

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 23  
2017  
№ 8

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ  
Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Аникин В. И., Аникина О. В.** Визуальное компонентное моделирование систем нечеткого вывода в электронных таблицах . . . . . 555
- Соснина О. А., Филинских А. Д., Коротаяева А. С.** Сравнение способов создания низкополигональной 3D-модели . . . . . 564

### ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Инютин С. А.** Метод вычисления характеристики отношения порядка для параллельных форматов данных . . . . . 569

### СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Тельпухов Д. В., Соловьев Р. А.** Метод поиска эквивалентных ошибок в логических схемах . . . . . 575

### ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Фонин Ю. Н.** Применение языка PDDL для автоматической генерации средств разработки процессоров . . . . . 583

### БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Асратян Р. Э., Лебедев В. Н.** Интернет-служба конвейерной обработки защищенных информационных запросов в мультисетевой среде . . . . . 589

### ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Кугаевских А. В.** Сравнение параметров фильтра Габора для эффективного извлечения краев . . . . . 598

### ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Авдошин С. М., Песоцкая Е. Ю.** Интернет вещей: ритейл . . . . . 606

### НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Печников Д. А.** Применение методов интервальной оценки результатов контроля знаний в компьютерных системах тестирования . . . . . 613
- Кадыева Л. М., Левин С. Е., Окренг Я. Н., Лоскутов Д. И., Романенков А. М.** Использование нейронечетких вычислений для диспетчеризации на железнодорожной станции со сложной топологией . . . . . 618

Главный редактор:  
СТЕМПКОВСКИЙ А. Л.,  
акад. РАН, д. т. н., проф.

Зам. главного редактора:  
ИВАННИКОВ А. Д., д. т. н., проф.  
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:  
БЫЧКОВ И. В., акад. РАН, д. т. н.  
ЖУРАВЛЕВ Ю. И.,  
акад. РАН, д. ф.-м. н., проф.  
КУЛЕШОВ А. П.,  
акад. РАН, д. т. н., проф.  
ПОПКОВ Ю. С.,  
акад. РАН, д. т. н., проф.  
РУСАКОВ С. Г.,  
чл.-корр. РАН, д. т. н., проф.  
РЯБОВ Г. Г.,  
чл.-корр. РАН, д. т. н., проф.  
СОЙФЕР В. А.,  
акад. РАН, д. т. н., проф.  
СОКОЛОВ И. А., акад.  
РАН, д. т. н., проф.  
СУЕТИН Н. В., д. ф.-м. н., проф.  
ЧАПЛЫГИН Ю. А.,  
акад. РАН, д. т. н., проф.  
ШАХНОВ В. А.,  
чл.-корр. РАН, д. т. н., проф.  
ШОКИН Ю. И.,  
акад. РАН, д. т. н., проф.  
ЮСУПОВ Р. М.,  
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:  
АВДОШИН С. М., к. т. н., доц.  
АНТОНОВ Б. И.  
БАРСКИЙ А. Б., д. т. н., проф.  
ВАСЕНИН В. А., д. ф.-м. н., проф.  
ВИШНЕКОВ А. В., д. т. н., проф.  
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.  
ДОМРАЧЕВ В. Г., д. т. н., проф.  
ЗАБОРОВСКИЙ В. С., д. т. н., проф.  
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.  
ЗАРУБИН В. С., д. т. н., проф.  
КАРПЕНКО А. П., д. ф.-м. н., проф.  
КОЛИН К. К., д. т. н., проф.  
КУЛАГИН В. П., д. т. н., проф.  
КУРЕЙЧИК В. В., д. т. н., проф.  
ЛЬВОВИЧ Я. Е., д. т. н., проф.  
МАРТЫНОВ В. В., д. т. н., проф.  
МИХАЙЛОВ Б. М., д. т. н., проф.  
НЕЧАЕВ В. В., к. т. н., проф.  
ПОЛЕЩУК О. М., д. т. н., проф.  
САКСОНОВ Е. А., д. т. н., проф.  
СОКОЛОВ Б. В., д. т. н., проф.  
ТИМОНИНА Е. Е., д. т. н., проф.  
УСКОВ В. Л., к. т. н. (США)  
ФОМИЧЕВ В. А., д. т. н., проф.  
ШИЛОВ В. В., к. т. н., доц.

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

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.  
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.  
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

# INFORMATION TECHNOLOGIES

## INFORMACIONNYYE TEHNOLOGII

Vol. 23  
2017  
No. 8

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

### CONTENTS

#### MODELING AND OPTIMIZATION

- Anikin V. I., Anikina O. V.** The Visual Component Modeling of Fuzzy Inference Systems in Spreadsheets . . . . . 555
- Sosnina O. A., Filinskikh A. D., Korotaeva A. S.** Comparing Approaches to Creating a Low Polygonal 3D Model . . . . . 564

#### COMPUTING SYSTEMS AND NETWORKS

- Inyutin S. A.** Method Calculation Characteristic of the Order Relation for Computer Modular Value . . . . . 569

#### CAD-SYSTEMS

- Telpukhov D. V., Solovyev R. A.** Development of a Method for Identical Fault Search in Logical Circuits . . . . . 575

#### SOFTWARE ENGINEERING

- Fonin Yu. N.** Using PDDL Language for Automatically Generation of the Tool Chain for Processors . . . . . 583

#### CRYPTOSAFETY INFORMATION

- Asratian R. E., Lebedev V. N.** Internet Service for Pipeline Processing of Protected Information Queries in Multinetwork Environment . . . . . 589

#### DIGITAL PROCESSING OF SIGNALS AND IMAGES

- Kugaevskikh A. V.** Comparison Gabor Filter Parameters for Efficient Edge Detection 598

#### APPLIED INFORMATION TECHNOLOGIES

- Avdoshin S. M., Pesotskaya E. Yu.** Internet of Things: Retail . . . . . 606

#### NEUROTECHNOLOGIES

- Pechnikov D. A.** Using the Methods of the Interval Estimation of the Results of the Control of Knowledge in Computer Testing Systems . . . . . 613
- Kadyeva L. M., Levin S. E., Okrent Ya. N., Loskutov D. I., Romanenkov A. M.** The Development of Neuro-Fuzzy Module Calculating the Priority Dispatching Tasks to Railway Station . . . . . 618

#### Editor-in-Chief:

Stempkovsky A. L., Member of RAS,  
Dr. Sci. (Tech.), Prof.

#### Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.  
Filimonov N. B., Dr. Sci. (Tech.), Prof.

#### Chairman:

Bychkov I. V., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Zhuravljov Yu. I., Member of RAS,  
Dr. Sci. (Phys.-Math.), Prof.  
Kuleshov A. P., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Popkov Yu. S., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Rusakov S. G., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Ryabov G. G., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Soifer V. A., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Sokolov I. A., Member of RAS,  
Dr. Sci. (Phys.-Math.), Prof.  
Suetin N. V.,  
Dr. Sci. (Phys.-Math.), Prof.  
Chaplygin Yu. A., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Shakhnov V. A., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Shokin Yu. I., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Yusupov R. M., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.

#### Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.  
Antonov B. I.  
Barsky A. B., Dr. Sci. (Tech.), Prof.  
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.  
Vishnekov A. V., Dr. Sci. (Tech.), Prof.  
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.  
Domrachev V. G., Dr. Sci. (Tech.), Prof.  
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.  
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.  
Zarubin V. S., Dr. Sci. (Tech.), Prof.  
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.  
Kolin K. K., Dr. Sci. (Tech.)  
Kulagin V. P., Dr. Sci. (Tech.), Prof.  
Kureichik V. V., Dr. Sci. (Tech.), Prof.  
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.  
Martynov V. V., Dr. Sci. (Tech.), Prof.  
Mikhailov B. M., Dr. Sci. (Tech.), Prof.  
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.  
Poleschuk O. M., Dr. Sci. (Tech.), Prof.  
Saksonov E. A., Dr. Sci. (Tech.), Prof.  
Sokolov B. V., Dr. Sci. (Tech.)  
Timonina E. E., Dr. Sci. (Tech.), Prof.  
Uskov V. L. (USA), Dr. Sci. (Tech.)  
Fomichev V. A., Dr. Sci. (Tech.), Prof.  
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

#### Editors:

Bezmenova M. Yu.  
Grigorin-Ryabova E. V.  
Lysenko A. V.  
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

# МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 519.6

**В. И. Аникин**, д-р техн. наук, проф., e-mail: anikin\_vi@mail.ru,  
Поволжский государственный университет сервиса,  
**О. В. Аникина**, канд. техн. наук, доц., e-mail: blue-waterfall@yandex.ru,  
Тольяттинский государственный университет

## Визуальное компонентное моделирование систем нечеткого вывода в электронных таблицах

*Доказано, что любая табличная модель представляет собой действующую граф-машину некоторого алгоритма. Разработана библиотека компонентов, симулирующих работу фрагментов этой граф-машины, для визуального табличного моделирования нечеткой логики без написания программного кода, по принципу "программирование без программирования". Предложенная технология компонентного моделирования систем нечеткого вывода в электронных таблицах по эффективности сопоставима с моделированием нечетких систем в MATLAB и R.*

**Ключевые слова:** нечеткая логика, электронные таблицы, компонентное моделирование, граф-машина алгоритма, визуализация алгоритмов

### Введение

В настоящее время нечеткое моделирование является одним из самых популярных направлений прикладных исследований в области управления и принятия решений. Нечеткое моделирование оказывается особенно полезным, когда в описании технических систем и процессов присутствует недостаточность или неопределенность знаний, либо когда поведение реальных систем/процессов невозможно представить математическими моделями [1].

Одним из лучших коммерческих программных пакетов для моделирования нечеткой логики является пакет Fuzzy Logic Toolbox в MATLAB ([www.mathworks.com](http://www.mathworks.com)) [2], содержащий функции, GUI-интерфейс и блоки Simulink для реализации нечетких рассуждений и воплощения их в системе нечеткого вывода (Fuzzy Inference System — FIS).

Похожий на MATLAB, но распространяемый по лицензии GNU, язык программирования R ([cran.r-project.org](http://cran.r-project.org)) включает несколько пакетов расширений для создания FIS, например, пакеты sets, frbs, FuzzyR и др., причем новейший пакет FuzzyR также имеет удобный пользовательский GUI-интерфейс.

Главными недостатками MATLAB и R являются сложность освоения языка программирования, основанного на матричных операциях, а также неразвитые средства визуализации данных этими программами — все, кому приходилось работать в них, наверняка замечали очевидное несоответствие

между высокой эффективностью вычислений в интерактивном окне команд и примитивным выводом получаемых результатов.

Естественно, что пользовательский GUI-интерфейс заметно облегчает работу по созданию модели FIS, но ограничивает свободу творчества разработчика предопределенными возможностями этого интерфейса.

Простые системы нечеткого вывода можно также создавать, используя электронные таблицы (ЭТ). Однако эти популярные и интуитивно понятные программы не имеют встроенных средств для нечеткого моделирования, поэтому разработчик должен вручную создавать структуру FIS и вводить необходимые формулы.

По этой причине на сегодняшний день нечеткое моделирование в ЭТ используется более чем скромно. Поиск литературных источников в Интернете позволил нам найти лишь две достойные упоминания ссылки.

1. Компания Microsoft предлагает пользователям Excel удобную бесплатную надстройку ([www.microsoft.com/en-us/download/details.aspx?id=15011](http://www.microsoft.com/en-us/download/details.aspx?id=15011)), позволяющую проводить поиск и выборку данных из двух таблиц, в которых схожие данные ключевых столбцов записаны с какими-то отличиями (сокращения, порядок слов, опечатки и т. п.).

2. Авторы работы [3] ([www.soa.org/Files/Research/Projects/research-2013-fuzzy-logic.pdf](http://www.soa.org/Files/Research/Projects/research-2013-fuzzy-logic.pdf)) разработали FIS в Excel для оценки риска недостойного пове-

дения консультантов по продажам товаров (побуждаемые высокими комиссионными от продаж консультанты могут скрывать от покупателей недостатки товара). Одним из недостатков этой FIS является то, что фактически она реализована на языке программирования VBA, а рабочие листы Excel выполняют в основном лишь функцию интерфейса пользователя с программным кодом. Кроме того, созданная FIS получилась громоздкой, неудобной в использовании и трудно модифицируемой.

Цель нашего исследования — разработка эффективной технологии визуального компонентного моделирования нечеткой логики в ЭТ без написания программного кода, по принципу "программирование без программирования". Впервые идея компонентного табличного моделирования была предложена и опробована нами в 2012 г. применительно к решению задач формальной логики [4].

Визуальное компонентное моделирование нечетких систем в ЭТ характеризуется рядом достоинств.

1. Табличная модель FIS не содержит программного кода VBA и, следовательно, имеет сравнительно высокое быстродействие несмотря на повышенные требования ЭТ к системным ресурсам.
2. Компонентная табличная модель FIS работает как в Microsoft Excel, так и в LibreOffice Calc.
3. Трудоемкость разработки компонентной табличной модели FIS сопоставима с трудоемкостью разработки аналогичной FIS с помощью GUI-интерфейсов Fuzzy Logic Toolbox и FuzzyR.
4. Библиотека компонентов нечеткого вывода в ЭТ предоставляет пользователю широкие возможности для креативного творчества и экспериментирования в области нечеткой логики.
5. Табличные модели FIS характеризуются впечатляющей визуализацией как данных, так и информационной структуры алгоритмов нечеткого вывода.

### Обоснование технологии компонентного табличного моделирования

Идея технологии визуального компонентного моделирования в электронных таблицах основывается на аналогии между процессом пересчета ячеек таблиц и выполнением программ в распределенных компьютерных системах.

Действительно, ЭТ можно представить себе как некий распределенный вычислитель. В процессе пересчета ЭТ каждая ее ячейка выполняет свою долю кооперативной работы по переработке и хранению данных, используя для вычислений формулу, ассоциированную с ячейкой, и саму себя — в качестве хранилища результата. Должным образом организуя связи между ячейками табличной модели и применяя в них подходящие формулы, можно "заставить" такой распределенный вычислитель решить любую вычислительную задачу, допускаемую

ресурсными возможностями ЭТ. Связи между ячейками табличной модели алгоритма отражают принцип причинности и образуют ациклический ориентированный граф [5].

Вместе с тем под информационной структурой компьютерной программы понимается совокупность сведений о том, как ее элементы данных связаны друг с другом. Эту структуру удобно описывать с помощью ациклического ориентированного графа алгоритма, в котором вершины соответствуют операциям (действиям) программы, а дуги отражают связи (отношения) между вершинами графа. Авторы работы [6] выделяют четыре графовые модели алгоритма: граф управления, информационный граф, операционно-логическая история и история реализации программы. Из них наибольший интерес представляет история реализации программы, и именно эту модель обычно называют *графом алгоритма*. Зная этот граф, можно получить практически любые сведения об алгоритме и его программной реализации.

Если пойти дальше и в каждую вершину графа алгоритма поместить исполнительное устройство (ИУ), выполняющее заданную операцию (действие) программы, то будет получена граф-машина, реализующая конкретный вычислительный алгоритм при заданных входных данных.

Между граф-машиной алгоритма и его табличной моделью существует глубокая аналогия, о которой мало кто из пользователей ЭТ знает (табл.).

Из этой аналогии следует важнейший теоретический вывод: *любая табличная модель реализует некоторый алгоритм и представляет собой ничто иное, как его действующую граф-машину.*

Аналогия "Граф-машина алгоритма — табличная модель"

Граф-машина алгоритма	Табличная модель
Исполнительное устройство в узле графа	Ячейка табличной модели
Входное устройство	Ячейка, содержащая исходные данные
Выходное устройство	Ячейка, содержащая результат вычислений
Граф-машина не имеет собственной выделенной памяти, результаты вычислений сохраняются в ее ИУ	Табличная модель не имеет собственной выделенной памяти, результаты вычислений сохраняются в ее ячейках
В процессе функционирования граф-машины каждое ИУ выполняет над данными заданную вычислительную операцию	В процессе пересчета ЭТ каждая ячейка выполняет над данными заданную вычислительную операцию
Граф алгоритма является ациклическим, каждое ИУ работает в режиме однократного срабатывания	Граф связей между ячейками табличной модели является ациклическим, каждая ячейка работает в режиме однократно-го срабатывания

Пользуясь, кроме того, обширным набором инструментов форматирования ячеек, пользователь может особым образом форматировать узлы графа алгоритма как статически, так и динамически в ходе выполнения алгоритма.

Эти уникальные достоинства электронных таблиц, недоступные другим средам программирования и моделирования, позволяют просто и изящно решать в ЭТ нетривиальную задачу визуализации информационной структуры и особенностей работы алгоритмов — задачу, которой, начиная с публикации знаменитых монографий Кнута и Вирта, увлекался не один любознательный программист.

Учитывая сказанное, под компонентом табличной модели будем понимать прямоугольный (для удобства копирования) интервал ячеек с формулами, кооперативно выполняющими некоторую полезную вычислительную функцию. Помимо ячеек с формулами, компонент должен содержать входные и выходные ячейки для обеспечения его взаимодействия с другими компонентами табличной модели.

В общем случае компонент табличной модели содержит ячейки пяти типов: 1) ячейки с формулами, симулирующими работу одного из фрагментов граф-машины алгоритма; 2) входные ячейки, ссылающиеся на другие компоненты; 3) ячейки с параметрами компонента; 4) выходные ячейки, хранящие результат работы компонента; 5) ячейки с комментариями компонента.

Требования, предъявляемые к компонентам:

- ссылки в формулах компонентов на другие ячейки должны быть относительными, чтобы компонент можно было копировать из библиотеки, не нарушая его работоспособности;
- компоненты одного типа по возможности должны иметь одинаковую структуру для обеспечения их взаимозаменяемости (необязательное требование).

### Библиотека компонентов нечеткого вывода в электронных таблицах

Разработка любой FIS включает ряд этапов, основанных на базовых положениях нечеткой логики [7]:

- фаззификация;
- формирование базы нечетких продукционных правил;
- агрегирование подусловий правил;
- активизация заключений правил;
- аккумулярование заключений правил;
- дефаззификация.

В настоящее время наибольшее распространение получили четыре алгоритма нечеткого вывода: Мамдани, Ларсена, Такаги—Сугено и Цукамото. Популярность этих алгоритмов объясняется их простотой и высоким быст-

родействием, обусловленным тем, что на этапе активизации заключений вместо трудоемкой интегральной операции свертки используются более экономные операции  $\min$  (алгоритм Мамдани), арифметическое умножение (алгоритм Ларсена), простые алгебраические функции (алгоритмы Такаги—Сугено и Цукамото).

В согласии с этими этапами разработанная библиотека компонентов нечеткого вывода в ЭТ делится на девять групп (категорий): *Universe*, *Membership functions*, *Fuzzy operations*, *Modifiers*, *Implications*, *Accumulations*, *Inference*, *Takagi—Sugeno* и *lingvar* — всего около 50 компонентов. Для примера на рис. 1 показаны по одному компоненту из каждой группы, за исключением трех последних групп, представленных всеми компонентами.

Компоненты библиотеки имеют стандартное оформление ячеек:

- ячейки розового цвета содержат формулу Excel и одновременно являются выходными ячейками компонента;
- темно-голубые ячейки являются входными и содержат ссылки на ячейки других компонентов;
- светло-голубые входные ячейки хранят значения параметров компонента;
- ячейки белого цвета содержат названия элементов компонентов или результаты промежуточных вычислений по формулам;
- ячейки серого цвета не задействованы и могут быть использованы для документирования компонента.

Группа *Universe* содержит два компонента: первый (рис. 1) используется для задания универсума непрерывного нечеткого множества и хранения четкого значения  $\text{in}$  входной переменной *lingvar*, второй — для задания нечеткой переменной типа *singleton*.

Группа *Membership functions* содержит 14 компонентов для задания функций принадлежности (ФП) термов лингвистических переменных. В целях обеспечения совместимости компоненты этой группы дублируют формулы и имена 12 функций прина-

<p>Компонент группы <i>Universe</i></p> <table border="1"> <thead> <tr> <th>lingvar</th> <th>ul</th> <th>ur/mf</th> <th>n</th> <th>h</th> <th>in</th> <th>x</th> </tr> </thead> <tbody> <tr> <td>lingname</td> <td>0,0</td> <td>10,0</td> <td>100,0</td> <td>0,1</td> <td>5,0</td> <td>5,0</td> </tr> </tbody> </table>	lingvar	ul	ur/mf	n	h	in	x	lingname	0,0	10,0	100,0	0,1	5,0	5,0	<p>Компонент группы <i>Accumulations</i></p> <table border="1"> <thead> <tr> <th>mf5</th> <th>mf4</th> <th>mf3</th> <th>mf2</th> <th>mf1</th> <th>p</th> <th>mf</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>0</td> <td>0,0</td> </tr> </tbody> </table>	mf5	mf4	mf3	mf2	mf1	p	mf						0	0,0						
lingvar	ul	ur/mf	n	h	in	x																													
lingname	0,0	10,0	100,0	0,1	5,0	5,0																													
mf5	mf4	mf3	mf2	mf1	p	mf																													
					0	0,0																													
<p>Компонент группы <i>Membership functions</i></p> <table border="1"> <thead> <tr> <th>lingval</th> <th>a1</th> <th>c1</th> <th>a2</th> <th>c2</th> <th>x</th> <th>mf(x)</th> </tr> </thead> <tbody> <tr> <td>trimf</td> <td>1,0</td> <td>3,0</td> <td>5,0</td> <td>5,0</td> <td></td> <td>0,0</td> </tr> </tbody> </table>	lingval	a1	c1	a2	c2	x	mf(x)	trimf	1,0	3,0	5,0	5,0		0,0	<p>Компонент группы <i>Inference</i></p> <table border="1"> <thead> <tr> <th>ul</th> <th>ur</th> <th>CoG</th> <th>CoA</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>0</td> <td>y</td> <td>lingvar</td> </tr> <tr> <td>2,0</td> <td><math>\int u dx</math></td> <td>-</td> <td></td> </tr> <tr> <td>1</td> <td>0,000</td> <td>0,00</td> <td></td> </tr> <tr> <td>2</td> <td>0,000</td> <td>0,00</td> <td></td> </tr> </tbody> </table>	ul	ur	CoG	CoA	i	0	y	lingvar	2,0	$\int u dx$	-		1	0,000	0,00		2	0,000	0,00	
lingval	a1	c1	a2	c2	x	mf(x)																													
trimf	1,0	3,0	5,0	5,0		0,0																													
ul	ur	CoG	CoA																																
i	0	y	lingvar																																
2,0	$\int u dx$	-																																	
1	0,000	0,00																																	
2	0,000	0,00																																	
<p>Компонент группы <i>Fuzzy operations</i></p> <table border="1"> <thead> <tr> <th>name</th> <th>lingval</th> <th>p</th> <th>x1</th> <th>x2</th> <th>t-norm</th> <th>s-norm</th> </tr> </thead> <tbody> <tr> <td>min/max</td> <td></td> <td></td> <td></td> <td></td> <td>0,0</td> <td>0,0</td> </tr> </tbody> </table>	name	lingval	p	x1	x2	t-norm	s-norm	min/max					0,0	0,0	<p>Компонент группы <i>Inference</i></p> <table border="1"> <thead> <tr> <th>aggRng</th> <th>actRng</th> <th>lingvar</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	aggRng	actRng	lingvar																	
name	lingval	p	x1	x2	t-norm	s-norm																													
min/max					0,0	0,0																													
aggRng	actRng	lingvar																																	
<p>Компонент группы <i>Modifiers</i></p> <table border="1"> <thead> <tr> <th>name</th> <th>lingval</th> <th>comment</th> <th>value</th> <th>func</th> <th>x</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>hedge</td> <td></td> <td>CON, DIL</td> <td>very</td> <td>=fm*2</td> <td></td> <td>0,0</td> </tr> </tbody> </table>	name	lingval	comment	value	func	x	y	hedge		CON, DIL	very	=fm*2		0,0	<p>Компонент группы <i>Inference</i></p> <table border="1"> <thead> <tr> <th>ul</th> <th>h</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>x</td> </tr> <tr> <td>0</td> <td>0,00</td> </tr> <tr> <td>1</td> <td>0,00</td> </tr> </tbody> </table>	ul	h	i	x	0	0,00	1	0,00												
name	lingval	comment	value	func	x	y																													
hedge		CON, DIL	very	=fm*2		0,0																													
ul	h																																		
i	x																																		
0	0,00																																		
1	0,00																																		
<p>Компонент группы <i>Implications</i></p> <table border="1"> <thead> <tr> <th>name</th> <th>lingval</th> <th>comment</th> <th>F</th> <th>mf(x)</th> <th>mf(y)</th> <th>mf(x,y)</th> </tr> </thead> <tbody> <tr> <td>Mamdani</td> <td></td> <td>min</td> <td>1,0</td> <td></td> <td></td> <td>0,0</td> </tr> </tbody> </table>	name	lingval	comment	F	mf(x)	mf(y)	mf(x,y)	Mamdani		min	1,0			0,0	<p>Компонент группы <i>Inference</i></p> <table border="1"> <thead> <tr> <th>lingvar</th> </tr> </thead> <tbody> <tr> <td>ямяя?</td> </tr> </tbody> </table>	lingvar	ямяя?																		
name	lingval	comment	F	mf(x)	mf(y)	mf(x,y)																													
Mamdani		min	1,0			0,0																													
lingvar																																			
ямяя?																																			
<p>Компонент группы <i>Takagi-Sugeno</i></p> <table border="1"> <thead> <tr> <th>name</th> <th>p0</th> <th>p1</th> <th>p2</th> <th>in1</th> <th>in2</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>Sugeno</td> <td>0,0</td> <td>0,0</td> <td>0,0</td> <td></td> <td></td> <td>0,0</td> </tr> </tbody> </table>	name	p0	p1	p2	in1	in2	y	Sugeno	0,0	0,0	0,0			0,0																					
name	p0	p1	p2	in1	in2	y																													
Sugeno	0,0	0,0	0,0			0,0																													

Рис. 1. Библиотека компонентов нечеткого вывода в электронных таблицах

длжности пакета Fuzzy Logic Toolbox MATLAB. Кроме того, для упрощения ввода к функциям MATLAB добавлены две простые кусочно-линейные функции принадлежности:

```
ztrmf(x, a1, c1) = ifelse(x <= a1; 1; ifelse(x <= c1; 1 - (x - a1) / (c1 - a1); 0))
strmf(x, a1, c1) = ifelse(x <= a1; 0; ifelse(x <= c1; (x - a1) / (c1 - a1); 1))
```

где  $x$  — значение входной переменной;  $a1, c1$  — параметры ФП (рис. 1).

