нейросетей. Таким образом, нечеткая нейронная сеть наряду с классическими нейронами должна включать в себя нечеткие нейроны "И" и "ИЛИ".

В данной работе была использована технология ANFIS (*Adaptive Network-based Fuzzy Inference System*) — адаптивная нейронная сеть, основанная на системе нечеткого вывода. Структура данной сети представлена на рис. 4. Слои нейронечеткой сети ANFIS выполняют следующие функции.

Слой 1 — входы объекта идентификации. Для исследуемого объекта рассматриваются три входных параметра (вход 1 — температура, вход 2 — давление, вход 3 — перепад давления). На входы сети подаются нормализованные сигналы от датчиков в виде безразмерных величин, характеризующих значения измеряемых параметров.

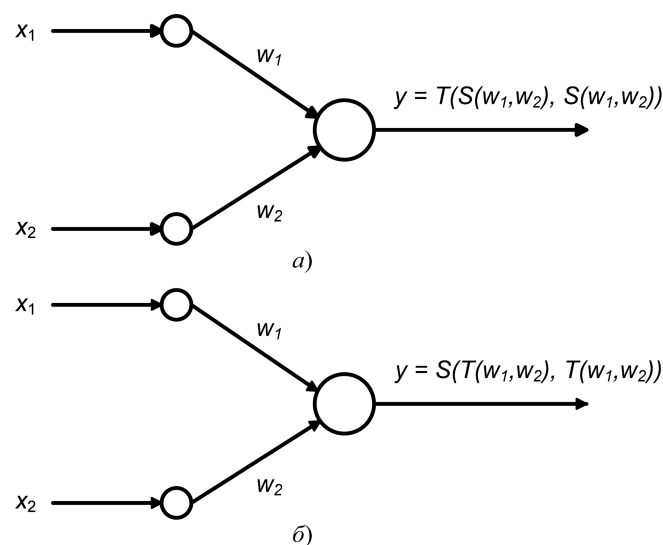


Рис. 3. Структура нечетких нейронов: а — "И"; б — "ИЛИ"

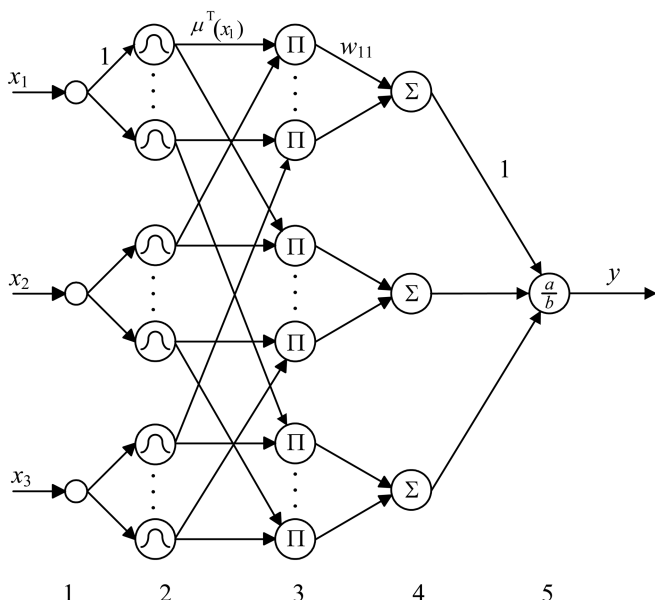


Рис. 4. Структура нейронечеткой сети

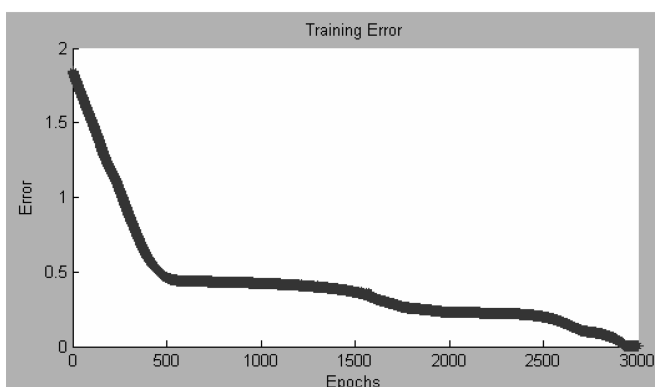


Рис. 6. График зависимости ошибки обучения от числа циклов обучения

Слой 2 представлен радиальными базисными нейронами и моделирует функции принадлежности.

Слой 3 — это слой И-нейронов, которые моделируют логическую связку И произведением $w_i = \mu^{T1}(x_1)\mu^{T2}(x_2)\mu^{T3}(x_3)$.

Слой 4 формирует значение выходной переменной

$$y(x_1, x_2, x_3) = \omega_i y_i = \omega_i (c_{i1}x_1 + c_{i2}x_2 + c_{i3}x_3).$$

Слой 5 выполняет операцию нахождения среднего, т. е. выполняет преобразование результатов нечеткого логического вывода в четкое число по формуле

$$y = \frac{w_1 y_1 + w_2 y_2 + w_3 y_3}{w_1 + w_2 + w_3}.$$

Далее находится ближайшее целое.

Дугам графа присваиваются следующие веса: единица (дуги между первым и вторым слоями);

функции принадлежности входа к нечеткому терму (дуги между вторым и третьим слоями); веса правил (дуги между третьим и четвертым слоями); единица (дуги между четвертым и пятым слоями) [2].

Принадлежность каждого точного значения к одному из термов лингвистической переменной определяется посредством функций принадлежности. Ее вид может быть абсолютно произвольным. Наиболее предпочтительными формами функции принадлежности являются треугольная и трапециевидная, поскольку нечеткий вывод значительно упрощается при использовании кусочно-линейных функций принадлежности [4, 5]. В данной работе применялись треугольные функции принадлежности.

В качестве алгоритма нечеткого вывода использовался алгоритм Сугено.

Моделирование системы принятия решений

Для моделирования гибридной сети используем пакет прикладных программ MatLab Fuzzy Logic Toolbox.

В данном пакете встроена технология ANFIS (*Adaptive Network-based Fuzzy Inference System*) — итерационная процедура для настройки функций принадлежности методом обратного распространения ошибки.

Приложение пакета MatLab ANFISedit позволяет сгенерировать данные из табл. 2.

Системой MatLab в соответствии с введенными условиями и параметрами формируется нейронечеткая сеть (рис. 5, см. третью сторону обложки), структура которой соответствует сети, представленной на рис. 4. Полученная структура сети имеет три входа (температура, давление, перепад давления), три внутренних слоя и один выход.

Последний слой сети преобразует результат нечеткого логического вывода, сделанного нейронечеткой сетью, в четкое число, которое должно соответствовать номеру инструкции, которой необходимо руководствоваться в той или иной сложившейся ситуации [1].

В качестве метода настройки параметров использовался метод "обратного распространения ошибки" (*back-propagation*) [4]; заданное число циклов обучения — 3000. Результат моделирования системы в виде графика ошибки сети в зависимости от числа проведенных циклов обучения представлен на рис. 6.

Результаты

После обучения выходное значение гибридной сети определяет характер протекания процесса гидрирования бутанола согласно матрице знаний (табл. 3). Число итераций обучения для трех входных параметров составило 2900 эпох; время обу-

Выходные значения гибридной сети после обучения

Вход 1 (температура, °С)	Вход 2 (давление, МПа)	Вход 3 (перепад давления, МПа)	Выход
140	2,0	0,07	0
133	2,3	0,09	0
153	2,1	0,07	1
160	1,9	0,09	1
145	1,7	0,08	2
125	1,8	0,02	2
130	2,2	0,11	3
137	2,1	0,13	3

чения для Pentium II 800 Mhz — 4 мин, ошибка обучения составила 0,00179, или 0,179 %.

Выход сети определяется четырьмя значениями {0, 1, 2, 3}.

Выходное значение сети "0" означает, что отклонений в процессе не наблюдается, значение "1" — превышение температуры внутри реактора выше нормы; значение "2" — снижение давления в реакторе ниже нормы; значение "3" — перепад давления в реакторе выше нормы.

Применив пакет расширения FlexTool for MatLab, можно сгенерировать код микроконтроллера для реализации полученной гибридной сети. Поставляемое с FlexTool for MatLab программное ядро совместимо с такими контроллерами, как 8096BH, 8096-90, 90196 KR, 90196MC, 80196NT/NQ и др. [2].

Выводы

Определена структура нейронечеткой сети для аппроксимации реакторной установки гидрирования бутанола.

Проверка построенной нечеткой модели гибридной сети показывает достаточно высокую степень ее адекватности реальным исходным данным (см. табл. 1), что позволяет сделать вывод о возможности ее практического применения для задачи поддержки принятия решений при управлении технологическим процессом любого объекта, характеризующегося нелинейной зависимостью.

Список литературы

1. Рогштейн А. П., Митюшкин Ю. И. Нейролингвистическая идентификация нелинейных зависимостей // Кибернетика и системный анализ. 2000. № 2. С. 37—44.
2. Дьяконов В., Круглов В. Математические пакеты расширения MatLab. СПб.: Питер, 2001.
3. Ярушкина Н. Г. Научная сессия МИФИ-2005 // VII Всероссийская научная конференция "Нейроинформатика-2005": Лекции по нейроинформатике. М.: МИФИ, 2005.
4. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика. М.: Горячая линия — Телеком, 2001. 382 с.
5. Омату С. Нейроуправление и его приложения. Кн. 2: Пер. с англ. Н. В. Батина / Под ред. А. И. Галушкина, В. А. Птичкина. М.: ИПРЖР. 2000. (Нейрокомпьютеры и их применение).

УДК 004.896

Т. И. Михеева, д-р техн. наук, доц.,
А. А. Потапкин, студент,
Р. В. Демьяненко, аспирант,
 Самарский государственный
 аэрокосмический университет,
 e-mail: mikheevati@mail.ru

Адаптивная кластеризация графа улично-дорожной сети

Рассматривается метод сжатия атрибутивного графа улично-дорожной сети, основывающийся на адаптивной нечеткой кластеризации. Приводятся критерии качества адаптивной кластеризации графа улично-дорожной сети и оценка степени сжатия атрибутивных данных.

Ключевые слова: граф улично-дорожной сети, нечеткая кластеризация, сжатие графа, транспортный маршрут.

Введение

Развитие международного рынка производства продукции стимулирует совершенствование транспортных систем, формирование международного рынка транспортных услуг, что в свою очередь диктует необходимость решения различного рода логистических задач. Большинство задач логистики приходится решать на огромных массивах данных, при этом используемые методы имеют экспоненциальную зависимость сложности алгоритма от размерности и количества данных. Перед аналитиком встает вопрос о соотношении между производительностью (временем работы) алгоритма и его эффективностью (точностью или достоверностью). При поиске оптимальных транспортных маршрутов с учетом всевозможных факторов, ограничений на время принятия решения и вычислительных ресурсов повышение эффективности принимаемых решений является важным аспектом. Итерационные алгоритмы поиска оптимальных транспортных мар-

шрутов, основывающиеся на глобальном переборе, являются весьма медленными. Одним из путей решения противоречия между скоростью обработки данных и их объемом является упрощение исходного набора данных — сужение пространства поиска.

Анализ модели данных предметной области "Организация дорожного движения" [1] позволил выявить наличие крупных сегментов данных, отличающихся однородностью распределения признаков анализа. Этот факт является основанием для применения методов кластерного анализа над графом сети дорог. Применение методов кластерного анализа для сжатия линейных многомерных пространств, по существу, является классическим. Специфика же рассматриваемой задачи состоит в географической распределенности числовых признаков, вследствие чего накладываются определенные качественные ограничения на функцию принадлежности элемента анализируемого множества к соответствующему кластеру.

Постановка задачи

Одним из объектов исследования в предметной области "Организация дорожного движения" является улично-дорожная сеть (УДС) [2].

Для описания движения транспортных средств по УДС резонно использовать математический аппарат теории графов. Геометрическим оргграфом G в пространстве \mathcal{R}^n будем считать совокупность непустого множества $N = \{n_i\}$ объектов некоторой природы в \mathcal{R}^n , называемых узлами графа, множества $E = \{e_i\}$ непрерывных самонепересекающихся кривых в \mathcal{R}^n , называемых дугами графа, инцидентора \tilde{P}_1 , сопоставляющего каждой дуге $e_i \in E$ узел $n_i \in N$, называемый началом дуги, инцидентора \tilde{P}_2 , ставящего в соответствие дуге $e_i \in E$ узел $m_i \in N$, называемый концом дуги.

Традиционно в графовых моделях, описывающих УДС, узлы $n_i \in N$ соответствуют перекресткам, а дуги $e_i \in E$ — перегонам УДС [3]. УДС любой сложности можно представить в виде графа методом редукции. В таком случае имеем дело с геометрическим, взвешенным оргграфом, при этом весовые функции дуг орграфа определяются параметрами реальной УДС и связанными с ней объектами транспортной инфраструктуры. Множество характеристик A УДС и транспортного потока биективно связано с дугами орграфа $\mu: A \rightarrow E$. Граф улично-дорожной сети мегаполиса имеет большую размерность, например, детальный граф г. Самары содержит более пятисот тысяч узлов и дуг. В целях уменьшения времени работы алгоритмов поиска оптимальных транспортных маршрутов требуется построить оргграф замещения $G' \langle N', E' \rangle$ с минимальной потерей информации

об исходном орграфе G . Таким образом, требуется найти такое минимальное ($n \rightarrow \min$) разбиение $G_1 \oplus G_2 \oplus \dots \oplus G_n = G$, чтобы для каждого G_k :

$$\sum_{i=1}^{|E^k|} \sum_{j=|E^k|-i}^{|E^k|} d(A_i^k, A_j^k) \rightarrow \min, \text{ где } d \text{ — евклидова мера.}$$

Анализ популярных алгоритмов кластеризации

Существует множество алгоритмов для решения задачи кластеризации: иерархические (дивизимные, агломеративные) и неиерархические (алгоритм *k-Means* и его модификации, алгоритм *Expectation Maximization*, карта Кохонена и др.) [4]. В методах, основанных на этих алгоритмах, имеются строгие ограничения на геометрию получаемых кластеров. Алгоритмы требуют возможность охвата каждого кластера отдельным выпуклым множеством. Такое ограничение налагается предположениями о существовании центров кластеров (алгоритм *k-Means*) или функции плотности вероятности для каждого кластера с соответствующим значением математического ожидания и дисперсией (алгоритм *Expectation Maximization*). В результате эти алгоритмы не в состоянии адекватно разбить на кластеры невыпуклые множества, тем более вложенные структуры.

Необходимость в построении таких кластеров возникает, например, для адекватной и информативной кластеризации данных исследуемой предметной области, описывающих зависимость интенсивности движения транспортных потоков от ширины проезжей части и качества покрытия дороги (рис. 1). Эту проблему решает алгоритм кластеризации на основе нечетких отношений, позволяющий группировать в кластеры элементы, между которыми существует последовательность "близких" друг к другу элементов, что соответствует интуитивному представлению о группировке [5].

В случае если пространство кластеризуемых элементов является векторным пространством

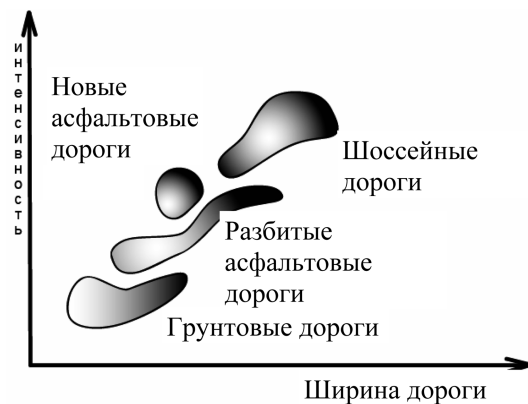


Рис. 1. Зависимость интенсивности движения от ширины проезжей части и качества покрытия дороги

с топологией, индуцированной метрикой, то полученные с помощью описанного алгоритма кластеры могут иметь произвольную геометрическую форму, в том числе состоять из невыпуклых множеств. Это является значительным преимуществом подхода в сравнении с известными алгоритмами кластеризации [7].

Математическое обоснование адаптивной кластеризации графа на базе построения нечеткого отношения эквивалентности

Пусть $G = \langle V, E, W \rangle$ — граф УДС, где V и $E = V^{(2)}$ — множество узлов и дуг улично-дорожной (транспортной) сети соответственно. С каждой дугой графа биективно связан вектор атрибутов $h: E \rightarrow W$, где W — метрическое пространство, и $d: W \rightarrow R$ — определенная на нем метрика. Положим $|E| = n$ и $|V| = m$. Топология графа описывается матрицей смежности $M^{m \times m}(G)$.

Предположим далее, что

$$\forall i \in \{1, \dots, n\} \exists j \in \{1, \dots, n\}: W_i \neq W_j. \quad (1)$$

Из условия (1) следует, что $\forall i \in \{1, \dots, n\}$ справедливо

$$\max\{d(W_i, W_k) | k \in \{1, \dots, n\}\} > 0. \quad (2)$$

Таким образом, для каждого индекса i можно определить функцию, описывающую меру сходства j -го элемента последовательности с i -м элементом:

$$\xi_i: \{1, \dots, n\} \rightarrow [0, 1];$$

$$\xi_i(j) := 1 - \frac{d(W_i, W_j)}{\max\{d(W_i, W_k) | k \in \{1, \dots, n\}\}}. \quad (3)$$

Для каждого индекса i определим функцию, описывающую меру сходства k -го и l -го элементов относительно i -го элемента:

$$\zeta_i: \{1, \dots, n\}^2 \rightarrow [0, 1];$$

$$\zeta_i(k, l) := 1 - |\xi_i(W_k) - \xi_i(W_l)|. \quad (4)$$

Определим функцию, описывающую меру сходства любых двух элементов последовательности относительно всех элементов последовательности:

$$\mu: \{1, \dots, n\}^2 \rightarrow [0, 1];$$

$$\mu(i, j) := \min\{\zeta_k(i, j) | k \in \{1, \dots, n\}\}. \quad (5)$$

Замечание 1. Вместо min-нормы можно взять любое отображение $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$, обладающее свойствами:

- симметричности $T(x, y) = T(y, x)$;
- ассоциативности $T(T(x, y), z) = T(x, T(y, z))$;
- монотонности $T(x, y) \leq T(x', y')$, при $x \leq x'$ и $y \leq y'$;
- $T(x, 1) = x, \forall x \in [0, 1]$.

Очевидно отношение, порождаемое мерой сходства μ , является α -толерантным, т. е. облада-

ет свойством четкой рефлексивности и нормальной α -симметричности [6].

С помощью полученного отношения α -толерантности можно построить классы нечеткой эквивалентности на множестве образцов данных и покрытие этого множества. Для этого необходимо найти транзитивно зависимые образцы данных на отношении α -толерантности.

Определим *транзитивное замыкание* нечеткого отношения R , определенного на множестве W :

$$R = \bigcup_{i=1}^{|W|} R^i. \quad (6)$$

Для указания правил вычисления транзитивного замыкания необходимо ввести ряд определений.

Замечание 2. В последующих определениях вместо max-нормы можно взять любое отображение $S: [0, 1] \times [0, 1] \rightarrow [0, 1]$, обладающее свойствами:

- симметричности $S(x, y) = S(y, x)$;
- ассоциативности $S(S(x, y), z) = S(x, S(y, z))$;
- монотонности $S(x, y) \leq S(x', y')$, при $x \leq x'$ и $y \leq y'$;
- $S(x, 0) = x, \forall x \in [0, 1]$.

Относительно min-нормы справедливо *замечание 1*.

Пусть R_1 — нечеткое отношение $R_1: (x \times y) \rightarrow [0, 1]$ между x и y , и R_2 — нечеткое отношение $R_2: (y \times z) \rightarrow [0, 1]$ между y и z . Нечеткое отношение между x и z , обозначаемое $R_1 \circ R_2$, определенное через R_1 и R_2 выражением

$$\mu_{R_1 \circ R_2}(x, z) = \max_y(\min(\mu_{R_1}(x, y), \mu_{R_2}(y, z))), \quad (7)$$

назовем *композицией* двух нечетких отношений. Композицию больше двух отношений определим по индукции. В частности,

$$R^n = \underbrace{R \circ \dots \circ R}_n = R^{n-1} \circ R, \quad (8)$$

где $R^1 = R$ — некоторое нечеткое отношение.

Объединением отношений R_1 и R_2 назовем отношение, определяемое следующим выражением:

$$\mu_{R_1 \cup R_2}(x, y) = \min(\mu_{R_1}(x, y), \mu_{R_2}(x, y)). \quad (9)$$

В теории нечетких множеств доказываются следующие утверждения относительно отношения α -толерантности.

Пусть R — отношение α -толерантности, тогда:

- $R \subseteq R^2 \subseteq \dots \subseteq R^n \subseteq \dots$;
- объединение отношений α -толерантности также является отношением α -толерантности;
- $R = \bigcup_{i=1}^{|W|} R^i = R^{|W|}$;
- транзитивное замыкание отношения α -толерантности порождает отношение α -квазиэквивалентности.

Таким образом, элементы матрицы степеней сходства, т. е. матрицы α -квазиэквивалентного отношения, определяются по следующему алгоритму:

1. Начальные условия: $\mu_{R^1}(i, j) = \mu_R(i, j)$, определяемые из (5).

2. Цикл по $q = \overline{2, n} : \lambda(W_i, W_j) = \mu_{R^q}(W_i, W_j) = \max_{k = \overline{1, n}} (\min(\mu_{R^{q-1}}(W_i, W_k), \mu_R(W_k, W_j)))$.

Итак, когда получена матрица нечеткого отношения эквивалентности, необходимо построить четкое отношение эквивалентности, которое в свою очередь определит классы эквивалентности для анализируемого множества. Введем необходимые определения.

Порогом α -квазиэквивалентности на множестве W назовем число $\alpha_{\text{порог}} \in [0, 1]$, при котором все элементы множества W являются квазиэквивалентными, а при $\alpha > \alpha_{\text{порог}}$ данное утверждение не соблюдается.

Уровнем α -квазиэквивалентности назовем число $\alpha_{\text{ур}} \geq \alpha_{\text{порог}}$.

Задав уровень α -квазиэквивалентности, можно перейти к четкому отношению эквивалентности. В классической теории множеств [6] доказывается, что отношение эквивалентности разбивает множество образцов данных W на попарно непересекающиеся классы эквивалентных элементов таким образом, что каждый элемент W принадлежит точно одному классу эквивалентности K_j . Таким образом, задавая различные значения параметру $\alpha_{\text{ур}}$, будем получать различные разбиения, составляющие градацию вложенных кластеров. Принципиальная разница между различными уровнями состоит только в изменении дисперсии исследуемых данных в пределах кластера.

Поскольку исследуемые данные составляют множество атрибутов дуг орграфа, от конечного разбиения следует требовать, чтобы элементы, входящие в один кластер, соответствовали *связному* орграфу. В таком случае, применив алгоритм поиска в глубину для каждого кластера атрибутов K_j в целях разделения компонент связности, получим новое разбиение K' , которое соответствует представлению о кластеризации графа как о разделении последнего на непересекающиеся связанные подграфы. Ясно, что $|K'| \geq |K|$.

Использование формального критерия качества адаптивной кластеризации

Очевидным является тот факт, что для больших значений параметра $\alpha_{\text{ур}}$, определяющего совокупность классов эквивалентности, достигается

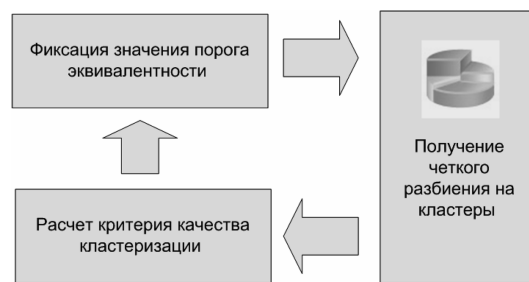


Рис. 2. Схема процесса адаптивной кластеризации

большая степень сохранности информации, но в то же время получаем меньшее значение степени сжатия, которое определяется значением мощности кластеров. Компромиссное решение можно получить, используя специальные критерии построения оптимального разбиения при нечеткой кластеризации. В [8, 9] рассматриваются формальные критерии качества, которые сочетаются с такими неиерархическими алгоритмами, как, например, *Fuzzy C-Means*. Адаптивная составляющая в данном случае зависит не только от матрицы принадлежности, но и от вектора центров кластеров. Обобщенная схема использования адаптивной кластеризации показана на рис. 2.

В нашем же случае имеем только совокупности транзитивно зависимых элементов. Определить для каждого нечеткого класса эквивалентности усредненный образец данных и принять его за центр можно только в том случае, если кластер имеет сферическую форму или хотя бы близкую к таковой. В рассматриваемой задаче допускаются кластеры произвольной формы, что в свою очередь усложняет получение критерия четкости разбиения.

Определим меру эквивалентности отдельного элемента набора данных четкому кластеру разбиения

$$\theta(W_i, K_j) = \min\{\lambda(W_i, W_j) | W_j \in K_j\},$$

где W_i — элемент данных; K_j — кластер четкого разбиения.

Наилучшим разбиением следует считать то, которое обладает наименьшим значением энтропии или, иными словами, имеет большую степень упорядоченности. То есть чем больше степень принадлежности элемента одному кластеру (и меньше степень принадлежности всем остальным кластерам), тем меньше значение энтропии и тем качественнее выполнена кластеризация [10].

Процедура поиска наиболее четкого разбиения является типичной задачей линейного программирования. Критерием оптимальности выступает нормализованная энтропия разбиения

$$PE = -\frac{\sum_{i=1}^{|W|} \sum_{j=1}^{|K|} \theta(W_i, K_j) \ln(\theta(W_i, K_j))}{|W| - |K|};$$

$$PE \in \left(0, \frac{|W| \ln |K|}{|W| - |K|}\right),$$

где W — множество образцов данных; K — множество кластеров.

Если учитывать биективное отображение атрибутивных данных на множество дуг графа, то энтропию разбиения следует вычислять по формуле

$$PE' = -\frac{\sum_{i=1}^{|W|} \sum_{j=1}^{|K'|} \theta(W_i, K'_j) \ln(\theta(W_i, K'_j))}{|W| - |K'|};$$

$$PE \in \left(0, \frac{|W| \ln |K'|}{|W| - |K'|}\right),$$

где K' — множество кластеров, полученных с учетом топологии орграфа.