Группа *Fuzzy operations* содержит 10 компонентов, симулирующих нечеткие операции T- и S-норм: min-пересечение и max-объединение Заде, вероятностное пересечение и объединение, граничное пересечение и объединение, пересечения и объединения Лукашевича, Эйнштейна, Хамакера, Ягера, Дюбуа-Прада, Вебера, Ю.

Группа *Modifiers* содержит шесть компонентов, реализующих модификаторы NOT, more/less, little, slightly, very, extremely, very/very нечетких высказываний: компонент hedge выполняет операции концентрирования (CON) или растяжения (DIL); компонент intensifier — операцию изменения контрастности (INT); компоненты — NOT, NOT(p>-1), NOT(p>0) — операции стандартного, параметрического Сугено и параметрического Ягера отрицания; repeater — операцию повторения (дублирования) входного значения.

Группа *Implications* содержит восемь компонентов, симулирующих ряд популярных операций нечеткой импликации: T-импликации Мамдани и Ларсена, S-импликации Заде, Лукашевича, Даэнса, Рейхенбаха, Ю и R-импликацию Геделя.

Группа *Accumulations* содержит четыре компонента, предназначенных для аккумуляции заключений продукционных правил посредством max-композиции. Компоненты различаются числом объединяемых термов (от 2 до 5) выходной лингвистической переменной.

Группа *Inference* содержит три компонента. Верхний компонент на рис. 1 — это функционально расширенный шаблон одноходовой таблицы данных (подстановки). В табличной модели FIS он хранит массив значений выходной лингвистической переменной  $lingvar(y_i + h/2), i = 1..n$ , вычисленных на универсуме  $Y$  с шагом дискретизации  $h = (ur - ul) / n$ , где  $n$  — число строк в таблице данных. Пользуясь этой таблицей, компонент выполняет также операцию дефазификации методами центра тяжести (Center of Gravity — CoG):

$$\bar{y} = \frac{\int_{\min}^{\max} y \mu(y) dy}{\int_{\min}^{\max} \mu(y) dy} \quad (1)$$

и центра площади (Center of Area — CoA):

$$\bar{y} = \frac{\int_{\min}^{\max} \mu(y) \cdot y dy}{\int_{\min}^{\max} \mu(y) dy} \quad (2)$$

Интегралы в формулах (1) и (2) вычисляются методом трапеций табличным способом, без применения программного кода VBA.

Второй и третий компоненты группы *Inference* выполняют операцию дефазификации в FIS, работающих по алгоритму Такаги—Сугено. Эти два компонента функционально равнозначны и отличаются друг от друга только реализацией.

Группа *Takagi-Sugeno* содержит один компонент, предназначенный для активизации продукционных правил в FIS, работающих по алгоритму Такаги—Сугено.

Наконец, группа *lingvar* содержит один компонент, представляющий собой шаблон одноходовой таблицы данных для хранения значений функций принадлежности  $k$  термов  $lingval(x_i)$  лингвистической переменной  $lingvar$ , вычисляемых на универсуме  $X$  с шагом дискретизации  $h$ . Данные этой таблицы используются для построения графиков функций принадлежности термов лингвистической переменной.

### Создание компонентных табличных моделей систем нечеткого вывода

Компонентная табличная модель FIS создается на базе файла шаблона *FIS\_Template.xlsm* и содержит два рабочих листа: *Info* и *FIS*.

Рабочий лист *Info* предназначен для постановки задачи нечеткого вывода, формализации и документирования создаваемой FIS. К структуре и форматированию этого листа не предъявляются никакие специальные требований. Он заполняется произвольно на этапе логического анализа задачи нечеткого вывода и общения разработчика FIS со специалистами и экспертами в предметной области.

Основной рабочий лист FIS шаблона содержит создаваемую табличную модель FIS и имеет структуру, показанную на рис. 2.

Строки 1—3 содержат название FIS и краткую информацию о ней (сведения о разработчике, дату разработки и др.); строки 4—17 содержат библиотеку компонентов нечеткого вывода, они размещены на втором, более низком уровне структуры рабочего листа и могут быть скрыты (и даже удалены, если не предполагается модификация созданной табличной модели FIS); в остальных строках рабочего листа размещается создаваемая модель FIS, далее эти строки будем называть *областью модели*.

Сама табличная модель FIS, кроме компонента дефазификации, располагается в столбцах A-I области модели, где столбцы B-H содержат компоненты модели, а столбцы A-I используются для дополнительных комментариев. Обычно столбцы A-I

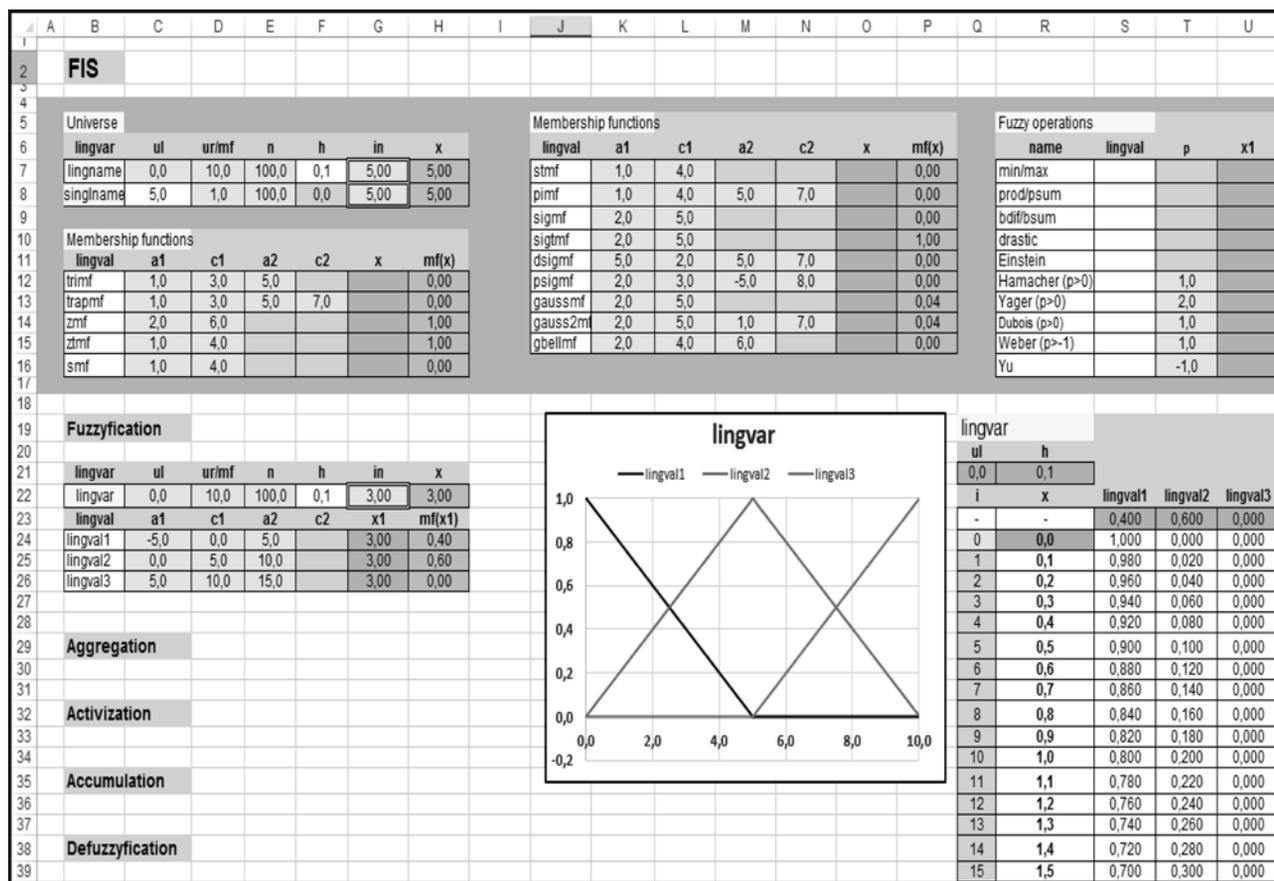


Рис. 2. Шаблон рабочего листа FIS

занимают левую половину экрана дисплея, поэтому в правой его половине, поверх столбцов J-P области модели размещают графики функций принадлежности термов лингвистических переменных.

Правее графиков функций принадлежности, начиная со столбца Q области модели, располагаются таблицы данных, используемые для построения графиков ФП и дефазификации результата нечеткого вывода. Быстрый переход на эти таблицы и обратно удобно осуществлять с помощью внутренних гиперссылок.

Технология визуального компонентного моделирования нечеткой логики тестировалась на трех FIS: *Tipper* [2] (алгоритмы Мамдани, Ларсена и Такаги—Сугено), *Pendulum* [8] (алгоритмы Мамдани и Такаги—Сугено), *Risk* [3] (алгоритм Мамдани).

Ниже мы будем, главным образом, опираться на FIS *Risk* по следующим причинам: 1) в ней используются не две, как обычно, а три входных лингвистических переменных; 2) база содержит небольшое число продукционных правил и является неполной; 3) в предпосылках правил, помимо нечетких опе-

раций AND и OR, используются модификаторы термов лингвистических переменных; 4) можно непосредственно сравнить нашу компонентную и эту табличные модели FIS по ряду критериев: трудоемкости создания, эффективности работы алгоритма, эргономичности и др.

Исходные данные для построения FIS *Risk* в ЭТ показаны на рис. 3, где *cost* — стоимость товаров,



Рис. 3. Рабочий лист Info табличной модели FIS Risk

Fuzzyfication						
lingvar	ul	ur	n	h	in	x
cost	0,0	5,0	100,0	0,1	1,91	1,9
lingval	a1	c1	a2	c2	x	mf(x)
low	-1,0	0,0	2,5		1,91	0,24
medium	0,0	2,5	5,0		1,91	0,76
high	2,5	5,0	6,0		1,91	0,00
lingvar	ul	ur	n	h	in	x
complex	0,0	10,0	100,0	0,1	7,37	7,4
lingval	a1	c1	a2	c2	x	mf(x)
low	-1,0	0,0	5,0		7,37	0,00
medium	1,0	5,0	7,0	8,0	7,37	0,63
high	5,0	10,0	12,0		7,37	0,47
lingvar	ul	ur	n	h	in	x
compens	0,0	10,0	100,0	0,1	7,92	7,9
lingval	a1	c1	a2	c2	x	mf(x)
low	-1,0	0,0	5,0		7,92	0,00
medium	1,5	5,0			7,92	0,15
high	5,0	10,0	12,0		7,92	0,58
lingvar	ul	ur	n	h	in	x
risk	0,0	10,0	100,0	0,1	7,00	7,0
lingval	a1	c1	a2	c2	y	mf(y)
low	-1,0	0,0	5,0		7,00	0,00
medium	3,0	1,5			7,00	0,38
high	5,0	10,0	12,0		7,00	0,40

Рис. 4. Фаззификация четких значений переменных модели:  
 cost=1.91, complex=7.37, compens=7.92, risk=7.00

проданных в прошедшем году благодаря нечестной рекламе; complex — сложность товара, мешающая покупателю самостоятельно оценить его качества; compens — размер комиссионных консультанту за проданный товар; risk — риск недостойного поведения консультанта по продажам.

Построение табличной модели FIS на рабочем листе FIS начинается с этапа фаззификации четких значений переменных. Это самый трудоемкий этап создания табличной модели FIS, занимающий большую часть времени разработки.

Для реализации этапа фаззификации требуемые компоненты из групп Universe и Membership functions библиотеки компонентов копируются и размещаются в столбцах В-Г области модели блоками, как показано на рис. 4, по блоку на каждую лингвистическую переменную. Затем в ячейки x компонентов Membership functions вводятся ссылки на четкое текущее значение соответствующей переменной in, уточняются параметры ФП нечетких множеств, редактируются комментарии компонентов.

Число интервалов дискретизации n универсумов лингвистических переменных (рис. 4) рекомендуется выбирать равным  $n = 100$ , чтобы, с одной стороны, таблицы данных, определяющие быстрдействие модели, были не слишком большими по размеру, с другой стороны, обеспечивалась приемлемая точность вывода графиков функций принадлежности и интегралов в формулах (1) и (2). Оценки показывают, что при  $n = 100$  и числе термов лингвистической переменной  $k \leq 7$  точность отображения графиков составляет около 5 %, а относительная ошибка значений интегралов — около 0,1 %.

В шаблоне табличной модели FIS (см. рис. 2) реализован этап фаззификации для абстрактной лингвистической переменной lingvar с тремя термами lingval1, lingval2, lingval3 и треугольными ФП. Для построения графиков ФП термов использовался компонент lingvar, который копировался из библиотеки компонентов в область данных, расширялся с помощью маркера заполнения Excel до трех столбцов и  $n + 1$  строк, а затем заполнялся данными (команда Excel Данные/Анализ 'что если'/Таблица данных...). На основе содержимого столбцов lingval1, lingval2, lingval3 этого компонента строились графики ФП термов лингвистической переменной lingvar шаблона.

Графики ФП термов табличной модели FIS Risk проще всего построить, используя реализованные в шаблоне графики ФП абстрактной переменной lingvar. Для этого таблица данных из столбцов Q-U копируется и трижды вставляется в область модели правее этих столбцов, затем копируется диаграмма и трижды вставляется в область модели ниже нее. Далее нужно настроить ссылки во входных ячейках полученных четырех таблиц данных и скорректировать ссылки на интервалы ячеек в рядах, выводимых на графиках.

Остальные этапы нечеткого вывода реализуются просто — нужно только скопировать из библиотеки, вставить в область модели и сгруппировать

Aggregation						
name	lingval	p	x1	x2	t-norm	s-norm
NOT	complex/low			=1-x	0,00	1,00
hedge	compens/high	CON, DIL	very	=fm*2	0,58	0,34
min/max	OR		1,0	0,34	0,34	1,00
NOT	cost/low			=1-x	0,24	0,76
min/max	AND		1,0	0,76	0,76	1,00
min/max	OR		0,5	0,00	0,00	0,47
min/max	AND		0,5	0,58	0,47	0,58
NOT	complex/high			=1-x	0,47	0,53
NOT	cost/high			=1-x	0,00	1,00
min/max	AND		0,5	1,00	0,53	1,00
NOT	compens/high			=1-x	0,58	0,42
min/max	AND		0,5	0,42	0,42	0,53
Activation						
name	lingval	comment	F	mf(x)	mf(y)	mf(x,y)
Mamdani	high	min	1,0	0,40	0,76	0,40
Mamdani	high	min	1,0	0,40	0,47	0,40
Mamdani	medium	min	1,0	0,38	0,42	0,38
Accumulation						
mf3	mf2	mf1	p	mf		
0,40	0,40	0,38	0	0,40		mf(tip)
Defuzzyfication						
CoG	CoA					
5,85	6,15					

Рис. 5. Табличная реализация правил нечеткого вывода (алгоритм Мамдани)

нужные компоненты, настроить параметры и ввести ссылки в их входные ячейки.

Табличная реализация этапов агрегирования, активизации и аккумуляции подусловий и условий нечетких правил для FIS Risk показана на рис. 5.

Обратим внимание на то, что дефаззификация нечетких данных осуществляется посредством компонента *Inference*, который размещается в области модели правее таблиц данных для ФП и с помощью маркера заполнения Excel расширяется до  $n$  строк данных. После настройки ссылок и заполнения таблицы этого компонента данными (команда Excel *Данные/Анализ 'что если'/Таблица данных...*) его выходные ячейки CoG и CoA содержат значения результата дефаззификации.

На рис. 5 в ячейках CoG и CoA отображаются копии этих значений.

### Результаты нечеткого моделирования в электронных таблицах

Анализ и тестирование табличной модели FIS из работы [3] показали, что в ячейках M20 и M23 ее рабочего листа *2.2 Misconduct Risk-Calc* содержатся ошибки. После их исправления эта и наша компонентная модели дали одинаковые результаты с точностью до 3-й значащей цифры (рис. 6).

Одним из недостатков табличной модели FIS [3] является то, что она работает в Excel, но не работает в LibreOffice, тогда как компонентная модель FIS работает в обеих электронных таблицах, однако наблюдается большая разница в ее быстроте действия в Excel и LibreOffice.

По нашему мнению, это различие обусловлено тем, что в Excel таблицы данных, автоматизирующие выполненные сценарии "что если...", реализованы с помощью формул массива, а в LibreOffice для этого используются совмещенные операции. Соответственно, вычисления в ячейках таблицы данных Excel выполняются по одинаковой формуле встроенным высокопроизводительным алгоритмом. В LibreOffice механизм выполнения сценариев "что если..." более гибкий — каждая ячейка таблицы данных может пересчитываться по независимой формуле. Платой за эту гибкость является то, что сценарии "что если..." в LibreOffice выполняются примерно в 10 раз медленнее, чем в Excel.

Так как в компонентной модели FIS Risk при каждом изменении входных данных автоматически пересчи-

тываются пять таблиц данных, то при  $n = 100$  любое изменение входов модели порождает пересчет около 1300 ячеек в этих таблицах (1200 ячеек — для графиков ФП, 100 ячеек — для операции дефаззификации). И если в Excel с табличной моделью Risk можно работать вполне комфортно, то в LibreOffice приходится ждать реакции системы на каждое действие пользователя в течение нескольких секунд.

К счастью, из этой неприятной ситуации есть простой выход: после настройки параметров функций принадлежности в ячейки таблиц данных, отвечающих за вывод графиков ФП, вместо формул можно вставить значения, вычисляемые по этим формулам. Это позволяет визуализировать графики ФП, как прежде, сократив при этом число пересчитываемых ячеек в таблицах данных с 1300 до 100.

Что касается практической значимости выводов, получаемых из табличной модели работы [3], то нужно заметить, что на рабочем листе *Disclaimer* спонсоры и разработчики пишут: "*The Joint Risk Management Section and the authors assume no responsibility for errors or omissions in the tool or related documentation. In no event shall the sponsor and the authors be liable to you or any third parties for any damages of any kind arising out of or in connection with the use of this tool*". Мы присоединяемся к этому мнению авторов, так

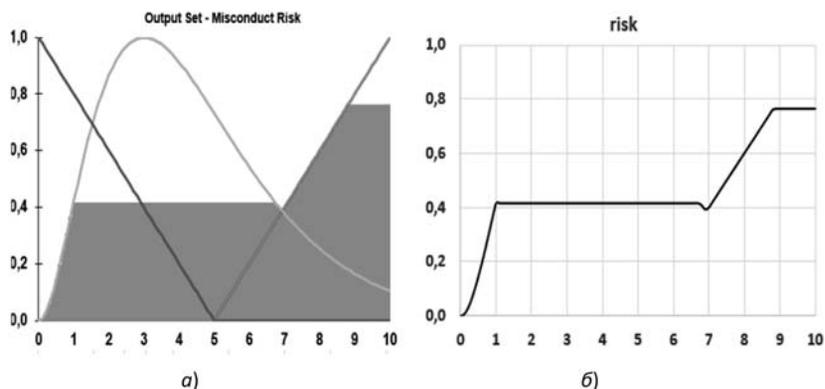


Рис. 6. Аккумуляция заключений для лингвистической переменной risk: а — табличная модель FIS из работы [4]; б — компонентная табличная модель FIS

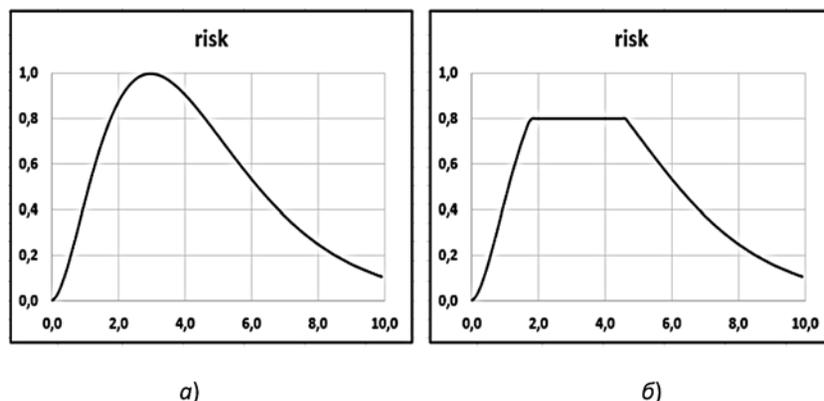


Рис. 7. Функция принадлежности выходной лингвистической переменной risk: а —  $cost \leq 2.5, complex = compens = 0$ ; б —  $cost = 3.0, complex = compens = 0$

Activation						
name	lingval	p	x1	x2	t-norm	s-norm
Yu	cheap	1,0	0,1	0,0	0,0	0,1
Yu	average	1,0	0,4	0,0	0,0	0,4
Yu	generous	1,0	0,5	1,0	0,5	1,0
Accumulation						
name	lingval	p	x1	x2	t-norm	s-norm
Yu	s-norm	1,0	0,0	0,0	0,0	0,0
Yu	s-norm	1,0	0,5	0,0	0,0	0,5
Defuzzyfication						
CoG	CoA					
20,53	24,30					

Рис. 8. Табличная реализация этапов активизации и аккумуляции для FIS *Tipper* с помощью параметрических t- и s-норм Yu

как считаем, что база продукционных правил данной FIS некорректна и нуждается в доработке.

Действительно, полагая  $complex=compens=0$  и увеличивая значения  $cost$  от 0 до 5, можно видеть, что ФП выходной переменной  $risk$  усекается по уровню все сильнее, начиная от графика, показанного на рис. 7, а, вплоть до значения  $risk=0$  при  $cost=5$ , что противоречит здравому смыслу. Такое же противоречие наблюдается и в двух других крайних случаях: 1)  $cost=0$ ,  $compens=0$ , изменяется  $complex$ , 2)  $cost=0$ ,  $complex=0$ , изменяется  $compens$ . При этом четкое значение CoG выход-

ной переменной, как и следует ожидать, возрастает, хотя и незначительно, с ростом значений  $cost$ .

Этот поучительный пример показывает важность наблюдения за поведением графика ФП выходной переменной для правильной интерпретации дефазифицированных результатов нечеткого вывода.

Помимо вывода двумерных графиков функций принадлежности мастер диаграмм Excel позволяет также строить трехмерные поверхности отображения декартового произведения двух входных переменных на универсум выходной лингвистической переменной (см. рис. 9). Однако FIS *Risk* имеет три входных переменных, поэтому построить поверхность отображения для нее невозможно.

Пожалуй, одной из самых привлекательных особенностей компонентного табличного моделирования в ЭТ является возможность креативного визуального экспериментирования с нечеткой моделью без программирования, что невозможно в GUI-интерфейсах Fuzzy Logic Toolbox и FuzzyR. Для примера на рис. 8, 9 показаны табличная реализация этапов активизации и аккумуляции для FIS *Tipper* [2] с использованием параметрических t- и s-норм Yu и поверхности отображения для четырех значений параметра  $p$ .

Видно, что наиболее близкий к алгоритму Мамдани результат получается при  $p = -1$ , тогда как при  $p < -1$  наблюдаются очень сильные отклонения поверхности отображения CoG (service, food) от интуитивно ожидаемой согласно базе правил нечеткого вывода поверхности.

## Заключение

Интерпретация табличной модели как действующей граф-машины алгоритма и предложенная авторами оригинальная технология компонентного табличного моделирования в сочетании с развитыми средствами форматирования ячеек (узлов графа алгоритма) делают электронные таблицы уникальной средой визуализации информационной структуры и поведения алгоритмов, в том числе и нечетких алгоритмов.

Разработанная легко расширяемая библиотека компонентов позволяет реализовывать и визуализировать в ЭТ алгоритмы и поведение систем нечеткого вывода без написания программного кода, по принципу "программирование без программирования".

Результаты экспериментальных исследований представляют интерес, как минимум, в двух обширных прикладных областях: 1) для программистов, разрабатывающих и изучающих новые алгоритмы и программы нечеткой логики; 2) в лабораторном практикуме по дис-

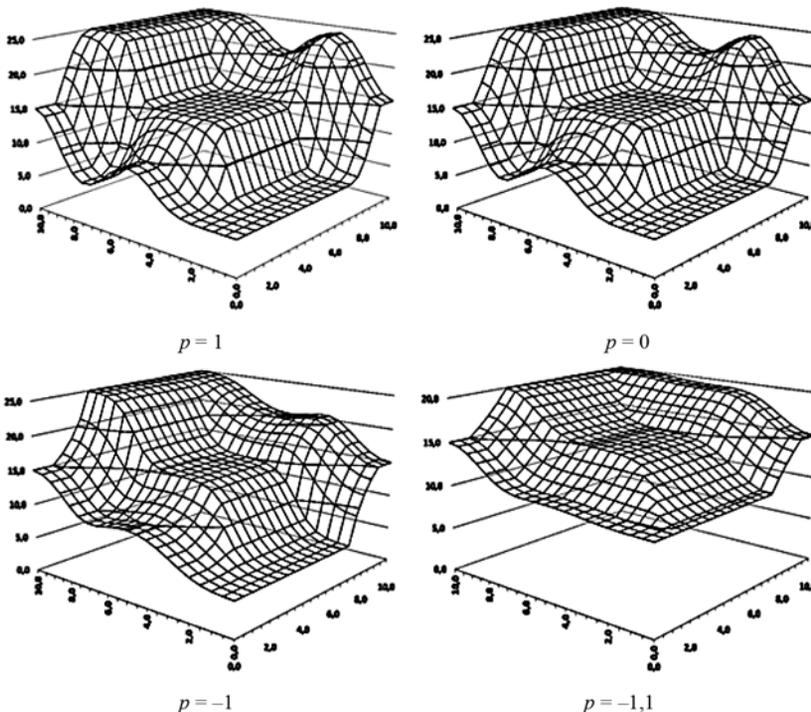


Рис. 9. Поверхности отображения CoG(service, food) для FIS *Tipper*: нечеткий вывод с помощью параметрических t- и s-норм Yu

циплинам искусственного интеллекта для визуализации теоретических закономерностей, алгоритмов и поведения систем нечеткой логики.

#### Список литературы

1. **Заде Л. А.** Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. С. 21—146.
2. **Fuzzy Logic Toolbox User's Guide: MATLAB R2012a.** The MathWorks Inc., 2012. P. 34—101.
3. **Shang K., Hossen Z.** Applying Fuzzy Logic to Risk Assessment and Decision-Making. Casualty Actuarial Society, Canadian Institute of Actuaries, 2013. P. 7—19.

4. **Аникин В. И., Воловач В. И., Сафронова К. В.** Компонентное моделирование формальной логики в Microsoft Excel // Сб. избр. трудов VIII Международной научно-практической конференции "Современные информационные технологии и ИТ-образование" / Под ред. проф. В. А. Сухомлина. М.: Изд-во МГУ, 2013. С. 281—289.

5. **Аникин В. И., Аникина О. В.** Моделирование и визуализация информационной структуры алгоритмов в Microsoft Excel // Математическое и программное обеспечение систем в промышленной и социальной сфере. 2014. № 1 (4). С. 21—28.

6. **Воеводин В. В., Воеводин Вл. В.** Параллельные вычисления. СПб.: БХВ-Петербург, 2002. С. 324—372.

7. **Леоненков А. В.** Нечеткое моделирование в среде MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2005. С. 33—220.

8. **Passino K. M., Yurkovich S.** Fuzzy Control. Addison Wesley, 1998. P. 24—51.

**V. I. Anikin**, D. Sc. Professor, e-mail: anikin\_vi@mail.ru,

Volga Region State University of Service,

**O. V. Anikina**, PhD, Assistant Professor, e-mail: blue-waterfall@yandex.ru,

Togliatti State University

## The Visual Component Modeling of Fuzzy Inference Systems in Spreadsheets

*It has been proved that any spreadsheet model is a graph — machine of some algorithm. The interpretation of the spreadsheet model as the algorithm graph-machine and the component modeling technology combined with the modern means of cells formatting do spreadsheets a unique environment for the visualization of algorithms structure, including fuzzy algorithms.*

*The library of components for the visual modeling of fuzzy logics without writing program code has been developed. The technology of component modeling of fuzzy logics has been tested on three fuzzy logics inference systems: Tipper (algorithms Mamdani, Larsen and Takagi—Sugeno), Pendulum (algorithms Mamdani and Takagi—Sugeno), Risk (algorithm Mamdani).*

*The presented results may find broad application in at least two areas: 1) for programmers developing new fuzzy logic algorithms and programs, 2) in laboratory practice on the disciplines of artificial intelligence to visualize the fuzzy logic theoretical bases, algorithms and systems.*

**Keywords:** fuzzy logic, spreadsheets, component modeling, algorithm's graph-machine, algorithms visualization

#### References

1. **Zade L. A.** *Ponjatie lingvisticheskoj peremennoj i ego primenenie k prinjatiju priblizhennyh reshenij* (The Concept of a Linguistic Variable and its Application to Approximate Reasoning), Moscow, Mir, 1976, pp. 21—146 (in Russian).
2. **Fuzzy Logic Toolbox User's Guide: MATLAB R2012a.** The MathWorks Inc., 2012, pp. 34—101.
3. **Shang K., Hossen Z.** Applying Fuzzy Logic to Risk Assessment and Decision-Making. — Casualty Actuarial Society, Canadian Institute of Actuaries, 2013, pp. 7—19 (in Russian).
4. **Anikin V. I., Volovach V. I., Safronova K. V.** Komponentnoe modelirovanie formalnoj logiki v Microsoft Excel (Component Modeling of a Formal Logic in Microsoft Excel) // *Sb. isbr. trudov VIII Mezh-dunarodnoj nauchno-prakticheskoj konferencii "Sovremennye informa-*

*cionnye tehnologii i IT-obrazovanie"* / Pod red. prof. V. A. Suhomlina, Moscow, MGU, 2013, pp. 281—289 (in Russian).

5. **Anikin V. I., Anikina O. V.** Modelirovanie i vizualizacija informacionnoj struktury algoritmov v Microsoft Excel (Modeling and Visualization of Algorithms' Information Structure in Microsoft Excel), *Matematicheskoe i programnoe obespechenie sistem v promyshlennoj i socialnoj sfere*, 2014, no. 1 (4), pp. 21—28 (in Russian).

6. **Voevodin V. V., Voevodin Vl. V.** *Parallelnye vychislenija* (Parallel computing). Sankt-Peterburg, BHV-Peterburg, 2002, pp. 324—372 (in Russian).

7. **Leonenkov A. V.** *Nechetkoe modelirovanie v MATLAB i fuzzyTECH* (Fuzzy Modeling in MATLAB and fuzzyTECH), Sankt-Peterburg, BHV-Peterburg, 2005, pp. 33—220 (in Russian).

8. **Passino K. M., Yurkovich S.** *Fuzzy Control*, Addison Wesley, 1998, pp. 24—51.

О. А. Соснина, канд. техн. наук, доц., e-mail: o.a.sosnina@mail.ru,

А. Д. Филинских, канд. техн. наук, доц., e-mail: alexfil@yandex.ru,

А. С. Коротаева, магистрант, e-mail: korotaeva.as@gmail.com,

Нижегородский государственный технический университет им. П. Е. Алексеева

## Сравнение способов создания низкополигональной 3D-модели

*Рассматривается вопрос создания низкополигональной модели, которая используется при разработке компьютерных видеоигр и анимации, где требуется экономия ресурсов компьютера и нет необходимости в высокой детализация трехмерного объекта. Для решения проблемы скорости моделирования и обеспечения точности построения рассмотрены различные способы моделирования, их преимущества и недостатки. Предложен способ, который упрощает и ускоряет процесс создания трехмерной низкополигональной модели на основе существующей физической модели. На примере создания трехмерной модели персонажа проведено практическое сравнение моделирования с использованием и без использования вспомогательной подложки, в результате которого было найдено эффективное решение проблемы создания низкополигональных 3D-моделей.*

**Ключевые слова:** низкополигональная модель, трехмерное моделирование, фотограмметрия, трехмерное сканирование, способы моделирования

### Введение

В настоящее время стремительно развивается область разработки компьютерных видеоигр и создания трехмерной анимации и визуализации. В таких сферах зачастую необходимо создавать трехмерные низкополигональные модели на основе существующей физической модели. На обработку низкополигональных моделей требуется меньше ресурсов компьютера, чем на обработку высокополигональной модели [1, 2]. Каждая 3D-модель состоит из отдельных полигонов, а группа полигонов образует полигональную сетку. В процессе моделирования важно помнить о топологии модели или о том, как именно полигоны формируют 3D-модель. Например, для создания такой модели лица для анимации необходимо, чтобы линии полигональной сетки соответствовали лицевым мышцам, т. е. требуется создать правильную топологию. Корректную полигональную сетку можно создать только в процессе построения низкополигональной модели. Далее из низкополигональной модели создают высокополигональную. В большинстве программных продуктов для трехмерного моделирования есть инструменты, с помощью которых можно легко сделать подобное преобразование. Применение таких инструментов увеличивает число полигонов трехмерной модели. Также существуют инструменты, которые, наоборот, позволяют преобразовать высокополигональную модель в низкополигональную, однако они не всегда помогают получить желаемую полигональную сетку, поэтому такой метод редко используется. Кроме того, в области трехмерного моделирования существует проблема скорости моделирования и обеспечения точности низкополигональной модели. Существует несколько

способов создания низкополигональной модели. Из них следует выбрать оптимальный, который способствовал бы решению проблемы скорости и точности моделирования [3, 4].

### 1. Постановка задачи исследования и метод решения задачи

Необходимо осуществить практическое сравнение разных способов низкополигонального моделирования на основе существующей физической модели и выбрать оптимальный вариант.

Чтобы провести практическое сравнение способов моделирования, была создана низкополигональная модель персонажа. Для его трехмерного низкополигонального моделирования использовали программу "3ds Max" компании "Autodesk" — полнофункциональную профессиональную программную среду для редактирования и создания трехмерной графики и анимации. Для трехмерной оцифровки использовали программу "ReMake" компании "Autodesk". Фотоматериалы будущей модели были получены с помощью цифрового фотоаппарата "Canon-500d". Для получения еще одной высокополигональной 3D-модели использовали программу для цифрового скульпинга "ZBrush" [10–12].

### 2. Описание полученных результатов исследования

Используют два основных способа создания низкополигональной модели: моделирование без вспомогательной подложки и моделирование с ее использованием. Моделирование с использованием вспомогательной подложки возможно в двух вариантах. Первый вариант предполагает использование 2D-подложки, которая представляет собой

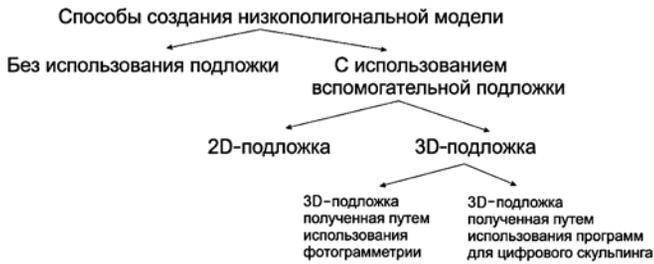


Рис. 1. Способы создания низкополигональной модели

две перпендикулярные друг другу плоскости с эскизами будущей модели. Второй вариант заключается в использовании 3D-подложки, являющей собой высокополигональную трехмерную модель, которая была получена либо путем использования фотограмметрического сканирования, либо программ для цифрового скульпинга (рис. 1) [13, 15].

Фотограмметрическое сканирование является одним из направлений технологии фотограмметрии. Оно заключается в создании группы фотографий объекта сканирования с различных точек и ракурсов. Далее с помощью специальных программ, предназначенных для трехмерной оцифровки, на основе полученных изображений создается трехмерная модель этого объекта. Таким образом, мы имеем возможность получить высокополигональный объект, который в дальнейшем можно использовать как вспомогательную 3D-подложку для моделирования низкополигональной модели. Способы и технические средства для получения фотоматериалов могут быть разными, что зависит от поставленной цели. Для получения фотоснимков небольшого объекта можно обойтись обычной камерой телефона, а для съемки объектов большой площади нужно использовать специальные средства (например, беспилотный летательный аппарат с установленной на нем камерой) [5–9].

**1. Создание низкополигональной модели без использования подложки.** Такой способ моделирования часто используется для создания трехмерных моделей простых форм. Для моделирования объектов сложных форм, таких как персонаж, необходимо использовать вспомогательные подложки. В противном случае точность модели будет на минимальном уровне, а время создания значительным.

**2. Создание низкополигональной модели с использованием подложки.**

2.1. Применение 2D-подложки. Это наиболее часто используемый метод моделирования трехмерных объектов. Его суть заключается в том, что перед началом моделирования объекта создают две плоскости, перпендикулярные друг другу (рис. 2). Каждая плоскость содержит изображение или эскиз будущей модели в анфас (вид спереди) и

в профиль (вид сбоку). Далее начинается процесс моделирования, и плоскости с двумя видами способствуют правильному созданию персонажа. Каждый трехмерный объект состоит из совокупности точек, ребер и полигонов. Изначально создается один полигон, далее он копируется со смещением и таким образом строится каркас персонажа в анфас и в профиль (рис. 3). После этого создается общая форма, и далее прорабатываются мелкие детали. При этом на каждом шаге необходимо контролировать положение точек объекта в пространстве, с тем чтобы конечная форма была максимально



Рис. 2. Вспомогательные плоскости для трехмерного моделирования персонажа в программе "3ds Max"



Рис. 3. Моделирование персонажа в программе "3ds Max"

близка к эскизу модели на плоскостях. Кроме того, в процессе моделирования важно помнить о топологии модели или о том, как именно полигоны формируют 3D-модель.

**Вывод:** моделирование с использованием вспомогательных плоскостей — это наиболее часто используемый метод моделирования. Он не требует знания каких-либо дополнительных программ, кроме программ для трехмерного моделирования. Тем не менее, применяя такой способ создания трехмерных моделей, не всегда получается добиться точного сходства с эскизом, особенно на начальном уровне знаний программы. Кроме того, чтобы добиться максимальной точности, потребуются большие навыки в области трехмерного моделирования.

## 2.2. Применение 3D-подложки.

**3D-подложка, полученная путем использования программ для цифрового скульптинга.** В программах для цифрового скульптинга с помощью специальных инструментов можно проводить различные манипуляции с 3D-моделями. В таких программах редактирование трехмерной модели можно сравнить с работой скульптора над обычной глиной. Их особенность заключается в том, что они позволяют создавать модели высокой детализации, что пока еще недостижимо традиционными методами 3D-моделирования. В результате получаются фотореалистичные

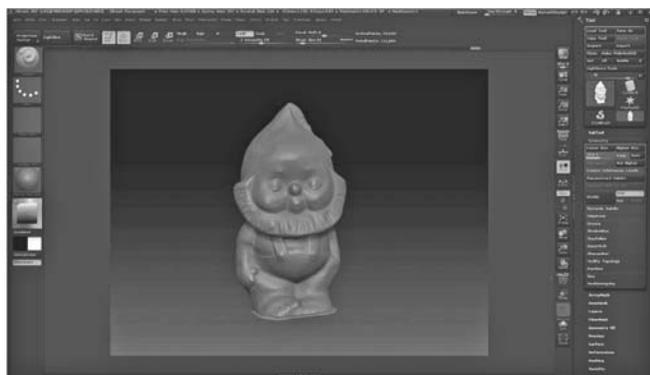


Рис. 4. Модель персонажа в программе "ZBrush"



Рис. 5. Создание низкополигональной модели персонажа в программе "3ds Max"

тичные трехмерные сцены и модели. В настоящее время программы для цифрового скульптинга используют в связке с пакетами низкополигонального моделирования. В программах скульптинга низкополигональная модель улучшается и усложняется за счет создания карт неровностей различного рода. Используя текстурные карты и карты нормалей, созданные в подобной программе, можно улучшить вид модели, добиться высокой степени реализма и при этом обеспечить экономию ресурсов компьютера. Суть данного способа заключается в том, что в программе для скульптинга сначала моделируется высокополигональная модель (рис. 4), далее эта модель экспортируется в программы для трехмерного моделирования, например "3ds Max", и наконец, с помощью ретопологии, т. е. создания топологии заново, на основе уже готовой высокополигональной формы создается низкополигональная сетка (рис. 5). В программах для трехмерного моделирования существуют инструменты, которые позволяют при создании новой низкополигональной модели привязываться к высокополигональной модели. Благодаря этому процесс моделирования упрощается, а конечная модель получается более точной [14].

**Вывод:** низкополигональное моделирование с использованием вспомогательной 3D-подложки или высокополигональной модели, созданной в программе для 3D-скульптинга, — распространенный способ создания трехмерных объектов. Он упрощает процесс моделирования низкополигональной модели, которая будет максимально схожа с высокополигональной моделью, созданной в программе скульптинга. Однако такой метод требует профессионального владения не только пакетами низкополигонального и высокополигонального моделирования, но и программой цифрового скульптинга. Освоение подобных программ — непростой и достаточно длительный процесс. Таким образом, используя данный способ моделирования, можно получить максимально точную низкополигональную модель, но процесс ее создания будет длительным, с учетом временных затрат на освоение программы для 3D-скульптинга и времени моделирования высокополигональной модели.

**3D-подложка, полученная путем использования фотограмметрии.** Данный способ моделирования новый по сравнению с предыдущими вариантами создания низкополигонального объекта. Он схож с вариантом использования вспомогательной высокополигональной модели, созданной в программе цифрового скульптинга. Суть данной технологии заключается в том, что создается группа фотографий моделируемого объекта с разных точек и ракурсов. В нашем случае было необходимо получить модель персонажа гнома. Для эксперимента использовали 43 фотографии. Все фотографии отправля-



Рис. 6. Трехмерная модель персонажа в программе "ReMake"

ли на обработку в программу "ReMake", где создавалась трехмерная модель по входным фотоизображениям. В результате получили трехмерную модель (рис. 6), которую можно экспортировать в программы для традиционного моделирования либо в низкополигональном, либо в высокополигональном виде (рис. 7, см. вторую сторону обложки). Как правило, экспортируется высокополигональная модель, так как она более детализирована и приближена к оригинальной модели. В определенных типах проектов подобные модели можно использовать и без редактирования полигональной сетки. Однако чаще возникает необходимость ее обработать (рис. 8, см. вторую сторону обложки). Процесс создания низкополигональной модели идентичен процессу создания модели в методе с использованием программ цифрового скульппинга. Кроме того, с использованием инструментов ретопологии в программе для традиционного моделирования, например, "3ds Max", создается низкополигональная модель персонажа (рис. 9, см. вторую сторону обложки).

**Вывод:** способ использования фотограмметрии в низкополигональном моделировании наиболее удобен. Во-первых, программа для трехмерной оцифровки фотоматериалов проста в изучении, поэтому ее могут легко освоить даже не продвинутые в области трехмерной графики пользователи. Чтобы получить высокополигональную модель, используемую в качестве 3D-подложки, необязательно тратить время на изучение программы для цифрового скульппинга. Полученную высокополигональную модель можно использовать в процессе низкополигонального моделирования для упрощения процесса и получения близкой к оригиналу модели.

## Заключение

Моделирование без использования вспомогательной подложки применимо в создании моделей простых геометрических форм или в области свободного моделирования без эскизов, когда окончательная 3D-модель является результатом творческого воображения художника. Моделирование с использованием 2D-подложки — это наиболее распространенный и универсальный метод моделирования. При таком способе в качестве вспомогательного инструмента используется 2D-эскиз будущего 3D-объекта. Эскизом может быть и фотография реального объекта, и чертеж детали, и набросок будущей 3D-модели. Данный метод применим для моделирования как существующих физических моделей, так и вымышленных. Моделирование с 3D-подложкой, полученной с использованием программ для цифрового скульппинга, применяют для моделирования реальных и вымышленных объектов. Как правило, такой метод используют в тех случаях, когда требуется наличие детализированной высокополигональной модели. Моделирование с 3D-подложкой, полученной путем использования фотограмметрии, возможно только для создания 3D-объекта на основе существующей физической модели.

На примере создания трехмерной модели персонажа было осуществлено практическое сравнение разных способов моделирования на основе существующей физической модели. В результате проведенной работы было выявлено, что при использовании фотограмметрии в области низкополигонального моделирования на основе существующей физической модели время на получение трехмерной модели с заданной точностью геометрических форм сократилось примерно на 40 % в сравнении с альтернативными способами моделирования (рис. 10, см. вторую сторону обложки).

## Список литературы

1. **Pavlovskis K.** What is Low Poly? URL: <http://electrocactus.com/index.php/2015/10/04/what-is-low-poly/> (дата обращения: 16.01.2017).
2. **Никонов В.** Что такое "low-poly"? URL: <http://nikonov.info/blog/low-poly/> (дата обращения: 16.01.2017).
3. **Бояршинов А., Ромин Е.** Методы оптимизации высокополигональных 3D-моделей. URL: [http://brainy.pro/blog-en/52-optimization\\_3d](http://brainy.pro/blog-en/52-optimization_3d) (дата обращения: 16.01.2017).
4. **Оптимизация 3D-моделей в "3ds Max" / 3d-Freedom.** URL: <http://3d-freedom.ru/index.php/2009-07-21-14-20-27/98-3d-max> (дата обращения: 16.01.2016).
5. **Назаров А. С.** Фотограмметрия: учеб. пособие для студентов вузов. Минск: "ТетраСистемс", 2006. 368 с.
6. **Карманов А. Г.** Фотограмметрия. Системные требования: AdobeAcrobatReader. URL: [http://open.ifmo.ru/images/6/6c/145531\\_photogrammetria.pdf](http://open.ifmo.ru/images/6/6c/145531_photogrammetria.pdf) (дата обращения: 16.01.2017).
7. **Walford A.** What is Photogrammetry? URL: <http://www.photogrammetry.com/> (дата обращения: 16.01.2017).