Сжатие графа

В соответствии с полученной моделью данных в виде множества кластеров разложим геометрическую модель транспортной сети в конечную сумму непересекающихся подграфов [11]:

$$\begin{aligned} G_1 \oplus G_2 \oplus \dots \oplus G_m &= G; \\ G_1 \otimes G_2 \otimes \dots \otimes G_m &= \emptyset, \end{aligned} \quad (10)$$

где m определяется величиной, характеризующей меру сходства элементов множества G_i .

Компоненты разбиения (10) в большинстве случаев являются несвязными графами, что в свою очередь препятствует созданию дискретной структуры замещения. Таким образом, следует увеличить степень дискретизации исходного разбиения посредством дробления несвязных элементов разбиения по компонентам их связности. Получим новое разбиение:

$$\begin{aligned} H_1 \oplus H_2 \oplus \dots \oplus H_s &= G; \\ H_1 \otimes H_2 \otimes \dots \otimes H_s &= \emptyset, \end{aligned} \quad (11)$$

где $s - m \geq 0$ и определяется степенью связности компонент разбиения (10).

Теперь каждый подграф $H_i = \langle N_i, E_i \rangle$ становится связным и имеет близкие значения весовых функций входящих в него дуг (рис. 3, 4, см. третью сторону обложки). Этот факт позволяет представить каждый подграф H_i в виде узла с определенной пропускной способностью, которую можно вычислять как среднее значение атрибутивно-го признака на всем множестве дуг подграфа H_i . Итак, проведя операцию сжатия по всем компонентам разбиения (11), получим множество узлов

N_c сжатого графа. Топологические связи между элементами множества N_c устанавливаются в соответствии с множеством узлов N_{bound} , граничных для $H_i, \{n \in N_{bound} | n \in N_i \wedge n \in N_j \wedge i \neq j\}$, а именно: множество дуг E_c сжатого графа определяется так:

$$\begin{aligned} \{(n_i, n_j) \in E_c | n_i, n_j \in N_c \wedge \\ \wedge (\exists n \in N_{bound}: n \in N_i, n \in N_j)\}. \end{aligned}$$

Таким образом, получили сжатый граф $G_c = \langle N_c, E_c \rangle$. Вследствие того, что направления или маршруты прохождения через узел принимаются изоморфными между собой, возникает ряд существенных ограничений на область применимости построенной модели [12].

Заключение

Достоинства метода:

- отсутствие необходимости в априорных предположениях относительно структуры данных и числа кластеров;
- отсутствие ограничений на геометрию кластеров;
- время выполнения алгоритма мало зависит от числа компонент входных векторов;
- матрица нечеткого отношения эквивалентности, сохраненная в постоянной памяти, позволяет оперативно строить разбиение данных и сжимать граф.

Ограничения метода:

- большое время выполнения, характеризующееся порядком $O(n^4)$ от числа дуг;
- метод не учитывает пространственной распределенности весовых характеристик в пределах кластера.

Вариантом использования модели является поиск оптимальных транспортных маршрутов. Если атрибуты дуг графа не являются аддитивными величинами при поиске маршрута, т. е. на общий вес маршрута оказывает влияние не сумма весов отдельных его участков, а только наибольший или наименьший из них, модель может считаться адекватной.

Примером однокритериальной задачи является поиск наиболее безопасного транспортного маршрута. Метод адаптивного сжатия графа улично-дорожной сети реализован при решении задач управления транспортными потоками в интеллектуальной транспортной системе г. Самара.

Список литературы

1. Михеева Т. И. Системный анализ при проектировании интеллектуальной транспортной системы региона // Математика. Компьютер. Образование: Сб. научных трудов. Т. 1 / Под ред. Г. Ю. Ризниченко. М.—Ижевск: Московский гос. ун-т, НИЦ "Регулярная и хаотическая динамика". 2006. С. 235—255.
2. Михеева Т. И. Построение математических моделей объектов улично-дорожной сети города с использованием геоин-

формационных технологий // Информационные технологии. 2006. № 1. С. 69—75.

3. **Автотранспортные** потоки и окружающая среда / Луканин В. Н., Буслаев А. П., Трофименко Ю. В. и др. / Под ред. В. Н. Луканина. М.: ИНФРА-М, 1998. 408 с.

4. **Технологии** анализа данных: Data Mining, Visual Mining, Text Mining, OLAP / А. А. Барсегян и др. СПб.: БХВ—Петербург, 2007. 384 с.

5. **Новак В., Перфильева И.** Математические принципы нечеткой логики. М.: Физматлит, 2006. 352 с.

6. **Лекции** по дискретной математике / Ю. В. Капитонова, С. Л. Кривой, А. А. Летичевский, Г. М. Луцкий. СПб.: БХВ—Петербург, 2004. 624 с.

7. **Jain A. K., Murty M. N., Flynn P. J.** Data clustering: a review. ACM Computing Surveys 31. 1999. P. 264—323.

8. **Jain A. K., Dubes R. C.** Algorithms for Clustering Data. Prentice Hall, 1988. 129 p.

9. **Hartuv E., Shamir R.** A clustering algorithm based on graph connectivity // Information Processing Letters 76. 2000. P. 175—181.

10. **Wagner D., Wagner F.** Between Min Cut and Graph Bisection. / Borzyszkowski A. M., Sokolowski S., eds. Lecture Notes in Computer Science. Springer-Verlag, 1993. P. 744—750.

11. **Михеева Т. И., Золотовицкий А. В.** Эвристические методы построения транспортных путей в задачах логистики // Тр. 6-й Междунар. научно-практ. конф. "Организация и безопасность дорожного движения в крупных городах". СПб.: С-ПБАДИ, 2004. С. 89—93.

12. **Кликовштейн Г. И., Афанасьев М. Б.** Организация дорожного движения. М.: Транспорт, 1992. 240 с.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ

УДК 519.6

Ю. А. Куперин, д-р физ.-мат. наук, проф.,

e-mail: yuri.kuperin@gmail.com,

Р. Р. Счастливцев, аспирант,

e-mail: srr2001@mail.ru,

Санкт-Петербургский государственный университет

Прогнозирование финансовых временных рядов с помощью комитетов искусственных нейронных сетей

Приведены разработанные методы получения высококачественных прогнозов временных рядов с помощью комитетов искусственных нейронных сетей (ИНС). Проанализированы стандартные методы объединения отдельных ИНС в комитеты и предложен новый метод формирования комитетов ИНС. Предложенные в работе методы протестированы на примере решения задачи высококачественного прогнозирования реальных данных фондового рынка США.

Ключевые слова: прогнозы, комитеты нейронных сетей, слабый прогноз, сильный прогноз, финансовые временные ряды.

Введение

Целью настоящей работы является разработка методов получения высококачественных нейропрогнозов финансовых временных рядов, в частности, прогнозирования цен акций с помощью специально разработанных комитетов искусственных нейронных сетей (ИНС). Нейронные сети в своей основе нелинейны, не требуют глубокого понимания связей между исходными данными и

получаемыми результатами и демонстрируют серьезные преимущества перед традиционными методами статистической обработки данных (см., например [1]). Многочисленные эксперименты показывают, что адаптивные сети на коротких промежутках времени прогнозируют всегда лучше, чем стандартные статистические модели [2, 3]. Однако в области прогнозирования финансовых данных отдельные ИНС практически любой архитектуры и при любом способе обучения зачастую не позволяют получить прогноз такого статистического качества, который был бы приемлем для принятия инвестиционных решений. Поэтому по-прежнему актуальна задача повышения статистического качества нейропрогнозов сложных временных рядов, в частности финансовых. В настоящей работе для решения этой проблемы применяется технология прогнозирования с помощью специально разработанных комитетов ИНС.

Эмпирически феномен повышения точности и надежности результата при объединении нескольких мнений в одно был замечен давно [4]. Первые математические основания для этого факта связываются с центральной предельной теоремой (ЦПТ) теории вероятностей. Хотя на практике применение ЦПТ может быть затруднено ввиду невыполнимости ее условий или проблем со сходимостью, эта теорема может рассматриваться как "не запрещающая" понижать ошибку путем комбинирования прогнозов.

В настоящей работе исследованы известные методы комбинирования прогнозов отдельных ИНС, такие как простое среднее или взвешенное среднее, когда веса определяются из качества прогноза на тестовом множестве [1, 4]. Предложен новый метод комбинирования прогнозов ИНС, который вместе с результатом прогнозирования позволяет оценить и статистическое качество прогноза.

1. Комитеты экспертов

В работе, как составная часть, рассматривается статистическая задача классификации в многомерных пространствах. Задача формулируется следующим образом. Имеется множество векторов из многомерного числового пространства. Для части векторов известно значение класса из оговоренного конечного набора классов. Требуется установить, к каким классам относятся оставшиеся векторы. Более строгая постановка задачи приведена в работах [5—7].

Существуют несколько методов комбинирования экспертов в комитет [7—12]. Под комитетом понимается модель, выход которой является комбинацией выходов отдельных экспертов. Строгое определение объединения алгоритмов можно найти, например, в работе [13]. Для демонстрации повышения качества работы комитета по сравнению с отдельным экспертом рассмотрим простой пример [4]. Пусть в распоряжении имеется три эксперта, каждый из которых решает задачу бинарной классификации с вероятностью успеха p . Если из данных экспертов образовать комитет большинства, принимающий решение простым голосованием, то вероятность успеха такого комитета составит $q = p^3 + 3p^2(1 - p)$. График зависимости $q(p)$ показан на рис. 1 штрих-пунктирной линией. Видно, что при $p > 0,5$ график $q(p)$ лежит выше пунктирной линии, которая соответствует вероятности верной классификации отдельным экспертом. Из этой зависимости можно сделать вывод, что при $p > 0,5$ комитет большинства имеет большую вероятность успеха, чем отдельный эксперт.

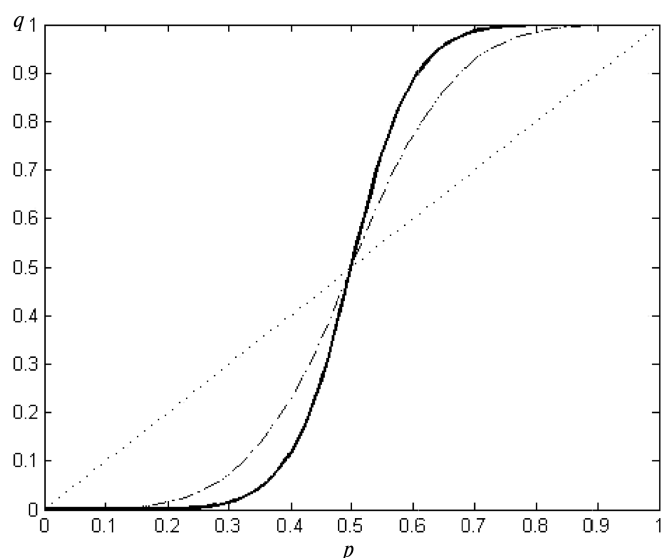


Рис. 1. Вероятность q верной классификации комитетом большинства (штрих-пунктирная кривая) и вероятность верной классификации согласованным комитетом (сплошная кривая) как функция точности отдельного эксперта (ось абсцисс). Прямая пунктирная линия соответствует отдельному эксперту

Рассмотрим случай, когда все три эксперта дают одинаковый выход. Тогда вероятность верной классификации составит величину $q = p^3/(p^3 + (1 - p)^3)$. Комитет, принимающий решение таким образом, назовем согласованным комитетом. Зависимость $q(p)$ для такого комитета изображена на рис. 1 сплошной линией. Из этой зависимости можно сделать вывод, что при $p > 0,5$ согласованный комитет имеет большую вероятность успеха, чем комитет большинства. Такой результат достигается за счет того, что принятие решения согласованным комитетом возможно только в случае, если все эксперты комитета принимают одинаковое решение, в то время как принятие решения комитетом большинства возможно при любой комбинации отдельных экспертов (если экспертов нечетное число).

Рассмотрим комитет из n экспертов, каждый из которых решает задачу бинарной классификации с вероятностью успеха $p > 0,5$. Пусть r — доля комитетов, показавших одинаковый результат, $0,5 \leq r \leq 1$. Тогда вероятность верной классификации составит

$$q^{r,n} = \frac{p^{(2r-1)n}}{p^{(2r-1)n} + (1-p)^{(2r-1)n}}. \quad (1)$$

Из этой зависимости видно, что при $p > 0,5$ комитет имеет тем большую вероятность успеха $q^{r,n}$, чем больше доля экспертов r , показавших одинаковый результат. Так как на практике априори оценить p или $q^{r,n}$ бывает достаточно сложно, а зависимость $q^{r,n}$ от r является взаимнооднозначной, то долю r можно интерпретировать как вероятность успеха комитета.

Из выражения (1) следует, что если даже отдельный эксперт является слабым классификатором, т. е. $p = 0,5 + \varepsilon$, где ε лишь немного больше 0, то для любых $r > 0,5$ при достаточно большом числе сетей, комитет будет являться сильным классификатором, поскольку

$$q^{r,n} \xrightarrow[n \rightarrow \infty]{} 1. \quad (2)$$

Однако последнее возможно, только если отдельные эксперты являются попарно независимыми. На практике реализовать достаточное число независимых экспертов является весьма сложной задачей, поэтому сходимость (2) является медленной или вообще не существует, а для быстрой сходимости требуется, чтобы p сильно превышало 0,5.

2. Численный эксперимент

Постановка задачи. Ниже в качестве численного эксперимента решается задача прогнозирования знака приращения финансового временного ряда. В качестве финансового ряда взяты наивыс-

шие цены акций компании *Alcola Inc* с временной нарезкой в один день (данные взяты с сайта <http://finance.yahoo.com>). Прогнозирование осуществлялось с помощью ИНС, которая выступает в качестве эксперта. Подробные сведения о теории ИНС и теории прогнозирования с помощью ИНС можно найти, например в работах [1, 14]. В настоящей работе архитектура ИНС — многослойный перцептрон. На вход ИНС подаются значения цен открытия, закрытия, наивысшей и наименьшей цен и технические индикаторы от этих цен за предыдущие несколько дней, а также цена открытия следующего дня. На выходе ИНС выдает знак приращения наивысшей цены на следующий день — значение 1 или -1. Таким образом, задача нейропрогнозирования сводится к задаче бинарной классификации. Все множество примеров разбивается на три последовательно упорядоченных множества. Первое множество — обучающее, его размер составляет 500...1500 точек ряда. Второе множество — валидационное, его размер составляет 50 точек ряда. Оно используется для отбора лучших сетей или вычисления коэффициентов для объединения сетей в комитет. Третье множество — тестовое, его размер составляет 100 точек ряда. На нем проверяется качество прогноза. Временной период, соответствующий тестовому множеству, в данном случае — 26 апреля — 15 сентября 2005 гг. Независимость сетей в комитете достигается за счет случайного выбора параметров сетей, таких как начальные веса, число нейронов, размер обучающего множества, число эпох обучения. Размер сети не превосходил трех скрытых слоев из 10 нейронов каждый.

Отбор лучших сетей. В целях тестирования работы комитета было обучено 10 000 сетей. Для оценки качества прогноза основным показателем являлась доля верно прогнозируемого знака приращения цены на тестовом множестве, дополнительными показателями — коэффициент корреляции и множественной детерминации между прогнозом сети и реальным значением цены. В результате эксперимента было получено в среднем 64 % верно спрогнозированного знака на тестовом множестве. Распределение доли верно спрогнозированного знака приращения цены на тестовом множестве изображено на рис. 2 кривой 1. Далее было отобрано из всех обученных ИНС 50 % лучших на валидационном множестве. Лучшей сетью считается та, для которой сумма всех трех перечисленных выше показателей, по которым оценивается качество прогноза, является максимальной. Для этих ИНС получилось в среднем 67 % верно спрогнозированного знака на тестовом множестве (распределение прогноза знака изображено на рис. 2 кривой 2). При отборе 10 % лучших ИНС было получено в среднем 68 % вер-

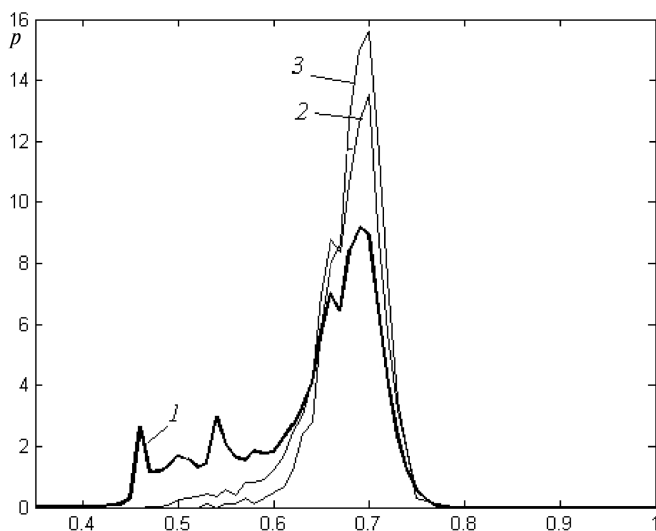


Рис. 2. Распределение доли верно спрогнозированного знака приращения цены на тестовом множестве среди всех сетей (кривая 1), среди 50 % лучших сетей, отобранных на валидационном множестве (кривая 2), среди 10 % лучших сетей на валидационном множестве (кривая 3)

но спрогнозированного знака на тестовом множестве (кривая 3).

Таким образом, показано, что при отборе лучших сетей на валидационном множестве качество прогноза на тестовом множестве улучшается. Однако сходимость такого процесса очень медленная. Так, при доле отобранных сетей в 50 % средняя доля верно спрогнозированного знака составляет 67 %. При уменьшении доли сетей до 10 % доля верно спрогнозированного знака увеличивается всего на 1 %. Из рис. 2 видно, что при любом отборе сетей доля верно спрогнозированного знака одной сетью не может превысить 78 %. При дальнейшем уменьшении доли отобранных сетей прогноз не становится более 72 %.

Простые комитеты. Рассмотрим комитет большинства. Решение комитет принимает по следующему правилу [4]:

$$h = \text{sign}(\sum_k h_k),$$

где h_k — решение отдельной сети. образуем три комитета большинства: из всех сетей, из 50 % лучших сетей и из 10 % лучших сетей от общего числа сетей 10 000. Получим 69; 71 и 72 % верно прогноза знака комитетом соответственно.

Рассмотрим комитет, в котором участие ИНС не является равноправным. Участие отдельной сети тем больше, чем лучше результаты она показала на валидационном множестве. Решение комитет принимает по следующему правилу [4]:

$$h = \text{sign}(\sum_k w_k h_k),$$

$$w_k = p_k / \sum_k p_k,$$

где p_k — доля верно спрогнозированного знака отдельной сети. образуем три комитета: из всех се-

Процент верного прогноза знака для различных комитетов

Доля сетей в комитете, %	Вид комитета		
	Отдельная сеть	Комитет большинства	Комитет с учетом доли угаданного знака
100	64	69	65
50	67	71	71
10	68	72	73

тей, из 50 % лучших сетей и из 10 % лучших сетей. Получим 65, 71 и 72 % верного прогноза знака комитетом соответственно.

Из таблицы видно, что комитеты сетей прогнозируют лучше, чем отдельные сети. Следует отметить, что комитеты с учетом доли верно спрогнозированного знака не показывают значительного улучшения результата по сравнению с комитетом большинства. Это связано с тем, что требования повышения точности отдельной сети и обеспечения независимости различных сетей комитета противоречат друг другу.

Согласованные комитеты. Исследуем, как зависит качество прогноза комитета от доли сетей, показавших одинаковое решение. Для этого все обученные ИНС были объединены в 100 комитетов по 100 сетей в каждом. Зависимость доли верно спрогнозированного знака от доли сетей, показавших одинаковое решение в среднем для всех комитетов, изображена на рис. 3, б, кривая 1. Доля верно спрогнозированного знака возрастает с 56 до 100 % при изменении доли сетей с 51 до 99 %. В момент, когда происходит резкое улучшение прогноза, примерно начиная с доли сетей от 90 %, одновременно резко уменьшается число дней, когда осуществляется прогноз (далее такие дни называются прогнозными), т. е. когда все эти сети показали одинаковый результат (см. рис. 3, б, кривая 1). Так, при доли сетей 90 % прогноз составляет 75 %, а доля прогнозных дней составляет 3,85 %. При доли сетей 99 % прогноз составляет 100 % при доли прогнозных дней 0,13 %. Такое поведение зависимости доли прогнозных дней от доли сетей вызвано тем, что среди ИНС встречаются сети с очень низкой и даже отрицательной корреляцией между собой. Ситуация же, когда все сети показывают одинаковый прогноз, очень редка, и в данном

эксперименте ни разу не встречалась. Для увеличения доли прогнозных дней рассмотрим комитеты, составленные из 50 % лучших ИНС на валидационном множестве. Зависимость верно спрогнозированного знака и распределение прогнозных дней показана на рис. 3 (кривая 2). Прогноз, сделанный такими комитетами, ухудшается, однако качественно меняется форма распределения прогнозных дней. Так, при доле сетей, равной 100 %, имеем прогноз 89 % при доле прогнозных дней 20 %.

Из результатов, полученных выше, следует, что наилучшее качество прогноза достигается, если все сети комитета показывают одинаковый результат. Назовем такой прогноз прогнозом согласованного комитета. Качество прогноза зависит от числа лучших сетей, включаемого в комитеты. Чем больше сетей в комитете, тем меньше между ними корреляция, тем выше результат. Рассмотрим зависимость качества прогноза согласованным комитетом от доли лучших сетей, включаемых в комитет. График зависимости до-

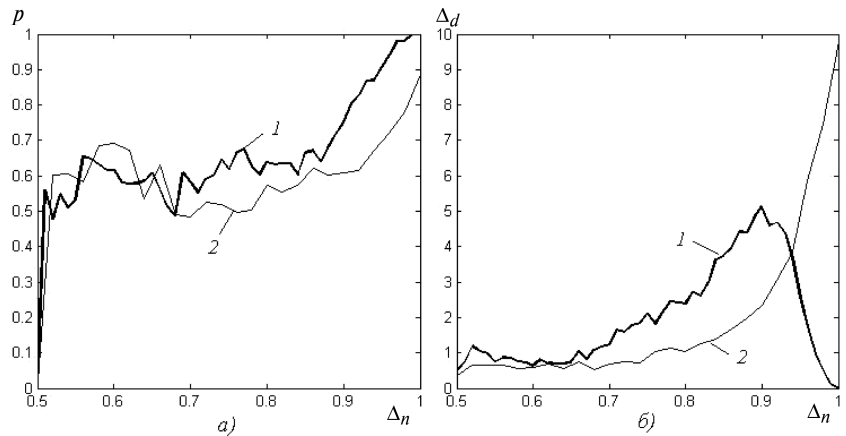


Рис. 3. Зависимость доли p верно спрогнозированного знака (а) и распределение доли дней Δ_d , в которых производился прогноз (б) на тестовом множестве, как функция доли сетей Δ_n , показавших одинаковый результат. Кривая 1 соответствует комитетам из всех сетей, кривая 2 соответствует комитетам из 50 % лучших сетей, отобранных на валидационном множестве

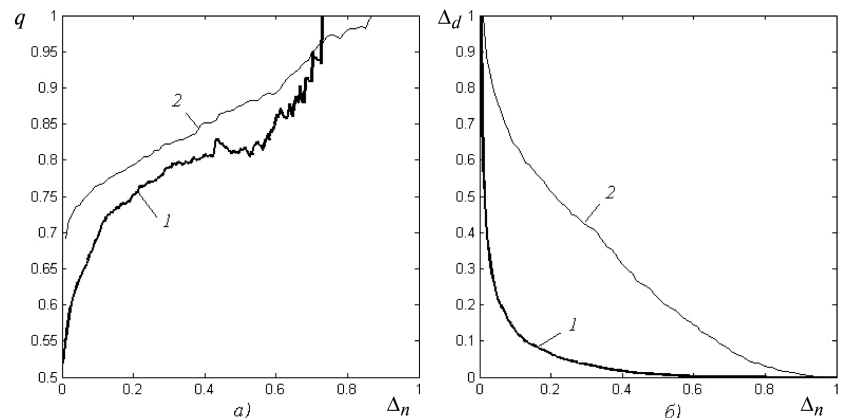


Рис. 4. Зависимость доли q верно спрогнозированного знака согласованными комитетами (а) и доля прогнозных дней (б) на тестовом множестве как функция доли лучших сетей Δ_n , участвующих в комитетах для наивысшей цены (кривая 2) и цены закрытия (кривая 1)

Слабый и сильный прогнозы

ли верно спрогнозированного знака согласованными комитетами как функции доли лучших сетей, участвующих в комитете, построен на рис. 4, а (кривая 2). Из графика видно, что согласованными комитетами можно достичь любого качества прогноза. Так, начиная с доли сетей, равной 81 %, прогноз абсолютно точный, т. е. доля верно спрогнозированного знака составляет 100 %. График зависимости доли прогнозных дней как функции доли лучших сетей построен на рис. 4, б (кривая 2). Это невозрастающая функция. При доле сетей, равной 87 %, когда достигается абсолютно точный прогноз, доля прогнозных дней составляет 1,46 %. Например, при доле сетей, равной 61 %, когда достигается качество прогноза в 90 %, доля прогнозных дней составляет 14 %.

Комитеты комитетов. Как было показано в предыдущем разделе, с помощью согласованных комитетов можно добиться любого качества прогноза. Однако с увеличением качества прогноза сильно падает число прогнозных дней. Для решения задачи увеличения доли прогнозных дней без ухудшения качества прогноза рассмотрим не один комитет, а несколько комитетов. Решение такой комитет комитетов будет принимать следующим образом. Если один из комитетов принимает решение, а остальные ему не противоречат, т. е. принимают такое же решение, или никакое решение (т. е. для них текущий день — непрогнозный день), то решение комитета комитетов и есть решение этого комитета. Если какой-то день для всех комитетов является непрогнозным, то и для комитета комитетов он является непрогнозным. Если в какой-то день различные комитеты принимают неодинаковые решения, то для комитета комитетов данный день является непрогнозным. Увеличение числа прогнозных дней происходит за счет того, что для разных комитетов прогнозные дни являются различными. Вместе с тем, за счет высокого качества прогноза отдельным комитетом увеличение непрогнозных дней для комитета комитетов за счет несогласованного прогноза отдельными комитетами является незначительным.