8. **Как выбрать** подходящую технологию 3D-сканирования? URL: <http://top3dshop.ru/blog/tehnologii-3d-skanirovaniya/> (дата обращения: 16.01.2017).

9. **Краснопецев Б. В.** Фотограмметрия. URL: <http://www.miigaik.ru/library.miigaik.ru/uchebnieposobiya/20160301150336-5998.pdf> (дата обращения: 16.01.2017).

10. **Autodesk Remake.** Официальный сайт компании "Autodesk". URL: <http://remake.autodesk.com/about> (дата обращения: 16.01.2017).

11. **ZBrush 4R7.** Официальный сайт компании "Pixologic". URL: <http://pixologic.com/zbrush/features/ZBrush4R7/> (дата обращения: 16.01.2017).

12. **3ds Max:** Программное обеспечение для 3D-моделирования, анимации и визуализации. Официальный сайт компании "Autodesk". URL: <http://www.autodesk.ru/products/3ds-max/overview> (дата обращения: 16.01.2017).

13. **Мэрдок К. Л.** Autodesk 3ds Max 2013. Библия пользователя. М.: Вильямс, 2013. 816 с.

14. **Слаква А.** Моделирование и скульптинг. URL: <http://blender3d.com.ua/tsifrovoy-i-traditsionnyy-skulpting/> (дата обращения: 16.01.2017).

15. **Горелик А. Г.** Самоучитель "3ds Max-2016". СПб.: "БХВ — Петербург", 2016. 528 с.

**О. А. Sosnina**, Assistant Professor, e-mail: o.a.sosnina@mail.ru,  
**А. Д. Filinskikh**, Assistant Professor, e-mail: alexfil@yandex.ru,  
**А. С. Korotaeva**, Master's Student, e-mail: korotaeva.as@gmail.com,  
Nizhny Novgorod State Technical University n.a. R. E. Alekseev

## Comparing Approaches to Creating a Low Polygonal 3D Model

*Considered in the article is the issue of creating low polygonal models, which are used in the development and animation of video games, when it is essential to save computer resources and when a detailed elaboration of the 3D object is not required. To address the issue of simulation performance and to ensure accuracy in modelling, the article studies diverse approaches to creating models and singles out their advantages and demerits. Proposed in the article is an approach that facilitates and expedites creating a low polygonal 3D model on the basis of an existing physical model. As exemplified by the creation of a fairy-tale character's model, the article makes a practical comparison of modelling with and without the use of auxiliary underlying planes carrying images. As a result, an effective way to create low polygonal 3D models is suggested.*

**Key words:** low polygonal model (low poly), 3D modelling, photogrammetry, 3D scanning, methods of modeling

### References

1. **Pavlovskis K.** What is Low Poly? URL: <http://electrocactus.com/index.php/2015/10/04/what-is-low-poly/> (date of access: 16.01.2017).

2. **Nikonov V.** Chto takoe chto takoe nizkopolygonal'noe modelirovanie (What is "low-poly"?). URL: <http://nikonov.info/blog/low-poly/> (date of access: 16.01.2017) (in Russian).

3. **Bojarshinov A., Romin E.** Metody optimizacii vysokopoligonal'nyh 3D modelej (Optimization methods "high-poly" 3D-models). URL: [http://brainy.pro/blog-en/52-optimization\\_3d](http://brainy.pro/blog-en/52-optimization_3d) (date of access: 16.01.2017) (in Russian).

4. **Optimizacija** 3D modelej v 3ds Max (Optimization 3D-models in "3ds Max") / 3D-Freedom. URL: <http://3d-freedom.ru/index.php/2009-07-21-14-20-27/98-3d-max> (date of access: 16.01.2016) (in Russian).

5. **Nazarov A. S.** Fotogrammetrija: Uchebnoe posobie dlja studentov vuzov (Photogrammetry: A manual for university students.) Minsk: TetraSistems, 2006, 368 pp. (in Russian).

6. **Karmanov A. G.** Fotogrammetrija (Photogrammetry). URL: [http://open.ifmo.ru/images/6/6c/145531\\_photogrammetria.pdf](http://open.ifmo.ru/images/6/6c/145531_photogrammetria.pdf) (date of access: 16.01.2017) (in Russian).

7. **Walford A.** What is Photogrammetry? URL: <http://www.photogrammetry.com/> (дата обращения: 16.01.2017).

8. **Как vybrat'** podhodjashhuju tehnologiju 3D skanirova? (How to choose the right technology 3D-scanning?). URL: <http://top3dshop.ru/blog/tehnologii-3d-skanirovaniya/> (date of access: 16.01.2017) (in Russian).

9. **Krasnopecev B. V.** Fotogrammetrija (Photogrammetry). URL: <http://www.miigaik.ru/library.miigaik.ru/uchebnieposobiya/20160301150336-5998.pdf> (date of access: 16.01.2017) (in Russian).

10. **Autodesk Remake** Oficial'nyj sajt kompanii (Autodesk Remake / Official website "Autodesk"). URL: <http://remake.autodesk.com/about> (date of access: 16.01.2017).

11. **ZBrush 4R7** / Oficial'nyj sajt kompanii "Pixologic" (ZBrush 4R7, Official website "Pixologic"). URL: <http://pixologic.com/zbrush/features/ZBrush4R7/> (date of access: 16.01.2017).

12. **3ds Max:** Programmnoe obespechenie dlja 3D modelirovaniya, animacii i vizualizacii, Oficial'nyj sajt kompanii "Autodesk" (3ds Max: Software for 3D-modeling, animation and rendering / Official website "Autodesk"). URL: <http://www.autodesk.ru/products/3ds-max/overview> (date of access: 16.01.2017).

13. **Mjerdok K. L.** Autodesk 3ds Max 2013. Biblija pol'zovatelja (Autodesk 3ds Max 2013. Bible user.). Vil'jams, 2013, 816 p. (in Russian).

14. **Slakva A.** Modelirovanie i skul'ping (Modeling and sculpting). URL: <http://blender3d.com.ua/tsifrovoy-i-traditsionnyy-skulpting/> (date of access: 16.01.2017) (in Russian).

15. **Gorelik A. G.** Samouchitel' "3ds Max-2016" (Tutorial "3ds Max-2016"), Sankt-Peterburg: BHV — St. Petersburg, 2016, 528 p. (in Russian).

С. А. Инютин, д-р техн. наук, профессор, inyutin\_int@mail.ru  
Московский авиационный институт (национальный исследовательский университет) (МАИ)

## Метод вычисления характеристики отношения порядка для параллельных форматов данных

*Разработан итерационный метод вычисления количественной характеристики отношения порядка для параллельных компьютерных форматов данных в модулярных реконфигурируемых вычислительных системах. Количественная характеристика предназначена для выполнения отдельных операций в параллельной компьютерной арифметике. Итерационный метод позволяет получить характеристику с линейной или логарифмической сложностью алгоритма.*

**Ключевые слова:** многопроцессорные реконфигурируемые системы SIMD-архитектуры, вычислительный процесс, линейная сложность вычисления, параллельные компьютерные форматы, числовая количественная характеристика

### Введение

Известно биективное отображение целых числовых величин из некоторого числового подмножества на множество векторов модулярного представления, лежащих в основе параллельных компьютерных форматов данных [1]:

$$A \in \left[ 0, \dots, \prod_{i=1}^n p_i \right) \leftrightarrow (\beta_1, \beta_2, \dots, \beta_n),$$

где  $\beta_i = |A|_{p_i}$  — наименьший неотрицательный вычет по модулю  $p_i$  целой числовой величины

$$A \in \left[ 0, \dots, \prod_{i=1}^n p_i \right), \text{ при условиях: } \forall i, j = 1, \dots, n, \text{ } NOD(p_i, p_j) = 1.$$

Целые простые (взаимно-простые) модули  $p_i$  называются основаниями модулярной (параллельной) системы счисления.

Из векторного модулярного представления числовых величин взаимно-однозначно получим нормированное модулярное представление:

$$A \in \left[ 0, \dots, \prod_{i=1}^n p_i \right) \leftrightarrow (\beta_1, \beta_2, \dots, \beta_n) \leftrightarrow (\alpha_1, \alpha_2, \dots, \alpha_n), \quad (1)$$

где компоненты модулярного вектора

$$\alpha_i = \left| \beta_i \left| \frac{1}{\prod_{j=1}^n p_j} \right|^{-1} \right|_{p_i} < p_i$$

Для нормированного модулярного представления В. М. Амербаевым введена функция от компо-

нент вектора — нормированный ранг, необходимый для выполнения ряда операций над модулярными компьютерными форматами данных [1]:

$$Z_A^{norm} = \left[ \sum_{i=1}^n \frac{\alpha_i}{p_i} \right]. \quad (2)$$

Для точного вычисления нормированного ранга вектора (1) необходим компьютерный диапазон  $\prod_{i=1}^n p_i$ , значительно превосходящий типовой машинный при больших значениях  $n$  и  $p_i$ , что приводит к большой сложности вычисления функции.

### 1. Свойства нормированного ранга

Для выявления свойств нормированного ранга (2) выполним преобразования, используя в качестве множителя один из модулей  $p_n$  модулярной системы [3]:

$$\begin{aligned} Z_A^{norm} &= \left[ \sum_{i=1}^n \frac{\alpha_i}{p_i} \right] = \left[ \frac{1}{p_n} \left( \sum_{i=1}^n \left[ \frac{\alpha_i p_n}{p_i} \right] + \sum_{i=1}^n \frac{|\alpha_i p_n|_{p_i}}{p_i} \right) \right] = \\ &= \left[ \frac{1}{p_n} \sum_{i=1}^n \left[ \frac{\alpha_i p_n}{p_i} \right] + \frac{1}{p_n} \sum_{i=1}^n \frac{|\alpha_i p_n|_{p_i}}{p_i} \right] = \left[ U + \frac{p_n - \omega}{p_n} + \frac{\mu}{p_n} \right], \quad (3) \end{aligned}$$

где  $U, \omega$  — целые;  $\mu$  — рациональное числа,

$$\begin{aligned} 0 \leq U &= \left[ \frac{1}{p_n} \sum_{i=1}^n \left[ \frac{\alpha_i p_n}{p_i} \right] \right] < n, \quad \omega = p_n - \left\lfloor \sum_{i=1}^n \left[ \frac{\alpha_i p_n}{p_i} \right] \right\rfloor_{p_n}, \\ 0 < \mu &= \sum_{i=1}^{n-1} \frac{|\alpha_i p_n|_{p_i}}{p_i} \leq n - 2. \end{aligned}$$

## Доказательство

*Замечание 1.* Нормированный ранг зависит от разности  $\mu - \omega$ . Если  $\mu - \omega < 0$ , то  $Z_A^{norm} = \left[ \sum_{i=1}^n \frac{\alpha_i}{p_i} \right] = U$ .

Если  $\mu - \omega \geq 0$ , то  $Z_A^{norm} = \left[ \sum_{i=1}^n \frac{\alpha_i}{p_i} \right] = U + 1$ .

*Замечание 2.* На нормированный ранг  $Z_A^{norm} = \left[ \sum_{i=1}^n \frac{\alpha_i}{p_i} \right] = U = n - 1$  разность  $\mu - \omega$  не влияет.

*Замечание 3.* Нормированный ранг  $Z_A^{norm} = \left[ \sum_{i=1}^n \frac{\alpha_i}{p_i} \right] = U$  при условии  $\omega > n - 1$ , так как  $\mu \leq n - 1$ . Следовательно, в процессе уточнения при вычислении нормированного ранга выполнение неравенства  $\omega > n - 1$  является критерием его завершения.

*Замечание 4.* Неравенство  $\omega \leq n - 1$  свидетельствует о необходимости процесса уточнения при вычислении нормированного ранга.

Актуальной является разработка алгоритма вычисления нормированного ранга, являющегося функцией от значений компонент модулярного представления числовой величины, имеющей линейную (или логарифмическую) сложность вычисления [2, 4].

Такая сложность вычисления нормированного ранга достигается, если для отдельных компонент модулярного представления получены разложения в конечные регулярные суммы. Это соответствует самой сути модулярной арифметики, в основе которой лежит идея покомпонентного распараллеливания машинных операций над данными в модулярных форматах данных. Разложения должны быть получены параллельно и независимо для каждого основания модулярной системы, задающего вычислительный тракт для операций по модулю в модулярном процессоре. Поэлементное суммирование слагаемых регулярных сумм позволяет получить линейную (или логарифмическую) сложность всего процесса вычисления нормированного ранга.

## 2. Основы итерационного алгоритма

Введем процесс итерационного уточнения при вычислении нормированного ранга и на примере одиночной компоненты модулярного представления получим необходимые соотношения.

*Лемма.* Для произвольной правильной дроби выполняется равенство

$$\frac{\alpha_i}{p_i} = \frac{\left(\frac{k_i}{r_i}\right)}{p_n} + \frac{1}{r_i p_1} = \frac{\left[\frac{k_i}{r_i}\right] + \frac{|k_i|_{r_i}}{r_i}}{p_n} + \frac{\xi_i}{p_n},$$

где целые числа  $k_i < p_n \alpha_i$ ,  $r_i < p_i$ ; рациональное число  $\xi_i < 1$ ;  $\forall i = 1, \dots, n$ .

Для взаимно-простых целых чисел  $a, p$ ,  $NOD(a, p) = 1$  выполняются диофантовы уравнения:  $ax - py = 1$  или  $a(p - x) - p(a - y) = 1$ . Следовательно, выполняются соотношения  $\frac{a}{p} = \frac{y}{x} + \frac{1}{xp}$

$$\text{или } \frac{a}{p} = \frac{a-y}{p-x} + \frac{1}{p(p-x)}.$$

Переходя к компонентам нормированного модулярного представления, получим следующие соотношения:

$$\text{для первого уравнения } r_i p_n \alpha_i - k_i p_i = 1$$

$$\text{или } \frac{\alpha_i}{p_i} - \frac{k_i}{r_i p_n} = \frac{1}{r_i p_n p_i}, \quad NOD(\alpha_i, p_n, p_i) = 1$$

$$\text{получим } \frac{\alpha_i}{p_i} = \frac{k_i}{r_i p_n} + \frac{1}{r_i p_n p_i};$$

$$\text{для второго уравнения } r_i p_n \alpha_i - k_i p_i = -1,$$

$$\frac{\alpha_i}{p_i} - \frac{k_i}{r_i p_n} = -\frac{1}{r_i p_n p_i}, \quad NOD(p_n, p_i) = 1,$$

$$\text{получим } \frac{\alpha_i}{p_i} = \frac{1}{p_n} \frac{(p_n \alpha_i - k_i)}{(p_i - r_i)} + \frac{1}{p_n} \frac{1}{(p_i - r_i) p_i}.$$

В полученных соотношениях выполняются неравенства  $k_i < p_n \alpha_i$ ,  $r_i < p_i$ .

*Следствие 1.* Для вторых слагаемых полученных равенств при  $p_i > n$  выполняются строгие неравенства

$$\frac{n}{r_i p_i p_n} < \frac{1}{r_i p_n} \quad \text{или} \quad \frac{n}{(p_i - r_i) p_i p_n} < \frac{1}{(p_i - r_i) p_n}.$$

Полученные неравенства позволяют пренебречь отдельными слагаемыми в процессе итерационного вычисления нормированного ранга.

## 3. Итерационный алгоритм

Построим итерационный процесс вычисления нормированного ранга на основе соотношений, установленных в лемме, и используя принципы модулярных рекурсивных вычислений [5]. Для первого этапа

итерационного процесса положим  $\frac{\alpha_i}{p_i} = \frac{\alpha_i}{p_i}$  и выпишем соотношения для следующих этапов:

$$\frac{\alpha_i^{j+1}}{p_i^{j+1}} = \frac{|k_i^j|_{r_i^j}}{r_i^j} \quad \text{или} \quad \frac{\alpha_i^{j+1}}{p_i^{j+1}} = \frac{|p_n \alpha_i^j - k_i^j|_{p_i^j - r_i^j}}{p_i^j - r_i^j} = \frac{|\tilde{k}_i^j|_{r_i^j}}{\tilde{r}_i^j},$$

$$j = 1, \dots, v.$$

Знаменатели дробей составляют монотонно убывающую, ограниченную последовательность  $p_i = p_i^1 > r_i^1 = p_i^2 > p_i^3 > \dots > p_i^v = 1$ , также выполняются

неравенства  $k_i^j < p_n \alpha_i^j$ ,  $|k_i^j|_{r_i^j} = |p_n \alpha_i^j|_{r_i^j} < r_i^j = p_i^{j+1}$ .

Эти соотношения обосновывают конечность итерационного разложения правильной дроби в регулярную сумму. Равенство  $p_i^{v+1} = r_i^v = 1$  является критерием завершения разложения.

При дальнейшем описании итерационного процесса вычисления неточного ранга учтем следующее.

Выбираем для шага  $j+1$  соотношение  $\frac{\alpha_i^{j+1}}{p_i^{j+1}} = \frac{|k_i^j|_{r_i^j}}{r_i^j}$ , если при вычислении с использованием расширенного алгоритма Евклида на предыдущем шаге  $j$  получено уравнение  $r_i^j p_n \alpha_i^j - k_i^j p_i^j = 1$ .

Учтем, что для вычисления слагаемого шага  $j$  разложения использовали соотношение

$$\frac{\alpha_i^j}{p_i^j} = \frac{k_i^j}{r_i^j p_n} + \frac{1}{r_i^j p_i^j p_n} = \frac{k_i^j}{r_i^j p_n} + \frac{1}{r_i^j p_i^j p_n} = \frac{k_i^j}{p_n} + \frac{\xi_i^j}{p_n},$$

$$U_i^j = \left[ \frac{k_i^j}{r_i^j} \right] \leq p_n - 1, \quad \xi_i^j = \frac{1}{r_i^j p_i^j}.$$

Выбираем для шага  $j+1$  соотношение  $\frac{\alpha_i^{j+1}}{p_i^{j+1}} = \frac{|p_n \alpha_i^j - k_i^j|_{p_i^j - r_i^j}}{p_i^j - r_i^j} = \frac{|\tilde{k}_i^j|_{r_i^j}}{\tilde{r}_i^j}$ , если на предыдущем шаге  $j$  получено уравнение  $r_i^j p_n \alpha_i^j - k_i^j p_i^j = -1$ .

Учтем, что для вычисления слагаемого  $j$  шага разложения использовалось соотношение:

$$\frac{\alpha_i^j}{p_i^j} = \frac{(p_n \alpha_i^j - k_i^j)}{p_n} + \frac{1}{(p_i^j - r_i^j) p_i^j p_n} = \frac{\tilde{k}_i^j}{p_n} + \frac{\tilde{\xi}_i^j}{p_n},$$

$$U_i^j = \left[ \frac{\tilde{k}_i^j}{\tilde{r}_i^j} \right] \leq p_n - 1, \quad \tilde{\xi}_i^j = \frac{1}{(p_i^j - r_i^j) p_i^j}.$$

Продолжим описание итерационного процесса с использованием дроби  $\frac{k_i^j}{r_i^j}$  и  $\xi_i^j$  или в соответствующих случаях, используя  $\frac{\tilde{k}_i^j}{\tilde{r}_i^j}$ ,  $\tilde{\xi}_i^j$ .

Для каждой правильной дроби  $\frac{\alpha_i}{p_i}$  на основе со-

отношений леммы может быть получено разложение в конечную сумму монотонно убывающих правильных дробей с знаменателями, являющимися степенями  $p_n$ , которые можно хранить в кэш-памяти модулярного процессора [6]:

$$\frac{\alpha_i}{p_i} = \frac{\left[ \frac{k_i^1}{r_i^1} \right]}{p_n} + \frac{U_i^2}{p_n^2} + \dots + \frac{U_i^v}{p_n^v} + \frac{\xi_i^1}{p_n} + \frac{\xi_i^2}{p_n^2} + \dots + \frac{\xi_i^v}{p_n^v} =$$

$$= \frac{U_i^1}{p_n} + \frac{U_i^2}{p_n^2} + \dots + \frac{U_i^v}{p_n^v} + \frac{\xi_i^1}{p_n} + \frac{\xi_i^2}{p_n^2} + \dots + \frac{\xi_i^v}{p_n^v},$$

где выполняются соотношения  $U_i^j = \left[ \frac{k_i^j}{r_i^j} \right] \leq p_n - 1$ ,

$1 \leq j \leq v$ ,

$$U_i^1 = \left[ \frac{\alpha_i p_n}{p_i} \right] = \left[ \frac{\alpha_i p_n}{p_i^1} \right] = \left[ \frac{k_i^1}{r_i^1} \right] \leq p_n - 1,$$

$$0 \leq \left[ \frac{\sum_{i=1}^n U_i^j}{p_n} \right] \leq n - 1, \quad 1 \leq j \leq v.$$

$$\xi_i^1 = \frac{1}{r_i^1 p_i^1} = \frac{1}{p_i^2 p_i^1}, \quad \xi_i^j = \frac{1}{r_i^j p_i^j} = \frac{1}{p_i^{j+1} p_i^j}.$$

При вычислении нормированного ранга на первом этапе, суммируя правильные дроби, получим

$$\sum_{i=1}^n \frac{\alpha_i}{p_i} =$$

$$= \sum_{i=1}^n \left( \frac{\left( \frac{k_i^1}{r_i^1} \right)}{p_n} + \frac{\xi_i^1}{p_n} \right) = \sum_{i=1}^n \left( \frac{\left[ \frac{k_i^1}{r_i^1} \right] + \frac{|k_i^1|_{r_i^1}}{r_i^1}}{p_n} + \frac{\xi_i^1}{p_n} \right),$$

где  $|k_i^1|_{r_i^1} = |p_n \alpha_i^1|_{r_i^1} < r_i^1 = p_i^2 < p_i^1 = p_i$ . Из следствия Леммы получим рациональное число  $\frac{1}{p_n} \sum_{i=1}^{n-1} \xi_i^1 = \frac{1}{p_n} \sum_{i=1}^{n-1} \frac{1}{r_i^1 p_i^1} < \frac{1}{2p_n}$ , неравенство учтем при

вычислении функции "целая часть числа".

Вычислим функцию "целая часть числа", учтем

соотношения (3) и  $\left[ \frac{[\alpha]}{p} \right] = \left[ \frac{\alpha}{p} \right]$ .

$$\begin{aligned} \left[ \sum_{i=1}^n \frac{\alpha_i}{p_i} \right] &= \left[ \sum_{i=1}^n \frac{\left[ \frac{k_i^1}{r_i^1} \right] + \frac{|k_i^1|_{r_i^1} + \left( \frac{1}{p_i} \right)}{r_i^1}}{p_n} \right] = \\ &= \left[ \frac{\sum_{i=1}^n \left[ \frac{k_i^1}{r_i^1} \right] + \sum_{i=1}^n \frac{|k_i^1|_{r_i^1}}{r_i^1} + \sum_{i=1}^{n-1} \frac{1}{r_i^1 p_i}}{p_n} \right] = \\ &= \left[ \frac{\sum_{i=1}^n \left[ \frac{k_i^1}{r_i^1} \right] + \left[ \sum_{i=1}^n \frac{|k_i^1|_{r_i^1}}{r_i^1} + \sum_{i=1}^{n-1} \frac{1}{r_i^1 p_i} \right]}{p_n} \right] = \\ &= \left[ \frac{\sum_{i=1}^n \left[ \frac{k_i^1}{r_i^1} \right] + \left[ \sum_{i=1}^n \frac{|k_i^1|_{r_i^1}}{r_i^1} \right]}{p_n} \right] = \\ &= \left[ \frac{\sum_{i=1}^n \left[ \frac{k_i^1}{r_i^1} \right]}{p_n} \right] + \left[ \frac{\sum_{i=1}^n \left[ \frac{k_i^1}{r_i^1} \right]_{p_n} + \left[ \sum_{i=1}^n \frac{\alpha_i^2}{p_i} \right]}{p_n} \right] = \\ &= \left[ \frac{1}{p_n} \sum_{i=1}^n U_i^1 \right] + \left[ \frac{\sum_{i=1}^n U_i^1}{p_n} + \left[ \sum_{i=1}^n \frac{\alpha_i^2}{p_i} \right]}{p_n} \right] = \left[ \frac{1}{p_n} \sum_{i=1}^n U_i^1 \right] + \\ &+ \left[ \frac{\sum_{i=1}^n U_i^1}{p_n} + \left[ \frac{1}{p_n} \sum_{i=1}^n U_i^2 \right] + \frac{\left[ \sum_{i=1}^n U_i^2 \right]_{p_n} + \left[ \sum_{i=1}^n \frac{\alpha_i^3}{p_i} \right]}{p_n} \right]. \end{aligned}$$

Выполняется неравенство

$$\left[ \frac{1}{p_n} \sum_{i=1}^n \left[ \frac{k_i^1}{r_i^1} \right] \right] = \left[ \frac{1}{p_n} \sum_{i=1}^n U_i^1 \right] \leq n - 1.$$

На первом этапе итерационного процесса необходима проверка неравенства  $\omega > n - 1$ , являющего критерием завершения процесса суммирования.

Для второго и последующих этапов в формуле вычисления нормированного ранга получим

$$\begin{aligned} \sum_{i=1}^n \frac{\alpha_i^2}{p_i} &= \sum_{i=1}^n \frac{|k_i^1|_{r_i^1}}{r_i^1} = \\ &= \sum_{i=1}^n \left( \left[ \frac{k_i^2}{r_i^2} \right] + \frac{\xi_i^2}{p_n} \right) = \sum_{i=1}^n \left( \frac{\left[ \frac{k_i^2}{r_i^2} \right] + \frac{|k_i^2|_{r_i^2}}{r_i^2}}{p_n} + \frac{\xi_i^2}{p_n} \right). \end{aligned}$$

Вычислим функцию "целая часть числа", учитывая

вая  $\left[ \frac{\alpha}{p} \right] = \left[ \frac{[\alpha]}{p} \right]$ :

$$\begin{aligned} \left[ \sum_{i=1}^n \frac{\alpha_i^2}{p_i} \right] &= \left[ \sum_{i=1}^n \frac{\left[ \frac{k_i^2}{r_i^2} \right] + \frac{|k_i^2|_{r_i^2}}{r_i^2}}{p_n} + \frac{1}{p_n} \sum_{i=1}^n \xi_i^2 \right] = \\ &= \left[ \sum_{i=1}^n \frac{\left[ \frac{k_i^2}{r_i^2} \right] + \frac{|k_i^2|_{r_i^2}}{r_i^2}}{p_n} \right] = \left[ \frac{1}{p_n} \sum_{i=1}^n \left[ \frac{k_i^2}{r_i^2} \right] + \frac{1}{p_n} \sum_{i=1}^n \frac{|k_i^2|_{r_i^2}}{r_i^2} \right] = \\ &= \left[ \frac{1}{p_n} \sum_{i=1}^n \left[ \frac{k_i^2}{r_i^2} \right] \right] + \left[ \frac{1}{p_n} \left[ \sum_{i=1}^n \left[ \frac{k_i^2}{r_i^2} \right] \right]_{p_n} + \frac{1}{p_n} \sum_{i=1}^n \frac{|k_i^2|_{r_i^2}}{r_i^2} \right] = \\ &= \left[ \frac{1}{p_n} \sum_{i=1}^n \left[ \frac{k_i^2}{r_i^2} \right] \right] + \left[ \frac{1}{p_n} \left[ \sum_{i=1}^n \left[ \frac{k_i^2}{r_i^2} \right] \right]_{p_n} + \frac{1}{p_n} \left[ \sum_{i=1}^n \frac{\alpha_i^3}{p_i} \right] \right] = \\ &= \left[ \frac{1}{p_n} \sum_{i=1}^n U_i^2 \right] + \left[ \frac{\sum_{i=1}^n U_i^2}{p_n} + \left[ \sum_{i=1}^n \frac{\alpha_i^3}{p_i} \right] \right]. \end{aligned}$$

Выполняется неравенство

$$\left[ \frac{1}{p_{n_i}} \sum_{i=1}^n \left[ \frac{k_i^2}{r_i^2} \right] \right] = \left[ \frac{1}{p_{n_i}} \sum_{i=1}^n U_i^2 \right] \leq n - 1.$$

На каждом этапе итерационного алгоритма необходима проверка неравенства  $\omega > n - 1$ , являющегося критерием завершения процесса суммирования.

Продолжая итерационный процесс, учтем, что

$$\frac{\alpha_i^3}{p_i^3} = \frac{|k_i^2|_{r_i^2}}{r_i^2}.$$

На следующих этапах итерационного процесса получим

$$\begin{aligned} & \left[ \sum_{i=1}^n \frac{\alpha_i}{p_i} \right] = \left[ \frac{1}{p_{n_i}} \sum_{i=1}^n U_i^1 \right] + \\ & + \left[ \frac{p_n - \omega^1}{p_n} + \frac{1}{p_n} \left[ \sum_{i=1}^n \frac{|k_i^1|_{r_i^1}}{r_i^1} \right] \right] = \left[ \frac{1}{p_{n_i}} \sum_{i=1}^n U_i^1 \right] + \\ & + \left[ \frac{p_n - \omega^1 + \left[ \frac{1}{p_{n_i}} \sum_{i=1}^n U_i^2 \right] + \frac{p_n - \omega^2 + [\dots] + \dots}{p_n}}{p_n} \right] = \\ & = \left[ \frac{1}{p_{n_i}} \sum_{i=1}^n U_i^1 \right] + \\ & + \left[ \frac{p_n - \omega^1 + \left[ \frac{1}{p_{n_i}} \sum_{i=1}^n U_i^2 \right] + \left[ \frac{p_n - \omega^2 + \left[ \frac{1}{p_{n_i}} \sum_{i=1}^n U_i^3 \right] + \frac{p_n - \omega^2 + [\dots] + \dots}{p_n}}{p_n} \right]}{p_n} \right]. \end{aligned}$$

В итерационном процессе вычисления нормированного ранга на всех этапах вычисляем целые величины:

$$\omega^j = p_n - \left\lfloor \frac{\sum_{i=1}^n U_i^j}{p_n} \right\rfloor = p_n - \left\lfloor \frac{\sum_{i=1}^n \left[ \frac{k_i^j}{r_i^j} \right]}{p_n} \right\rfloor,$$

что позволяет получить два критерия завершения процесса суммирования.

**Первый критерий.** Процесс завершается, если на некотором этапе выполняется неравенство

$$\left\lfloor \frac{\sum_{i=1}^n U_i^j}{p_n} \right\rfloor = \left\lfloor \frac{\sum_{i=1}^n \left[ \frac{k_i^j}{r_i^j} \right]}{p_n} \right\rfloor \leq p_n - n, \quad 1 \leq j \leq v,$$

так как соответствующие последующие суммы дают величину строго меньшую  $n - 1$ , которая не оказывает влияние на ранее полученную величину.

**Второй критерий.** Процесс завершается, если на некотором этапе выполняется неравенство

$$p_n - \omega^j + \left[ \frac{1}{p_{n_i}} \sum_{i=1}^n U_i^{j+1} \right] < n - 1, \text{ так как соответствующие последующие суммы дают величину строго меньшую двух, которая не оказывает влияние на ранее полученную сумму.}$$

## Заключение

Для модулярного процессора итерационный процесс вычисления нормированного ранга состоит из параллельных вычислений разложения каждой компоненты модулярного представления в регулярную сумму по степеням одного (наибольшего) из модулей и затем суммирование соответствующих слагаемых этих разложений с проверкой критериев останова процесса. Сложность алгоритма разложения правильных дробей в регулярные суммы равна сложности расширенного алгоритма Евклида, выполняемого параллельно и независимо менее  $m = \lceil \log_2 p_n \rceil + 1$  раз в каждом из  $n$  вычислительных трактов модулярного процессора, т. е.  $O(mSAE(m))$  (где  $SAE(m)$  — сложность алгоритма Евклида для  $m$ -разрядных чисел) или при оценке сложности алгоритма Евклида  $O(mM(m)\log_2 m)$  (где  $M(m)$  — сложность умножения  $m$  разрядных чисел) [7]. Полученные разложения позволяют выполнить менее чем  $m$  раз суммирование соответствующих целых слагаемых полученных разложений, не более чем  $n$  раз последовательно или  $k = \lceil \log_2 n \rceil + 1$  раз в форме пирамидального суммирования на  $n$  вычислительных трактах модулярного процессора. Верхняя граница сложности вычисления на модулярном процессоре нормированного ранга рассмотренным итерационным процессом равна  $O(m(SAE(m) + k))$  или  $O(m(M(m)\log_2 m + k))$ . Следовательно, сложность процесса оценивается линейно при последовательных суммированиях или логарифмом от числа компонент векторного модулярного представления числовой величины в варианте пирамидального суммирования.

## Список литературы

1. **Амербаев В. М.** Теоретические основы машинной арифметики. Алма-Ата: Наука, 1976. 320 с.
2. **Инютин С. А.** Основы модулярной алгоритмики. Ханты-Мансийск: Полиграфист, 2009. 237 с.
3. **Инютин С. А.** Метод вычисления количественной характеристики модулярной величины // Информационные технологии. 2016. Т. 22, № 5. С. 343–347.
4. **Инютин С. А.** Анализ сложности многоразрядных вычислительных процессов // Научные труды МАТИ. 2014. Вып. 22 (94). С. 154–159.
5. **Амербаев В. М., Стемпковский А. Л., Соловьев Р. А.** Принципы рекурсивных модулярных вычислений // Информационные технологии. 2013. № 2. С. 22–27.
7. **Столярский Е. З., Шилов В. В.** Организация и работа кэш-памяти // Информационные технологии. 2000. № 7. С. 2–8.
7. **Ноден П., Китге К.** Алгебраическая алгоритмика. М.: Мир, 1999. 720 с.

## Method Calculation Characteristic of the Order Relation for Computer Modular Value

*An iterative method for calculating the quantitative characteristic of the order relation for parallel computer data formats in modular reconfigurable computing systems is developed. The quantitative characteristic is intended for performance of separate operations in parallel computer arithmetic. The complexity of calculating the quantitative characteristic is a critical parameter for any software or technical implementations of modular arithmetic, which allows parallel and independent execution of ring algebraic operations on modular data formats on computational SIMD architectures. In the method, deductions for individual modules using diophantine relations are decomposed into regular finite sums, the terms of which are then element-wise added together with the verification of the criterion for completing the process of calculating the quantitative characteristic. The simulation of the method shows the convergence for 2–3 stages of summation. The iterative method allows under certain conditions to obtain a characteristic with a linear or logarithmic complexity of the algorithm for pyramidal summation on a parallel architecture.*

**Keywords:** multiprocessor reconfigurable SIMD architecture systems, parallel computational process, linear and logarithmic computational complexity, parallel computer formats, numerical quantitative characteristic, regular finite sums

### References

1. Amerbaev V. M. *Teoreticheskie osnovy mashinnoy arifmetiki* (Theoretic base computer arithmetic). Alma-Ata, Nauka, 1976, 320 p. (in Russian).
2. Inyutin S. A. *Osnovy modulyarnoy algoritmiki* (Base at modular algorithmic). Hantyi-Mansiysk; Poligrafist, 2009, 237 p. (in Russian).
3. Inyutin S. A. Analiz slojnosti mnogorazryadnyh vyichislitelnyh protsessov (Analysis many digital calculation process), *Nauchnyye trudy MATI*, 2014, vol. 22 (94), pp. 154–159 (in Russian).
4. Inyutin S. A. Metod vyichisleniy kolichestvenoy harakteristic modulyarnoy velichina (Method calculation quantitative characteri-
5. Amerbaev V. M., Stempkovsky A. L., Solovjev R. A. Printsiipy rekursivnyh modulyarnyih vyichisleniy (Principles of recursive modular computing), *Informacionnyie tehnologii*, 2013, no. 2, vol. 5, pp. 22–27 (in Russian).
6. Stolyarskiy E. Z., Shilov V. V. Organizatsiya i rabota kesh-pamyati (Planning work Kech-memory), *Informacionnyie tehnologii*, 2000, no. 7, pp. 2–8 (in Russian).
7. Noden P., Kitte K. Algebraicheskaya algoritmika (Algebra algorithmic), Moscow, Mir, 1999, 720 p. (in Russian).

*Теоретический и прикладной научно-технический журнал*

## ПРОГРАММНАЯ ИНЖЕНЕРИЯ

ISSN 2220-3397

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях человеческой деятельности.

**Журнал распространяется только по подписке.**

Журнал входит в перечень периодических изданий, рекомендованных ВАК РФ для публикации основных результатов диссертаций на соискание степеней кандидата и доктора наук, а также в базу данных Российского индекса научного цитирования (РИНЦ).

**Оформить подписку можно через подписные Агентства  
или непосредственно в редакции журнала.**

Подписные индексы по каталогам:  
"Роспечать" — 22765; "Пресса России" — 39795

107076, Москва, Стромьинский пер., 4  
Тел./факс: (499) 269-55-10  
e-mail: prin@novtex.ru  
<http://www.novtex.ru/prin/rus>

УДК 004.272.2

**Д. В. Тельпухов**, канд. техн. наук, зав. отделом, e-mail: nofrost@inbox.ru,  
**Р. А. Соловьев**, канд. техн. наук, вед. науч. сотр.,  
Институт проблем проектирования в микроэлектронике РАН, г. Зеленоград

## Метод поиска эквивалентных ошибок в логических схемах

*Данная статья посвящена вопросам предварительной обработки логических схем для сокращения вычислительных затрат на оценку параметров сбоеустойчивости, а также затрат на верификацию и тестирование схемы. Был предложен метод нахождения классов эквивалентности для различных ошибок логической схемы, позволяющий оценивать влияние только одной ошибки из соответствующего класса при последующих процессах верификации и оценки.*

**Ключевые слова:** эквивалентные схемы, ICCAD Contest, инжектирование ошибок, поиск эквивалентных ошибок

### Введение

В рамках различных уровней обеспечения надежности интегральных схем особенно активно в настоящее время развивается направление обеспечения сбоеустойчивости на уровне логических вентилей [1—4]. Этому факту способствует ряд взаимосвязанных трендов. Во-первых, уменьшение размеров транзисторов и увеличение плотности их размещения приводит к тому, что вероятность сбоев и отказов в схеме резко возрастает. Во-вторых, увеличение тактовых частот, на которых работают схемы, также приводят к повышению вероятности защелкивания сбоев в регистрах схемы. В-третьих, долгое время считалось что сбои в комбинационных схемах не существенны по сравнению со сбоями в элементах памяти, и в настоящее время возникла ситуация острой нехватки методов и средств автоматизации проектирования сбоеустойчивых логических схем [5].

Зачастую в задачах верификации и тестирования, а также для сокращения времени, требуемого для нахождения параметров сбоеустойчивости [6, 7], требуется провести некоторый предварительный анализ схемы. Одним из основных этапов такого предварительного анализа является задача обнаружения ошибок, под действием которых схема ведет себя абсолютно одинаково [8, 9].

Проблема ставится следующим образом: в определенной схеме, которую будем считать эталонной, инжектируются одна или несколько ошибок. Получаемая в этом случае схема считается дефектной. В рамках данной задачи будем рассматривать только одиночные сбои, поэтому инжектировать ошибки в эталонную схему будем по одной. Как

правило, вследствие такого воздействия инжектированные ошибки будут проявлять свое влияние на некоторых выходных портах при некоторых входных комбинациях.

Однако некоторые ошибки будут вызывать одинаковые сбои на выходах схемы. Другими словами, независимо от того, какие значения подаются на входные порты, выходы у разных дефектных схем могут быть одинаковыми. Поскольку выходные значения неисправных схем с различными ошибками являются идентичными при одинаковых входных воздействиях, мы будем называть эти ошибки идентичными. Учитывая этот факт, можно сделать вывод, что только одна ошибка из такого класса эквивалентности идентичных ошибок может быть рассмотрена для определенных верификационных задач. Это позволяет экономить вычислительные ресурсы и ускорять нахождение параметров сбоеустойчивости.

### Формальная постановка задачи поиска эквивалентных ошибок в логических схемах

Вносимые ошибки разделяются на одиннадцать типов, которые представлены в табл. 1.

Рассмотрим пример: пусть имеется некоторая логическая схема, реализующая схему проверки на четность (рис.1).

Мы будем инжектировать две ошибки. Одна меняет драйвер сигнала N6 на NXOR, как представлено на рис. 2. Будем называть это "ошибка 1". Другая ошибка, "ошибка 2", меняет драйвер сигнала N7 на NXOR.

Если провести моделирование, можно убедиться, что ошибка 1 и ошибка 2 абсолютно эквива-

Таблица 1

Типы вносимых в схему ошибок

Имя	Описание
SA0	Замыкание на 0
SA1	Замыкание на 1
NEG	Инверсия сигнала
RDOB_AND	Замена драйвера на AND
RDOB_NAND	Замена драйвера на NAND
RDOB_OR	Замена драйвера на OR
RDOB_NOR	Замена драйвера на NOR
RDOB_XOR	Замена драйвера на XOR
RDOB_NXOR	Замена драйвера на NXOR
RDOB_NOT	Замена драйвера на NOT (только если тот был BUFF)
RDOB_BUFF	Замена драйвера на BUFF (только если тот был NOT)

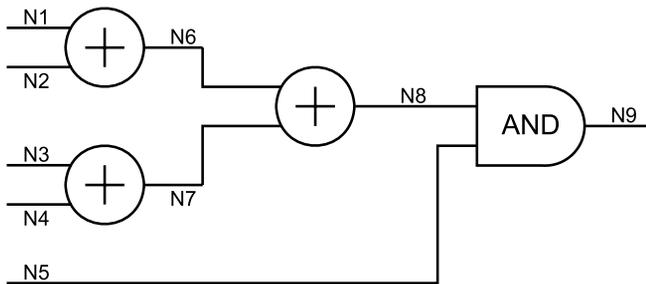


Рис. 1. Логическая схема операции проверки на четность

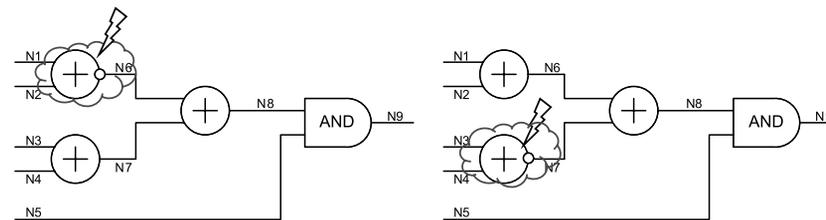


Рис. 2. Пример двух эквивалентных ошибок для схемы проверки на четность

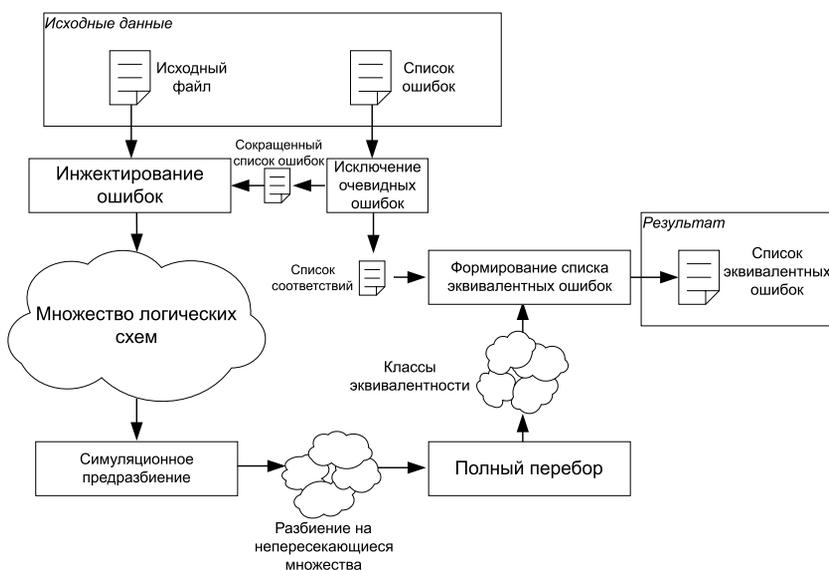


Рис. 3. Маршрут метода поиска эквивалентных ошибок

лентны на всех входных наборах воздействий. Таким образом, обе эти ошибки являются эквивалентными.

Задача ставится следующим образом: имеется эталонная схема и список различных ошибок, где под ошибкой понимается пара — имя узла, тип ошибки. Имя узла эталонной схемы определяет местоположение внедряемой ошибки, в то время как тип ошибки определяет ее тип в соответствии с табл. 1. Требуется разбить весь список ошибок на классы эквивалентности, где в рамках одного класса все ошибки являются эквивалентными, а любые две ошибки из разных классов не являются эквивалентными. Очевидно, что такое разбиение является единственным. Для оценки работы алгоритма был сформирован комплексный критерий, который включает точность получаемых разбиений, а также временные затраты и затраты памяти в долях 60:30:10 соответственно. Более подробно о системе оценок и начислении очков можно прочитать на официальном сайте организаторов конкурса [10].

Особенность поставленной задачи заключается в том, что требуется реализовать эффективный алгоритм поиска классов эквивалентности большого набора ошибок (тысячи ошибок) для достаточно больших логических схем (сотни логических элементов) — как комбинационных, так и последовательных.

### Метод поиска эквивалентных ошибок в логических схемах.

#### Базовая структура алгоритма

Предлагается методика, реализующая разбиение заданного набора ошибок на некоторые классы эквивалентности, где в рамках каждого класса находятся ошибки, эквивалентные относительно заданной эталонной схемы.

Структура разработанного метода, а также основные этапы представлены на рис. 3.

Кратко опишем базовые этапы представленного маршрута.

- **Инжектирование ошибок.** На данном этапе ошибки по одной вносятся в эталонную схему, формируя при этом множество новых схем. Задача тем самым сводится к поиску эквивалентных схем в полученном множестве.
- **Исключение очевидных эквивалентных ошибок.** Под очевидными ошибками понимаются такие ошибки, которые могут быть признаны эквивалентными на основе тривиального анализа структуры схемы, местоположения и типа ошибок. Например, очевидно, что идентичными могут быть признаны

## Структурные методы поиска эквивалентных пар ошибок

следующие две ошибки: замена драйвера сигнала с OR на NOR и инверсия на этом узле. Как показали исследования, данный этап позволяет отсеивать до 30 % всех ошибок, затрачивая на это совершенно незначительные вычислительные ресурсы.

- Предварительная группировка ошибок с помощью симуляции. Данный этап заключается в подаче некоторых воздействий на входы всех оставшихся схем и последующем сравнении выходных векторов. Это позволяет отсеивать уникальные схемы и объединять оставшиеся в предварительные группы для последующего более тщательного анализа.
- Полный перебор внутри каждой группы на основе аналитических методов проверки схем на эквивалентность. На данном этапе каждая предварительная группа подвергается тщательному попарному сравнению на эквивалентность. Сложность алгоритма на данном этапе варьируется от  $O(n)$  в лучшем случае до  $O(n(n - 1)/2)$  в худшем.
- Формирование списка эквивалентных ошибок. На данном этапе в уже сформированные группы с эквивалентными ошибками добавляются исключенные из рассмотрения "очевидные ошибки". Формируется выходной файл с описанием искомым классов эквивалентности.

На первом этапе обработки списка ошибок целесообразно как можно раньше исключить из рассмотрения наибольшее число "очевидных ошибок". Под очевидными ошибками понимаются такие ошибки, эквивалентность которых можно установить простым анализом структуры схемы. Данный этап позволяет без значительных затрат существенно сократить размерность задачи и соответственно время, необходимое для реализации последующих этапов. Полный список всех "очевидных ошибок", отслеживаемых в нашем алгоритме, представлен в табл. 2.

Все "очевидные ошибки" из представленного списка можно разделить на два класса. Первый тип "очевидных ошибок" (1—8) связан с ошибками только на одном элементе, в то время как ошибки второго типа (9—18) затрагивают два соседних элемента без разветвлений. Приметы ошибок первого и второго типов представлены на рис. 4. Для данного примера использована эталонная схема, представленная на рис. 1.