Применим предложенную здесь технологию для прогнозирования знака наивысших дневных цен акций компании *Alcola Inc.* Рассмотрим комитет комитетов, состоящий из 100 комитетов по 100 сетей каждый. В результате увеличивается доля прогнозных дней. Для сравнения, качество прогноза в 90 % верно спрогнозированного знака достигается при доле прогнозных дней 20 %, а абсолютно точный прогноз — при доле прогнозных дней 18 %. Таким образом, увеличение числа ИНС и объединение их в комитеты комитетов значительно увеличивает долю прогнозных дней.

Как было показано в разд. 2, теоретически с помощью согласованного комитета можно добиться сильного прогноза (при условии независимости сетей в комитете). Выше данное утверждение было проверено экспериментально. Прогноз с качеством в среднем 64 % был улучшен до 100 %. Рассмотрим теперь прогноз худшего качества и покажем, что с помощью согласованных комитетов его можно улучшить до абсолютно точного. Для того чтобы получить прогноз худшего качества, будем вместо наивысшей цены прогнозировать цену закрытия. Практика показывает, что наивысшие и наименьшие цены прогнозируются лучше, чем цены открытия и закрытия. Это интуитивно понятно, так как наивысшие и наименьшие цены — некие усредненные характеристики рынка за определенный период, тогда как цены открытия и закрытия — сугубо локальные. В проведенном эксперименте было обучено 90 000 сетей и в среднем было получено 52 % правильного спрогнозированного знака приращенния цены закрытия на тестовом множестве. Теперь объединим сети в 300 комитетов по 300 сетей. Увеличение числа сетей в отдельном комитете до 300 по сравнению со 100 для наивысшей цены вызвано тем, что прогноз цены закрытия отдельной сетью значительно слабее и сходимость ожидается более медленная.

Зависимость доли q верно спрогнозированного знака согласованными комитетами и доля прогнозных дней Δ_d как функция доли лучших сетей Δ_n , участвующих в комитетах для наивысшей цены, изображены на рис. 4 (кривая 1). Из рис. 4 заключаем, что вид обеих зависимостей не изменился для цены закрытия по сравнению с наивысшей ценой. Принципиальное отличие — это значительное уменьшение доли прогнозных дней. Для сравнения, прогноз с качеством 90 % верно спрогнозированного знака достигается при доле прогнозных дней в 0,1 %, а абсолютно верный прогноз достигается при доле прогнозных дней в 0,03 %. Применение комитета комитетов (300 комитетов по 300 сетей) увеличивает эти значения до 3 % в обоих случаях.

Заключение

В работе теоретически показано, что комитеты экспертов имеют превосходство в качестве классификации по сравнению с отдельными экспертами. Разработана модель объединения экспертов в комитет, который является сильным классификатором, тогда как отдельные эксперты могут быть лишь слабыми классификаторами (при условии независимости отдельных экспертов).

Теоретические выводы подтверждены экспериментально. В качестве экспертов были взяты ИНС. Решалась задача прогнозирования знака приращения временного ряда на один отсчет вперед на примере наивысших цен акций компании *Alcola Inc.* Было обучено 10 000 сетей и показано, что простые комитеты улучшают качество прогноза с 64—68 % до 65—73 % верно спрогнозированного знака приращения ряда. Согласованный комитет показал увеличение прогноза с 64 до 100 %. Получение 100 % точного прогноза связано с тем, что принятие решения согласованным комитетом возможно только в случае, если все эксперты комитета принимают одинаковое решение, в то время как принятие решения простыми комитетами возможно при любой комбинации отдельных экспертов. Был проведен второй численный эксперимент и обучено 90 000 ИНС со средним качеством прогноза 52 %, где в качестве временного ряда были взяты цены закрытия акций компании *Alcola Inc.* Как и в первом случае, согласованный комитет показал 100 % верно спрогнозированных знаков приращения ряда. Отличие первого эксперимента от второго состоит в том, что при качестве прогноза отдельного эксперта, равного 64 %, наблюдалось 18 % прогнозных дней, при качестве же отдельного эксперта, равного 52 %, наблюдалось всего 3 % прогнозных дней. Полученные результаты позволяют сделать вывод о применимости прогнозов согласованных комитетов для построения надежных инвестиционных стратегий.

1. **Ежов А. А., Шумский С. А.** Нейрокомпьютинг и его применение в экономике и бизнесе. — М.: Изд-во МИФИ, 1998. 222 с.
2. **Couper C., Couper P.** Neural network and statistics: a naïve comparison // *Belgian journal of operations research and computer sciences*. 1997. N 4. 36 p.
3. **Sarle W. S.** Neural networks and statistical models // Proc. of the nineteenth Annual SAS users Group international conference, Cary, NC, SAS Institute. 1994. April 3—4. P. 1538—1550.
4. **Терехов С. А.** Гениальные комитеты умных машин // Лекции по нейроинформатике. Ч. 2. IX Всероссийская научная конференция "Нейроинформатика — 2007". Москва, МИФИ, 2007. С. 11—42.
5. **Bishop C. M.** *Neural Networks for Pattern Recognition*. — Oxford University Press, 1995. 504 p.
6. **Vapnik V. N.** *The Nature of Statistical Learning Theory*. — Springer, 1995. 314 p.
7. **Freund Y., Schapire R. E.** A short introduction to boosting // *Journal of Japanese Society for Artificial Intelligence*. 1999. N 14(5). P. 771—780.
8. **Ratsch G., Warmuth M. K.** Efficient margin maximizing with boosting // *March. Learn. Res.* 2005. 6(Dec.). P. 2131—2152.
9. **Hastie T., Tibshirani R., Friedman J.** *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. — Springer, 2001. 520 p.
10. **Tsybal A., Puuronen S.** Bagging and boosting with dynamic integration of classifiers // In Proc. of PKDD 2000. Lyon, France. *Lecture Notes in Artificial Intelligence*. 2000. Vol. 1910. Springer Verlag. P. 116—125.
11. **Jordan M. I., Jacobs R. A.** Hierarchical mixtures of experts and the EM algorithm // *Neural Computation*. 1994. N 6. P. 181—214.
12. **Breiman L.** Pasting bites together for prediction in large data sets // *Machine Learning*. 1999. N 36(2). P. 85—103.
13. **Журавлев Ю. И.** Об алгебраическом подходе к решению задач распознавания или классификации // *Проблемы кибернетики*. 1978. Т. 33. С. 5.
14. **Haykin S.** *Neural Networks: A Comprehensive Foundation*. 2 ed. IEEE. 1999. 700 p.

УДК 621.385.63

Л. А. Серков, канд. физ.-мат. наук, доц., зав. отд.,
Европейско-Азиатский институт управления и предпринимательства,
e-mail: kpkg94@mail.ru

Синергетические аспекты моделирования инновационных процессов

В рамках синергетического подхода изучалась стохастичность инновационного процесса и влияние на этот процесс инновационной глобализации, являющейся системным фактором. Авторами показано, что роль случайностей (шума) заключается в индуцировании ими явления, называемого самоорганизованной критичностью. Вовлечение в инновационный процесс все большего числа участников (инновационная глобализация) приводит к некоторому подавлению флуктуаций, к уменьшению роли случайности в появлении и развитии инноваций.

Ключевые слова: синергетика, инновации, диффузия инноваций, технологический трансфер, детерминированность, стохастичность, бифуркации, самоорганизованная критичность, инновационная активность, логистическая кривая.

Введение

Синтез и анализ синергетических моделей управления социально-экономическими системами становится в последнее время особенно актуальным. Это происходит вследствие того, что все

социально-экономические объекты являются открытыми системами и обладают способностями к самоорганизации и адаптации, поэтому представляется перспективным подход к управлению этими объектами через распознавание, анализ, про-

гнозирование и управление процессами самоорганизации. Этот подход является главной составной частью системного управления и предполагает рассмотрение системы управления организацией как целостной совокупности элементов. В этой сфере синергетические модели нужны для выявления «узких мест», возникающих в ходе развития системы. В случае построения адекватных моделей могут быть проанализированы альтернативные экономические стратегии и дан прогноз развития системы при различных управляющих воздействиях.

Теоретические исследования в области синергетического моделирования в экономике проводятся в разных направлениях. При этом все динамические модели, как аналитические, так и допускающие лишь вычислительный эксперимент, можно условно разделить на две группы. К первой группе относятся модели экономического роста, ко второй группе — модели экономического цикла или, в более широком смысле, экономических колебаний. Хотя колебательная динамика в большей мере соответствует реальным процессам по сравнению с поведением моделей роста [1], тем не менее выявление "узких мест" последних, в том числе влияние управляющих воздействий на поведение траекторий моделей роста, представляет также научный и практический интерес.

Одной из распространенных нелинейных моделей экономического роста является модель распространения (диффузии) инноваций. Заметим, что моделирование инновационных процессов является необходимым для понимания природы и организации системного управления нововведениями. В работе А. И. Яблонского [2] высказано предположение о возможности использования *s*-образных кривых (логистическая, Гомпертца, модифицированная экспоненциальная и др.) и уравнений типа Лотки—Вольтера для моделирования процессов технологического развития. Экспериментальные исследования [3, 4] показали, что процесс диффузии, выраженный в виде доли выпуска продукции определенного технологического уровня или доли фирм, освоивших рынок новой продукции, также описывается логистической кривой или ее модификациями (рис. 1).

Логистическая кривая является решением уравнения

$$dx/dt = \lambda x(Y - x), \quad (1)$$

где x — зависимый от времени показатель эволюции объекта; λ — параметр, определяющий скорость изменений (тренд эволюции); Y — параметр, определяющий предел изменений в рамках текущего состояния рынка. Фактически уравнение (1) описывает эволюцию объекта в условиях ограниченных ресурсов и конкуренции, ограни-



Рис. 1. Динамика развития сетей автомобильных дорог в США, подчиняющаяся логистическим закономерностям [3] (в относительных единицах в зависимости от времени)

чивающей рост. При этом начало и конец инновационного производства, смена технологических укладов приводят к появлению скачков (рис. 1) на логистических кривых. Использование положительных свойств логистической кривой для описания жизненного цикла макроструктур (продуктовых инноваций в экономике США, определяющих динамику ВВП) предложено в работе В. И. Маевского [5].

Недостатком практически всех моделей является отсутствие учета влияния системообразующих и субъективных причин (в том числе влияние состояния окружающей социально-экономической среды) на эволюцию инноваций. Между тем, такие характерные особенности инновационных процессов, как лавинообразный характер начала процесса развития, природа скачков на логистических кривых, субъективность (стохастичность процесса), невозможно объяснить в рамках эконометрических и имитационных моделей. Эти особенности связаны с нелинейностью инновационных процессов и для их объяснения можно использовать синергетический подход. Заметим, что хотя синергетические модели позволяют лишь качественно (на концептуальном уровне) описать исследуемые явления, тем не менее с их помощью можно выявлять различные нелинейные особенности изучаемых процессов, что невозможно сделать в рамках имитационных моделей. Кроме того, ценность этих моделей состоит в том, что они позволяют исследовать влияние различных эндогенных и экзогенных факторов на поведение нелинейных траекторий изучаемой системы.

В предлагаемой работе в рамках синергетического подхода исследуются стохастичность инновационного процесса и влияние на этот процесс

такого системообразующего фактора, как взаимосвязь исследуемой системы с другими подсистемами. Исходное уравнение для исследуемой модели инновационного роста в детерминированном случае запишем в виде

$$dx/dt = A + p/Nx(N - x) - \beta x, \quad (2)$$

где $x = x(t)$ — число участников (компаний) региона, участвующих в инновационном процессе в момент времени t ; N — общее число участников; p ($p > 9$) — коэффициент роста числа участников. Второй член уравнения учитывает конкуренцию между участниками инновационного процесса. Последний член уравнения учитывает отторжение инноваций некоторыми компаниями-участниками; β — коэффициент отторжения ($\beta > 0$); A — постоянная скорость появления новых участников инновационного процесса, в том числе и из других регионов в силу открытости системы. В безразмерных переменных $x' = x/N$, $t' = t\beta$ уравнение (2) будет иметь вид:

$$dx'/dt' = \alpha + \gamma x'(1 - x') - x', \quad (3)$$

где $\alpha = A/\beta N$, $\gamma = p/\beta$. В дальнейшем будем опускать штрихи в уравнении (3) и называть переменную x инновационной активностью.

Модели, соответствующие уравнению (3), встречаются в популяционной генетике и имеют химическую реализацию [6]. Если для простоты положить $\alpha = 1/2$, то физический смысл имеет стационарное значение

$$\bar{x} = [\gamma - 1 + (\gamma^2 + 1)^{1/2}]/(2\gamma). \quad (4)$$

Это значение задает взаимно-однозначное отображение интервала $(-\infty, +\infty)$ на интервал $(0, 1)$ [6]. Анализ устойчивости по линейному приближению показывает асимптотическую глобальную устойчивость стационарных состояний (4), т. е. в детерминированном случае в модели инновационного процесса, описываемой уравнением (3), не происходит потери устойчивости. Следовательно, любая потеря устойчивости в этой модели, наблюдающаяся при флуктуациях среды, является чисто индуцированным шумом эффектом, соответствующим качественному изменению макроскопических свойств [6].

Анализ стохастической модели инновационных процессов

Возникновение инноваций всегда сопряжено со случайностями, так что параметр тренда и является флуктуирующим. Заметим, что этот параметр характеризует как внутренние свойства инновационной системы, так и внешние свойства окружающей систему среды. В предположении, что флуктуации довольно быстры, заменим параметр γ стационарным случайным процессом

$\gamma_t = \gamma + \sigma\zeta_t$, где гауссов белый шум ζ_t имеет нулевое среднее значение и интенсивность σ^2 , т. е.

$$\langle \zeta_i(t) \rangle = 0, \langle \zeta_i(t) \zeta_i'(t') \rangle = \sigma^2 \delta(t - t'). \quad (5)$$

Заметим также, что инновационный процесс протекает в пространстве множества технологий. Это можно учесть путем введения взаимодействия между участниками технологического трансфера, отражающего инновационную глобализацию. Учитывая сказанное, запишем стохастическое дифференциальное уравнение для участников инновационного процесса (уравнение Ланжевена):

$$dx_i/dt = f(x_i) - D/M \sum_{j=1}^M (x_i - x_j) + g(x_i)\zeta_i(t); \quad (6)$$

$$f(x_i) = \alpha + \gamma x_i(1 - x_i) - x_i, \quad g(x_i) = x_i(1 - x_i), \quad (6a)$$

где x_i — доля участников процесса в пространстве i -й технологии (инновационная активность); D — агрегированная величина взаимодействия (коммуникационного и производственного) между участниками технологического трансфера ($D > 0$); M — размерность пространства технологий (число инновационных технологий). Белый шум $\zeta_i(t)$ определяется как

$$\langle \zeta_i(t) \rangle = 0, \langle \zeta_i(t) \zeta_j(t') \rangle = \sigma^2 \delta_{ij} \delta(t - t'). \quad (7)$$

В дальнейшем будем рассматривать случайный процесс в интерпретации Стратоновича [6] и интересоваться динамикой макроскопической переменной $\langle x \rangle = 1/M \sum_i x_i$ и ее стационарным значением $\langle x \rangle^* = \mu$, используемым в качестве параметра порядка. Для упрощения расчетов примем в дальнейшем без потери общности $\alpha = 1/2$. В приближении среднего поля (это приближение является точным в случае глобального взаимодействия при $M \rightarrow \infty$), заменяя x_j средним значением μ , систему уравнений (6)—(7) можно записать в виде

$$dx_i/dt = f(x_i) - D(x_i - \mu) + g(x_i)\zeta_i(t), \quad (8)$$

где μ определяется с помощью самосогласованного уравнения

$$\mu = \int_0^1 x P_{st}(x, \mu) dx \equiv \Phi(\mu), \quad (9)$$

с использованием стационарной плотности вероятности $P_{st}(x, \mu)$. Последнюю определим из стационарного уравнения Фоккера—Планка [7]

$$0 = d/dx_i(-f(x_i) + D(x_i - \mu) + \sigma^2/2g(x_i)d/dx_j(g(x_j)))P_{st}(x_i). \quad (10)$$

Из последнего уравнения (опуская нижний индекс i) найдем $P_{st}(x)$ [7]:

$$P_{st}(x) = Z^{-1} \exp \left\{ \int_0^x dy (f(y) - \sigma^2/2g(y)g'(y) - D(y - \mu)(\sigma^2/2g^2(y))^{-1}) \right\}, \quad (11)$$

где Z — нормированная постоянная ($Z = \int_0^1 P_{st}(x, \mu) dx$); $g'(y)$ — обозначение производной.

Из уравнения (11) для $\alpha = 1/2$ получим

$$P_{st}(x, \mu) = (Zg(x))^{-1} \exp \{ 2/\sigma^2 [-(1 + 2Dx + 2D\mu(1 - 2x))/(2x(1 - x)) - \gamma \ln((1 - x)/x) + D \ln(x/(1 - x))(2\mu - 1)] \}. \quad (12)$$

Исследуем сначала инновационный процесс, описываемый уравнением (8) при $D = 0$, т. е. в отсутствие взаимосвязей между участниками технологического трансфера с разными технологиями. В этом случае из уравнения

$$f(\bar{x}) - (\sigma^2/2g(\bar{x})g'(\bar{x})) = 0 \quad (13)$$

легко найти экстремумы стационарной плотности вероятности \bar{x} [7]. Например, в случае $\gamma = 0$ (для простоты) стационарное решение детерминированного уравнения (3) $\bar{x} = 1/2$ ($\alpha = 1/2$). При флуктуирующем параметре γ из уравнения (13) получим [6]

$$\bar{x}_1 = 1/2, \quad \bar{x}_{2,3} = (1/2)[1 \pm (1 - 4/\sigma^2)^{1/2}]. \quad (14)$$

Таким образом, при $\sigma^2 > 4$ стационарная плотность вероятности имеет три экстремума, из которых $\bar{x}_{2,3}$ — максимумы, а наиболее вероятное детерминированное состояние $\bar{x}_1 = 1/2$ в стохастическом случае превращается в минимум (рис. 2). То есть при превышении критического значения интенсивности шума $\sigma_c^2 = 4$ ($\gamma = 0$) плотность вероятности становится бимодальной. В асимметричном случае $\gamma \neq 0$ ситуация качественно остается такой же, только пороговое значение интенсивности шума возрастает с увеличением $|\gamma|$ (рис. 3). В этом случае при $\gamma > 0$ пик, соответствующий в детерминированном уравнении стационарному состоянию, движется в направлении $x = 1$ по мере увеличения σ^2 , и если σ^2 превышает некоторое пороговое значение $\sigma_{пор}^2 > 4$, то на конечном расстоянии от первоначального пика, вблизи другой границы пространства состояний ($x = 0$) появляется второй пик (рис. 3). Иначе говоря, любой переход в рассматриваемой модели инновационного процесса всегда является чисто шумо-

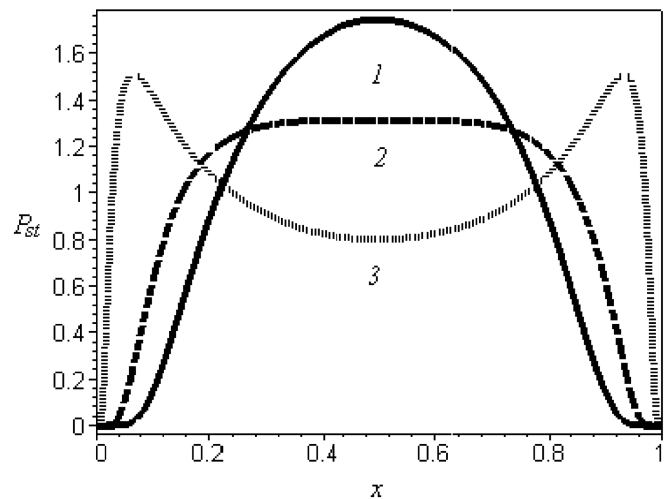


Рис. 2. Зависимость стационарной плотности вероятности распределения $P_{st}(x)$ от инновационной активности x при $\gamma = 0$, $D = 0$. Значение интенсивности шума: для кривой 1 — $\sigma^2 = 1$; для кривой 2 — $\sigma^2 = 4$; для кривой 3 — $\sigma^2 = 16$

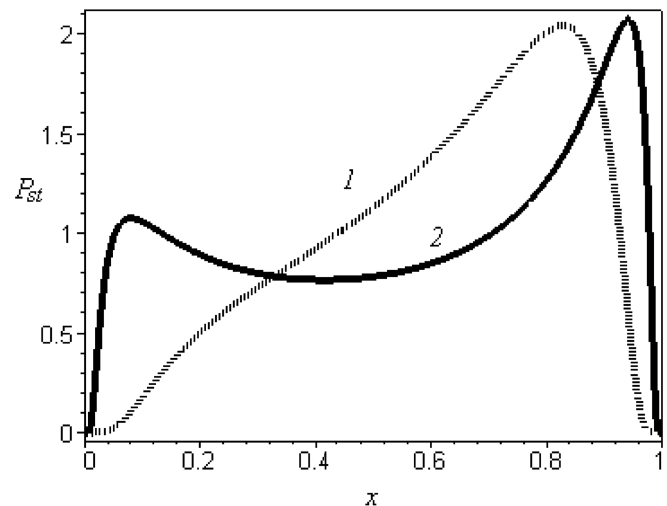


Рис. 3. Зависимость стационарной плотности вероятности распределения $P_{st}(x)$ от инновационной активности x при $\gamma = 1$, $D = 0$. Значение интенсивности шума: для кривой 1 — $\sigma^2 = 4$, для кривой 2 — $\sigma^2 = 16$

вым процессом. В работе [6] резюмируется, что экстремумы стационарной плотности вероятности $P_{st}(x)$ претерпевают в верхней полуплоскости (γ, σ^2) катастрофу типа сборки с критической точкой (острием клюва) в $(0, 4)$. В отличие от работы [6] автор настоящей публикации настаивает на том, что все переходы в исследуемой модели как при $\gamma = 0$, так и при $\gamma \neq 0$ являются мягкими (критическими). Субкритических переходов (скачкообразных), характерных для катастроф типа сборки, автором не обнаружено.

На рис. 4 приведены решения уравнения (9) при $\gamma = 0$, $D = 16$ и $\sigma^2 = 16$. На приведенном графике присутствуют решения, соответствующие лишь критической точке при $\bar{x} = 1/2$. Субкрити-

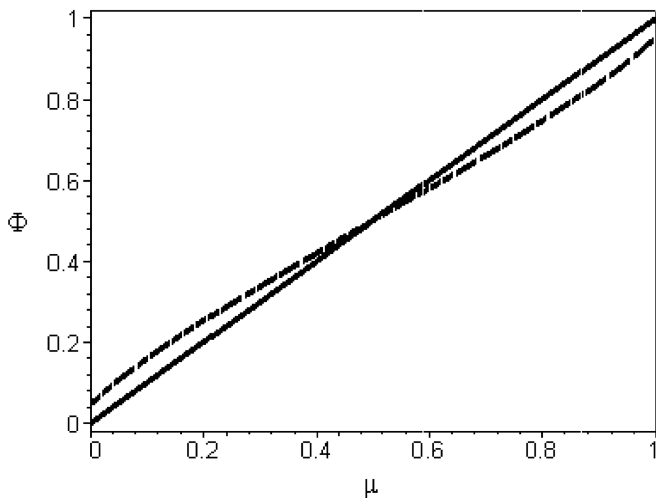


Рис. 4. Решения самоогласованного уравнения (9) при $\gamma = 0$, $D = 16$ и $\sigma^2 = 16$

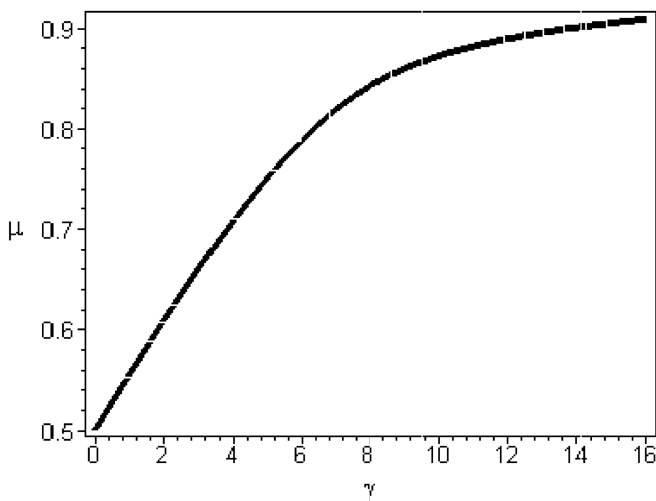


Рис. 5. Зависимость параметра порядка μ от параметра тренда γ . Значение интенсивности шума $\sigma^2 = 16$, параметр взаимодействия $D = 16$

ческих точек, соответствующих скачкообразным переходам, не обнаружено. В пользу приведенных утверждений свидетельствует и поведение параметра порядка μ в зависимости от параметра тренда γ (рис. 5), характерное для критических явлений (отсутствие разрывов на кривой зависимости μ от γ).

Автор считает, что описанные выше закономерности, индуцированные мультипликативным белым шумом в уравнении (6), характерны для явления "самоорганизованной критичности" [8]. В пользу последнего утверждения свидетельствует и приведенная на рис. 6 зависимость спектральной плотности $S(f)$ решений стохастического уравнения (6) от частоты. При этом спектр плотности имеет на низких частотах степенной вид $S(f) \sim f^{-\beta}$, где $\beta \sim 1$. Подобная зависимость называется фликкер-шумом [8]. Наличие послед-

него означает, что значительная часть энергии исследуемой системы связана с очень медленными процессами и в такой системе возможны очень большие флуктуации. Это позволяет предположить, что система находится в окрестности критической точки, или точки бифуркации, где обычно и происходят такие явления. В пользу самоорганизованной критичности свидетельствует также наблюдающееся на практике лавинообразное протекание инновационных процессов в период замен одних технологий на другие более совершенные и приводящее к скачкам на логистических кривых (см. рис. 1). Кроме того, одной из закономерностей самоорганизованной критичности является наличие прерванного равновесия, или перемежаемости, заключающееся во вспышках высокой инновационной активности, прерывающих состояние относительного покоя, когда ее уровень низок. Заметим, что явление перемежаемости происходит в результате самоорганизации системы и приводит к наблюдающимся в реальной практике инновационным циклам (циклам Шумпетера) [3].