На верхней части рис. 4 представлен пример эквивалентных ошибок типа AND(RDOB\_NAND)(NEG) (№ 1 в табл. 2). Как и было сказано выше, ошибка зависит только от ошибок на одном элементе эталонной схемы — AND.

Таблица 2

Список всех "очевидных" ошибок

Порядковый номер	Обозначение	Описание
1	AND(RDOB_NAND)(NEG)	Инверсия на выходе элемента AND эквивалентна замене этого элемента на NAND
2	NAND(RDOB_AND)(NEG)	Инверсия на выходе элемента NAND эквивалентна замене этого элемента на AND
3	OR(RDOB_NOR)(NEG)	Инверсия на выходе элемента AND эквивалентна замене этого элемента на NAND
4	NOR(RDOB_OR)(NEG)	Инверсия на выходе элемента NOR эквивалентна замене этого элемента на OR
5	XOR(RDOB_NXOR)(NEG)	Инверсия на выходе элемента XOR эквивалентна замене этого элемента на NXOR
6	NXOR(RDOB_XOR)(NEG)	Инверсия на выходе элемента NXOR эквивалентна замене этого элемента на XOR
7	BUFF(RDOB_NOT)(NEG)	Инверсия на выходе элемента BUFF эквивалентна замене этого элемента на NOT
8	NOT(RDOB_BUFF)(NEG)	Инверсия на выходе элемента NOT эквивалентна замене этого элемента на BUFF
9	(SA0)NAND(SA1)	Замыкание на ноль на входе элемента NAND эквивалентно замыканию на единицу на его выходе
10	(SA1)OR(SA1)	Замыкание на единицу на входе элемента OR эквивалентно замыканию на единицу на его выходе
11	(SA0)AND(SA0)	Замыкание на ноль на входе элемента AND эквивалентно замыканию на ноль на его выходе
12	(SA1)NOR(SA0)	Замыкание на единицу на входе элемента NOR эквивалентно замыканию на ноль на его выходе
13	(NEG)XOR(RDOB_NXOR)	Инверсия сигнала на входе элемента XOR эквивалентна замене элемента на NXOR
14	(NEG)NXOR(RDOB_XOR)	Инверсия сигнала на входе элемента NXOR эквивалентна замене элемента на XOR
15	(SA0)INV(SA1)	Замыкание на ноль на входе элемента INV эквивалентно замыканию на единицу на его выходе
16	(SA1)INV(SA0)	Замыкание на единицу на входе элемента INV эквивалентно замыканию на ноль на его выходе
17	(SA1)BUFF(SA1)	Замыкание на единицу на входе элемента BUFF эквивалентно замыканию на единицу на его выходе
18	(SA0)BUFF(SA0)	Замыкание на ноль на входе элемента BUFF эквивалентно замыканию на ноль на его выходе

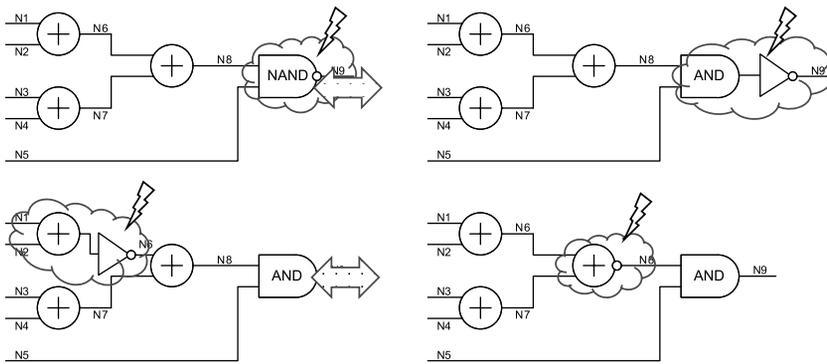


Рис. 4. Пример очевидных ошибок первого и второго типов

Внизу на рис. 4 приведен пример "очевидных ошибок" второго типа, а именно (NEG)XOR (RDOB\_NXOR) (№ 13 в табл. 2). Здесь видно, что анализируются ошибки не только на элементе XOR (№ 8), но и ошибки на предыдущих элементах XOR (№ 6 и № 7). Дело в том, что выход для элемента XOR (№ 6) является входом элемента XOR (№ 8), и, следовательно, ошибку вида "инверсия сигнала" для элемента XOR (№ 6) можно интерпретировать как инверсию сигнала на входе элемента XOR (№ 8). Соответственно, можно использовать правило, которое заключается в том, что инверсия сигнала на входе элемента XOR эквивалентна замене элемента на NXOR. Единственный нюанс, касающийся "очевидных ошибок" второго типа, заключается в том, что входной сигнал для нашего элемента не должен быть входным сигналом для других вентилей схемы. В противном случае ошибка на драйвере этого сигнала повлияет также и на другие участки схемы, и эквивалентность этих "очевидных ошибок" нарушится.

#### Итеративный метод предварительного разбиения на непересекающиеся подмножества

Базисом для итеративного метода предварительного разбиения на непересекающиеся подмножества служит процедура симуляции схем. Суть его заключается в подаче  $n$  (число симуляций) одинаковых тестовых векторов на все  $k$  схем рассматриваемого множества. Тестовые векторы формируются случайным образом. На выходе данной процедуры мы получаем  $k$  битовых векторов длиной  $n \cdot out\_num$ , где  $out\_num$  — число выходов эталонной схемы. Каждый битовый вектор представляет собой реакцию соответствующей схемы на входные воздействия.

Дальнейший анализ полученных битовых векторов строится на основе следующих вполне очевидных утверждений:

- если два битовых вектора различаются, то соответствующие им схемы заведомо находятся в разных классах эквивалентности;

- если два битовых вектора идентичны, то соответствующие им схемы могут быть эквивалентны, причем вероятность того, что эти схемы эквивалентны, тем выше, чем выше  $n$ .

После однократного прохода рассматриваемой процедуры ( $n$  симуляций) и анализа полученных реакций схем получаем некоторое предварительное разбиение исходного множества на непересекающиеся подмножества. Причем, если две схемы лежат в разных подмножествах, то это гарантирует, что схемы лежат в разных классах эквивалентности, однако нахождение двух схем в одном подмножестве

не гарантирует их эквивалентности. Таким образом, на последующих этапах данные подмножества могут только уменьшаться и дробиться, но не могут укрупняться и сливаться.

Важным вопросом на этапе предварительного разбиения является вопрос выбора  $n$  — числа симуляций. Выбрав слишком маленькое число симуляций, мы получаем очень мало крупных подмножеств, анализ которых на последующих этапах займет слишком много ресурсов. В то же время, выбирая неоправданно большое число симуляций, можно существенно замедлить этап предварительного разбиения.

Также на этапе предварительного разбиения следует учитывать тот факт, что после какого-то числа симуляций некоторые подмножества могут состоять всего из одной схемы, следовательно, дальнейшие симуляции и анализ этой схемы нецелесообразны. С учетом вышесказанного был предложен итеративный алгоритм для предварительного разбиения схем на непересекающиеся подмножества. Схема алгоритма представлена на рис. 5.

Подробнее опишем представленный алгоритм. На вход алгоритма поступает исходный массив схем. Первое предварительное разбиение происходит с заданным параметром  $n$  (число симуляций), равным  $first\_tst\_num$ . На выходе данного разбиения получаем разбиение на подгруппы. Далее следует проверка максимального числа схем в подгруппах. Если максимальное число схем в подгруппах не превышает некоторого значения  $max\_el$ , то работа алгоритма заканчивается. В противном случае выбирается конкретная подгруппа для разбиения. Далее следуют две проверки для данной подгруппы. Во-первых, проверяется число схем в подгруппе, и если оно меньше некоторого  $min\_el$ , то подгруппа не рассматривается. Во-вторых, проверяется история разбиений этой подгруппы. В случае, если несколько раз подряд были проведены попытки разбиения этой подгруппы, а число элементов в ней не уменьшилось, группа исключается из рассмотрения. Число попыток устанавливается параметром  $threshold$ . После успешного прохождения проверок

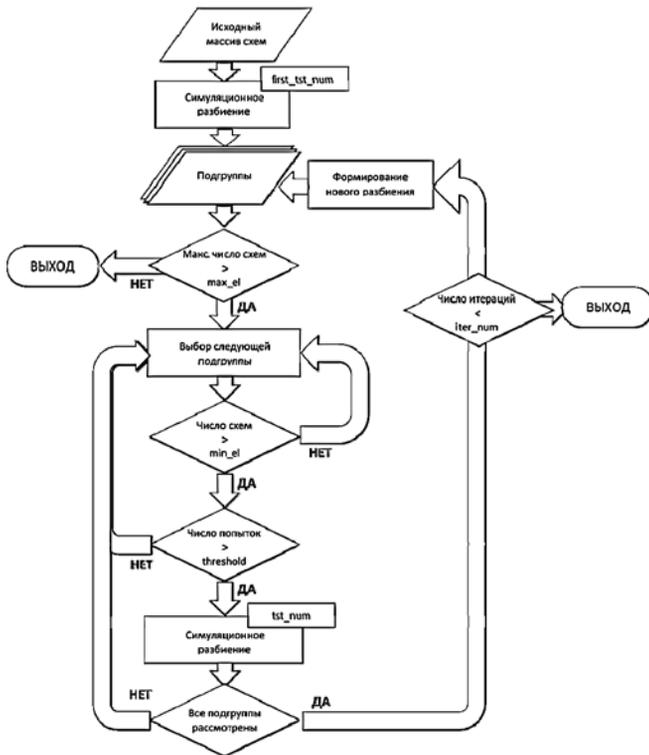


Рис. 5. Алгоритм предварительного разбиения на непересекающиеся подмножества

происходит непосредственно симуляционное разбиение данной подгруппы с параметром  $n = \text{tst\_num}$ . Если не все подгруппы рассмотрены, то выбирается следующая подгруппа, и весь процесс повторяется. В противном случае проверяется установленный порог числа итераций  $\text{iter\_num}$ , и в случае если он не превышен, происходит новый цикл алгоритма.

Можно видеть, что данный алгоритм имеет шесть параметров, от настройки которых существенно зависит эффективность всего алгоритма. И если необходимость наличия параметров  $\text{first\_tst\_num}$ ,  $\text{tst\_num}$ ,  $\text{iter\_num}$  не требует пояснения, то на остальных параметрах остановимся чуть подробнее.

- Параметр  $\text{max\_el}$  — это, по сути, некоторый порог числа элементов для всех подгрупп. Если все группы состоят из меньшего числа элементов, перегруппировка завершается. Это необходимо для того, чтобы не тратить лишнего времени на этап перегруппировки и быстрее перейти к следующему этапу в том случае, когда подгруппы уже очень малы.
- Параметр  $\text{min\_el}$  устанавливает порог для конкретной подгруппы. Если число элементов в подгруппе меньше этого значения, то подгруппа не рассматривается. Тут подразумевается, что на последующем этапе попарного полного перебора такая маленькая подгруппа будет проанализирована быстрее.
- Параметр  $\text{threshold}$  устанавливает лимит на число неуспешных попыток разбиения. Если за такое

число итераций подряд группа не разбилась, группа больше не рассматривается. Тут подразумевается, что при длительных неуспешных попытках разбиения высока вероятность того, что все схемы в подгруппе являются эквивалентными, и не имеет смысла тратить вычислительные ресурсы на их симуляцию.

Поиск оптимального сочетания данных критериев существенно зависит от вида эталонной схемы, числа ее входов, выходов и элементов и в общем случае требует существенных вычислительных и временных затрат. Для нахождения некоторого приближения к оптимуму была разработана некоторая эмпирическая схема.

Были реализованы генератор произвольных логических схем, а также генератор списка ошибок. С помощью данных генераторов был создан большой набор схем с различными параметрами. Дальнейший эксперимент строился следующим образом: выбирались некоторые наборы параметров и для каждой схемы при каждом наборе параметров определялось время выполнения всего маршрута

поиска классов эквивалентности  $T_j = \sum_{i=1}^n t_{ij}$ , кото-

рое определяет эффективность набора параметров  $j$  для всех рассматриваемых  $n$  схем (табл. 3). Для поиска наилучшего набора параметров учитывалось именно время выполнения всего маршрута поиска классов эквивалентности, так как это время определяет не только время работы алгоритма, но и качество получаемого разбиения.

Таблица 3

Схема эксперимента для нахождения параметров предварительного разбиения

Номер схемы	Набор параметров № 1	...	Набор параметров № k
1	$t_{11}$	...	$t_{1k}$
2	$t_{21}$	...	$t_{2k}$
...	...	...	...
n	$t_{n1}$	...	$t_{nk}$
Время выполнения всего маршрута	$T_1 = \sum_{i=1}^n t_{i1}$	...	$T_k = \sum_{i=1}^n t_{ik}$

Было проведено большое число экспериментов, итеративно приближающих к области наилучшего разбиения. В итоге для алгоритма были использованы следующие параметры:

// число тестов для первого подразбиения

`int first_tst_num = 50;`

// число итераций, после которых завершается перегруппировка

`int iter_num = 50;`

// число тестов для остальных итераций

`int tst_num = 50;`

```

// если число элементов в группе меньше этого значения, группа не рассматривается
int min_el = 5;
// если все группы состоят из меньшего числа элементов, предгруппировка завершается
int max_el = 20;
// если за такое число итераций подряд группа не разбилась, группа больше не рассматривается
int threshold = 3;

```

### Алгоритм разбиения на классы эквивалентности методом полного попарного сравнения

На данном этапе каждая предварительная группа, полученная на предыдущем этапе, подвергается тщательному попарному сравнению на эквивалентность. Эквивалентность логических схем проверялась с помощью средств академического открытого САПР Berkeley abc [11]. На рис. 6 для примера изображены четыре схемы, которые нужно проверить на эквивалентность. Связи между ними символизируют операцию формальной проверки на эквивалентность.

По очевидным причинам эквивалентность двух схем есть отношение эквивалентности. Учитывая свойство симметричности (если  $a \sim b$ , то  $b \sim a$ ) для отношения эквивалентности, число проверок в общем случае можно вычислить по формуле  $\frac{n(n-1)}{2}$ , где  $n$  — число схем в предварительной группе. Однако, учитывая свойство транзитивности (если  $a \sim b$  и  $b \sim c$ , то  $a \sim c$ ), число операций сравнения можно еще сократить.

Продемонстрируем этот факт на конкретном примере. Допустим, схемы 1 и 3 (рис. 6) в дейст-

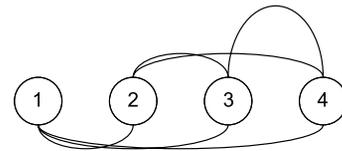


Рис. 6. Процесс разбиения на классы эквивалентности методом полного попарного сравнения

вительности эквивалентны. Без учета свойства транзитивности порядок проверок на эквивалентность был бы таким: (1,2) (1,3) (1,4) (2,3) (2,4) (3,4). Однако, если на второй проверке получен положительный результат (1,3), дальнейшее рассмотрение схемы 1 не имеет смысла, так как все последующие сравнения со схемой 1 дадут тот же результат, что и сравнение со схемой 3. Иными словами, сравнения (1,4) и (3,4) дадут заведомо идентичный результат, так же как и (1,2) и (2, 3).

Таким образом, сложность алгоритма на данном этапе варьируется от  $O(n)$  в лучшем случае до  $O(n(n-1)/2)$  в худшем. Это зависит как от числа эквивалентных схем, так и от порядка схем в подгруппе. К примеру, если все схемы в подгруппе окажутся эквивалентными, то алгоритм сработает за  $n$  итераций.

Этот факт открывает перспективы по разработке разного рода эвристик, связанные с предварительной сортировкой схем в подгруппе.

### Результаты конкурса

Рассмотренная задача выполнялась в рамках ежегодного международного конкурса по автоматизации проектирования интегральных схем, проводимого

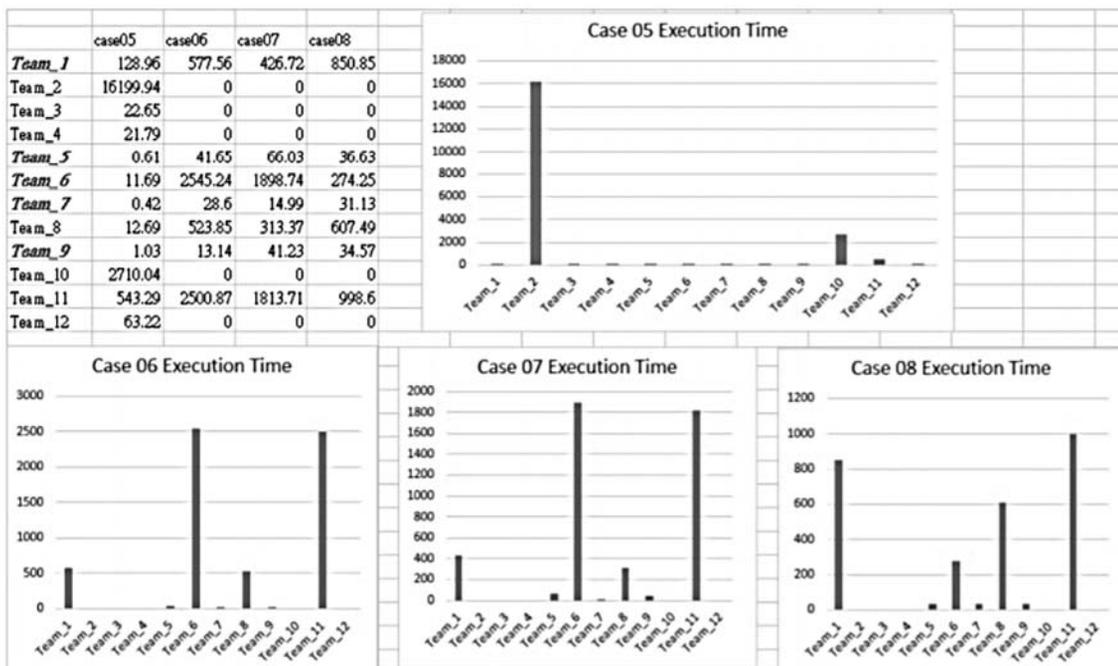


Рис. 8. Оценка времени выполнения алгоритма

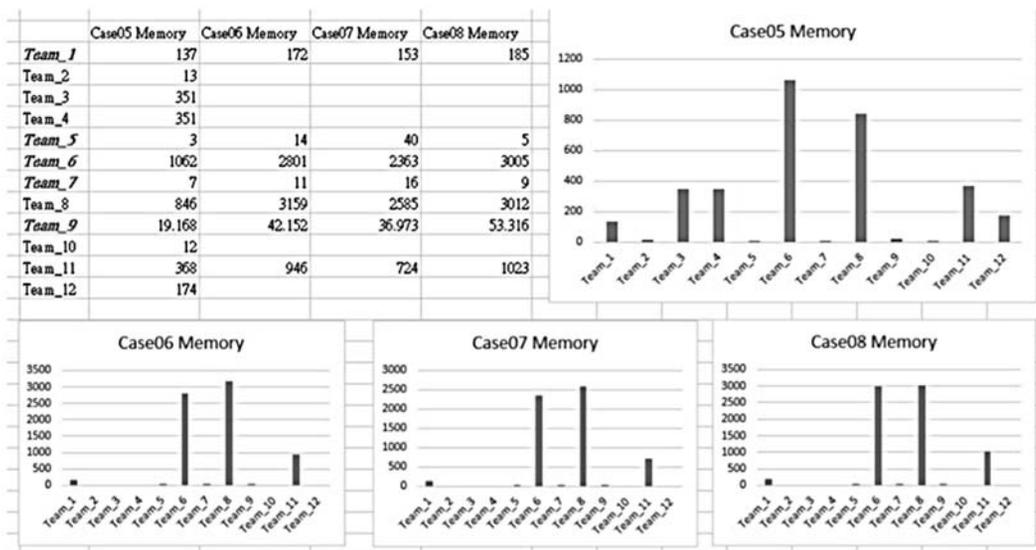


Рис. 9. Оценка затрат памяти алгоритма

в рамках крупнейшей конференции ICCAD 2016. Разработанная программная модель продемонстрировала отличные результаты по быстродействию, памяти и точности результатов, благодаря чему заняла призовое место, а также была удостоена почетным упоминанием на церемонии награждения с вручением сертификатов. Результаты точности, времени выполнения, а также затрат оперативной памяти для четырех кейсов представлены на рис. 7 (см. четвертую сторону обложки), рис. 8 и 9. Представленное в статье решение в списке фигурирует под именем Team\_6.

Как можно видеть, разработанная программа поиска эквивалентных ошибок единственная из представленных на конкурсе программ, обнаружила все пары идентичных ошибок — во всех тестовых кейсах со 100 %-ной точностью. Также следует отметить, что команда из ИППМ РАН — единственная команда из Российской Федерации, занявшая призовое место в этом году.

## Заключение

Вопросы обеспечения надежности цифровых интегральных схем в настоящее время обретают все возрастающую актуальность. Понятие надежности является сложным комплексным понятием и включает в себя как конструктивные, схемотехнические и технологические решения, так и решения на уровне логических вентилях, регистровых передач и микросистемы в целом.

В статье представлен метод поиска эквивалентных ошибок в логических схемах, необходимый для сокращения вычислительных затрат на оценку параметров сбоеустойчивости, а также затрат на верификацию и тестирование схем. Также эти данные могут быть использованы для построения самотестируемых схем.

Рассматриваемая задача была инспирирована конкурсом ICCAD Contest 2016, в рамках которого было написано программное обеспечение для нахождения классов эквивалентных ошибок. Представленное решение получило четвертое место в конкурсе и было удостоено почетным упоминанием на церемонии награждения. Следует отметить, что с точки зрения точности построения классов эквивалентности решение было лучшим из представленных и единственное достигло 100 %-ной точности.

В статье представлен базовый маршрут алгоритма, а также приведены детали реализации базовых его этапов. Разработанный метод совместно с результатами исследований выложены в открытый доступ: исходный код программы, реализующей описанный метод поиска эквивалентных ошибок, опубликован в открытом доступе по адресу [12]; видеозапись с описанием принципов работы метода доступна по адресу [13].

## Список литературы

1. Asadi H., Tahoori M. B., Fazeli M., Miremadi S. G. Efficient algorithms to accurately compute derating factors of digital circuits // *Microelectron Reliab.* 2012. 52 (6). P. 1215—26.
2. George N. and Lach J. Characterization of logical masking and error propagation in combinational circuits and effects on system vulnerability // *Dependable Systems Networks (DSN), 2011 IEEE/IFIP 41st International Conference on.* 2011. P. 323—334.
3. Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А. Повышение сбоеустойчивости логических схем на основе частичного ресинтеза схемы // *Информационные технологии.* 2016. Т. 22, № 7. С. 515—522.
4. Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А., Мячиков М. В. Повышение отказоустойчивости логических схем с использованием нестандартных мажоритарных элементов // *Информационные технологии.* 2015. Т. 21, № 10. С. 749—756.
5. Mahatme N. N., Jagannathan S., Loveless T. D., Massengill L. W., Bhuvu B. L., Wen S.-J. et al. Comparison of combinational and sequential error rates for a deep submicron process // *IEEE Trans Nucl Sci (TNS).* 2011. 58 (6). P. 2719—25.

6. Тельпухов Д. В., Соловьев Р. А., Мячиков М. В. Разработка практических метрик для оценки методов повышения сбоеустойчивости комбинационных схем // Информационные технологии и математическое моделирование систем 2015. Труды международной научно-технической конференции. 2015. С. 79–81.

7. Стемповский А. Л., Тельпухов Д. В., Соловьев Р. А., Мячиков М. В., Тельпухова Н. В. Разработка технологически независимых метрик для оценки маскирующих свойств логических схем // Вычислительные технологии. 2016. Т. 21, № 2. С. 53–62.

8. Akers S. B., Joseph C., and Krishnamurthy B. On the Role of Independent Fault Sets in the Generation of Minimal Test Sets // Proc. Int. Test Conf. August 1987. P. 1100–1107.

9. Prasad A. V. S. S., Agrawal V. D., Atre M. V. A new algorithm for global fault collapsing into equivalence and dominance sets // Proceedings of ITC. October 2002. P. 391–397.