Еще одной отличительной чертой систем, в которых наблюдается самоорганизованная критичность, являются степенные законы распределения вероятностей, т. е. статистические характеристики происходящих в них событий обыкновенно имеют плотность вероятности вида $p(x) \sim x^{-(1+\theta)}$. При этом показатель θ обычно лежит в диапазоне от нуля до единицы [8]. Степенное распределение имеют характеристики многих явлений, в том числе и научная продуктивность исследований [8], имеющая отношение к инновациям. Наконец, о роли степенных законов и самоорганизованной критичности в эпоху инноваций сказано в работе [9],

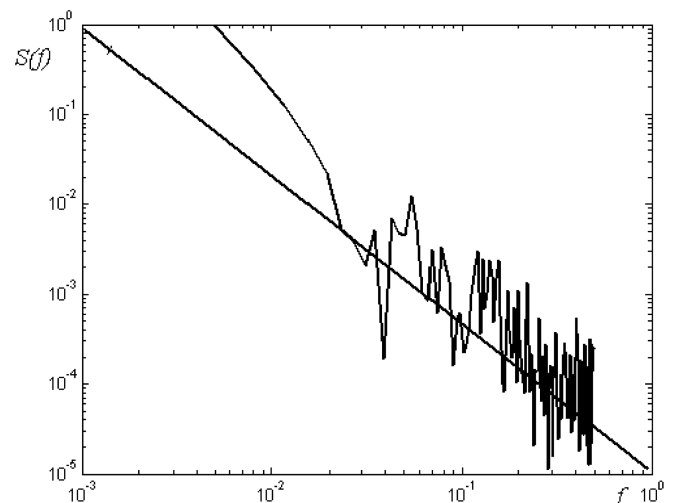


Рис. 6. Зависимость спектральной плотности S решений стохастического уравнения (6) от частоты f в двойном логарифмическом масштабе. Значения параметров: $\alpha = 1/2$, $\gamma = 1$, $D = 0$, $\zeta(t) = 1$

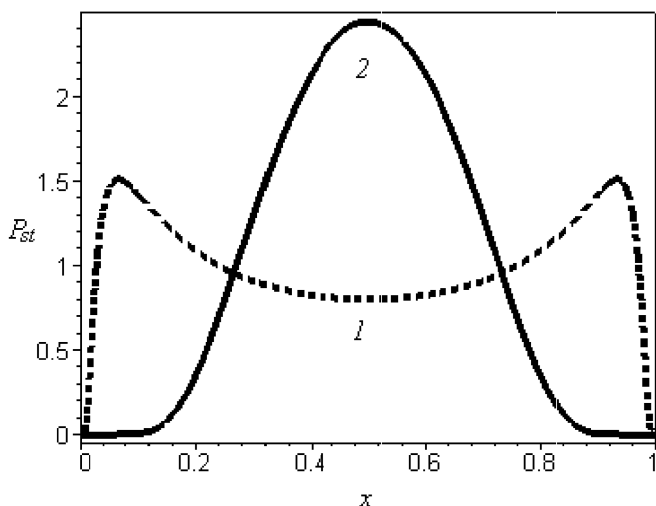


Рис. 7. Зависимость стационарной плотности вероятности распределения $P_{st}(x)$ от инновационной активности x при $\sigma^2 = 16$, $\gamma = 0$. Значение параметра взаимодействия: для кривой 1 $D = 0$, для кривой 2 $D = 16$

где отмечается, что приспособляемость организации к нововведениям является наивысшей как раз в области самоорганизованной критичности.

Одним из главных проявлений наличия взаимодействия между участниками технологического трансфера ($D > 0$) является то, что включение этого взаимодействия приводит к подавлению шумового эффекта. Это отчетливо видно на рис. 7, где приведены кривые стационарной плотности вероятности P_{st} при интенсивности шума $\sigma^2 = 16$ и различных значениях D . Если при $D = 0$ зависимость $P_{st}(x)$ имеет бимодальный характер, то при $D = 16$ — кривая зависимости $P_{st}(x)$ имеет один максимум. Таким образом, увеличение взаимодействия между участниками инновационного процесса (инновационная глобализация) приводит к некоторому подавлению флуктуаций, к уменьшению роли случайности в появлении и развитии инноваций. Это означает, что, регулируя значение этого взаимодействия, можно управлять инновационным процессом в условиях неопределенности. Заметим, что в настоящее время на практике доминирует как раз парадигма все большей взаимосвязанности различных процессов в области инноваций, что облегчается развитием информационных и коммуникационных технологий. Новые технологии появляются в результате слияния специализированных направлений, например, геномика, биоинформатика или комбинаторная химия [9]. Важнейшим направлением в современном развитии инновационной глобализации является формирование международных инновационных сетей, совместное научно-техническое творчество разных стран и их корпораций в интересах не только бизнеса, но и экономического и научно-технического развития всего мира.

Нарастание активности в использовании информационных технологий и Интернета убеждает каждого, что прежнее мышление на основе старых экономических парадигм больше не является надежным. Нелинейное поведение экономики становится все более очевидным фактом. В результате прежняя парадигма менеджмента устарела. Возникла необходимость в разработке новых приемов менеджмента, основанных, в том числе, на нелинейном поведении инновационных процессов. Поэтому исследование нелинейных процессов в экономике и факторов, влияющих на них, особенно актуально. В настоящей работе в рамках синергетического подхода изучались стохастичность инновационного процесса и влияние на этот процесс инновационной глобализации, являющейся системным фактором. Автором показано, что роль случайностей (шума) заключается в индуцировании ими явления, называемого самоорганизованной критичностью. Роль ее проявляется в наблюдающемся на практике лавинообразном протекании инновационных процессов в период замен одних технологий на более совершенные и приводящем к скачкам на логистических кривых. Кроме того, одной из закономерностей самоорганизованной критичности является наличие прерванного равновесия или перемежаемости, заключающееся во вспышках высокой инновационной активности, прерывающих состояние относительного покоя, когда ее уровень низок. Вовлечение в инновационный процесс все большего числа участников (инновационная глобализация) приводит к некоторому подавлению флуктуаций, к уменьшению роли случайности в появлении и развитии инноваций.

Список литературы

1. Буланичев В. А., Серков Л. А. Самоорганизация экономических систем с детерминированным хаосом // Математическое моделирование. 2007. Т. 19. С. 116–126.
2. Яблонский А. И. Математические модели в исследовании науки. — М.: Мысль, 1986.
3. Grubler A. Time for a Change: On the Pattern of Diffusion of Innovation // Daedalus. 1996. N 1. P. 19–42.
4. Полтерович В. М., Хенкин А. А. Диффузия технологий и экономический рост. М.: Наука, 1988.
5. Маевский В. И. Введение в эволюционную макроэкономику. — М.: Изд-во "Япония сегодня", 1997.
6. Horsthemke W., Lefever R. Noise-Induced Transitions. Berlin: Springer, 1984.
7. Van den Broeck C., Parrondo J. M. R., Toral R., Kawai R. Nonequilibrium phase transitions induced by multiplicative noise // Phys. Rev. E. 1997. V. 55. P. 4084–4094.
8. Малинецкий Г. Г., Потапов А. Б., Подлазов А. В. Нелинейная динамика. Подходы, результаты, надежды. — М.: КомКнига, 2006.
9. Янсен Ф. Эпоха инноваций. — М.: Инфра-М, 2002.

А. С. Соловецкий, аспирант,
Институт экономики и организации
промышленного производства СО РАН,
г. Новосибирск,
e-mail: solovetski@gmail.com

Мобильные информационные технологии в маркетинге и рекламе

Отражены результаты, полученные автором в ходе исследования рынка услуг мобильного маркетинга. Целью исследования являлась оценка особенностей использования мобильных информационных технологий в маркетинге и рекламе.

Приведенные данные по различным региональным рынкам позволяют сделать вывод о перспективности услуг мобильного маркетинга на мировом рынке и, в частности, в России. Отмечены преимущества мобильного маркетинга и его особенности по сравнению с классическими рекламными инструментами. Подробно разобраны мобильные технологии, применяемые в маркетинге, и предложены методы их использования. Также рассмотрены этические и технические проблемы, возникающие при использовании инструментов мобильного маркетинга.

Статья будет интересна специалистам, работающим в сферах мобильной связи, маркетинга и рекламы, исследователям, занимающимся проблематикой отрасли информационных технологий, а также всем интересующимся мобильной связью и новыми инструментами маркетинга.

Ключевые слова: мобильный маркетинг, мобильная реклама, мобильные информационные технологии, мобильная связь, услуги с добавленной стоимостью, мобильный контент.

Настоящая статья отражает результаты, полученные автором в ходе исследования рынка услуг мобильного маркетинга. Цель исследования заключалась в оценке особенностей использования мобильных информационных технологий в маркетинге и рекламе.

Мобильный маркетинг (ММ) — это совокупность мероприятий, связанных с продвижением товаров и услуг с использованием мобильных информационных технологий. Мобильные информационные технологии включают в себя мобильную связь и сопутствующие ей технологии.

Термины "мобильный маркетинг" и "мобильная реклама" в периодической печати и Интернете чаще всего используются как эквивалентные. В настоящей статье эти термины используются в том же ключе.

О перспективах ММ можно судить по действиям крупных компаний, задающих общую тенденцию в IT-индустрии. Например, Microsoft в мае 2007 г. приобрела рекламную компанию Screen-Tonic. А компания Google, в свою очередь, начала использовать мобильную рекламу на своем поисковом сервере (www.google.com) и в картографической системе Google Maps.

Группа мобильных компаний Vodafone и испанский сотовый оператор Telefonica выкупили часть компании Amobee Media Systems, занимающейся мобильной рекламой, в целях предотвращения угрозы вытеснения с этого рынка онлайн-новыми гигантами Google и Yahoo.

Согласно прогнозам ABI Research, к 2013 г. мировой рынок услуг ММ составит \$24 млрд. В 2007 г. ММ принес \$1,8 млрд. Опрос тысячи взрослых американцев, проведенный исследовательской компанией GFK/NOP Research, показал, что "каждый третий мобильный пользователь хотя бы раз видел или слышал мобильную рекламу в четвертом квартале 2007 г. При этом SMS-сообщения значились доминантным средством передачи рекламы. Одна треть людей, получивших мобильную рекламу, перезвонили в ответ; из них 41 % — это женщины и 27 % — мужчины [1].

Согласно данным компании eMarketer, затраты на ММ только на рынке США вырастут до \$4,8 млрд в 2011 г. (с \$421 млн в 2006 г.), т. е. с 2,6 до 12 % общего рекламного рынка страны. В 2007 г., как считает eMarketer, объем рынка ММ в США удвоился и составил \$878 млн. А к 2010 г. ММ и контент, связанный с широкоэвещательным ТВ, составят почти \$1 млрд. Росту способствует высокий процент отклика в этом сегменте рекламного рынка, который оборачивается высоким CPM (стоимость миллиона показов рекламы) — \$41 в 2006 г. [2].

По данным JupiterResearch, к 2012 г. доход мобильной рекламы на европейском рынке достигнет \$1,93 млрд (1,2 млрд евро) [3]. В отчете исследования сообщается, что кроме SMS-маркетинга многие "мобильные маркетинговые стратегии все еще находятся в зачаточном состоянии, и им придется преодолеть множество препятствий, чтобы выйти на должный уровень".

Компания ABI Research сообщает, что к 2012 г. рынок мобильной рекламы и маркетинга в Японии достигнет \$1,2 млрд. По мнению ABI Research, рост южнокорейского рынка позволит достигнуть \$684 млн также к 2012 г. [4].

По данным компании Mobile Direct, в 2008 г. объем рынка мобильной рекламы в России достигнет \$3—5 млн. В ближайшие годы рынок будет, как минимум, удваиваться, опережая темпы роста рынка рекламы в обычном Интернете [5].

Приведенные данные по различным региональным рынкам услуг позволяют сделать вывод о

перспективности услуг ММ на мировом рынке и, в частности, в России.

Преимущества мобильного маркетинга

Мобильный телефон находится рядом со своим владельцем 24 ч в сутки. С помощью своего телефона человек принимает и осуществляет звонки, отправляет и получает сообщения, звонит в сервисные службы, пользуется мобильным Интернетом, загружает и пересылает другим абонентам информационные ресурсы (контент) — текст, графику, музыку, звук, видео.

Благодаря ММ каждый абонент мобильной связи имеет возможность получать информацию об интересующих его товарах и услугах с помощью своего телефона.

ММ обладает рядом преимуществ в сравнении с традиционными маркетинговыми инструментами.

- **Интерактивность.** Абоненты, являющиеся активными участниками акций ММ, имеют возможность влияния на ход маркетинговой компании, реагируя на рекламное воздействие через свой мобильный телефон. Таким образом, рекламодатель может узнавать мнение потребителя о рекламируемых товарах и услугах и его отношении к проводимой рекламной акции.
- **Четкий фокус на целевую аудиторию.** ММ позволяет показывать рекламу именно той аудитории, которую интересуется рекламируемый продукт. Например, если человек заходит на мобильный сайт, то ему будет демонстрироваться реклама продуктов, связанных с тематикой этого сайта.
- **Точность оценки эффективности рекламной компании.** С помощью мобильной рекламы можно с точностью до человека отследить число откликнувшихся на эту рекламу, потому что существует возможность мгновенной обратной связи с потребителем (использование мобильных купонов на скидку, нажатие (клик) на рекламный баннер на мобильном сайте).
- **Высокая степень психологического воздействия на потребителя.** Мобильный телефон, наверное, единственное устройство, которое находится рядом с человеком все время на протяжении дня во включенном состоянии, и информация на экране мобильного телефона зачастую полностью занимает внимание человека по причине небольших размеров этого экрана.
- **Широкая рекламная аудитория.** По оценке Мининформсвязи [6], сотовой связью в России пользуются примерно 98 млн человек (67,6 % населения)¹. Это колоссальная аудитория по-

ребителей товаров и услуг. В ближайшие годы численность этой аудитории будет только увеличиваться.

Технологии, используемые в мобильном маркетинге

Рассмотрим основные технологии и способы их применения в ММ.

SMS (Short Message Service — служба коротких сообщений), **MMS** (Multimedia Message Service — служба мультимедийных сообщений).

SMS — это короткие текстовые сообщения, рассылаемые абонентам — самый старый вид мобильной рекламы (ему более десяти лет). С помощью SMS производители товаров проводят мгновенные розыгрыши призов, для этого покупателю нужно отправить код, напечатанный на упаковке купленного товара.

MMS — технология передачи текста, графики, звука и видео в формате мультимедийных сообщений. В России MMS-сообщения пока не получили широкого распространения в сфере ММ. Тем не менее, у MMS — большое будущее, особенно с развитием сетей связи третьего поколения.

Bluetooth. Технология ближней (до 30 м) беспроводной радиосвязи в диапазоне частот 2,4 ГГц. Позволяет передавать голос и данные между мобильными устройствами. В ходе маркетинговых акций практикуется загрузка рекламного контента (видеороликов, купонов со скидкой) через Bluetooth-соединение. Абонент, находящийся в радиусе действия Bluetooth-передатчика, получает на свой телефон запрос о загрузке рекламного контента. Он может отклонить запрос либо принять его и загрузить контент на свой телефон.

JAVA. JAVA представляет собой программную платформу, на основе которой создаются приложения для мобильных телефонов. В JAVA-играх используется баннерная реклама и продакт-плейсмент (использование в сюжете игры известных торговых марок и продуктов в целях скрытой рекламы). Новый инструмент мобильной рекламы на основе технологии JAVA — реклама на "спящем" экране (Idle Screen ad). Реклама на "спящем" экране представляет собой рекламную заставку, которая проигрывается на экране телефона, когда он не используется абонентом. Такие заставки могут транслироваться напрямую из Интернета.

WAP (Wireless Application Protocol — протокол беспроводного доступа). WAP позволяет просматривать мобильные сайты и порталы на экране телефона. *Портал* — это web-сайт, предназначенный для предоставления интегрированной информации. Порталы реализованы как многоуровневое объединение различных ресурсов и сервисов и содержат ссылки на другие сайты, контент которых

¹ Это оценка числа реальных абонентов, а не числа SIM-карт на руках у населения, которых по состоянию на конце декабря 2007 г. по данным компании АСМ-Consulting в России насчитывалось 172,8 млн шт.

отвечает интересам посетителей портала. Мобильные порталы (WAP-порталы) имеют урезанный функционал и формат для удобства их просмотра на мобильном телефоне. Обычно на мобильных порталах используется баннерная реклама. Как правило, тематика такой рекламы совпадает с тематикой портала. Очень часто можно встретить CRM-решения (решения по управлению взаимодействием с клиентами) в онлайн-овых мобильных магазинах.

Рассмотрим в качестве примера CRM-возможности, которые предлагают порталы по продаже таких товаров, как книги, CD, DVD:

- настройка интерфейса сайта под предпочтения пользователя;
- накопительная система скидок;
- показ клиенту аналогичных по тематике товаров на основе ранее сделанных заказов;
- автоматическая авторизация при заходе клиента на сайт (сайт "узнает" клиента);
- отображение лишь тех товаров, которые клиент может купить в настоящий момент;
- возможность заказа "в один клик" с использованием параметров предыдущих заказов. Клиенту нужно лишь выбрать товар для заказа, а система предложит использовать параметры предыдущих заказов (адрес доставки, способ оплаты, способ доставки товара).

Многие компании создают мобильные версии своих промопорталов для продвижения товаров и услуг. Например, очень часто киностудии заказывают такие порталы контент-провайдерам в целях продвижения своих новых фильмов.

На промопорталах размещается бесплатный контент. Роль такого контента заключается в напоминании человеку о рекламируемом товаре и побуждении его покупать этот товар и сопутствующие товары. Например, если речь идет о фильме, то такими товарами будут билеты в кино, DVD с фильмом, CD с музыкой из фильма, сувениры с символикой фильма. Часто бесплатные видеоклипы содержат десятисекундные рекламные вставки от спонсора.

Также реклама используется в результатах поиска на WAP-порталах. В этом случае пользователь видит рекламу, которая тематически связана с его поисковым запросом (контекстная реклама).

Некоторые компании организуют на своих WAP-порталах мобильную социальную сеть в целях увеличения посещаемости портала, а следовательно, и продаж (см., например [7]). *Социальные сети* представляют собой онлайн-овые сообщества людей с общими интересами. Например, компания, продающая горнолыжную экипировку, может создать на своем мобильном сайте сообщество любителей горнолыжного спорта.

Другие компании создают мобильные социальные сети в поддержку своих продуктов. В такие

сети входят пользователи лояльной компании и ее продукции. Таким образом, сеть формирует положительный имидж компании и ее продуктов в глазах пользователей. Такую сеть, к примеру, создала компания Nokia.

GPRS (General Packet Radio Services — пакетная радиосвязь общего назначения) и **EDGE** (Enhanced Data rates for GSM [Global] Evolution — расширенная передача данных для [глобальной] эволюции GSM-сетей). GPRS и EDGE — технологии мобильного Интернета в GSM-сетях. GPRS обеспечивает передачу данных со скоростью до 171 Кбит/с, EDGE — до 384 Кбит/с. Так как EDGE не требует больших доработок существующих сетей сотовой связи, она считается переходной к сетям третьего поколения (3G), в частности, к технологии HSPA. GPRS и EDGE обеспечивают подключение к Интернету в мобильных сетях второго поколения (2G, 2.5G).

HSPA (High-Speed Packet Access — высокоскоростная пакетная передача данных). HSPA — технология беспроводной широкополосной радиосвязи, использующая пакетную передачу данных и являющаяся надстройкой к мобильным сетям третьего поколения WCDMA/UMTS. Обеспечивает передачу данных с пропускной способностью 384 Кбит/с — 3,6 Мбит/с. Теоретическая максимальная пропускная способность — 14,4 Мбит/с. По своей функциональности аналогична Wi-Fi — обеспечивает доступ к Интернету в мобильном телефоне. В России эта технология начнет внедряться после запуска сетей третьего поколения.

Wi-Fi (Wireless Fidelity — беспроводная точность). Технология Wi-Fi позволяет мобильным устройствам, поддерживающим эту технологию, подключаться к беспроводному Интернету на скоростях до 100 Мбит/с, что сравнимо со скоростью стационарного Интернета. Следует отметить, что технология Wi-Fi, а также GPRS, EDGE, HSPA не используются напрямую в ММ, а служат абонентам мобильных сетей в качестве платформы для доступа к мобильному Интернету.

USSD (Unstructured Supplementary Service Data — неструктурированные дополнительные служебные данные). USSD — технология текстовых запросов вида *100# или *108*12345#. Позволяет абонентам получать различную справочную информацию и подключать услуги. После набора определенного кода абонент нажимает кнопку посылы вызова. Ответом на запрос абонента является SMS-сообщение.

USSD может использоваться для участия абонентов в рекламных акциях. Код указывается в рекламном сообщении. Набрав указанный код, абонент таким способом подключается к рекламной акции.

Производитель товара может запустить акцию для защиты своего товара от подделки. Чтобы

проверить подлинность купленного товара, покупатель отправляет USSD-запрос, содержащий код с упаковки товара.

Cell Broadcast (сотовое вещание). Технология обеспечивает рассылку базовой станцией сообщений на все сотовые телефоны, находящиеся в ее зоне покрытия. Например, владелец ресторана может заключить договор с оператором мобильной связи, которому принадлежит базовая станция в районе местонахождения ресторана, чтобы каждый абонент этого оператора при нахождении в области обслуживания данной станции получал на свой мобильный телефон сообщение с предложением посетить этот ресторан и получить скидку на обслуживание. Скидка предоставляется в том случае, если абонент демонстрирует менеджеру сообщение на экране своего телефона. Сообщение должно содержать уникальный код, который дает право на скидку.

IVR (Interactive Voice Response — интерактивный голосовой ответ). Данная технология чаще всего используется операторами мобильной связи. Абонент, следуя указаниям автоответчика, нажимает определенные кнопки на телефоне в тональном режиме и получает интересующую его информацию. Услугу на основе этой технологии часто называют *интерактивным автоответчиком* или *голосовым порталом*.

Наиболее прогрессивная разновидность IVR может распознавать голосовые команды звонящего, а использование технологии HSPA позволяет организовать video IVR, когда абонент не только слышит голос, но и видит на экране изображение говорящего человека. В маркетинге IVR часто используется для информирования абонентов об условиях рекламных акций. Также IVR может применяться в качестве CRM-решения в службе технической поддержки.

NFC (Near-Field Communication — связь на ближней дистанции) и **RFID** (Radio Frequency Identification — радиочастотная идентификация). Технология RFID позволяет помечать различные объекты миниатюрными перепрограммируемыми RFID-чипами, которые содержат информацию об этих объектах. NFC — технология связи на ближней дистанции (0,1—0,2 м), наподобие Bluetooth, позволяющая считывать данные с RFID-чипов и передавать данные с устройств, оснащенных таким чипом.

В Японии функционирует система мобильных платежей Felica на основе технологии NFC, позволяющая оплачивать товары и услуги с помощью своего мобильного телефона. Для оплаты товара или услуги нужно провести своим телефоном, оснащенным RFID-чипом, над считывающим устройством и с вашего счета спишется необходимая денежная сумма.

Использование мобильного телефона с RFID-чипом в качестве электронного кошелька позволяет абоненту участвовать в дисконтных программах и розыгрышах призов среди покупателей различных товаров и услуг. Используя телефон с поддержкой NFC, можно считывать информацию с рекламных постеров и товаров, оснащенных RFID-чипом.

DVB-H (Digital Video Broadcasting [for] Handheld — цифровая видеотрансляция для мобильных устройств). DVB-H — европейский стандарт цифрового мобильного телевидения. Позволяет принимать видеотрансляции в цифровом формате на мобильных телефонах, поддерживающих данный стандарт. DVB-H позволяет использовать телефон в качестве рекламного носителя практически так же, как и телевизор.

Преимущество рекламы на мобильном телевидении заключается в ее *интерактивности*. Интерактивность дает возможность получения мгновенной обратной связи от абонента. Компания-рекламодатель может выступать спонсором разнообразных интерактивных игр, викторин, конкурсов, розыгрышей, голосований и опросов, проводимых в эфире мобильного телевидения. Тестовые трансляции в формате DVB-H проводятся в нескольких российских городах.

QR codes (Quick Response Codes — "коды быстрого ответа"), или QR-коды, — двумерный матричный код, в котором зашифрована различная информация. QR-код выглядит, как квадратная область, состоящая из очень мелких черных и белых прямоугольников, подобно штрих-коду. QR-коды печатают в журналах, газетах, рекламных проспектах, постерах, на визитных карточках, наносят на упаковки с товарами.

QR-код может быть распознан с помощью мобильного телефона со специальным программным обеспечением. Для этого нужно навести камеру телефона на код и сфотографировать его. После этого программное обеспечение телефона, распознав код, предложит владельцу телефона выполнить какое-либо действие, зашифрованное в коде, — зайти на определенный сайт, загрузить рекламный видеоклип из Интернета, отправить SMS-сообщение или совершить звонок на специальный номер. В Европе QR-коды транслируют по телевидению — абонент может активировать код, поднеся свой телефон к экрану телевизора.

GPS (Global Positioning System — система глобального позиционирования). Технология GPS позволяет определять местоположение мобильного устройства с точностью до 10 м. Координаты мобильного устройства определяются с помощью навигационных спутников и наземных станций слежения. Использование GPS в ММ имеет принципиальную схожесть с использованием технологий Bluetooth и Cell Broadcast.

Использование GPS в мобильном поиске предоставляет абоненту рекламную информацию на основе его местоположения. Например, в случае поиска абонентом информации о кинотеатрах поисковая система показывает абоненту информацию о тех кинотеатрах, которые находятся с ним рядом. Часто такой поиск осуществляется с использованием интерактивных географических карт в навигационном программном обеспечении. Такие маркетинговые инструменты получили название *маркетинга на основе местоположения* (location based marketing).

Также перспективным направлением маркетинга являются интерактивные игры на основе GPS, проводимые в целях поддержки продаж нового товара.

Морально-этические аспекты использования мобильного маркетинга

Мобильный телефон — устройство очень личное, почти интимное, и мобильная реклама в некоторых случаях может вызывать более негативную реакцию потребителей, чем реклама телевизионная. Поэтому негативный эффект от получения нежелательных SMS гораздо выше, чем эффект от спама в электронной почте. Если в Интернете человек может завести другой почтовый ящик, то для избавления от нежелательных сообщений, приходящих на мобильный телефон, приходится блокировать прием всех сообщений или менять номер телефона.