10. URL: [http://cad-contest-2016.el.cycu.edu.tw/Problem\\_A/default.html](http://cad-contest-2016.el.cycu.edu.tw/Problem_A/default.html)

11. Brayton R., Mishchenko A. ABC: An Academic Industrial-Strength Verification Tool. Touili T., Cook B., Jackson P. (eds.) Computer Aided Verification. CAV 2010 // Lecture Notes in Computer Science. 2010. Vol. 6174. Springer, Berlin, Heidelberg.

12. URL: [https://github.com/IDMIPPM/ICCAD\\_2016\\_Contest](https://github.com/IDMIPPM/ICCAD_2016_Contest)

13. URL: <https://www.youtube.com/watch?v=ABIGIiwc29A>

D. V. Telpukhov, Ph. D., Head of Department of IPPM RAS, e-mail: e-mail: nofrost@inbox.ru,  
R. A. Solovye, Ph. D., Leading Researcher,  
Institute for Design Problems in Microelectronics, Zelenograd, 124681, Russian Federation

## Development of a Method for Identical Fault Search in Logical Circuits

*The problems of IC reliability are now becoming increasingly important. The idea of reliability is a complex integrated concept and includes structural, schematic and technological solutions, as well as solutions at the level of logic gates, register transmissions and microsystem as a whole.*

*This paper is devoted to preprocessing of logical circuits aimed at the reduction of computational costs of evaluating fault tolerance parameters, as well as the costs of circuit verification and testing. A method for identical fault search in logical circuits have been proposed, which makes it possible to estimate the effect of only one fault from the corresponding equivalence class of identical faults in subsequent verification and evaluation processes. In addition, these data can be used to construct self-test circuits.*

*This task was inspired by ICCAD 2016 Contest, within the frame of which software for identical fault search was written. The presented decision won the fourth place in the contest and was awarded honorable mention at the award ceremony. Note that in terms of correctness of equivalence classes' identification this solution was the best one presented, and was the only one that reached 100 % accuracy.*

**Key words:** equivalent circuits, ICCAD Contest, fault injection, identical fault search

### References

1. Asadi H., Tahoori M. B., Fazeli M., Miremadi S. G. Efficient algorithms to accurately compute derating factors of digital circuits, *Microelectron Reliab*, 2012; 52 (6), pp. 1215–26.

2. George N. and Lach J. Characterization of logical masking and error propagation in combinational circuits and effects on system vulnerability, *Dependable Systems Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*, 2011, pp. 323–334.

3. Stempkovskiy A. L., Telpukhov D. V., Soloviev R. A. Povyshenie sboeustojchivosti logicheskikh shem na osnove chastichnogo resinteza shemy, *Informacionnye tehnologii*, 2016, vol. 22, no. 7, pp. 515–522 (in Russian).

4. Stempkovskiy A. L., Telpukhov D. V., Soloviev R. A., Myachikov M. V. Povyshenie otkazoustojchivosti logicheskikh shem s ispol'zovaniem nestandardnykh mazhoritarnykh jelementov, *Informacionnye tehnologii*, 2015, vol. 21, no. 10, pp. 749–756 (in Russian).

5. Mahatme N. N., Jagannathan S., Loveless T. D., Massengill L. W., Bhava B. L., Wen S.-J. et al. Comparison of combinational and sequential error rates for a deep submicron process, *IEEE Trans Nucl Sci (TNS)*, 2011, 58 (6), pp. 2719–25.

6. Telpukhov D. V., Soloviev R. A., Myachikov M. V. Razrabotka prakticheskikh metrik dlja ocenki metodov povysheniya sboeustojchivosti kombinacionnykh shem, *Informacionnye tehnologii i matematicheskoe modelirovanie sistem*, 2015, pp. 79–81 (in Russian).

7. Stempkovskiy A. L., Telpukhov D. V., Soloviev R. A., Myachikov M. V., Telpukhova N. V. Razrabotka tehnologicheskikh nezavisimykh metrik dlja ocenki maskirujushchikh svoystv logicheskikh shem, *Vychislitel'nye tehnologii*, 2016, vol. 21, no. 2, pp. 53–62 (in Russian).

8. Akers S. B., Joseph C., and Krishnamurthy B. On the Role of Independent Fault Sets in the Generation of Minimal Test Sets, *Proc. Int. Test Conf.*, August 1987, pp. 1100–1107.

9. Prasad A. V. S. S., Agrawal V. D., Atre M. V. A new algorithm for global fault collapsing into equivalence and dominance sets, *Proceedings of ITC*, October 2002, pp. 391–397.

10. URL: [http://cad-contest-2016.el.cycu.edu.tw/Problem\\_A/default.html](http://cad-contest-2016.el.cycu.edu.tw/Problem_A/default.html)

11. Brayton R., Mishchenko A. ABC: An Academic Industrial-Strength Verification Tool. Touili T., Cook B., Jackson P. (eds.) Computer Aided Verification. CAV 2010. *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2010, vol. 6174.

12. URL: [https://github.com/IDMIPPM/ICCAD\\_2016\\_Contest](https://github.com/IDMIPPM/ICCAD_2016_Contest)

13. URL: <https://www.youtube.com/watch?v=ABIGIiwc29A>

# ПРОГРАММНАЯ ИНЖЕНЕРИЯ SOFTWARE ENGINEERING

УДК 004.4'22

Ю. Н. Фонин, науч. сотр., e-mail: yurifonin@gmail.com,  
Московский физико-технический институт (государственный университет), г. Долгопрудный

## Применение языка PDDL для автоматической генерации средств разработки процессоров

*Представлено описание языка PDDL и возможностей компилятора языка в части автоматической генерации средств разработки процессоров, а также охарактеризован опыт использования PDDL для создания средств разработки ПО для процессоров семейства NeuroMatrix.*

**Ключевые слова:** язык PDDL, архитектура микропроцессора, средства разработки, языки описания архитектур, средства программирования

### Введение

Одной из проблем, связанной с разработкой сложных интегральных цифровых микросхем, является проблема совместной и синхронной разработки схемотехники и топологии микросхемы, с одной стороны, и необходимого системного программного обеспечения (ПО) — с другой [1]. В то же время процесс проектирования современных систем на кристалле (СнК) часто носит итеративный характер [2]. Для современных разработчиков СнК практически всегда стоит задача поиска оптимального решения по быстродействию, форм-фактору, потребляемой мощности и т. д. Необходимость многократного итеративного изменения и аппаратного описания и системного ПО приводит к существенному росту времени и затрат на разработку микросхем.

Сократить время проектирования СнК позволяют высокоуровневые языки описаний архитектур. Такие языки позволяют из единого описания автоматически генерировать как описание аппаратной платформы на языках VHDL и Verilog, так и программные компоненты: симуляторы, ассемблеры, отладчики и т. д. Одним из таких языков является PDDL [3]. Язык PDDL был разработан инженерами компании "РЕДА-Софт" в рамках проекта по созданию средств моделирования перспективных архитектур СнК и использовался в первую очередь для создания моделей процессоров и автоматической генерации средств разработки к ним. Впоследствии в компиляторе PDDL был реализован генератор Verilog моделей, что позволило осуществлять

прототипирование на ПЛИС ядер микропроцессоров, описанных на PDDL.

В настоящее время существуют три различных стиля для описания архитектуры процессора: структурный, поведенческий и смешанный. Зачастую при описании конкретной архитектуры используются все три типа стиля.

Структурное описание используется в основном для создания аппаратуры, поведенческое, как правило, — для программного моделирования систем и генерации системного ПО. Смешанные языки предоставляют возможности для создания и программной (модели, системное ПО), и аппаратной (VHDL, Verilog) составляющих СнК. Язык PDDL можно отнести к категории смешанных языков описания архитектур.

Структурное описание соответствует стилю описания, используемому в языках регистровых передач (*register transfer level* — *RTL*), когда архитектура представляется в виде иерархии связанных компонентов. Для спецификации аппаратуры на уровне регистровых передач используются языки HDL (*Hardware Description Languages*), наиболее известными из которых являются Verilog [4] и VHDL [5].

Примером высокоуровневого языка описания архитектур структурного типа является язык MIMOLA [6]. Описание процессора на языке MIMOLA представляется в виде набора блоков, таких как арифметико-логическое устройство, регистры и шины для доступа к памяти, а также набора соединений между блоками. Анализировать такое описание для генерации компилятора или операци-

онной системы сложно. Поэтому в языке MIMOLA предусмотрена возможность ввода дополнительной спецификации компонентов. Такие спецификации позволяют получить дополнительную информацию об элементах процессора, необходимую для автоматической генерации компиляторов, операционных систем и создания отладочного ПО.

Для решения задачи автоматической генерации системного программного обеспечения предназначены языки поведенческого класса ADL (*Architecture Description Languages*), в которых функциональность каждой инструкции описана в явном виде. Рассмотрим более подробно некоторые из них. К языкам класса ADL относится язык nML [7]. В nML система команд процессора описывается с помощью атрибутивных грамматик. Атрибуты включают в себя поведение (*action*), ассемблерный синтаксис (*syntax*) и отображение в машинные коды (*image*). В исходном варианте nML отсутствовали механизмы описания многотактовых команд. В последних версиях nML поддерживается синтез VHDL-описания.

Языки смешанного типа похожи на поведенческие языки, однако они позволяют при описании инструкций специфицировать использование конкретных аппаратных блоков, например, арифметико-логического устройства или специальных умножителей, а также описывать конвейер инструкций. В смешанных языках, как и в поведенческих, инструкции описываются в явном виде. Но при создании дерева инструкций могут возникать трудности, связанные с извлечением информации о функциональности внешних блоков.

Примером языка смешанного описания является язык LISA [8]. Он разрабатывался в качестве средства описания аппаратуры для генерации симуляторов. К ключевым характеристикам LISA можно отнести подробное описание конвейера на уровне операций с возможностью задания зависимостей и блокировок. Каждая команда специфицируется в виде набора операций, которые определяются как регистровые пересылки за время одного такта синхронизации. Описание аппаратуры на языке LISA состоит из двух основных частей: спецификации ресурсов и описания операций. LISA позволяет детально описать архитектуру процессора, что важно для синтеза микросхемы. Вместе с тем, поскольку инструкции процессора описаны в явном виде, создание систем автоматической генерации системного программного обеспечения и средств разработки на основе EXPRESSION или LISA-описаний процессора существенно проще, чем создание таких систем на основе структурных

языков. К недостаткам решения на основе этих языков по сравнению с поведенческими языками следует отнести относительную трудоемкость описания архитектуры, связанную с необходимостью детально описывать структурную составляющую. В этом смысле LISA стоит между чистыми поведенческими ADL решениями (типа nML) и структурными описаниями уровня HDL.

Перечисленные выше языки описания архитектур были разработаны в начале—середине 2000-х годов и предназначены для описания относительно простых ядер процессоров. Ядра современных микропроцессоров имеют достаточно сложную архитектуру, включающую в себя аппаратную многопоточность, длинное слово команд (VLIW), параллельное выполнение инструкций над разными данными (SIMD), а также механизм выполнения инструкций out-of-order. Все перечисленные технологии можно реализовать и с помощью указанных языков описания архитектур. Однако наличие специальных конструкций в языке описания архитектур для быстрой реализации данных технологий в процессорах, а также поддержка на уровне генерации аппаратуры и системного ПО способна существенно ускорить процесс проектирования современных СнК.

### Описание языка PPDL

Язык PPDL относится к третьему типу языков. Описание на PPDL позволяет генерировать как программные симуляторы, так и синтезируемое описание на языке Verilog.

Для декларации ядра процессора используется ключевое слово DEVICE. Декларация имеет следующий формат:

```
DEVICE ProcessorName {  
    // описание процессора  
}
```

Здесь ProcessorName — это название процессора. Генератор симулятора в случае успешной компиляции производит C++ класс с именем ProcessorName.

Описание процессора в PPDL можно условно разделить на описание ресурсов процессора и описание функциональности. Ресурсы процессора описываются множеством регистров или массивом регистров, флагов, шин для работы с памятью и портов ввода-вывода. Функциональность описывается набором (списком) инструкций и описанием процедур Operate (эмуляция такта работы процессора) и RESET (реакция на сигнал сброса процессора).

### **Описание регистров процессора**

Регистры процессора описываются в соответствии со следующим шаблоном:

```
Reg type name[array_size],  
"asm_dbg_name" [mutable];
```

Здесь `type` — это тип данных регистра, `name` — имя регистра, `array_size` — размер массива регистров (для регистровых массивов), `asm_dbg_name` — ассемблерное имя регистра. Этот параметр можно опустить, если регистр (или регистровый массив) не является аргументом инструкции или его содержание не должно быть видно в отладчике.

### **Описание флагов**

Флагом в PDDL является именованный бит регистра процессора. Флаг декларируется в соответствии со следующим шаблоном:

```
Flag name, Reg[bit_of_reg];
```

Здесь `name` — имя флага в симуляторе, `Reg` — имя регистра флагов, `bit_of_reg` — целочисленная константа, номер бита в регистре флагов.

### **Описание шины памяти**

Операции обращения к памяти процессора осуществляются через шины памяти. При описании модели в целом шины могут быть соединены с блоками памяти или мультиплексорами, разделяющими запросы к памяти по разным блокам. Шина памяти декларируется в соответствии с шаблоном

```
MemSpace name, AddrSize, DataSize;
```

Здесь `name` — имя шины, `AddrSize` — размер шины адреса, `DataSize` — размер шины данных.

### **Описание списков аргументов инструкций**

Для удобства и компактности описания и анализа инструкций ресурсы процессора можно сгруппировать в списки аргументов. Списки декларируются в соответствии с одним из следующих шаблонов:

```
ArgList name, "assembly_template",  
{arg1, arg2, arg3};
```

```
ArgList name, {"asm1", "asm2", "asm3"},  
{arg1, arg2, arg3};
```

Здесь `ArgList` — ключевое слово, обозначающее начало списка аргументов, `name` — имя списка, `"assembly_template"` — шаблон в языке ассемблера для описания аргумента в ассемблере; `"asm1", "asm2", "asm3"` — множество ассемблерных конструкций, каждая из которых соответствует элементу из спи-

ска; `arg1, arg2, arg3` — множество выражений — элементов списка. Способы использования списков будут описаны ниже.

### **Описание поведенческой модели процессора**

Поведение процессора в PDDL задается описанием работы инструкций процессора и описанием процедуры `Operate`. Процедура `Operate` эмулирует работу процессора в течение одного такта работы. В `Operate` описываются процессы чтения команды из памяти, ее декодирования, реакции системы на прерывания и т.п. И инструкции, и процедура `Operate` разделяются на ступени конвейера, что позволяет также задать параллельность исполнения инструкций.

Способы описания базовых операций в PDDL (арифметические и логические операции, присваивание, ветвление и циклы) аналогичны операциям в языках C\C++. Однако, в отличие от C\C++, в PDDL добавлен ряд специфичных операций:

1) оператор конкатенации подвыражений `Оператор [exp1, exp2]` объединяет пару выражений `exp1` (старшая часть) и `exp2` (младшая часть) в значение (если они в правой части выражения) или переменную (если они в левой части выражения). Размер объединенной переменной в битах равен сумме размеров составляющих выражений. Например, если мы имеем два регистра `ar0` и `gr0` размером по 32 бита каждый, то в конструкции `int64 x = [ar0,gr0]`; в старшие 32 бита `x` будет записано значение `ar0`, а в младшие — значение `gr0`;

2) операторы блокировки и разблокировки регистров: `lock` — заблокировать доступ к ресурсу, `unlock` — разблокировка доступа к ресурсу, `is_locked` — возвращает 1, если доступ к ресурсу заблокирован, 0 — в противном случае. Операторы блокировки-разблокировки никак не влияют на операции чтения-записи соответствующих ресурсов, а только позволяют управлять задержками и инкрементацией счетчика тактов.

### **Конвейер**

Конвейер инструкций предназначен для описания потактового разбиения инструкций. Различают два типа конвейера: конвейер инструкций и конвейер непосредственно процессора. В обоих случаях конвейер декларируется в соответствии с шаблоном

```
PIPE { stage1,stage2, ... stageN };
```

где `stage1, stage2, ... stageN` — перечисление символьных названий ступеней конвейера.

### Набор инструкций

Язык PDDL позволяет описать несколько наборов инструкций, а также вложенные инструкции, что дает возможность сформировать сложные VLIW-инструкции.

Набор инструкции описывается следующим образом:

```
ISET name
{
// описание конвейера
PIPE { stage1, stage2, ... stageN };
// описание набора инструкций
}
```

Здесь ISET — ключевое слово, определяющее начало нового набора инструкций, name — название набора инструкций.

### Описание инструкции

Инструкция в PDDL описывается в соответствии со следующим шаблоном:

```
INST Assembly_template, (OpCode_Descr)
Disasm { disasm_directives; }
assembler { assembler_directives; }
{
Decode_instruction_functionality
stage1:
Instruction_functionality_stage1
stageN:
Instruction_functionality_stageN
}
```

Здесь INST — ключевое слово, определяющее начало инструкции, assembly\_template — шаблон представления инструкции в ассемблере, OpCode\_Descr — описание формата кода команды, disasm — описание последовательности операций над аргументами (полями инструкции) на этапе дизассемблирования инструкции. Instruction\_functionality\_stageX — поведенческое описание команды на ступени конвейера stageX, Decode\_instruction\_functionality — поведенческое описание команды на этапе декодирования. Формат инструкции описывается в соответствии со следующим шаблоном:

```
(inst_size,
Type_1 argval_1 [= List_or_iset],
Type_2 argval_2 [= List_or_iset],
)
```

Здесь inst\_size — размер инструкции, argval\_N — значение N-го аргумента. Type\_N — тип аргумента, допустимы только целочисленные типы int1 — intN.

Аргумент argval может быть представлен в одном из следующих форматов: в виде целочисленной константы, представленной в десятичном, шест-

надцатеричном или двоичном виде, или именем. В таком случае при описании семантики команды это имя можно использовать в качестве аргумента операции. В ассемблере данная конструкция будет представлена как численный аргумент команды.

В формате "name = list", где name — это имя аргумента инструкции, а list — название списка аргументов или массива регистров. В ассемблере данная конструкция будет представлена как ассемблерное имя одного из аргументов списка или в виде элемента массива регистров.

В формате "name = iset", где iset — имя ранее определенного набора инструкций. В ассемблере конструкция \$name будет представлена как вложенная ассемблерная инструкция. На этапе декодирования бинарный код, соответствующий полю name, будет декодирован как вложенная инструкция. Такая вложенная инструкция декодируется в процессе декодирования основной инструкции и затем выполняется параллельно с основной. Это дает возможность строить иерархию наборов инструкций и организовывать "длинное" слово команды, которое (слово) включает в себя несколько параллельно исполняющихся инструкций.

Если аргумент arg\_val представляется в виде целочисленной константы, то допустима дополнительная спецификация типа релокации метки. Тип релокации метки определяет, как рассчитывать значение константы относительно адреса метки. В PDDL предусмотрены два типа релокации — абсолютная, при которой в код инструкции вставляется абсолютный адрес метки, и релокация относительно счетчика команд. Абсолютная релокация задается по умолчанию. Для задания относительной релокации используется ключевое слово RELOC\_PC. Например, при описании аргумента int32 branch\_addr RELOC\_PC адрес метки будет записан как смещение относительно адреса инструкции.

Пример описания инструкции:

```
INST "ADD $dst, $src1, $src2",
(32,
int4 0,
int4 src1 = r, // первое слагаемое —
регистр из списка r
int4 src2 = r, // второе слагаемое —
регистр из списка r
int4 dst = r, // результат операции —
регистр из списка r
int4 0,
int12 0)
```

Формат шаблона команды ассемблера представляется в виде текстовой строки, в которой могут присутствовать любые символы, за исключением символов перехода строки. Для задания аргумен-

тов команды используется символ "\$idx", где idx — это число от 1 до 9, определяющее номер аргумента в коде команды (начиная с первого). По умолчанию, если с аргументом в описании кода команды не сопоставлен какой-либо список или массив регистров, аргумент "\$idx" подразумевает целую константу.

Ниже приводится пример реализации простой инструкции на PDDL — сложения значений двух регистров (src1 и src2) из списка RALL и запись суммы в регистр dst из списка RALL.

```
INST "ADD $1,$2,$3", (32,  
int4 0x0,  
int4 src1 = r,  
int4 src2 = r,  
int4 dst = r,  
int16 0x0)  
{  
ra: // первый такт выполнения инструкции —  
    сложение T1 = src1 + src2;  
rb: // второй такт выполнения инструкции —  
    запись в регистр dst = T1;  
}
```

Здесь r — Reg r[16],32,"r[\$]"; массив из 32-битных регистров общего назначения.

В инструкции содержатся три аргумента src1, src2, dst. Значение каждого из аргументов на языке ассемблера задается как регистр из массива r. Инструкция работает следующим образом: на шаге "ra" сумма значений регистров, заданных аргументами src1 и src2, сохраняется в регистре T1. На втором шаге значение регистра T1 сохраняется в регистре dst. Пример инструкции на языке ассемблера:

```
ADD r[0],r[2],r[3]
```

### **Функция Operate**

Процедура Operate вызывается на каждый такт работы процессора. В ней описывается цикл загрузки и выполнения инструкций, а также обработчики прерываний.

Помимо стандартных арифметических, логических операций, операций условного перехода и чтения/записи в память, в функции Operate доступен следующий набор вызовов для управления процессом декодирования и исполнения команд:

ISET\_NAME.decode(inst) — функция декодирования инструкции из набора ISET\_NAME. Здесь inst — это код инструкции. Функция decode возвращает размер декодированной команды в битах или 0, если входное значение не может быть декодировано как команда;

ISET\_NAME.isset\_stage() — функция, которая запускает ступень iset\_stage конвейера набора ин-

струкций ISET\_NAME на исполнение. В случае, если на предыдущем такте инструкция не была декодирована или на предыдущей стадии дальнейшего прохождения инструкции было остановлено, iset\_stage не выполняет никаких операций.

### **Функция RESET**

Функция RESET вызывается при аппаратном сбросе процессора, т. е. при подаче внешнего сигнала RESET. В функции RESET инициализируются регистры, переменные флаги и пр.

### **Результаты применения языка PDDL при создании средств разработки для процессоров NeuroMatrix**

Апробация языка PDDL была осуществлена в рамках работ по созданию средств разработки для семейства процессоров NeuroMatrix компании ЗАО НТЦ "Модуль".

В процессе работы была создана функциональная модель процессора на языке PDDL. На основе созданной модели были автоматически сгенерированы:

- симуляторы процессоров NM6403, NM6405, NM6407 в виде C++ классов;
- симулятор процессора NM6407 на основе QEMU. С помощью компилятора PDDL был сгенерирован код функций;
- функция кодирования инструкций для GNU ассемблера;
- функция декодирования инструкций для утилиты GNU objdump, а также для дизассемблера в отладчике GDB.

Все перечисленные выше компоненты были успешно протестированы на нескольких тысячах разработанных тестов. Кроме того, для тестирования связки "компилятор — ассемблер — линковщик — симулятор — отладчик" были использованы тесты deja-gnu, которые показали корректность работы ассемблера с автоматически сгенерированным кодировщиком инструкций, симуляторов и отладчика gdb.

Разработка полного описания процессора на языке PDDL, включая отладку кода симулятора, заняла без учета затрат на разработку собственно компилятора языка PDDL, а также затрат на модификацию платформу-независимых частей QEMU и GNU assembler и разработку стаба GDB примерно 6 человеко-месяцев. Трудозатраты на модификацию платформу-независимых частей можно исключить из общих затрат, поскольку эти модификации сделаны единожды для всех архитектур, которые генерируются автоматически с помощью компилятора PDDL. Дополнительно было затрачено 0,5 человеко-

месяца на отладку ассемблерных мнемоник инструкций в части написания корректных шаблонов ассемблерных инструкций на PPDL. Для реализации описанных выше компонент без использования средств автоматизации потребовалось бы не менее 18 человеко-месяцев.

### Заключение

В статье описан новый язык описания ядер микропроцессоров PPDL. С помощью PPDL был разработан набор средств разработки для процессоров семейства NeuroMatrix. Результаты применения PPDL для создания средств разработки показали существенное сокращение времени и стоимости работ по созданию средств разработки для микропроцессоров.