Доклад автора по теме ММ на конференции молодых ученых, проходившей в г. Барнауле в апреле 2007 г., вызвал оживленную дискуссию в аудитории. Большинство присутствующих выразили нежелание получать рекламу на свои мобильные телефоны и предоставлять какую-либо информацию о своих потребительских предпочтениях.

По способу воздействия на потребителя реклама может осуществляться двумя методами:

- методом "проталкивания" (push). Это традиционная массовая реклама, ориентированная на широкую аудиторию без учета индивидуальных интересов потребителей;
- методом "вытягивания" (pull). Это реклама с учетом информации, "вытянутой" из конкретного потребителя о его интересах и предпочтениях относительно товаров и услуг.

Для того чтобы снизить до минимума эффект нежелательности рекламы для потребителей, фирмам, предоставляющим услуги ММ, нужно отдать предпочтение pull-инструментам, так как подобные инструменты вовлекают мобильного пользователя в рекламную акцию таким образом, что он участвует в ней с желанием и интересом. Очень важно, чтобы у пользователя всегда была простая возможность отказаться от получения или просмотра мобильной рекламы.

Мобильные операторы имеют доступ к личной информации своих клиентов, такой как имя, пол, возраст, место проживания, хотя они и предпочитают не разглашать ее. Системы мобильных платежей владеют информацией о том, какие товары и услуги люди оплачивают с помощью мобильного телефона. Например, в Японии, где действует платежная система Felica, многие люди практически все товары и услуги приобретают с помощью мобильного телефона. Естественно, доступ к этой информации ограничен. Тем не менее, существует риск кражи такой информации, так же как это иногда происходит с номерами кредитных карт. Поэтому в целях защиты персональной информации требуются принятие на государственном уровне соответствующих законов и разработка необходимых технологий компаниями-участниками отрасли мобильных информационных технологий.

Проблемы использования мобильного маркетинга

Помимо морально-этических проблем существуют также специфические проблемы технического и экономического характеров при использовании инструментов ММ.

- **Небольшие размеры мобильных устройств.** Экраны и клавиатуры мобильных телефонов вследствие своих малых размеров не слишком удобны по сравнению с персональным компьютером. Поэтому нужно свести к минимуму неудобства пользователей, связанные с получением визуальной информации на экране телефона и установлением обратной связи, например, когда абонент пишет отзыв на сайт со своего телефона. С этой целью интерфейс мобильных порталов должен быть предельно эргономичным, а подключение различных услуг и участие в рекламных акциях абонентов должно требовать минимальных и простых для понимания абонентом действий с его телефоном.
- **Сложность мобильных информационных технологий для пользователя.** Большинство абонентов мобильной связи используют свой телефон в основном для совершения звонков. Приучить таких людей к участию в рекламных мероприятиях с использованием всевозможных мобильных информационных технологий достаточно сложно. В этой ситуации мобильные рекламные агентства должны найти способ заинтересовать абонентов и побудить их к освоению новых для себя технологий. С этой целью необходимо наличие услуг справочно-разъяснительного характера, которые позволили бы абонентам быстро осваивать новые возможности, которые предоставляют их мобильные устройства.
- **Низкая скорость и нестабильность передачи данных в мобильном Интернете.** Это обстоятельство накладывает ограничения на размеры

загружаемого пользователем контента и лишает возможности смотреть мобильное видео в режиме реального времени. С появлением в России 3G-сетей мобильной связи и развитием Wi-Fi-сетей ситуация должна измениться к лучшему.

- **Зависимость рынка услуг ММ от операторов мобильной связи.** Мобильные операторы хотят получать процент со всех услуг, которые оказываются с использованием их сетей связи. Часто значение этих процентов заставляет рекламные агентства завышать стоимость своих услуг, что делает их невыгодными для рекламодателей. Перемещение деятельности агентств мобильной рекламы из сетей мобильных операторов в мобильный Интернет должно уменьшить влияние операторов на рынок услуг ММ.
- **Существование правовых претензий со стороны правообладателей на отчисления за использование контента (музыки, видео и изображений) в ММ.** Необходимо введение стандартных ли-

цензий на использование контента, что должно значительно упростить работу мобильным рекламным агентствам.

Список литературы

1. **К 2013 году** рынок мобильного маркетинга достигнет \$24 млрд // Мобильный контент. <http://www.procontent.ru/news/7159.html>
2. **Marketer** предсказывает светлое будущее мобильному маркетингу и рекламе // Мобильный контент. <http://www.procontent.ru/news/3665.html>
3. **Европейский** рекламный рынок достигнет \$1,93 млрд к 2012 году // Мобильный контент. <http://www.procontent.ru/news/7166.html>
4. **Рынок** мобильной рекламы в Японии — \$1,2 млрд // Мобильный контент. <http://www.procontent.ru/news/7007.html>
5. **Гетманов М.** Компания Mobile DIRECT, "Текущие тенденции мобильной рекламы" {PDF-презентация}. <http://www.procontent.ru/media/presentations/iv-vas-conference/mobile-direct.pdf>
6. **Цуканов И., Дядко Т.** Компьютер победил трубку // Ведомости, 11 февраля 2008 г., № 24 (2046).
7. **eBay** запустил социальную сеть // Компьюлента. <http://net.compuenta.ru/335352/>

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

УДК 004.942

А. В. Сарафанов, д-р техн. наук,
проф., проректор, e-mail: IAD@stu-kras.ru,
В. А. Комаров, аспирант,
e-mail: vladimir@ire.krgtu.ru,
Сибирский федеральный университет,
г. Красноярск

Разработка математической модели многопользовательского режима функционирования аппаратно-программных комплексов с удаленным доступом

Рассматривается применение математического моделирования для анализа многопользовательского режима функционирования аппаратно-программных комплексов с заданным доступом.

Ключевые слова: многопользовательский режим, математическое моделирование, аппаратно-программный комплекс с удаленным доступом (АПК УД).

Активно внедряемые в последние 10 лет автоматизированные лабораторные практикумы с удаленным доступом (АПК УД), обеспечиваю-

щие дистанционное выполнение лабораторного эксперимента на реальном оборудовании, разрабатываются на основе собственных или унифицированных технологий [1]. В связи с этим в настоящее время стала актуальной проблема применения многопользовательского режима их функционирования и работа таких практикумов в сети *Internet*.

Составной частью систем АПК УД являются аппаратно-программные комплексы с удаленным доступом (АПК УД), построение которых осуществляется на основе обобщенной функциональной схемы, представленной на рис. 1 (см. четвертую сторону обложки).

В качестве *объекта исследования* в АПК УД могут выступать лабораторные стенды, макеты, промышленные установки и другие устройства, с помощью которых проводятся экспериментальные лабораторные исследования.

Исполнительные устройства обеспечивают автоматизированное управление объектом исследования, а также внесение структурных и параметрических изменений в его конфигурацию. Они могут включать в свой состав ключи различных типов, управляемые резистивные матрицы, приводы и т. д.

Интерфейс — функциональный узел, обеспечивающий сопряжение по уровням сигналов устройства сбора данных, исполнительных устройств и объекта исследования.

Устройство сбора данных выполняет функции ввода-вывода измерительных аналоговых и цифровых управляющих сигналов, которые передаются с исследуемого объекта и на него.

Программное обеспечение (ПО) АПК УД распределяется между измерительным ПЭВМ-сервером — серверное ПО и ПЭВМ удаленного пользователя (ПЭВМ-клиент) — клиентское ПО. Подробно базовые конфигурации аппаратно-программных средств рассмотрены в работе [2].

Измерительное ПО, устанавливаемое на ПЭВМ-сервере, непосредственно управляет процессом измерений и формированием массивов (пакетов) результатов измерений, передаваемых клиенту. Измерительное ПО клиента содержит, как правило, программы лицевых панелей виртуальных измерительных приборов и/или виртуальных лабораторных стендов, путем настройки которых формируется траектория измерения, задается конфигурация объекта исследования и соответствующий ей пакет заданий, передаваемых на измерительный ПЭВМ-сервер.

Многопользовательское функционирование АПК УД, обеспечивающее *одновременную работу* с лабораторным оборудованием нескольким пользователям, достигается за счет реализации специализированного алгоритма программного обеспечения серверного уровня. В данном режиме обеспечивается буферизация и очередность доступа к объекту исследования в порядке поступления заданий на измерение. Подготовка эксперимента (выбор структуры объекта исследования, настройка параметров конфигурации, задание параметров тестирующих сигналов) проводится на уровне клиентского ПО. Подготовленное задание на измерение передается на сервер-измеритель лабораторного стенда и в случае занятости лабораторного стенда измерением ставится в очередь. После освобождения измерителя происходит реконфигурация объекта исследования, формирование тестовых сигналов, съем исследуемых сигналов в соответствии с новым заданием, передача массива результатов измерения обратно пользователю и начинается обработка следующего задания. Таким образом, основной идеей, используемой при реализации многопользовательского режима, является то, что *монопольный захват* объекта исследования осуществляется *только на время измерения*. Именно этот факт позволяет говорить об увеличении пропускной способности лабораторного оборудования и, как следствие, о повышении экономической эффективности его использования.

В ФГОУ ВПО "Сибирский федеральный университет" (СФУ), на базе регионального инновационного центра "Центр технологий *National Instruments*" [www.sfu-kras.ru/studies/sdo/ni], на основе изложенного подхода были разработаны АПК

УД "Тракт усиления звуковой частоты", "Электроника", "Устройства приема и обработки сигналов", "Схемотехника аналоговых электронных устройств", "Цифровая схемотехника", функционирующие во многопользовательском режиме и активно внедряемые в учебный процесс [2, 3]. Построение АПП УД с использованием разработанных АПК УД осуществляется на основе унифицированной схемы, представленной на рис. 2, в соответствии с ранее выработанной концепцией [4].

Опыт работы центра показывает, что в процессе разработки и внедрения АПК УД в учебный процесс возникает необходимость решения следующих задач:

1. Расчет качественных параметров функционирования готового АПК УД в целях определения особенностей его использования в учебном процессе (например, возможность подключения дополнительных групп удаленных пользователей).

2. Определение требований к АПК УД и его функциональным узлам на этапе разработки в соответствии с заданными (требуемыми) качественными параметрами функционирования, такими как число одновременно работающих удаленных пользователей, среднее время ожидания результатов измерения.

3. Определение методов повышения качественных параметров функционирования АПК УД.

Решение обозначенных выше задач может осуществляться экспериментальным и теоретическим путями. Экспериментальное решение заключается в проведении серии экспериментов в целях определения и построения соответствующих необходимых зависимостей для каждого АПК УД, что достаточно трудоемко и не может быть применимо на этапе разработки. Теоретическое решение основывается на построении математической модели многопользовательского режима работы АПК УД, что позволяет наиболее полно решить поставленные задачи.

Рассмотрим обобщенную временную диаграмму обработки запроса на измерение (рис. 3).

Значения временных отрезков на диаграмме t_5-t_6 и t_1-t_2 определяются временем прохождения пакета данных по каналу связи (*Internet, Intranet*). Они зависят от его пропускной способности и объема передаваемых данных. Введем обозначение $t_{св} = (t_6 - t_5) + (t_2 - t_1)$. В общем случае $t_{св}$ является случайной величиной.

Участок диаграммы t_2-t_3 характеризуется временем нахождения запроса в очереди и зависит от того, сколько уже запросов находится в очереди на момент поступления нового. Обозначим $t_{оч} = t_3 - t_2$. Параметр $t_{оч}$ является случайной величиной.

Участок диаграммы t_3-t_4 определяется временем реконфигурации устройства в соответствии с поступившим заданием (внесение параметриче-

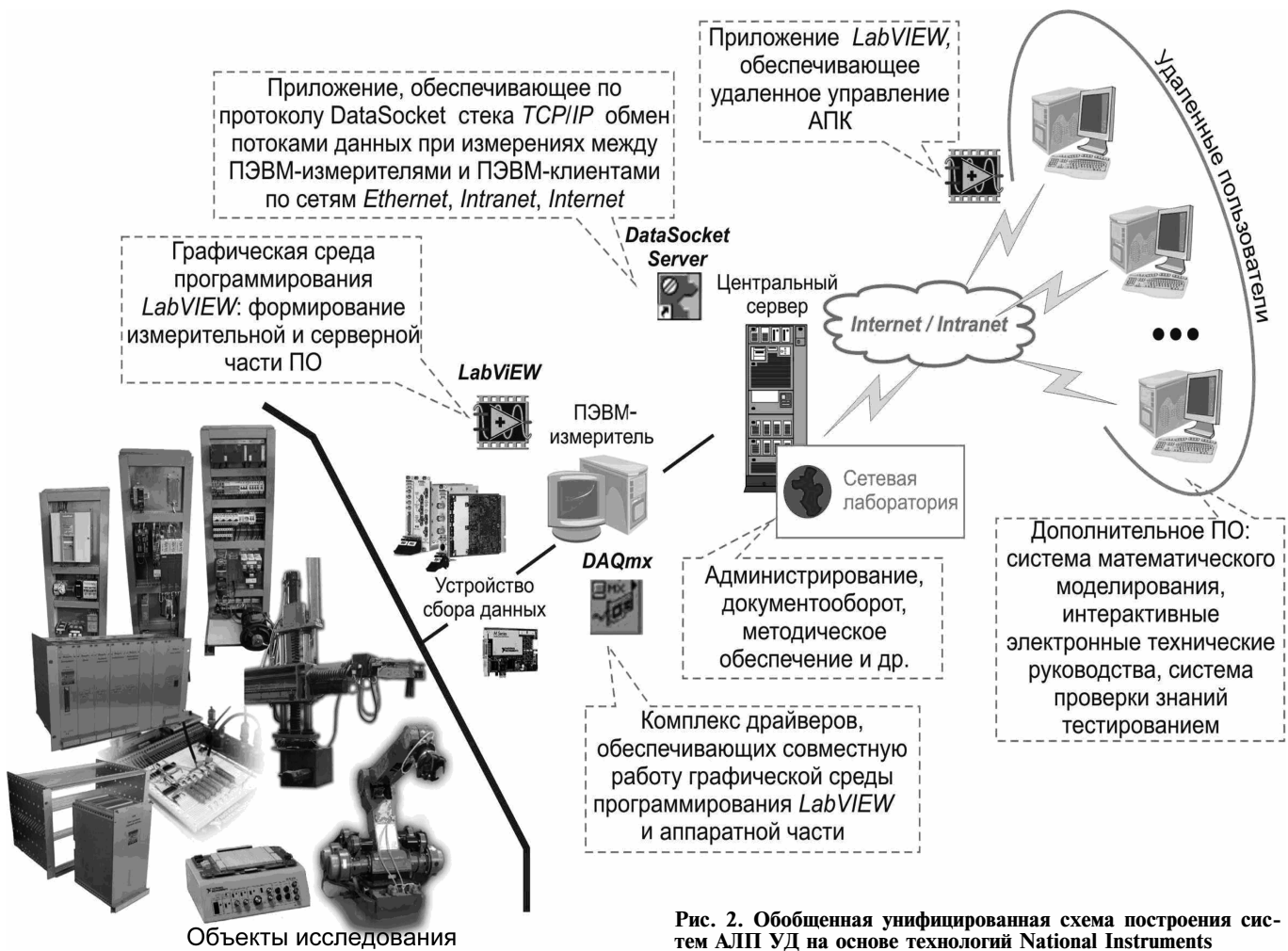


Рис. 2. Обобщенная унифицированная схема построения систем АПК УД на основе технологий National Instruments

ских и структурных изменений в объект исследования) и зависит от характеристик объекта исследования. Введем обозначение $t_{рек} = t_4 - t_3$. Значение $t_{рек}$ может быть фиксированным или определяться дискретным набором детерминированных значений (для конкретного объекта исследования). В общем случае $t_{рек}$ является случайной величиной.

Участок диаграммы $t_4 - t_5$ определяется временем, затраченным на измерение задаваемых параметров и/или характеристик. Данная величина может быть фиксированной (для конкретного объекта исследования) или определяться дискретным набором детерминированных значений (в зависимости от исследуемых характеристик лабораторного макета, типа, метода измерений). Введем обозначение $t_{изм} = t_5 - t_4$. В общем случае $t_{изм}$ является случайной величиной.

Таким образом, время ожидания результатов измерения $t_{ож}$ определяется выражением

$$t_{ож} = t_{св} = t_{оч} + t_{рек} + t_{изм}. \quad (1)$$

Работа по исследованию динамики работы АПК УД, определению $t_{ож}$ и построению статистической модели многопользовательского режи-

ма работы была проведена учеными из Казанского государственного технического университета [5]. В своих исследованиях они описали зависимости временных характеристик обслуживания от следующих факторов: быстродействия информационной сети (скорости передачи данных на участке сервер—пользователь), дней недели (рабочие дни с понедельника по пятницу), времени суток.

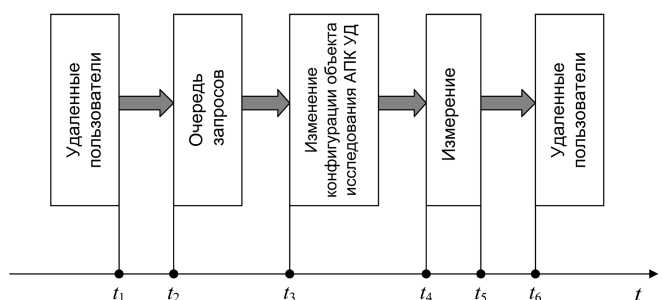


Рис. 3. Обобщенная временная диаграмма обработки запроса на измерение: t_1 — отправка запроса на измерение ПЭВМ-измерителю; t_2 — получение запроса сервером АПК УД, запись в очередь; t_3 — извлечение запроса из очереди, изменение конфигурации объекта исследования АПК УД; t_4 — начало измерения; t_5 — отправка результатов измерения пользователю; t_6 — получение результатов измерения пользователем

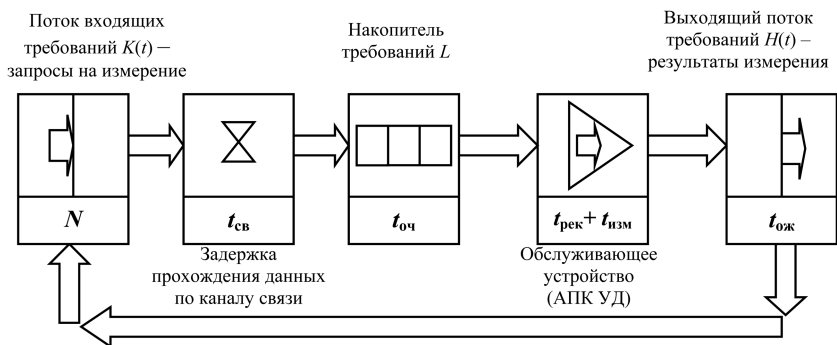


Рис. 4. Обобщенная структура модели функционирования многопользовательского АПК УД

Построение модели основывается на предположении, что время измерения $t_{изм}$ намного меньше времени задержки прохождения данных по каналу связи $t_{св}$, поэтому $t_{изм}$ при анализе не учитывалось. Но ввиду актуальности разработки и внедрения многопользовательских АПК УД данная модель требует доработки в связи с тем, что при современном состоянии пропускной способности информационных сетей время транспортной задержки передачи данных от сервера к удаленному пользователю ($t_{св}$), при среднем объеме передаваемых данных реального учебного эксперимента 50 Кбайт составляет менее 1 с, в то время как процесс реконфигурации и измерения может занимать большее или соизмеримое с ним время.

Обобщенная структура модели функционирования многопользовательских АПК УД как системы массового обслуживания (СМО) приведена на рис. 4.

Объект (предприятие, организация и др.), деятельность которого связана с многократной реализацией исполнения каких-то однотипных задач и операций, классифицируется как *система массового обслуживания* [6, 7]. В нашем случае в качестве СМО выступает АПК, а *обслуживаемой системой* являются ПЭВМ-клиенты (ИВЦ кафедр, институтов, филиалов, центры коллективного пользования (ЦКП), домашние ПЭВМ и т. д.).

В качестве параметра, характеризующего обслуживаемую систему, выступает общее (максимальное) число клиентских ПЭВМ — N . В момент функционирования системы N является постоянной величиной и меняется только при вводе в эксплуатацию новых групп или отдельных пользователей (клиентов). Следовательно, при анализе можно принять, что максимальное число источников требований (запросов на измерение) в системе ограничено и равно N .

После получения результатов измерения пользователь может повторить запрос, т. е. обслуженное требование через некоторое время возвращается в обслуживающую систему. Следовательно, система является замкнутой.

Рассматриваемая система является СМО с ожиданием, так как все поступившие в систему требования не покидают ее до тех пор, пока не будут обслужены. Максимальный объем накопителя очереди требований (L) задается программно, в общем случае он должен превышать N для того, чтобы исключить потерю запросов на измерение. Таким образом, относительно N объем накопителя можно считать бесконечным, так как N никогда не превысит L . Обслуживание выполняется по принципу *first in, first out (FIFO)*

(первым пришел — первым обслужен).

Обслуживающим прибором является АПК УД, от характеристик и параметров которых зависят величины $t_{изм}$ и $t_{рек}$. Значение $t_{рек}$ можно определить по следующей формуле:

$$t_{рек} = t_{упр} + t_{п.пр}, \quad (2)$$

где $t_{упр}$ — время взаимодействия интерфейса, исполнительных устройств и объекта исследования для установки выбранной пользователем конфигурации; $t_{п.пр}$ — время переходных процессов, протекающих в объекте исследования после изменения конфигурации.

Распределение $t_{ож}$ может быть получено на основе имитационного моделирования с использованием специализированных программных пакетов [8] или аналитическим методом.

В качестве примера рассмотрим аналитический вариант анализа на основе упрощенной модели функционирования АПК УД, полагая, что значения $t_{рек}$, $t_{изм}$ и $t_{св}$ фиксированы (для некоторых АПК $t_{рек}$, $t_{изм}$, $t_{св}$ можно принять постоянными в связи с их незначительным изменением).

Таким образом, рассматриваемый частный случай модели представляет собой одноканальную замкнутую СМО с фиксированной величиной $t_{обсл} = t_{рек} + t_{изм}$.

Функционирование любой системы массового обслуживания можно представить через все возможные ее состояния и интенсивность перехода из одного состояния в другое. Исходными параметрами, характеризующими систему массового обслуживания [6], являются:

- число требований (число клиентов) N ;
- интенсивность поступления требований на обслуживание, т. е. число поступлений требований в единицу времени, — λ ;
- интенсивность обслуживания поступивших требований μ .

Интенсивность обслуживания требований определяется как величина, обратная времени обслуживания одного требования. В данном случае

$$\mu = \frac{1}{t_{\text{рек}} + t_{\text{изм}}} \quad (3)$$

Очевидно, что для полного и успешного проведения эксперимента каждому пользователю необходимо выполнить в среднем X_0 измерений за рассматриваемый период времени Δt (данный параметр может быть определен на основе теории планирования эксперимента). То есть за время эксперимента Δt должно быть проведено $X_0 N$ измерений. Таким образом, среднюю интенсивность поступления запросов можно принять равной

$$\lambda = \frac{NX_0}{\Delta t} \quad (4)$$

Вероятность нахождения системы в k -м состоянии вычисляется по формуле

$$P_k = \frac{N!}{(N-k)!} \rho^k P_0, \quad (5)$$

где $\rho = \frac{\lambda}{\mu}$ — коэффициент загрузки системы; P_0 — вероятность простоя СМО ввиду отсутствия требования в ней.

Постоянная вероятность P_0 определяется из условия нормировки [7] $\sum_{k=0}^{\infty} P_k = 1$:

$$P_0 = \frac{1}{\sum_{k=0}^N \frac{N! \rho^k}{(N-k)!}} \quad (6)$$

Среднее число требований, находящихся в системе (на обслуживании и в очереди), определяется по формуле

$$L_S = \sum_{k=1}^N k P_k \quad (7)$$

Коэффициент использования объектов (источников требований)

$$\alpha_2 = 1 - \frac{L_S}{N} \quad (8)$$

Среднее время ожидания обслуживания в очереди требованием

$$t_{\text{оч.ср}} = \frac{1 - \alpha_2}{\lambda \alpha_2} - \frac{1}{\mu} \quad (9)$$

Таким образом, среднее время ожидания результатов измерения

$$t_{\text{ож.ср}} = \frac{1 - \alpha_2}{\lambda \alpha_2} - \frac{1}{\mu} + t_{\text{обсл}} + t_{\text{св}} \quad (10)$$

На рис. 5, 6 приведены семейства зависимостей $t_{\text{ож.ср}} = f(N)$ на примере следующих исходных данных: $\Delta t = 3600$ с; $X_0 = 50$; $t_{\text{св}} = 0,5$ с.

На рис. 7 приведены зависимости $t_{\text{ож.ср}} = f(N)$ для АПК УД "Электроника" [2] и "Тракт 'УЗЧ'" на отрезке времени функционирования $\Delta t = 3600$ с, $t_{\text{св}} = 0,5$ с.

Анализ данных зависимостей показывает, что АПК УД "Электроника" имеет большую нагрузочную способность (большое число одновременно работающих пользователей), чем АПК УД "Тракт 'УЗЧ'". Это следует обязательно учитывать при использовании АПК УД в учебном процессе.

Фактическое число измерений, осуществляемое каждым пользователем в течение рассматриваемого промежутка времени Δt

$$X = \frac{\Delta t}{t_{\text{ож.ср}}} \quad (11)$$

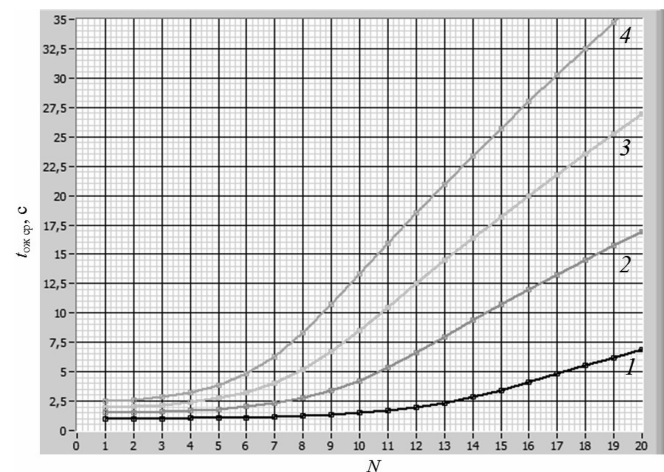


Рис. 5. Зависимость $t_{\text{ож.ср}} = f(N)$ при различных значениях $t_{\text{обсл}}$: 1 — $t_{\text{обсл}} = 0,5$ с; 2 — $t_{\text{обсл}} = 1$ с; 3 — $t_{\text{обсл}} = 1,5$ с; 4 — $t_{\text{обсл}} = 2$ с

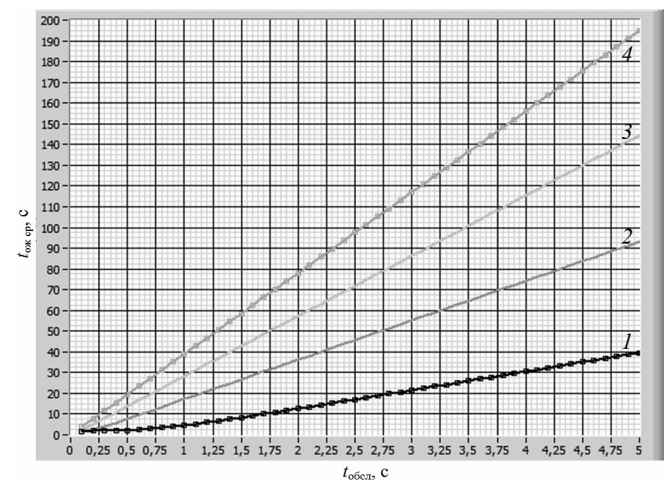


Рис. 6. Зависимость $t_{\text{ож.ср}} = f(t_{\text{обсл}})$ при различных значениях N : 1 — $N = 10$; 2 — $N = 20$; 3 — $N = 30$; 4 — $N = 40$

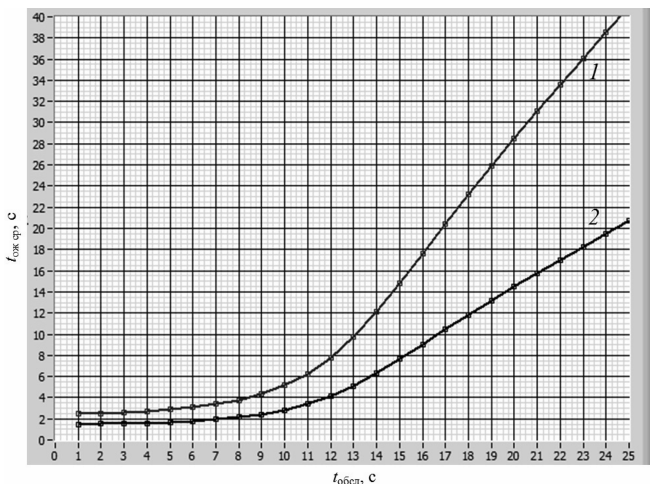


Рис. 7. Зависимости $t_{ож.ср} = f(N)$:
 1 – АПК УД "Тракт УЗЧ" ($X_0 = 15$; $t_{обсл} = 2$ с; $t_{св} = 0,5$ с); 2 – АПК УД "Электроника" ($X_0 = 30$; $t_{обсл} = 1$ с; $t_{св} = 0,5$ с)

Очевидно, что для полного и успешного проведения эксперимента необходимо, чтобы $X \geq X_0$. Таким образом, должно выполняться неравенство

$$\frac{\Delta t}{t_{ож.ср}} \geq X_0. \quad (12)$$

Из неравенства (12) следует, что

$$\frac{\Delta t}{X_0} \geq t_{ож.ср}. \quad (13)$$

Выражение (13) является критерием полного и успешного проведения эксперимента.

Используя выражения (10) и (13), получаем неравенство

$$\frac{1 - \alpha_2}{\lambda \alpha_2} - \frac{1}{\mu} + t_{рек} + t_{изм} + t_{св} \leq \frac{\Delta t}{X_0}. \quad (14)$$

Данное неравенство должно быть использовано на этапе проектирования для определения соответствия техническому заданию. Анализ выражения (14) позволяет определить область допустимых значений $t_{рек}$, $t_{изм}$, $t_{св}$ и их оптимальное соотношение, на основе которых могут быть синтезированы функциональные узлы АПК УД в соответствии с требованиями технического задания.

Математическая модель также позволяет провести повторный расчет нагрузочной способности АПК УД при изменении условий эксплуатации (новые группы пользователей, изменение временного интервала функционирования), что очень важно при использовании АПК УД в учебном процессе.

На рис. 8 приведена блок-схема алгоритма проектирования АПК УД на основе предлагаемой математической модели.

Существуют следующие общеизвестные подходы, применяемые в практике и направленные

на увеличение пропускной способности АПК УД:

- уменьшение $t_{упр}$ может быть достигнуто использованием грамотной схемотехнической реализации узлов интерфейса и исполнительных устройств, современной элементной базы, позволяющей в течение миллисекунд реконфигурировать объект исследования, в то же время на $t_{п.пр}$ повлиять достаточно сложно (так как оно зависит от объекта исследования);
- уменьшение $t_{св}$ может быть достигнуто увеличением пропускной способности каналов связи, а также минимизацией объема передаваемых данных за счет применения специализированных алгоритмов цифровой обработки (имеет смысл при соизмеримости $t_{обсл}$ и $t_{св}$);
- уменьшение $t_{изм}$ достигается применением специализированных алгоритмов измерения и последующей постобработкой результатов измерения.

При достижении практического предела пропускной способности АПК УД применяется корректировка ТЗ:

- уменьшение N , достигается временным разделением групп удаленных пользователей;
- уменьшение X_0 , может быть достигнуто применением скорректированных методик проведения эксперимента (за одно обращение к серверу измеряется максимально необходимое число данных);
- увеличение Δt — увеличение временного интервала работы АПК УД (например круглосуточное функционирование).

Для определения эффективности мероприятий, направленных на увеличение нагрузочной способности АПК УД, определим относительные коэффициенты чувствительности [9]:

$$S_{t_{обсл}}^{t_{ож.ср}} = \frac{\partial t_{ож.ср}}{\partial t_{обсл}} \cdot \frac{t_{обсл}}{t_{ож.ср}};$$

$$S_{t_{св}}^{t_{ож.ср}} = \frac{\partial t_{ож.ср}}{\partial t_{св}} \cdot \frac{t_{св}}{t_{ож.ср}};$$

$$S_N^{t_{ож.ср}} = \frac{\partial t_{ож.ср}}{\partial N} \cdot \frac{N}{t_{ож.ср}};$$

$$S_{X_0}^{t_{ож.ср}} = \frac{\partial t_{ож.ср}}{\partial X_0} \cdot \frac{X_0}{t_{ож.ср}};$$

$$S_{\Delta t}^{t_{ож.ср}} = \frac{\partial t_{ож.ср}}{\partial \Delta t} \cdot \frac{\Delta t}{t_{ож.ср}}.$$

На рис. 9 (см. четвертую сторону обложки) приведены результаты расчета $S_{t_{обсл}}^{t_{ож.ср}}$, $S_N^{t_{ож.ср}}$,

$S_{X_0}^{t_{ож.ср}}$, $S_{\Delta t}^{t_{ож.ср}}$, $S_{t_{св}}^{t_{ож.ср}}$ методом численного дифференцирования на примере АПК УД "Электроника"

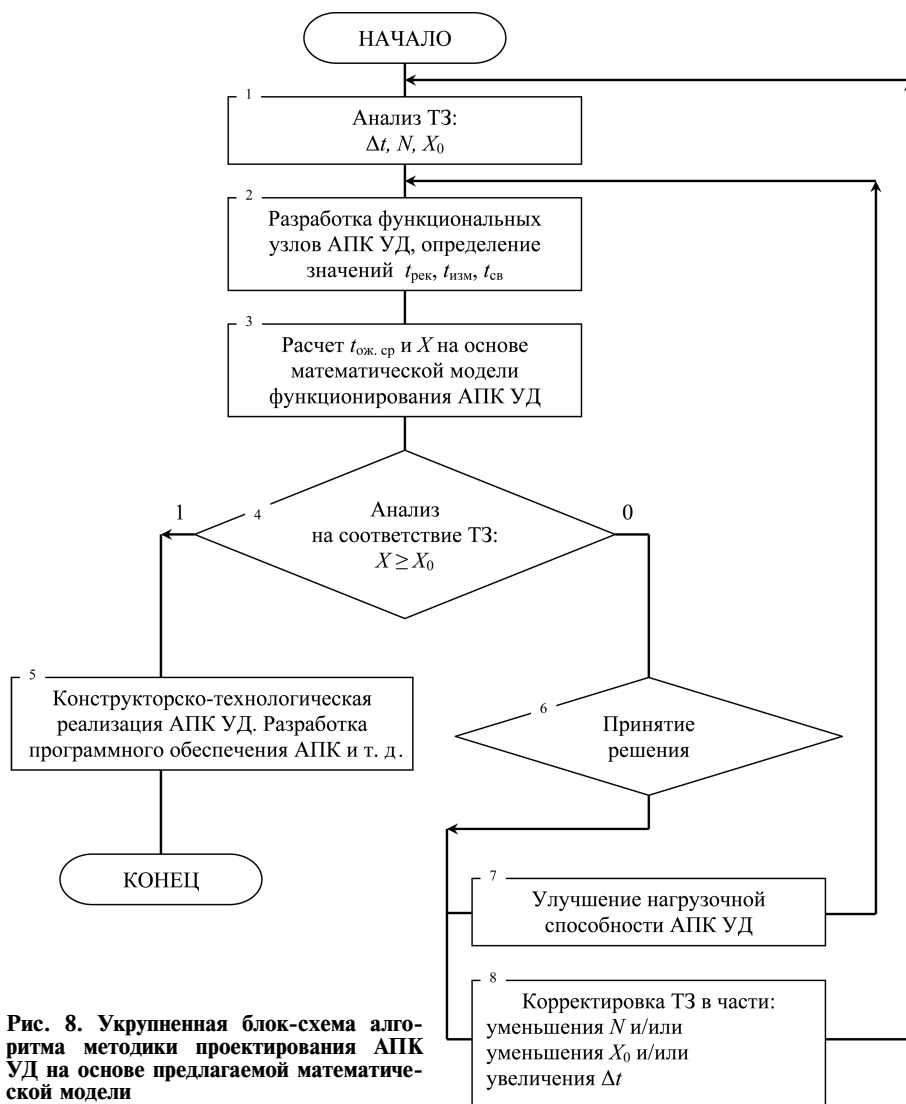


Рис. 8. Укрупненная блок-схема алгоритма методики проектирования АПК УД на основе предлагаемой математической модели

[2] ($t_{\text{обсл}} = 1$ с; $t_{\text{св}} = 0,5$ с; $X_0 = 30$) и для следующих условий эксплуатации: $\Delta t = 3600$ с; $N = 20$.

Таким образом, результаты расчета $S_i^{t_{\text{ож.ср}}}$ являются математическим подтверждением того, что наиболее предпочтительным и действенным способом увеличения нагрузочной способности АПК УД является уменьшение $t_{\text{обсл}}$ и N . Но уменьшение N является крайней мерой и не всегда целесообразно, следовательно, основной задачей при разработке АПК УД должна быть минимизация $t_{\text{обсл}}$.

Довольно часто на этапе разработки измерительного программного обеспечения возникает необходимость повышения точности измерений методом усреднений. Реализация данного метода может осуществляться следующими способами.

Способ 1. Необходимое число измерений k осуществляется на измерительном ПЭВМ-сервере, результаты усредняются, среднее значение отправляется ПЭВМ-клиенту. При использовании

данного способа значение $t_{\text{изм}}$ увеличивается в k раз, в связи с необходимостью проведения k измерений.

Способ 2. Усреднение осуществляется на ПЭВМ-клиенте, т. е. каждое последующее измерение усредняется с предыдущими измерениями. При использовании данного способа значение $t_{\text{изм}}$ остается неизменным, но увеличивается число необходимых обращений X_0 к измерительному ПЭВМ-серверу в k раз.

Анализ относительных коэффициентов чувствительности $S_i^{t_{\text{ож.ср}}}$ показывает, что величина $t_{\text{ож.ср}}$ менее чувствительна к изменению X_0 , чем к изменению $t_{\text{изм}}$, следовательно, более предпочтительным для реализации является второй способ. Таким образом, расчет относительных коэффициентов чувствительности $S_i^{t_{\text{ож.ср}}}$ на этапе разработки измерительного программного обеспечения позволяет выявить и использовать наиболее оптимальный алгоритм измерения.

Следует также отметить, что ввиду сложности функциональной зависимости $t_{\text{ож.ср}} = f(t_{\text{обсл}}, N, X_0, \Delta t, t_{\text{св}})$ [см. выражения (3)–(10)] значения относительных коэффициентов чувствительности $S_i^{t_{\text{ож.ср}}}$ при различных сочетаниях аргументов будут отличаться, поэтому для каждого АПК УД и конкретных условий эксплуатации необходим свой вариант расчета $S_i^{t_{\text{ож.ср}}}$ на основе математической модели функционирования.

Исходя из результатов приведенных выше исследований можно сделать следующий вывод. Проектирование и создание АПК УД целесообразно вести с использованием предварительных расчетов нагрузочной способности разрабатываемого комплекса на основе характеристик его функциональных узлов, что позволит получить АПК УД с максимальной нагрузочной способностью. Предлагаемая методика была использована при проектировании АПК УД "Электроника", "Схемотехника аналоговых электронных устройств" и "Тракт усиления звуковой частоты",

функционирующих во многопользовательском режиме [3]. Например, решение задачи выбора оптимального метода измерений с использованием анализа относительных коэффициентов чувствительности $S_i^{t_{ож.ср}}$ для АПК УД "Схемотехника аналоговых электронных устройств" позволило снизить погрешность измерений в 5 раз, при увеличении $t_{ож.ср}$ всего на 30 %. Применение представленной выше методики проектирования на основе математической модели позволило снизить $t_{обсл}$ для АПК УД "Электроника" [2] с 5 с до 1...1,5 с за счет применения специализированных алгоритмов и методов измерения, а также последующей постобработки информации, что существенно уменьшило $t_{ож.ср}$ и увеличило нагрузочную способность АПК УД в целом (см. рис. 5, 9).

Список литературы

1. **Отраслевой** стандарт 9.2—98. Учебная техника для образовательных учреждений. Системы автоматизированного лабо-

раторного практикума. Основные положения. — М.: Изд. Росстандарт, 1998.

2. **Исследование** параметров и характеристик полупроводниковых приборов с применением интернет-технологий: учеб. пособие / А. С. Глинченко, Н. М. Егоров, В. А. Комаров, А. В. Сарафанов. М.: ДМК-Пресс, 2008. — 352 с.

3. **Сетевая** лаборатория Сибирского федерального округа [сайт]. URL: <http://www.alpsib.ru>

4. **Подлесный С. А.** Концепция типовых решений при построении автоматизированных лабораторных практикумов с удаленным доступом (на примере дисциплин радиотехнических специальностей) / С. А. Подлесный, А. В. Сарафанов, В. А. Комаров. — Красноярск: ИПЦ КГТУ, 2005. — 40 с.

5. **Евдокимов Ю. К.** LabVIEW для радиоинженера: от виртуальной модели до реального прибора. Практическое руководство для работы в программной среде LabVIEW / Ю. К. Евдокимов, В. Р. Линдваль, Г. И. Шербаков — М.: ДМК-Пресс, 2007. — 400 с.

6. **Клейнрок Л.** Теория массового обслуживания / Л. Клейнрок; Пер. с англ. И. И. Грушко; ред. В. И. Нейман. — М.: Машиностроение, 1979. — 432 с.

7. **Гнеденко Б. В.** Введение в теорию массового обслуживания, 3-е изд., испр. и доп. / Б. В. Гнеденко, И. Н. Коваленко. — М.: КомКнига, 2005. — 400 с.

8. **Кудрявцев Е. М.** GPSS World. Основы имитационного моделирования различных систем / Е. М. Кудрявцев — М.: ДМК-Пресс, 2004. — 320 с.

9. **Кофанов Ю. Н.** Теоретические основы конструирования, технологии и надежности радиоэлектронных средств: учеб. для вузов / Ю. Н. Кофанов. — М.: Радио и связь, 1991. — 360 с.

УДК 004.81:159.953.52

И. П. Норенков, д-р техн. наук,
проф., зав. каф.,
e-mail: norenkov@rk6.bmstu.ru,
Н. К. Соколов, инженер,
МГТУ им. Н. Э. Баумана

Синтез индивидуальных маршрутов обучения в онтологических обучающих системах

В процессе развития баз учебных материалов, включая объединение баз, созданных разными преподавательскими коллективами, число разделяемых единиц контента, определяющих одни и те же понятия, будет неуклонно увеличиваться. Этот процесс ведет к резкому росту числа возможных маршрутов обучения при одних и тех же целевых концептах. В статье предложены модель и метод построения множества маршрутов обучения, что является необходимым для решения задачи выбора оптимальных траекторий обучения в автоматизированных обучающих системах.

Ключевые слова: автоматизированная система обучения, технология разделяемых единиц контента, онтология, маршрут обучения.

Введение

К настоящему времени создано большое число автоматизированных обучающих систем (АОС) как отечественных (Competentum, МАГИСТР 2008, СДТ REDCLASS, HyperMethod, Прометей, КАДИС и др.), так и зарубежных WebCT, Learning Space 5.0, Microsoft Class Server 3.0, Raptivity и др.). Для расширения перечня и качества услуг, предоставляемых обучаемым, необходимо развитие АОС по ряду направлений.

Одним из направлений является пополнение баз образовательных ресурсов (ОР), в том числе путем интеграции баз из разных систем. Для решения проблемы интеграции ОР разработан ряд технологий и описывающих их стандартов. К ним относятся концепции LOM (Learning Object Metadata) и семантического Веба, стандарты RDF/RDFS, протоколы RSS и т. п. На их основе стало возможным создание организаций и объединений, аккумулирующих образовательные ресурсы и обеспечивающих к ним открытый доступ. Таковыми консорциум OCW (Open Course Ware) [1], появившийся по инициативе ЮНЕСКО, интегрированная система доступа к образовательным ресурсам OER (Open Educational Resources Commons) [2], система российских федеральных образовательных порталов [3]. В систему OCW в настоящее время входит более 100 университетов. В системе

OER собрано несколько тысяч электронных учебных пособий. В систему российских образовательных порталов включены порталы "Российское образование", "Естественно-научный", "Экономика, социология, менеджмент", "Инженерное образование", "Информационно-коммуникационные технологии в образовании" и др.

Наряду с интеграцией ОР, в современных АОС актуальной является проблема индивидуализации обучения, заключающаяся в создании и обеспечении обучаемых учебными пособиями, соответствующими их индивидуальным запросам и особенностям предварительной подготовки. Решение этой проблемы представляется возможным при условии мелкомодульного построения баз образовательных ресурсов. Эти условия соблюдены в модели обучения SCORM (Shareable content object reference model) [4] и в развиваемой в МГТУ им. Н. Э. Баумана технологии разделяемых единиц контента (ТРЕК) на основе онтологического подхода [5]. Обе технологии ориентированы на оперативную компиляцию новых учебных пособий из предварительно разработанных элементов, называемых разделяемыми единицами контента (РЕК) или модулями.

В модели SCORM подразумевается, что РЕК являются автономными фрагментами учебных пособий и, следовательно, не связанными с другими РЕК системой гиперссылок. Тем самым, системы, реализующие SCORM, не рассчитаны на выполнение таких функций, как автоматизация, во-первых, преобразования текстов пособий в гипертекст и, во-вторых, упорядочения РЕК в составе пособия в соответствии с отношениями "определяющее понятие — определяемое понятие".

Выполнение таких функций в альтернативной технологии стало возможным, благодаря введению в модель информационно-образовательной среды онтологий рассматриваемых предметных областей. Онтологическая ТРЕК реализована в системе "База и генератор образовательных ресурсов" (БиГОР) [6].

Наличие онтологий позволяет ставить и решать ряд дополнительных задач в части индивидуализации обучения, а именно задач построения оптимальных маршрутов обучения конкретных пользователей и выражающих эти маршруты учебных пособий.

Одна из таких задач рассмотрена в работе [7], в которой оптимизируется размещение контролируемых РЕК в последовательности основных РЕК.

В данной статье предлагается метод синтеза маршрутов обучения в расширенных базах учебных материалов (БУМ), в которых возможно наличие нескольких модулей, поясняющих одно и то же понятие. Такие модули могут различаться теми или иными параметрами метаданных, на-

пример, именами авторов модулей, датами создания РЕК, сложностью изложения материала, степенью связности с другими модулями и т. п. Чем богаче БУМ, тем разветвленнее семантическая сеть ОР, тем больше мощность множества возможных траекторий обучения. Построение маршрутов обучения и синтез поддерживающих учебных пособий в этих условиях становится не тривиальной задачей.

Модель образовательного пространства в онтологических базах образовательных ресурсов

При описании модели образовательного пространства и метода синтеза учебных пособий будем использовать следующие определения.

Целевой концепт — концепт (понятие), который должен быть изучен с помощью синтезируемого пособия.

Исходный концепт — концепт, уже изученный пользователем.

Семантическая сеть концептов — сеть, вершинам которой соответствуют концепты, а дугам — отношения между ними. Так, если X — определяющее понятие, а Y — определяемое, то дуга направлена от вершины X к вершине Y .

Семантическая сеть модулей — сеть, вершинам которой соответствуют модули. Дуга $X \rightarrow Y$ имеет место, если некоторое понятие определено в модуле X и используется в модуле Y . Понятие, определяемое в модуле, называется *выходным понятием* этого модуля, а понятия, используемые в модуле для пояснения выходного понятия, называются *входными понятиями* этого модуля.

Переменная концепта (*k-переменная*) — логическая переменная, принимающая значение *true*, если концепт входит в синтезируемый маршрут обучения, иначе *k-переменная* равна *false*. Аналогично определяется *m-переменная* — логическая переменная модуля.

Модель образовательного пространства представляет собой И-ИЛИ-граф, образованный объединением семантических сетей концептов и модулей.

Задача синтеза маршрута обучения формулируется следующим образом.

Задано множество целевых концептов $T_{\text{цел}}$, известно также исходное множество концептов $T_{\text{исх}}$. Нужно построить подграф И-ИЛИ-графа, содержащий пути от исходных концептов (или от концептов, не имеющих входов) к каждому целевому концепту. Очевидно, что в общем случае задача имеет множество решений. Среди них нужно выбрать одно, оптимальное по заданному критерию.

Концепты и модули, вошедшие в искомые пути, должны поясняться в пособии, т. е. их переменные должны иметь значения *true*. Пояснения концептов содержатся в модулях. Для того чтобы был изучен концепт k_j , нужно в маршруте обуче-

Алгоритм синтеза маршрутов обучения

Синтез маршрутов обучения начинается с выбора множества целевых и исходных концептов. Этот выбор выполняет преподаватель, ориентируясь на индивидуальные особенности обучаемого и/или на учебную программу курса. При самообразовании задание цели исходит от самого обучаемого, которому нужно пособие для ее достижения.

Далее выполняется циклический вычислительный процесс выбора модулей на основе формулы (1) для каждого целевого концепта с последовательной заменой в правых частях формул k -переменных на m -переменные и новые k -переменные. Этот процесс продолжается до тех пор, пока в правых частях не будет переменных других концептов, кроме исходных.

При формировании маршрутов используются следующие очевидные правила:

$$k * k = k \text{ и } a * m + m = m, \quad (2)$$

где $*$ — знак конъюнкции, $+$ — знак дизъюнкции.

По окончании циклического процесса и раскрытия всех скобок имеем дизъюнктивную нормальную форму, в которой каждый дизъюнкт представляет один из альтернативных маршрутов обучения. Среди них нужно выбрать маршрут, оптимальный по одному из критериев, которыми могут быть те или иные функции метаданных модулей, вошедших в маршрут, например, сложность освоения, современность материала, степень взаимосвязи с модулями других предметных областей и т. п.

Рассмотрим пример синтеза маршрута обучения, иллюстрируемый графом рисунка. Пусть в качестве целевых концептов заданы $k1, k2, k3$, исходные концепты на рисунке не показаны. Обозначим функцию, характеризующую множество маршрутов обучения, через \mathbf{M} . Каждый дизъюнктивный член этой функции отображает один из маршрутов обучения.

Исполнение алгоритма начинается с описания \mathbf{M} в виде конъюнкции целевых концептов:

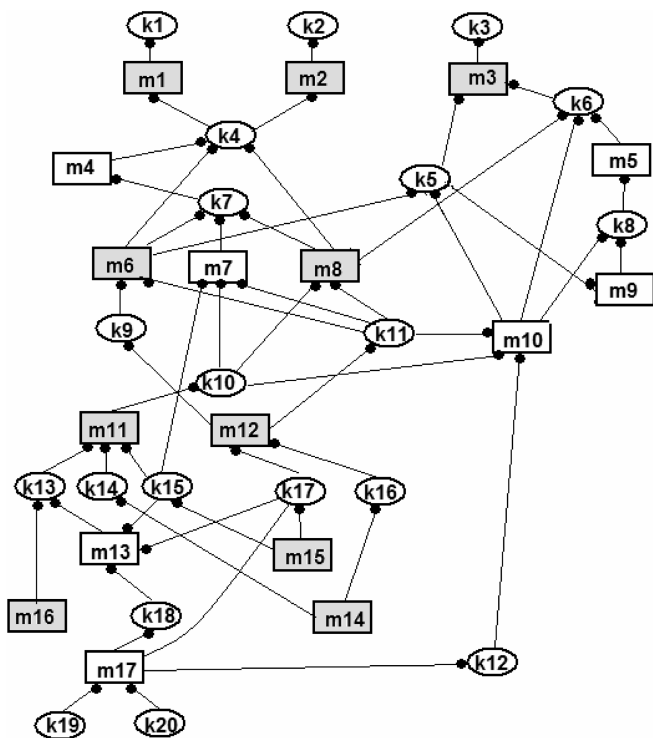
$$\mathbf{M} = k1 * k2 * k3.$$

Далее следует применение правила (1):

$$\begin{aligned} \mathbf{M} &= m1 * k4 * m2 * k4 * m3 * k5 * k6 = \\ &= m1 * k4 * m2 * m3 * k5 * k6, \end{aligned}$$

поскольку $k4 * k4 = k4$. Продолжая замены переменных, получаем

$$\begin{aligned} \mathbf{M} &= m1 * (m4 * k7 + m6 * k9 * k11 + \\ &+ m8 * k10 * k11) * m2 * m3 * (m6 * k9 * k11 + \\ &+ m10 * k10 * k11 * k12) * (m8 * k10 * k11 + \\ &+ m10 * k10 * k11 * k12 + m5 * k8) = \\ &= m1 * (m4 * (m6 * k9 * k11 + m7 * k10 * k11 * k15 + \\ &+ m8 * k10 * k11) + m6 * m12 * k16 * k17 + \\ &+ m8 * m11 * k13 * k14 * k15 * m12 * k16 * k17) * m2 * \\ & * m3 * (m6 * m12 * k16 * k17 + \end{aligned}$$



ния иметь хотя бы один модуль, поясняющий этот концепт. Другими словами, чтобы k_i -переменная имела значение *true*, требуется принятие значения *true* хотя бы одной из m -переменных тех модулей, в которых определяется k_j . Вместе с тем, чтобы модуль с пояснением концепта k_i (пусть это модуль m_{ij}) был понятен обучаемому, обучаемый должен знать или предварительно изучить входные для модуля m_{ij} концепты. Переменные этих концептов должны иметь значение *true*.

Следовательно, для синтезируемого маршрута должно соблюдаться правило (для краткости здесь и далее в формулах вместо k_i -переменных и m_j -переменных записываются символы k_i и m_j соответственно):

$$k_i = \sum_{j \in \mathbf{J}} m_{ij} \prod_{r \in \mathbf{R}_j} k_{jr} = \text{true}, \quad (1)$$

где k_i — целевой концепт; Σ и Π — знаки логических сложения и умножения; \mathbf{J} — множество номеров модулей, в которых определен концепт k_i ; \mathbf{R}_j — множество номеров входных для модуля m_j концептов.

Из выражения (1) следует, что вершинам концептов соответствует связка дизъюнкции, и поэтому такие вершины названы вершинами типа ИЛИ, а вершинам модулей соответствует связка конъюнкции, и потому эти вершины — вершины типа И.

Пример модели приведен на рисунке, где вершины концептов показаны в виде овалов, а вершины модулей — в виде прямоугольников.

$$\begin{aligned}
& + m_{10} * m_{11} * k_{13} * k_{14} * k_{15} * m_{17} * k_{19} * k_{20} * \\
& * (m_8 * m_{11} * k_{13} * k_{14} * k_{15} * m_{12} * k_{16} * k_{17} + \\
& + m_{10} * m_{11} * k_{13} * k_{14} * k_{15} * m_{12} * k_{16} * k_{17} * \\
& * m_{17} * k_{19} * k_{20} + m_5 * (m_9 * k_5 + \\
& + m_{10} * k_{10} * k_{11} * k_{12}))
\end{aligned}$$

После раскрытия скобок с учетом правил (2) получаем следующий список из четырех альтернативных маршрутов:

$$\begin{aligned}
& m_1 * m_2 * m_3 * m_4 * m_7 * m_{10} * m_{11} * m_{12} * \\
& * m_{14} * m_{15} * m_{16} * m_{17} * k_{19} * k_{20}; \\
& m_1 * m_2 * m_3 * m_6 * m_8 * m_{11} * m_{12} * m_{14} * m_{15} * m_{16}; \\
& m_1 * m_2 * m_3 * m_6 * m_{10} * m_{11} * m_{12} * m_{14} * m_{15} * \\
& * m_{16} * m_{17} * k_{19} * k_{20}; \\
& m_1 * m_2 * m_3 * m_8 * m_{10} * m_{11} * m_{12} * m_{14} * m_{15} * \\
& * m_{16} * m_{17} * k_{19} * k_{20}.
\end{aligned}$$

Выбор одного из них проводится в соответствии с некоторым критерием оптимальности. Например, если пользователя интересует наиболее короткое учебное пособие и длина пособия измеряется числом входящих в него модулей, то будет выбран маршрут

$$m_1 * m_2 * m_3 * m_6 * m_8 * m_{11} * m_{12} * m_{14} * m_{15} * m_{16},$$

модули которого на рисунке показаны слегка затененными.

Заключение

Предложенный алгоритм позволяет генерировать варианты маршрутов обучения, состоящие из подмножества небольших модулей (разделяемых единиц контента) учебного материала, и входит составной частью в алгоритмы полного или частичного перебора маршрутов для выбора наилучшего из них. В системах типа БиГОР автоматически создается учебное пособие, реализующее выбранный маршрут.

Список литературы

1. **Open** Course Ware — <http://www.ocwconsortium.org/>.
2. **Open** Educational Resources Commons. — <http://www.oercommons.org/>.
3. **Российское** образование. Федеральный портал — http://www.edu.ru/db/portal/sites/portal_page.htm
4. **SCORM**. Shareable Content Object Reference Model. 2d Edition. — Advanced Distributed Learning, 2004.
5. **Норенков И. П., Зимин А. М.** Информационные технологии в образовании. М.: Изд-во МГТУ им. Н. Э. Баумана, 2004.
6. **Норенков И. П., Уваров М. Ю.** База и генератор образовательных ресурсов // Информационные технологии, 2005. № 9. С. 60—65.
7. **Доррер Г. А., Попов А. А., Рудакова Г. М., Сысенко К. В.** Оптимальная группировка разделяемых единиц контента в учебные модули на базе системы БиГОР // Информационные технологии. 2008. № 8. С. 70—73.

УДК 519.673

А. Е. Рыбаков, аспирант,
 Марийский государственный технический университет,
 e-mail: alexey.rybakov@cpslabs.net

Моделирование структуры учебных курсов и подсистемы навигации для систем управления обучением, поддерживающих стандарт SCORM

Рассматривается модель подсистемы навигации в электронных образовательных ресурсах. Предлагается структуризация учебного материала и подход к моделированию навигации, которые согласуются с международными спецификациями электронного обучения SCORM и IMS, дополняя их конкретными адаптивными алгоритмами для навигации по учебным курсам на основе использования модели интегральной оценки знаний обучаемого.

Ключевые слова: электронное обучение, структуризация, учебные материалы, навигация, курсы, обучаемый, SCORM, IMS.

Введение

Современный этап развития систем управления обучением связан с использованием тех возможностей, которые предоставляются глобальными компьютерными сетями и технологическими стандартами, разработанными в области дистанционного образования [1].

Среди всех продуктов стандартизации электронного образования SCORM и IMS [2, 3] получили самое широкое признание. Эти специфика-

ции используются при создании систем обучения, опирающихся на ресурсы Интернета. Эталонная модель SCORM состоит из четырех частей: введения и обзорной части; описания модели интеграции содержания; описания среды выполнения программ; описания управления навигацией по контенту и упорядочения содержания. Последняя часть стандарта описывает, как должны быть организованы навигация и предоставление компонентов учебного материала в зависимости от действий учащегося.

Требования SCORM к содержанию и организации навигации по учебным материалам позволяют упорядочивать его в соответствии с индивидуальными особенностями обучаемого. Опыт использования различных компьютерных средств обучения подтверждает взаимосвязь факторов, определяющих эффективность познавательной деятельности, с наличием средств их адаптации к личности обучаемого [4]. Поэтому одной из важных задач при создании системы управления обучением является возможность адаптации системы к индивидуальным особенностям обучаемого при организации контроля знаний и построения индивидуальной образовательной траектории.

В статье рассматриваются модель реализации последовательности индивидуального изучения учебных элементов и построение динамических связей между ними, позволяющие организовать рациональную в дидактическом плане навигацию в электронном курсе. Также предлагается метод интегральной оценки знаний для модели обучаемого.

Модель структуризации учебного материала

Предлагаемая модель содержания адекватна концепциям SCORM [2]. В ней планируемый для изучения учебный материал разбивают на отдельные *учебные элементы* (УЭ). Под УЭ понимают объекты, явления, методы деятельности, темы курса, отобранные из соответствующей науки и внесенные в программу учебной дисциплины или раздела учебной дисциплины. Совокупность УЭ представляют в виде структурной схемы, которую называют *графом содержания* (ГС) учебного материала. Узлами (вершинами) графа являются УЭ, ребрами — связи между ними. Понятие УЭ и представление структуры учебного материала в виде ГС эквивалентны соответственно понятию совместно используемых объектов содержания (Sharable Content Objects — SCOs) и их агрегациям в SCORM.

Параллельно с построением ГС составляют спецификацию (таблицу) УЭ, в которую вносят наименования УЭ. После структурирования и отбора содержания учебного материала для каждого УЭ формулируют требования по уровню усвоения, которые включают в спецификацию УЭ. Уровень освоения определяется результатами выполнения тестовых заданий и представляет собой рациональную величину, лежащую в пределах $\{0...1\}$.

По этому показателю заполняют два столбца таблицы УЭ. В первом указывают начальное значение показателя (требуемый уровень освоения учебных элементов, необходимых для изучения данного УЭ), во втором — конечное значение показателя (требуемый минимальный уровень после

обучения). Заметим, что первые версии SCORM (например, SCORM 1.2) не содержали подобных элементов дидактического целеполагания. В последней версии (SCORM 2004) этот пробел был частично восполнен включением в характеристики SCOs дидактических целей, основанных на таксономии уровня знаний Блума [5].

Совокупность ГС и спецификации УЭ являются моделью содержания учебного материала электронного образовательного ресурса. В предложенной модели содержания используются особенности технологии разработки электронных энциклопедий [6]. При создании учебного курса предварительно разрабатывается онтология предметной области, т. е. определяется система понятий (сущностей) и отношений между ними. Таким образом, каждый узел ГС может дополнительно содержать совокупность ключевых понятий курса (I, O), где I — множество входных (предпосылочных) понятий, а O — множество выходных понятий. Это могут быть элементарные единицы знания, термины, разделы.

Выходным понятием является понятие, определяемое в данном УЭ, а входным — понятие, используемое в УЭ для определения выходного.

Схематично модель учебного курса представлена на рис. 1. На рисунке изображены учебные элементы разного уровня иерархии (окружности 1–7), стрелками даны иерархические связи между ними. Каждый УЭ имеет множество входных и выходных понятий, они изображены пунктирными прямоугольниками.

Переходы между учебными элементами могут формироваться динамически в зависимости от уровня знаний и умений обучаемого. Такая структуризация позволяет строить адаптивную последовательность подачи учебного материала. Возможность перехода определяется уровнем освоения

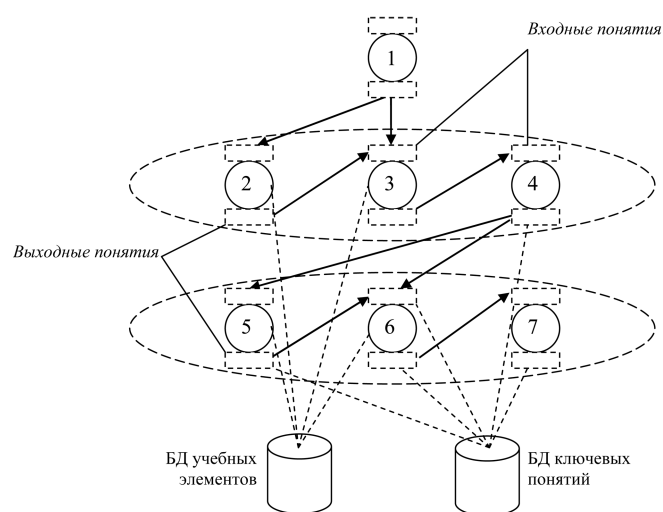


Рис. 1. Модель учебного курса

входных ключевых понятий: если $I_j \subset K_{\text{обуч}}$, где $K_{\text{обуч}}$ — множество ключевых понятий, освоенных обучаемым, то переход к j -му УЭ возможен.

На рис. 1 мы видим, что после изучения УЭ 1, можно перейти как к учебному элементу 2, так и к 3. В данном случае переход к УЭ 2 предполагает меньший уровень знаний обучаемого, т. е. множество входных (предпосылочных) понятий УЭ 2 меньше множества входных понятий УЭ 3: $I_2 \subset I_3$. Если $I_3 \subset K_{\text{обуч}}$, то возможен переход сразу к УЭ 3. Если $I_2 \subset K_{\text{обуч}}$, то обучаемый изучает УЭ 2. После успешного изучения во множество понятий обучаемого добавляются понятия выходного множества УЭ 2: $K_{\text{обуч}} = K_{\text{обуч}} \cup O_2$. Теперь $I_3 \subset K_{\text{обуч}}$, и обучаемый может перейти к УЭ 3.

Таким образом, преимуществом предложенной модели структуризации является возможность динамического создания связей между УЭ в ГС для автоматизированного построения индивидуальной последовательности подачи учебного материала.

Модель последовательности освоения материала на базе орграфа

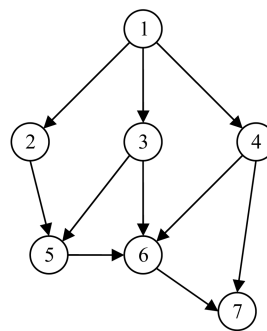
Рассмотрим математическую модель для построения оптимальной, с точки зрения времени изучения материала [7], траектории в ГС. Пусть G — орграф, описывающий ГС, $VG = \{v_i\}$ — множество вершин и $VE = \{e_i\}$ — множество дуг этого орграфа. Орграф G не должен содержать:

- петли, т. е. дуги (v_i, v_i) ;
- циклы, т. е. такие маршруты $v_1, e_1, v_2, e_2, \dots, e_k, v_{k+1}$, в которых $v_1 = v_{k+1}$ (где v_i, e_i — соответственно номера вершин и дуг, входящих в маршрут);
- несвязные вершины или подграфы.

Для определения наличия несвязных вершин (подграфов) орграф рассматривается как неориентированный граф. Неориентированный граф является связным тогда и только тогда, когда для произвольной фиксированной вершины v существует маршрут (v, \dots, u) , где u — любая другая вершина графа.

Использование графов дает возможность решить оптимизационную задачу, связанную с определением целесообразной последовательности изучения тем как в рамках курса, так и в нескольких взаимосвязанных курсах (дисциплинах). Модель содержания отражает логические связи между УЭ. Соответственно, последовательность освоения тем должна быть такова, чтобы к началу изучения некоторого УЭ все предшествующие ему УЭ были уже изучены.

Эта задача сводится к классу задач раскраски вершин орграфа. Рассмотрим произвольную



а)

	1	2	3	4	5	6	7
1		1	1	1			
2					1		
3					1	1	
4						1	1
5						1	
6							1
7							

б)

Рис. 2. Граф содержания:

а — граф содержания; б — матрица смежности

функцию вида $f: VG = \{1, 2, \dots, k\}$, где k — число различных красок. В данном случае решением задачи определения последовательности изучения тем является такая раскраска орграфа G , при которой для любого маршрута $v_1, e_1, v_2, e_2, \dots, e_k, v_{k+1}$, вершины которого раскрашены цветами l_1, l_2, \dots, l_k , верно утверждение $l_i < l_j$, если $i < j$, где l — длина пути, и в нашем случае она равна числу переходов.

Чтобы решить задачу раскраски, необходимо знать кратчайший путь из одной вершины в другую. Рассмотрим решение данной задачи (рис. 2) для пары вершин $v_{\text{нач}}, v_{\text{кон}}$ (вершины 1—7). Множество дуг VE представим в виде матрицы смежности $S = [s_{ij}]$ порядка n , где n — число вершин графа и

$$s_{ij} = \begin{cases} 1, & \text{если вершина } v_i \text{ соединена с } v_j; \\ 0, & \text{в противном случае.} \end{cases}$$

Введем целочисленные переменные x_{ij} , $i = 1 \dots n, j = 1 \dots n$, где

$$x_{ij} = \begin{cases} 1, & \text{если кратчайший путь включает переход из вершины } v_i \text{ в } v_j; \\ 0, & \text{в противном случае.} \end{cases}$$

Целевая функция имеет следующий вид:

$$\sum_{i=1}^n \sum_{j=1}^n s_{ij} x_{ij} \rightarrow \min.$$

В ней вычисляется число переходов между вершинами в кратчайшем пути. Сформулируем ограничения для данной задачи поиска кратчайшего пути.

Первая пара ограничений задает условия для начальной вершины пути $v_{\text{нач}}$. В искомом пути в эту вершину не должно быть входа, но должен быть один выход:

$$\sum_{i=1}^n s_{i \text{ нач}} x_{i \text{ нач}} = 0, \quad \sum_{j=1}^n s_{j \text{ нач}} x_{j \text{ нач}} = 1.$$

	1	2	3	4	5	6	7
1		1	1	1	2	2	2
2					1	2	3
3					1	1	2
4						1	1
5						1	2
6							1
7							

Рис. 3. Матрица кратчайших путей (достижимости)

Вторая пара ограничений задает условия для конечной вершины пути $v_{\text{кон}}$. В нее должен быть один вход, но не должно быть выхода:

$$\sum_{i=1}^n s_{i \text{ кон}} x_{i \text{ кон}} = 1, \quad \sum_{j=1}^n s_{j \text{ кон}} x_{j \text{ кон}} = 0.$$

Для всех остальных вершин (кроме $v_{\text{нач}}$, $v_{\text{кон}}$) устанавливаются ограничения, задающие равенство числа входов и выходов в каждую из них в исходе кратчайшем пути:

$$\sum_{i=1}^n s_{ik} x_{ik} - \sum_{j=1}^n s_{kj} x_{kj} = 0, \\ k = \overline{1, n}, k \neq \text{нач}, k \neq \text{кон}.$$

И последнее, число входов и выходов для каждой вершины не должно быть более одного:

$$\sum_{i=1}^n s_{ik} x_{ik} \leq 1, k = \overline{1, n}; \quad \sum_{j=1}^n s_{kj} x_{kj} \leq 1, k = \overline{1, n}.$$

Найденная таким образом матрица кратчайших путей (рис. 3) определяет таблицу переходов между учебными материалами в графе содержания. Это позволяет строить индивидуальную последовательность освоения учебного курса и достигать заданного уровня компетентности за минимальное число шагов в учебном курсе.

Далее рассмотрим подход для расчета оценки уровня знаний наиболее слабо изученных областей курса. Он позволяет более точно определять текущую оценку знаний обучаемых и на ее основе определять рациональную последовательность подачи учебного материала.

Метод интегральной оценки уровня знаний обучаемого

В состав систем управления учебным процессом входит модель обучаемого [4]. Эта модель пользователя, предназначенная для адаптации процесса обучения. Она содержит информацию о пользователе, выводимую на основе его взаимодействия с системой. Основным параметром модели обучаемого является уровень знаний обучаемого, который вычисляется на основе анализа освоенности учебных элементов.

Точная оценка уровня знаний необходима для определения индивидуальных способностей обучаемого. Она определяет адекватность выбранной образовательной траектории, а значит, и эффективность процесса обучения.

Анализ уровня знаний учебных материалов обучаемых широко используется при реализации адаптивных образовательных сред. Так, в работе Беспалько и Белкина [8] рассматривается несколько уровней осознанности полученных знаний. Эти показатели классифицируют глубину проникновения и качество владения учащимися учебным материалом. Это позволяет четко формулировать дидактические цели при проектировании учебного комплекса и на их основе определять его состав. Дело в том, что часть элементов знания учащийся должен уметь применять при решении задач (для этого необходимы тренажеры, учебные пакеты прикладных программ), а с какими-то элементами ему достаточно лишь познакомиться.

Для определения уровня знаний УЭ в предложенной модели дополнительно учитывается уровень знаний понятий, входящих в рассматриваемую выше модель структуризации учебного материала. По окончании изучения каждого УЭ вычисляется степень освоения его выходных понятий. Для автоматизированного вычисления оценки авторами предложен метод интегральной оценки освоенности УЭ.

Интегральная оценка каждого выходного понятия складывается из его базовой оценки и дополнительной, учитывающей оценки понятий, которые являются входными для данного УЭ.

Базовая оценка выставляется по результатам сеанса контроля УЭ. Придадим i -му понятию вес v_i , отражающий его важность в УЭ. Тогда базовая оценка $U_{\text{баз}}$ степени освоения рассчитывается по результатам тестирования:

$$U_{\text{баз}} = \frac{\sum_{i=1}^n m_i v_i}{\sum_{i=1}^n n_i},$$

где n_i — число вопросов в тесте, содержащих i -е понятие; m_i — число правильных ответов на вопросы, содержащих i -е понятие; n — общее число понятий, содержащихся в контрольных вопросах.

Входные понятия определяют знания, необходимые при изучении данного учебного элемента, а выходные — определяют новые знания обучаемого, представленные в данном учебном элементе. Таким образом, справедливо предположить,

что каждое понятие $O_n \in O_{\text{ВЫХ}}$ зависит (ссылается) от каждого из входных понятий $I_{\text{ВХ}}$.

Для вычисления дополнительной оценки понятий используется ранг понятия R . Так как все понятия учебного курса имеют различную важность, для правильной оценки знаний обучаемого необходимо это учитывать. Пусть каждое понятие имеет свой ранг, определяющий его важность во всем учебном курсе. Если на понятие имеется ссылка, т. е. оно входит во множество входных понятий какого-либо УЭ, его ранг повышается. Таким образом, ранг понятия тем выше, чем чаще оно встречается во входных множествах УЭ.

Для вычисления рангов понятий применяется схема алгоритма ранжирования страниц, предложенного Клейнбергом, который является известным представителем группы методов, использующих информацию о связях между страницами [9]. Этот подход позволяет использовать две различные роли понятий — входной как первоисточника информации и выходной как посредника.

Интегральная оценка уровня знаний понятия O_1 рассчитывается по следующей формуле:

$$U_{\text{инт}}(O_1) = U_{\text{тест}}(O_1) + \sum_{i=1}^k U(I_i)R(I_i),$$

где $U(I_i)$ — уровень знаний входного понятия I_i ; $R(I_i)$ — ранг входного понятия I_i .

Предложенный метод интегральной оценки позволяет в автоматическом режиме определять уровень знаний обучаемого на основе анализа оценок ключевых понятий учебного курса. Этот метод отличается тем, что в нем корректируют базовую оценку, определяемую сеансом контроля обучаемого, используя уровень знания понятий, от которых зависит понимание ссылающихся на него понятий. Это позволяет более точно определять текущий уровень знаний обучаемого, предоставлять информацию обучаемым о пробелах в знаниях, формировать адекватную последовательность подачи учебного материала, наиболее точно соответствующего способностям обучаемого,

тем самым, повышая эффективность познавательной деятельности.

Заключение

Рассмотренная в данной статье модель структуризации учебного материала позволяет устанавливать динамическую связь между отдельными учебными блоками. Модель позволяет строить гибкую последовательность подачи учебного материала, учитывающую индивидуальные способности обучаемого и отвечающую различным запросам пользователей.

Предлагаемый подход к моделированию навигации согласуется с международными спецификациями электронного обучения SCORM и IMS, дополняя их конкретными адаптивными алгоритмами для навигации по учебным курсам на основе использования модели интегральной оценки знаний обучаемого.

Список литературы

1. Башмаков А. И., Башмаков И. А. Разработка компьютерных учебников и обучающих систем. М.: Феликс, 2003.
2. **Advanced** Distributed Learning Initiative. <http://www.adlnet.org>
3. **IMS** Global Learning Consortium, Inc. <http://www.imsglobal.org>
4. Растринин Л. А. Адаптивное обучение с моделью обучаемого. Рига: Зинанте. 1988. 160 с.
5. Bloom B. S. (Ed.) Taxonomy of educational objectives: The classification of educational goals: Handbook I, cognitive domain. New York: Longman, 1956.
6. Норенков И. П. Технологии разделяемых единиц контента для создания и сопровождения информационно-образовательных сред // Информационные технологии. 2003. № 8. С. 34–39.
7. Рыбаков А. Е. Применение модуля решения задач линейного программирования для поиска оптимального пути в графе, определяющего последовательность изучения образовательного курса // Материалы конференции "Технологии Microsoft в теории и практике программирования". Нижний Новгород: Изд. Нижегородского госуниверситета. 2007. С. 248–251.
8. Беспалько В. П. Основы теории педагогических систем. Воронеж: Изд-во Воронежского ун-та, 1977. 303 с.
9. Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew S. Tomkins. The Web as a graph: Measurements, models, and methods // In Proc. 5th Annual Int. Conf. Computing and Combinatorics, COCOON 1999. N 1627. Springer-Verlag.

УДК 004.942

С. П. Сердобинцев, д-р техн. наук,
проф., зав. каф. АПП, e-mail: serdobinzev@mail.ru,
Ф. В. Коломейко, инженер-программист,
e-mail: fed@atlant.baltnet.ru,
Калининградский государственный
технический университет

Применение информационных технологий в прогнозировании распределения объектов рыболовства

Изложены принципы построения автоматизированной системы, результаты анализа временных рядов на основе промысловой статистики для прогнозирования пространственно-временного распределения гидробионтов.

Ключевые слова: рыбный промысел, автоматизированная система, прогнозирование, база данных, временной ряд, спектральный анализ, локальная аппроксимация.

Вопросы математического моделирования пространственно-временного распределения биологических популяций и сообществ пока не получили такого развития, как моделирование динамики их численности [1]. В данной статье предложен подход к разработке пространственно-временных моделей на основе промысловой статистики, представленной в виде временного ряда. При этом широко используются программы, учитывающие не только статистическую информацию, но и мнение эксперта. Благодаря скорости и точности автоматизированных прогнозов возможно снижение непроизводительных затрат флота на поиск скоплений промысловых объектов.

Эффективность работы программного обеспечения в значительной степени зависит от рационального выбора алгоритмов прогнозирования динамики пространственно-временного распределения объектов лова по предшествующим и текущим результатам промысла.

В научных организациях рыбной отрасли за последние десятилетия накоплены большие объемы информации по промысловому вылову различных гидробионтов и результатам биологических анализов, проводимых в научно-исследовательских рейсах. В большинстве случаев эти данные можно представить в виде временных рядов.

Для решения научно-технических задач в настоящее время используются спектральный, корреляционный и другие методы анализа и прогнозирования временных рядов.

В рыбной отрасли широко применяются методы прогнозирования, основанные на уравнениях регрессии. Последние базируются на достаточно жестких предположениях о характере исследуемых процессов (стационарности, законе распределения и т. д.). Проверка выполнения этих предположений для биологических объектов затруднительна.

Для эффективного использования собранной информации целесообразно применять системы управления базами данных (СУБД) с программами, обеспечивающими автоматизированный анализ временных рядов с целью прогнозирования пространственно-временного распределения объектов рыболовства.

В зависимости от масштабов пространственно-временной изменчивости изучаемых явлений выделяются различные уровни рыболовства прогнозирования [1]. Для управления флотом важное значение имеет среднесрочное прогнозирование. Качественное решение этой задачи может быть достигнуто применением автоматизированной системы прогнозирования пространственно-временного распределения объектов лова (рис. 1). Автоматизированная система предназначена для:

- систематизации и архивирования оперативных данных состояния морской поверхности океана;
- применения созданных алгоритмов, моделей и программного обеспечения для рыболовства прогнозов и доведения фактической и прогнозной информации до потребителей с использованием современных средств передачи информации;
- обеспечения и сопровождения проведения отраслевых, межотраслевых и международных научных программ по исследованию состояния океана как среды обитания водных биоресурсов.

Разрабатываемая автоматизированная система прогнозирования распределения объектов рыболовства представляет собой комплекс технических

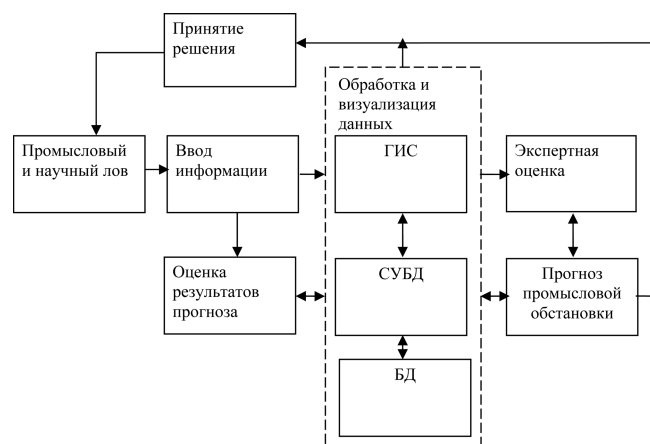


Рис. 1. Автоматизированная система прогнозирования пространственно-временного распределения рыболовства объектов

средств с программно-математическим обеспечением, осуществляющим сбор, анализ, обработку информации, прогноз и отображение промысловой обстановки.

В подсистему ввода автоматизированной системы поступает информация, получаемая средствами дистанционного (космического) зондирования и контактных измерений с научных и иных судов, океанологических буев, оснащенных автоматизированными датчиками.

Подсистема обработки и визуализации данных включает программно-аппаратные средства, необходимые для комплексной обработки информации, направляемой в СУБД, содержащую набор архивов по исследуемым объектам и районам промысла. Для визуализации пространственно-координированных данных используется географическая информационная система (ГИС), осуществляющая построение карт распределения параметров, характеризующих состояние поверхности океана и океанической среды. Базы данных (БД), СУБД и ГИС функционируют в тесном информационном взаимодействии.

Результаты прогнозирования и текущая ситуация в промысловых районах проходят экспертную оценку, с использованием которой принимаются решения о перспективности районов промысла.

Автоматизированная комплексная обработка ретроспективной и текущей информации позволяет выделить перспективные районы рыбного промысла. От полноты, оперативности и целенаправленности информационного обеспечения в значительной мере зависит технический уровень и конкурентоспособность промысловых прогнозов. Использование современных информационных технологий, включая СУБД и ГИС, упрощает процедуры сбора и визуализации данных, уменьшает вероятность ошибок и улучшает обмен данными между всеми участниками, обеспечивающими процесс рыбного лова.

Построение карт распределения параметров океанической среды, промысловых и биологических данных основывается на обработке нерегулярных временных рядов. Для анализа динамики временного ряда и прогнозирования пространственно-временного распределения гидробионтов в представленной системе используется сочетание мощных и быстро развивающихся методов сингулярного спектрального анализа (ССА) (в частности, модификация ССА под названием "Гусеница") и локальной аппроксимации. В совокупности эти методы обладают преимуществами при прогнозировании нерегулярных, нестационарных временных рядов, имеющих место в статистических наблюдениях за объектами лова. Важная особенность описанного подхода — возмож-

ность использования диалога исследователя, включая экспертов, с персональным компьютером.

Метод ССА с использованием разложения временного ряда на аддитивные составляющие позволяет решать различные задачи, включая выделения тренда в изменениях уловов, обнаружение периодик в распределении объектов промысла (сезонность), сглаживание исходного ряда. Предварительная фильтрация исходных данных методом ССА используется для последующей экстраполяции ряда методом локальной аппроксимации (ЛА).

Пространственно-временной анализ, в качестве примера, проводился применительно к акватории зоны (3404) Мавритании — району, широко эксплуатируемому судами России. В качестве исходных данных для моделирования и прогнозирования промысловой ситуации использовалась оперативная промысловая статистика за 1996—2007 гг., сформированная в базу данных. Объектом лова была выбрана скумбрия.

Для анализа временного ряда выбирается целый параметр K — длина окна — и строится траекторная матрица, столбцами которой являются скользящие отрезки ряда длиной K : с первой точки по K -ю, со второй — по $(K + 1)$ -ю и т. д. Траекторная матрица раскладывается в сумму элементарных матриц, задаваемых набором из собственного числа и двух сингулярных векторов: факторного и собственного.

Вся акватория разделяется на одноградусные квадраты и с помощью созданного на языке *Delphi* программного обеспечения информационно-справочной системы [3] и СУБД *Microsoft SQL Server* уловы суммируются в каждом квадрате ежемесячно. В итоге получается набор вре-

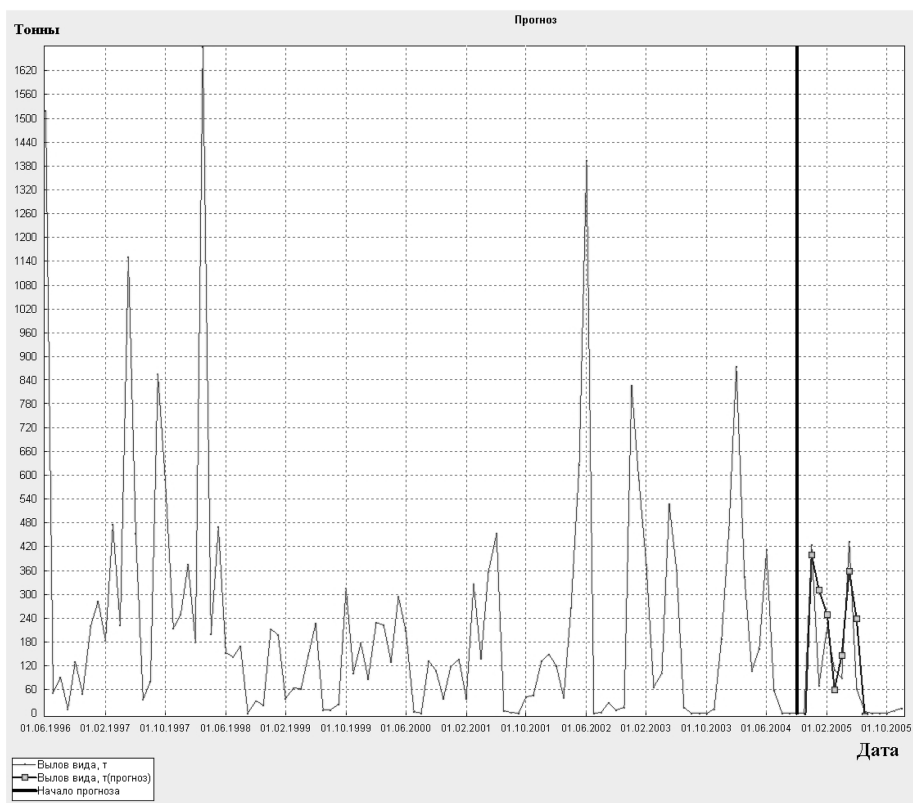


Рис. 2. Прогноз вылова скумбрии в тоннах на фоне фактических выловов в квадрате с центром в $18,5^\circ$ северной широты и $-16,5^\circ$ западной долготы

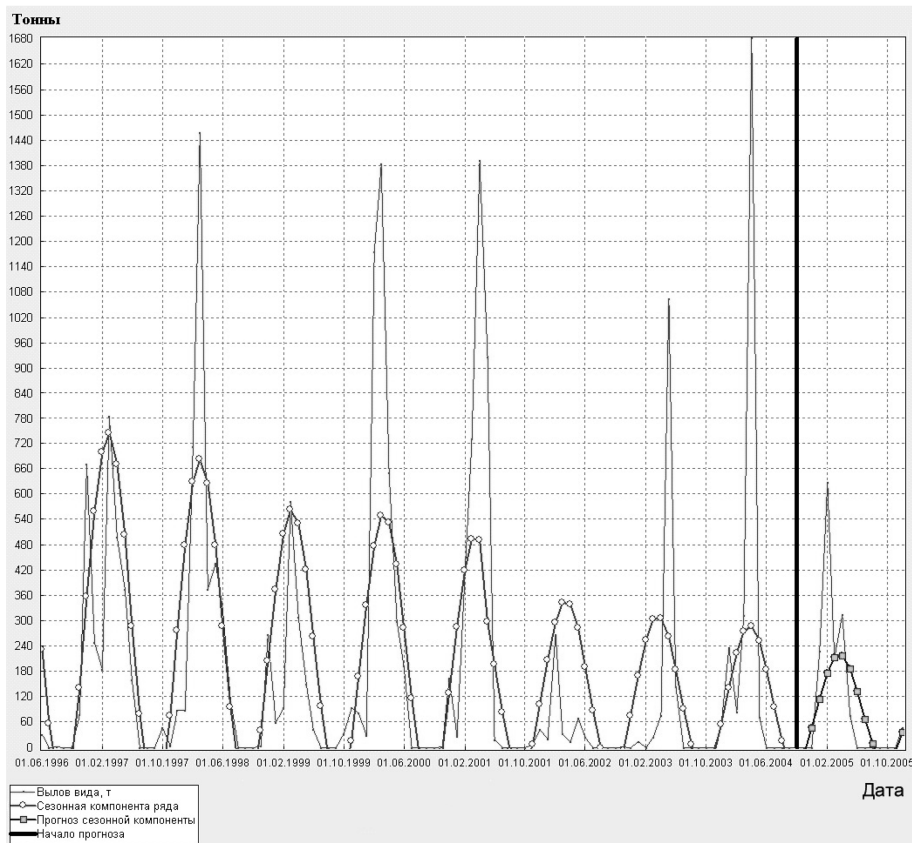


Рис. 3. Годичная периодичность в изменениях уловов скумбрии в квадрате акватории с центром в 16,5° широты и -16,5° долготы

менных рядов, характеризующих динамику вылова в каждом квадрате акватории:

$$Z_{lf} = \sum_{f=1}^r \sum_{l=1}^n v_{lf}$$

где Z_{lf} — значение формируемого временного ряда; v — вылов за одно траление; l — номер квадрата акватории; f — номер временного интервала; n — число тралений в месяц; r — общее число временных интервалов.

Суммарный вылов привязывается к центру квадратов. Пример части сформированного временного ряда для квадрата с центром (18,5° северной широты, -16,5° западной долготы) приведен в таблице.

Выбор "длины окна" K , близкой к половине ряда и кратной периодической составляющей, в соответствии с рекомендациями метода "Гусеница" [2] позволяет в процессе анализа очистить сформированный временной ряд от шума, выделить тренд и сезонные составляющие в выловах судов.

После построения K -мерной траекторной матрицы Z по обычной схеме проводится анализ главных компо-

Пример сгруппированного временного ряда

Широта	Долгота	Вылов, т	% вида	Дата
18,5	-16,5	1518,43	35,51	06.96
18,5	-16,5	51,9	6,36	07.96
18,5	-16,5	91	9,24	08.96

нент ковариационной матрицы $U = (1/k)Z^T Z$, вычисляются собственные числа и собственные векторы матрицы U в виде $U = PLP^T$, где L — диагональная матрица, составленная из собственных чисел, а P — ортогональная матрица собственных векторов матрицы U . Матрица P представляется как матрица перехода к главным компонентам: $ZP = Y = (y_1, y_2, \dots, y_M)$. Разложение матрицы $Z = Y \cdot P^T$ по ортогональным составляющим образует главные компоненты, среди которых выделяются составляющие, образующие тренд, периодичность, шум.

Разложение временного ряда, очистка от шума, выявление тренда и периодичности выполняются для каждого квадрата акватории. В дальнейшем очищенный от шума ряд для каждого квадрата прогнозируется с помощью метода ЛА.

Сравнение результатов прогноза с фактическими данными для одного из квадратов акватории $1^\circ \times 1^\circ$ представлено на рис. 2. Видно, что прогноз достаточно хорошо описывает динамику фактических выловов в 2005 г. В результате численного эксперимента выяснено, что меньшую дискретность разбиения по времени и пространству акватории брать нецелесообразно.

При меньшей пространственно-временной дискретизации статистические закономерности результатов улова отдельных районов часто перекрываются и смазываются флуктуациями.

Адекватность прогноза зависит от компромисса между точностью аппроксимации и близостью соседних точек, выбранных для ЛА. Суждение о точности аппроксимации [4] выполняется с использованием относительной квадратичной ошибки, определяющей расхождение между результатами прогнозируемых и фактических уловов.

Используемый подход удовлетворительно решает задачи оперативного прогнозирования (временные интервалы — месяц, квартал).

Очистка от шума временного ряда промысловой статистики позволяет выявить сезонные, годовичные и даже многолетние периодичности в распределении уловов скумбрии. Выявленная годичная периодичность в значениях уловов скумбрии показана на рис. 3. Полученные результаты качественно совпадают с данными биологических исследований [5].

Итогом применения алгоритма прогноза является пространственная сетка точек со значениями прогнозируемого вылова в рассматриваемом районе для заданного времени (первая половина 2006 г.). Сетки точек с использованием ГИС трансформируются в итоговые карты распределения фактического и прогнозируемого выловов.

Для сравнения и выявления качества полученной прогнозной карты применяются методы распознавания образов, в частности, корреляционное сравнение по

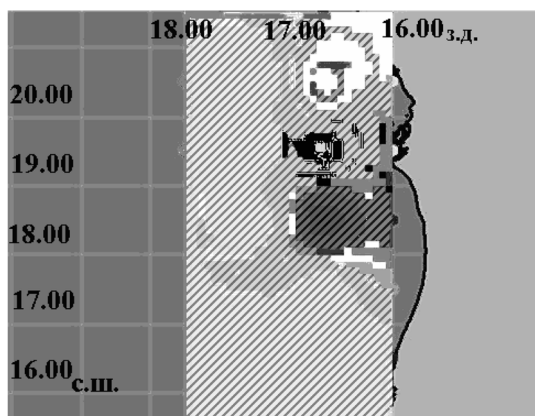


Рис. 4. Результаты программной обработки и сравнения карт фактических и прогнозируемых уловов скумбрии в мае 2006 г. (с. ш. — северная широта, з. д. — западная долгота).

уровню яркости между соответствующими пикселями. Для реализации данного алгоритма была создана программа (на языке C++), в которую передача карт из ГИС и их анализ происходит автоматически (рис. 4). Обработка и сравнение попиксельно карт проводится в пространстве цветов *RGB*. Яркости каждого пикселя вычисляются по формуле $I = 0,3R + 0,59G + 0,11B$, где *R* — красная, *G* — зеленая, *B* — синяя составляющие цветового пространства *RGB*. Прогноз для областей, показанных на рис. 4 штриховкой, является хорошим (коэффициент корреляции составляет более $80 \pm 0,05$). Области, где коэффициент корреляции составляет менее $80 \pm 0,05$, делятся на два вида: с заниженным (показаны черным) и завышенным (показаны белым цветом) прогнозом относительно фактического вылова. Эти области занимают соответственно 1,12 и 4 % площади исследуе-

мой акватории, однако и в этих районах коэффициент корреляции ни разу не был ниже 60. Значения фактических уловов показаны различными оттенками серого цвета — более высоким уловам соответствует более темный цвет. По вычисленному коэффициенту корреляции экспертом принимается решение о возможных корректировках расчетов для повышения достоверности прогноза.

Таким образом, автоматизированная система мониторинга объектов рыбопромыслового лова, созданная с помощью современных информационных технологий, может использоваться для эффективного целеуказания при направлении судов на промысел и прогнозирования результативности лова. Комплексная обработка данных мониторинга объектов и условий рыбопромыслового лова автоматизированной системой позволяет выявлять и закономерности изменения районов обитания, нереста, размерно-массовых характеристик гидробионтов.

Список литературы

1. Бочаров Л. Н. Системный анализ в краткосрочном рыбопромысловом прогнозировании. Л.: Наука, 1990.
2. Голяндина Н. Э. Метод "Гусеница"-SSA: анализ временных рядов: учеб. пособие. — СПб.: Изд-во СПбГУ, 2004.
3. Коломейко Ф. В., Зуев А. В., Чур В. Н. К вопросу о принципах построения и функционирования информационно-справочной системы АтлантНИРО // Материалы IX Всероссийской конференции по проблемам рыбопромыслового прогнозирования. — Мурманск, 2004.
4. Дли М. И., Круглов В. В., Осокин М. В. Локально-аппроксимационные модели социально-экономических систем и процессов. — М.: Наука, Физматлит, 2000.
5. Доманевский Л. Н. Рыбы и рыболовство в неритической зоне Центрально-Восточной Атлантики. — Калининград, 1998. — 195 с.

Информация

Представляем книгу

Российская энциклопедия CALS. Авиационно-космическое машиностроение / Гл. ред. А. Г. Братухин. М.: ОАО "НИЦ АСК", 2008. 608 с.: илл.

Состоялась презентация уникального издания, освещающего методы и средства автоматизации всех этапов жизненного цикла сложной наукоемкой продукции авиационно-космической отрасли промышленности.

Как отметил в предисловии к книге генеральный директор НПО "Энергомаш имени академика В. П. Глушко" академик РАН Б. И. Каторгин, "...энциклопедия CALS преследует главную цель — обобщить накопленный опыт, провести первичную систематизацию материалов и наметить пути дальнейшего совершенствования технологий обеспечения разработки, производства и эксплуатации конкурентоспособной машиностроительной продукции в области авиации и космонавтики".

Главы энциклопедии посвящены вопросам стратегии CALS, особенностям жизненного цикла и электронному представлению изделий, конструкторско-технологической подготовке и организации производства, политике, эксплуатации и экологии авиационно-космической техники, кадровому обеспечению CALS.

В написании книги участвовали более 100 известных ученых НИИ, КБ, предприятий авиакосмической индустрии, технических университетов.

Энциклопедия адресована специалистам не только авиационной, ракетно-космической, но и другой наукоемкой техники, а также преподавателям и студентам высших технических учебных заведений.

CONTENTS

Vyukova N. I., Galatenko V. A., Samborskij S. V. Generalization of the Software Pipelining Problem to Loop Nests	2
---	---

The paper discusses an approach to software pipelining of loop nests based on unrolling of the outermost loop which is then interchanged with the inner ones and moved down to the level of the innermost loop. The authors propose a combination of loop unrolling and software pipelining using methods of Integer Linear Programming, applicable to the case when the loop being unrolled cannot be interchanged with some of its nested ones.

Keywords: code optimization, software pipelining, module scheduling, loop nest, loop unrolling, Integer Linear Programming (ILP).

Laptev V. V., Morozov A. V. Expert System for Software Design Model Quality Assurance	8
--	---

Approach for automation of software quality improving process discussed in this article. A number of significant software quality characteristics identified here. The method for automation of quality measurement over the metrics calculation proposed. The main idea of that method is using fuzzy classification methods and software metrics as input. Estimating software quality characteristics is useful for improving design model and reduces development quality of software.

Keywords: software, design model, quality characteristics, metrics, fuzzy classification, neural network.

Zuev A. S., Petrov Yu. I. The Description of Windows Explorer Address Line Modification	11
--	----

Paper presents the description of the model introducing modification of an address line of the Windows operating system explorer and allowing to simplify navigation on logical disks and folders. Results of comparison of the model with similar interface elements of other applications are provided, its competitiveness in comparison with analogues is proven.

Keywords: human-computer interaction, graphical user interface, ergonomics of the software, designing of graphical interfaces, optimization of graphical interfaces, modification of an address line, the contextual interactive help, window of operational system Windows navigation on logical disks and folders, navigation in a network the Internet.

Gavrilov S. V., Kagramanyan E. R., Khodosh L. S. Evolutional Trends of Standard Cell Models for Static Timing Analysis of Digital VLSI Circuits	20
--	----

The evolutional trends of standard cell models are considered with respect to their usage for static timing analysis. The advantages and disadvantages of different types of models are discussed, and the limits of their application are mentioned in terms of continuous scaling of CMOS-technology down to deep submicron and nanometer regions.

Keywords: timing analysis of VLSI circuits, characterization of standard digital cells, non-linear delay model, NLDM, scalable polynomial delay model, SPDM, current source model, ECSM, CCS.

Shmyrev N. V. Modelling and Control Tools for Distributed Realtime Systems	24
---	----

The purpose of this paper is to describe the method of the distributed real-time system modeling. The combination of static analysis and runtime assertions checking allows a developer to verify the correctness of the formal model implementation.

Keywords: distributed real-time systems, modeling, managed execution.

Bimakov V. A. High Speed Processing of n-Dimensional Dot Geometric Models	29
--	----

The way to construct high speed devices for processing of n-dimensional images and n-dimensional scenes is described. The way is based on advantages of dot method of geometric modeling to execute the scene procedures, which are most "problematic" from the standpoint of time for their performance, with deep parallelism.

Keywords: high-speed method of processing of images and scenes, n-dimensional geometrical dot models, modelling of geometrical transformations, architecture of the specialised SIMD-system.

Arkhipov O. P., Zikova Z. P. Identity Output of Different Color Printers	37
---	----

The problem of providing maximal identity of output of color images printed on different printers, which have the stable color transmission, is being considered. For its decision new method, based on test of color

transmission, description and consistency of color spaces of printers is proposed. This method can be used not only by manufactures, but also by ordinary users of printers. Manufacturers can apply to write printer's drivers in concern of determining the function of color separation. The printer's user can use proposed method to work together with special soft. Special soft, giving new determination of color separation function, allows transformation existing RGB-image in order to get maximally identical outputs for different printers.

Keywords: color printing, scanning, print outputs of color images, color spaces, color transmission.

Koldaev A. I. *Modelling of Intellectual System of Maintenance of Decision-Making at Management of Technological Process* 42

The intellectual system of neuro-fuzzy conclusion for the decision of a problem of maintenance of decision-making is offered at management of technological process of reactor installation of hydrogenation of butanol.

Keywords: fuzzy logic, neuro-fuzzy network.

Mikheeva T. I., Potapkin A. A., Demyanenko R. V. *Adaptive Clustering of Graph of Street-Road Network* 46

The attributive street-road network graph compression method based on adaptive fuzzy clustering is analyzed. The performance criterion of adaptive clustering and evaluation of attributive data compression ratio are considered.

Keywords: street-road network graph, fuzzy clustering, compression graph, transport route.

Kuperin Yu. A., Schastlivtsev R. R. *Forecasting Financial Time Series by Committees from Artificial Neural Networks* 51

In the present paper methods of high quality time series forecasting by means of artificial neural networks (ANN) committees have been elaborated. Conventional approaches to aggregation of ANN's into committees have been analyzed and a novel approach to forming of ANN committees has been suggested. The proposed methods have been tested in order to demonstrate the high quality the USA stock market time series forecasting.

Keywords: forecasts, neural networks committees, week forecast, strong forecast, financial time series.

Serkov L. A. *Synergetic Aspects of Modelling of Innovative Processes* 56

In frameworks of the synergetic approach stochasticity of innovative process and influence on this process of the innovative globalization was studied. By authors it is shown, that the role of noise consists in noise-induced phenomenon named self-organized criticality. Involving in innovative process of the increasing number of participants (innovative globalization) leads to some suppression of fluctuations, to reduction of a role of accident in occurrence and development of innovations.

Keywords: synergetics, innovations, diffusion of innovations, technological transfer, determinancy, stochasticity, bifurcations, the self-organized criticality, innovative activity, logistical curve.

Solovetski A. S. *Mobile Informational Technologies in Marketing and Advertising* 62

Results, received by author during the mobile informational technologies (MIT) marketing research are shown in article. The goal of research was evaluation of MIT use in marketing and advertising.

Quoted data about different regional markets allows to draw a conclusions on availability of MIT marketing services worldwide and particularly in Russia.

MIT advantages and features as compared to classical advertising instruments are noted in article.

Mobile technologies used in marketing are inquired in detail as well as methods of its use in marketing are proposed.

Author is considering ethical and technical problems appear during use of mobile marketing instruments.

This article would be interesting for experts who work in mobile telecommunications, marketing and advertising, for researchers engaged in IT industry problems as well for everyone interested in mobile telecom and new marketing instruments.

Keywords: mobile marketing, mobile ad, mobile informational technology, mobile telecom, value-added services, mobile content.

Sarafanov A. V., Komarov V. A. *Designing Mathematical Model Multiuser Operating Mode of the Hardware and Software Remote-Access Complex* 67

Application mathematic simulation for analysis of the hardware and software remote-access complex.

Keywords: multiuser mode, mathematic simulation, hardware and software remote-access complex (H&SRAC).

Norenkov I. P., Sokolov N. K. *Creation of Individual Learning Routes in Ontology Education Systems* 74

If the number of shareable content modules is increased in a base of educational resources diversity of possible learning routes is increased too. Then the problem of optimal learning route synthesis is arised. The paper

is devoted to the model and method of synthesis of textbooks by the technology of shareable content modules for support of individual learning routes.

Keywords: computer instruction system, technology of shareable content modules, ontology, learning route.

Rybakov A. E. *Modeling to Courses Structure and Subsystem of Navigation for SCORM Compatible Learning Management Systems* 77

The navigation system in e-learning courses is considered. Structure of learning courses and modeling of navigation are suggested. They are compatible with SCORM and IMS standards. They are expanding them with the specific adaptive algorithms for navigation in learning courses. Algorithms are based on model of integral mark of student's knowledge.

Keywords: e-learning, sequence, education, navigation, courses, learner, SCORM, IMS.

Serdobintsev S. P., Kolomeyko F. V. *The Usage of Information Technologies in Fishery Objects Allocation Prediction*.. . . . 82

In this article the following questions are considered: automation system construction principles and time series analysis results which are based on fishery statistics for prediction of time and space hydrobionts allocation.

Keywords: fishery, automation system, prediction, data base, time series, singular spectrum analysis, local approximation.

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *О. А. Ефремова*.
Корректор *Т. В. Арбузова*

Сдано в набор 29.12.2008. Подписано в печать 16.02.2009. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л.10,78. Уч.-изд. л. 11,84. Заказ 137. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.
Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"
142110, Московская обл., г. Подольск, ул. Кирова, 15