### Список литературы

1. **Kempf T., Ascheid G., Leupers R.** Multiprocessor Systems on Chip: Design Space Exploration. Springer, 2016. 206 p.

2. **Bricau P.** Reuse Methodology Manual for System-On-A-Chip Designs. Kluwer, 2002. 292 p.

3. **Дроздов А. Ю., Фонин Ю. Н.** Применение языка PPDL для автоматической генерации средств разработки процессоров семейства NeuroMatrix // Международный форум "Микроэлектроника—2016", 2-я научная конференция "Интегральные схемы и микроэлектронные модули". Республика Крым, г. Алушта, 26—30 сентября 2016. С. 93.

4. IEEE Standard Verilog Hardware Description language, IEEE Std 1364-2001. URL: <https://inst.eecs.berkeley.edu/~cs150/fa06/Labs/verilog-ieee.pdf>

5. VHDL Reference Manual. URL: <http://www.ics.uci.edu/~jmoorkan/vhdlref/Synario%20VHDL%20Manual.pdf>

6. **Bashford S., Bieker U., Harking B., Leupers R., Marwedel P., Neumann A., Voggenauer D.** The MIMOLA Language Version 4.1. Technical report, Lehrstuhl Informatik XII, University of Dortmund, Dortmund, 1994. URL: [https://www.researchgate.net/publication/2272489\\_The\\_MIMOLA\\_Language\\_Version\\_41](https://www.researchgate.net/publication/2272489_The_MIMOLA_Language_Version_41)

7. **Fauth A., van Praet J., Freericks M.** Describing instruction set processors using nML // Proc. of the European Design and Test Conference (ED & TC), 1995. P. 503—507.

8. **Zivojnovic V., Pees S., Meyr H.** LISA — Machine Description Language and Generic Machine Model for HW/SW Co-Design // Proc. of the IEEE Workshop on VLSI Signal Processing, 1996. P. 126—137. URL: <https://www.ice.rwth-aachen.de/fileadmin/publications/Zivojnovic96vlsi.pdf>

**Yu. N. Fonin**, Research Associate, e-mail: [yurifonin@gmail.com](mailto:yurifonin@gmail.com),  
Moscow Institute of Physics and Technology (State University)

## Using PPDL Language for Automatically Generation of the Tool Chain for Processors

*The modern system on chip development technologies requires an iterative approach, where the hardware and software is modified several times to achieve the best characteristics for the specific benchmark. To accelerate the iterative development process a special high-level architecture description language PPDL has been developed. The paper describes the syntax of PPDL. A set of constructions for resource description and behavior of processor is present. Constructions for the declaration of registers, flags, memory buses, IO channels as well as groups of resources, called in PPDL as sets, are given. A behavioral part of the processor's description is present by instructions's description and pipeline description. Beside the PPDL constructions description, the paper describes experience of using PPDL for automatically generation of the tool chain components for NeuroMatrix processor. The example of using PPDL for the NeuroMatrix tool chain development showed that PPDL allows to accelerate a tool chain development about three times compare to manual development of the separated components.*

**Keywords:** PPDL, architecture of microprocessor, tool chain, architecture description language

### References

1. **Kempf T., Ascheid G., Leupers R.** Multiprocessor Systems on Chip: Design Space Exploration. Springer, 2016. 206 p.

2. **Bricau P.** Reuse Methodology Manual for System-On-A-Chip Designs. Kluwer, 2002. 292 p.

3. **Drozдов А. Ю., Фонин Ю. Н.** Применение языка PPDL для автоматической генерации средств разработки процессоров семейства NeuroMatrix, *Международный форум "Микроэлектроника—2016", 2-я научная конференция "Интегральные схемы и микроэлектронные модули"*. Республика Крым, г. Алушта, 26—30 сентября 2016, p. 93 (in Russian).

4. IEEE Standard Verilog Hardware Description language, IEEE Std 1364-2001. URL: <https://inst.eecs.berkeley.edu/~cs150/fa06/Labs/verilog-ieee.pdf>

5. VHDL Reference Manual. URL: <http://www.ics.uci.edu/~jmoorkan/vhdlref/Synario%20VHDL%20Manual.pdf>

6. **Bashford S., Bieker U., Harking B., Leupers R., Marwedel P., Neumann A., Voggenauer D.** The MIMOLA Language Version 4.1. Technical report, Lehrstuhl Informatik XII, University of Dortmund, Dortmund, 1994. URL: [https://www.researchgate.net/publication/2272489\\_The\\_MIMOLA\\_Language\\_Version\\_41](https://www.researchgate.net/publication/2272489_The_MIMOLA_Language_Version_41)

7. **Fauth A., van Praet J., Freericks M.** Describing instruction set processors using nML, *Proc. of the European Design and Test Conference (ED & TC)*, 1995. pp. 503—507.

8. **Zivojnovic V., Pees S., Meyr H.** LISA — Machine Description Language and Generic Machine Model for HW/SW Co-Design, *Proc. of the IEEE Workshop on VLSI Signal Processing*, 1996, pp. 126—137. URL: <https://www.ice.rwth-aachen.de/fileadmin/publications/Zivojnovic96vlsi.pdf>

# БЕЗОПАСНОСТЬ ИНФОРМАЦИИ CRYPTOSAFETY INFORMATION

УДК 004.738

**Р. Э. Асратян**, канд. техн. наук., вед. науч. сотр., e-mail: rea@ipu.ru,  
**В. Н. Лебедев**, канд. техн. наук., зав. лаб., e-mail: lebvini@ipu.ru,  
Институт проблем управления им. В. А. Трапезникова РАН

## Интернет-служба конвейерной обработки защищенных информационных запросов в мультисетевой среде

*Рассмотрены принципы организации сетевой службы, предназначенной для реализации защищенной обработки запросов в распределенных информационных системах, ориентированных на работу в сложных мультисетевых средах со многими обрабатываемыми серверами. Описано применение этой службы для выполнения "конвейерной" обработки запросов в цепочке серверов и маршрутизации запросов между удаленными частными сетями.*

**Ключевые слова:** распределенные системы, Web-технологии, Интернет-технологии, информационное взаимодействие, информационная безопасность

### Введение

Появление сетевой архитектуры .Net и технологии Web-сервисов, безусловно, открыло новый этап в теории и практике создания распределенных информационных систем. Признанные достоинства этой технологии (высокая гибкость в определении сервисных функций, высокое быстродействие, эффективная поддержка в целом ряде современных систем программирования и т. п. [1, 2]) позволили ей быстро занять ведущее положение в этой области. Тем не менее, разработчики распределенных систем нередко испытывают трудности с обеспечением защиты данных в сети. Эти трудности чаще всего бывают связаны с отсутствием в архитектуре .NET встроенных средств защиты и аутентификации сетевых сообщений и более всего проявляются в разработках систем, предназначенных для работы в сложных, мультисерверных и мультисетевых средах [3] в условиях высоких требований к информационной безопасности [4, 5].

В работе [6] описана новая сетевая служба PMS (Protected Message Service), разработанная в целях преодоления вышеуказанного недостатка. Суть подхода заключается в тесной интеграции функций сетевого информационного обмена с функциями защиты и аутентификации данных. Внешне эта интеграция проявляется в том, что отмеченные функции входят в набор методов главного программного класса службы — класса "Защищенное сообщение", отображающего электронный документ (информационный запрос или ответ), снабженный одной или несколькими удостоверяющими

электронными цифровыми подписями (ЭЦП). В отличие, например, от технологии Web-сервисов, описываемая служба опирается не на модель вызова методов удаленных объектов, а на модель обмена сообщениями. В данном случае это означает, что все сервисные обрабатываемые функции (методы) имеют одинаковую, жесткую спецификацию: они получают объект класса "Защищенное сообщение" в качестве параметра и возвращают объект того же класса. Эти обрабатываемые функции группируются в одну или несколько динамических библиотек, которые подключаются к серверу PMS в момент его запуска (каждая библиотека может рассматриваться как отдаленный аналог Web-сервиса в .NET), и становятся доступными для клиентских компонент.

Базисное свойство PMS — фиксированная спецификация сервисных функций с объектами одного и того же класса на входе и на выходе функции — дает принципиальную возможность организации "программного конвейера", основанного на направлении объекта класса "Защищенное сообщение" с выхода одной функции на вход другой. Другими словами, речь идет об обработке информационного запроса не одной сервисной функцией, а цепочкой функций таким образом, что защищенное сообщение, возвращенное каждой сервисной функцией, или передается непосредственно на вход следующей функции в цепочке, если она имеется, или возвращается клиенту (PMS-конвейер). Разумеется, здесь имеется прямая аналогия с известным еще с первых версий системы UNIX механизмом "трубопровода" (pipeline), основанном

на последовательном соединении стандартных выводов и вводов у нескольких процессов в компьютере. Однако поскольку в данном случае речь идет о сетевой службе, сетевых сообщениях и распределенных системах, наибольший интерес представляет мультисерверная организация конвейера, в которой сервисные функции, задействованные в обработке информационного запроса, могут выполняться на разных серверах (в том числе на серверах, размещенных в разных сетях).

В данной работе рассматриваются два связанных между собой механизма в составе PMS: конвейерная мультисерверная обработка информационных запросов и межсерверная маршрутизация защищенных сообщений в сложных мультисетевых средах.

### Краткие сведения о PMS

Как и всякая сетевая служба, основанная на базовом сетевом протоколе TCP/IP [7], PMS поддерживается клиентским и серверным программным обеспечением. Сервер PMS представляет собой постоянно активную программу, обслуживающую запросы на обработку от клиентов (по умолчанию используется порт 8132). Клиентское программное обеспечение представляет собой библиотеку функций PmsBase.dll, реализующих прикладной программный интерфейс (API) к PMS. Этот интерфейс является "лицом" PMS с точки зрения пользователя.

На рис. 1 представлен фрагмент кода в нотации C#, иллюстрирующий простое обращение к обрабатывающей функции с применением средств защиты данных. Две первые строки кода определяют две переменные типа PmsMessage, представляющего собой главный класс PMS "Защищенное сообщение". Первой переменной (Request) присваивается значение: объект класса PmsMessage, инициализированный символьной строкой (например, содержащий XML-документ информационного запроса). Вторая переменная (Reply) предназначена для хранения

результата обработки. В третьей строке определяется инициализируется переменная MyConn класса PmsConnection, предназначенного для создания и прекращения сетевого соединения с сервером.

В четвертой и пятой строках кода выполняется формирование подписей в запросе. Сначала определяется переменная SenderCerts класса PmsCertList. Этот класс предназначен для хранения в памяти списков сертификатов с открытыми ключами в стандарте X509 и содержит несколько конструкторов для загрузки сертификатов из файлов или из системных хранилищ с поиском по имени владельца или по серийному номеру. В переменную SenderCerts загружаются два сертификата, соответствующие именам владельцев "Иванов" и "Петров", для последующего формирования двух ЭЦП в запросе. (Такой способ использования означает, что с этими сертификатами обязательно должны быть связаны парные им закрытые ключи, иначе формирование ЭЦП закончится неудачно.) Вызов метода AddSignatures позволяет сформировать две ЭЦП в сообщении Request.

Собственно вызов обрабатывающей функции начинается с вызова метода Connect, устанавливающего сетевое соединение с сервером с указанным сетевым именем или адресом. Далее выполняется первая сетевая операция — запрос сертификата сервера (метод GetServerCertificate) с занесением результата в переменную ReceiverCert уже знакомого нам класса PmsCertList для последующего шифрования информационного запроса. Сразу же после успешного получения сертификата сервера выполняется вызов удаленной функции с помощью применения метода Process к переменной Request с использованием все того же соединения и полученного сертификата. Предполагается, что к серверу PMS подключена библиотека MyLib.dll, содержащая код функции MyFunc. При запуске функции на сервере ей передаются значение переменной Request и опциональный строковый параметр param в качестве фактических параметров. Результат обработки заносится в переменную Reply. Подчеркнем, что шифрование запроса и дешифрование ответа выполняются автоматически в методе Process.

Последующие строки обеспечивают последовательную проверку всех ЭЦП в полученном ответе сервера (вызов метода VerifySignature) и запись в стандартный вывод сведений о подписантах.

Пример заканчивается записью результата обращения в стандартный вывод (предполагается, что результат, как и запрос, имеет форму символьной строки) и закрытием соединения с сервером с помощью метода Disconnect, так как в данном примере оно больше не нужно.

Необходимо отметить, что из обоих фрагментов кода намеренно удалены операторы обработки исключений.

```
PmsMessage Request= new PmsMessage("<request> ... </request>");
PmsMessage Reply;
PmsConnection MyConn = new PmsConnection();
PmsCertList SenderCerts = PmsCertList( new string [] {"Иванов", "Петров"});
Request.AddSignatures(SenderCerts);

MyConn.Connect("MyServer");
PmsCertList ReceiverCert = MyConn.GetServerCertificate();

Reply=Request.Process (MyConn, "MyLib.MyFunc param", ReceiverCert);
if(Reply != null)
{
    string Signer;
    for(int i=0; Reply.Signatures.Length; i++)
        if(Reply.VerifySignature(i, out Signer) >= 0)
            Console.WriteLine ("Ответ подписал: " + Signer);
    Console.WriteLine (Reply.GetString());
}
else
    Console.WriteLine ("Ошибка: " +MyConn.ErrMsg);
MyConn.Disconnect();
```

Рис. 1. Пример использования PMS

Как уже отмечалось, все сервисные обрабатывающие функции группируются в одну или несколько динамических библиотек, которые подключаются к серверу PMS в момент его запуска. Никакие специальные конструкции типа Web-сервисов не используются. Поиск и загрузка всех динамических библиотек выполняются при запуске сервера PMS из каталога, указанного в конфигурации сервера. В каждой найденной библиотеке проводятся поиск и подключение всех функций, имеющих строго определенную спецификацию:

**PmsMessage имя\_функции (PmsMessage Inpt, string [] Conf, ref string Msg),**

в которой параметр Inpt — входное сообщение (запрос); параметр Conf — конфигурационные данные в форме массива строк вида "имя = значение"; строковый параметр Msg на входе принимает любое значение, а возвращает диагностическое сообщение функции. С этого момента все библиотечные функции, соответствующие данной спецификации, начинают играть роль сервисных функций, доступных для клиентских программ. Все остальные функции попросту игнорируются.

### Сетевой протокол PMS

Класс PMSMessage содержит всего два поля: массив байтов (Data), представляющий собой "контейнер" обрабатываемых данных, и массив ЭЦП (Signatures). Каждая ЭЦП представлена объектом класса Signature и содержит свертку (digest) массива Data, зашифрованную закрытым ключом подписанта, и сертификат структуры X509, содержащий парный открытый ключ. Важно отметить, что пользователю совершенно необязательно знать устройство этого класса.

PMS в полной мере использует двоичную природу TCP/IP [7, 8]. Взаимодействие между клиентом и сервером PMS осуществляется по специальному, достаточно простому PMS-протоколу, ориентированному на передачу двоичных сетевых сообщений (PMS-сообщений) в обоих направлениях. Каждое такое сообщение в общем случае содержит два массива байтов: заголовок сообщения и тело сообщения (рис. 2). Первые четыре байта заголовка или тела сообщения содержат целое число — его длину. При передаче запроса от клиента к серверу в заголовок сетевого сообщения помещается строка, содержащая полное имя вызываемой функции, а в тело сообщения упаковывается структура PmsMessage в открытой или зашифрованной форме, содержащая информационный запрос. Строка заголовка используется сервером для организации вызова соответствующей обрабатывающей функции. При передаче результата обработки от сервера к клиенту в заголовок сетевого сообщения помещается строка диагностического сообщения (значение параметра Msg, сформированное обрабатывающей

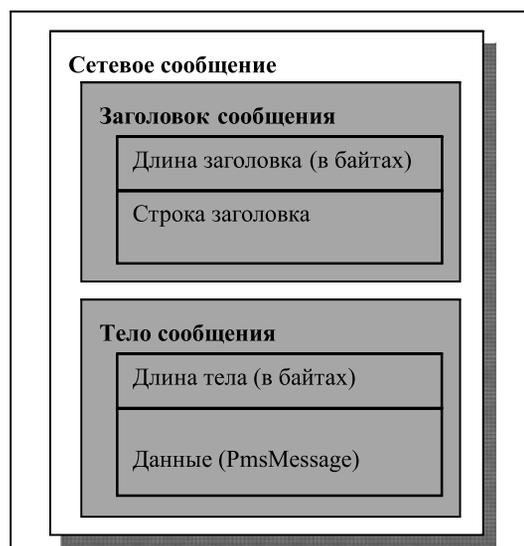


Рис. 2. Формат сетевого сообщения

функцией), а в тело сообщения упаковывается структура PmsMessage, содержащая ответ сервера в открытой или зашифрованной форме, предварительно подписанный собственным закрытым ключом сервера. Никакие двоично-текстовые преобразования (типа base64) не применяются. Полученное от сервера диагностическое сообщение автоматически присваивается члену ErrMsg объекта класса PmsConnection на стороне клиента (см. рис. 1).

### Принципы организации PMS-конвейера

Как уже отмечалось, главное свойство сервисных функций заключается в том, что они получают объект класса PmsMessage (защищенное сообщение) в качестве основного параметра и возвращают объект того же класса в качестве результата. Это создает принципиальную возможность организации конвейерной обработки сообщений на сервере PMS, т. е. последовательной обработки входного сообщения цепочкой сервисных функций.

Предположим, например, что в библиотеке Flib.dll имеются пять сервисных функций:

- функция List получает во входном сообщении строку, содержащую спецификацию папки на сервере, и возвращает построчный список имен файлов в папке;
- функция Filter получает построчный список имен файлов и оставляет в нем только файлы с указанным расширением имени;
- функция Sort сортирует список файлов;
- функция Cut оставляет в списке файлов только заданное число элементов;
- функция GetFiles получает список имен файлов во входном сообщении и упаковывает все эти файлы в выходное сообщение в форме массива байтов (способы упаковки нескольких файлов в

один байтовый массив мы не будем рассматривать в этой статье).

Рассмотрим следующее обращение к серверу PMS, в котором параметр метода Process содержит несколько имен сервисных функций, разделенных запятыми ("конвейерный список"):

```
PmsMessage Request=PmsMessage("c:\\PmsSrv\\Datafiles");
Request.AddSignatures(SenderCerts);
MyConn.Connect("alpha");
...
PmsMessage Reply = Request.Process(MyConn,
"Flib.List, Flib.Filter jpg, Flib.Sort, Flib.Cut 10,
Flib.GetFiles", ReceiverCert);
```

Здесь переменные MyConn, SenderCerts и ReceiverCert имеют тот же смысл, что и на рис. 1. Результатом данного обращения является отсортированный список имен файлов с расширением "jpg", содержащихся в папке "c:\PmsSrv\Datafiles" на сервере PMS, усеченный до 10 элементов. Предполагается, что функция Filter получает требуемое значение расширения ("jpg") через строковый формальный параметр Msg (см. предыдущий раздел). Точно так же функция Cut получает максимально допустимое число элементов в списке (в данном случае 10). Процесс обработки для данного приме-

ра проиллюстрирован на рис. 3. Легко видеть, что функции сетевого взаимодействия и функции криптозащиты (формирования ЭЦП, шифрования, дешифрования и проверки ЭЦП) в данном случае должны применяться только при передаче запроса серверу и возврате результата клиенту. Если бы клиентская программа вызывала сервисные функции последовательно и поочередно, то все это пришлось бы повторять при каждом вызове. Именно в этом заключается положительный эффект применения PMS-конвейера с точки зрения скорости обработки.

На рис. 4 отображены экспериментальные оценки быстродействия конвейерной и поочередной обработок для рассмотренного примера в форме диаграммы в двух режимах: при применении только ЭЦП и при применении ЭЦП вместе с шифрованием данных на основе криптосистемы "КриптоПро CSP" версии 3.6. Эксперимент проводился в скоростной локальной сети (100 мбит/с) с использованием четырехъядерного сервера с тактовой частотой 2,4 ГГц и операционной средой Window 2003 Server, а также одноядерной клиентской рабочей станции с тактовой частотой 2,8 ГГц. Хотя приведенные оценки относятся к конкретному примеру (из удаленного каталога было получено 10 маленьких картинок в формате JPG общим размером 50 Кбайт), они показывают, что выигрыш в быстродействии от применения конвейерной обработки может быть весьма значительным (т. е. измеряться в разгах, а не в процентах).

Применение PMS-конвейера для распределенной, мультисерверной обработки реализуется по следующим правилам.

- Кроме имен сервисных функций в конвейерном списке может использоваться еще один элемент — "вызов удаленного сервера". Этот элемент имеет следующий формат:

"имя\_сервера/(имя<sub>1</sub>, имя<sub>2</sub>, ..., имя<sub>N</sub>)",

где каждое "имя" может, в свою очередь, представлять собой имя функции или же вложенный вызов удаленного сервера. Если  $N = 1$ , то скобки можно отбросить: "имя\_сервера/имя<sub>1</sub>".

- Как и всякая многоканальная служба, сервер PMS создает отдельный обрабатывающий программный поток ("нить") для обслуживания каждого запроса на сетевое соединение от клиента. Этот поток оснащается собственной структурой данных, обеспечивающей выполнение не только серверных, но и клиентских функций с использованием классов PmsConnection и PmsMessage. Обработка каждого вложенного вызова удаленного сервера, содержащегося в конвейерном списке, выливается в обращение к другому серверу с передачей ему на обработку соответствующего фрагмента конвейерного списка.

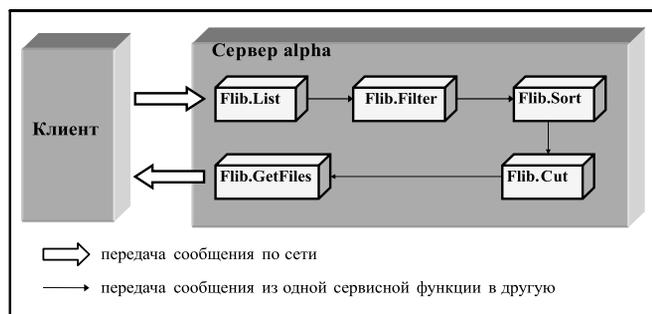


Рис. 3. Пример конвейерной обработки

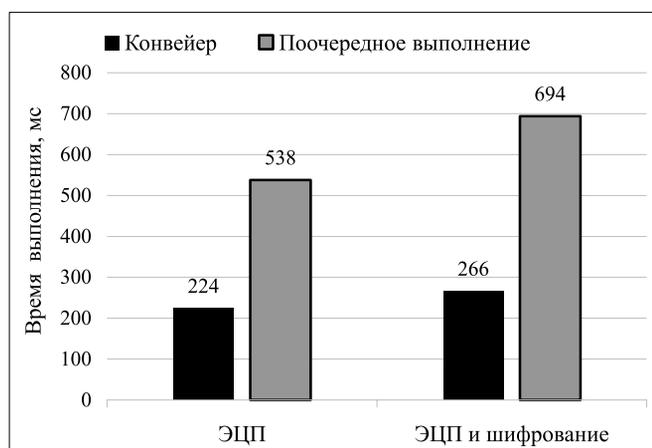


Рис. 4. Пример соотношения скоростей конвейерной и последовательной поочередной обработок

- Организация обращения к удаленному серверу включает формирование собственной ЭЦП обрабатывающего сервера для передаваемого защищенного сообщения (с помощью метода AddSignatures), соединение с удаленным сервером (с помощью метода Connect) и получение сертификата удаленного сервера с помощью метода GetServerCertificate. После завершения обработки проводится проверка всех ЭЦП в возвращенном защищенном сообщении с прерыванием обработки в случае неудачи.

Важно подчеркнуть, что проверка ЭЦП на сервере PMS может включать не только аутентификацию входящего защищенного сообщения, но и выборочную проверку реквизитов подписантов (имя, организация, должность и т. п.) по специальным спискам доступа, содержащимся в конфигурационных данных сервера.

Формирование собственной ЭЦП сервера PMS осуществляется с помощью сертификата, указанного в его конфигурационных данных: серийный номер сертификата в одном из системных хранилищ или имя файла, содержащего сертификат (разумеется, с этим сертификатом должен быть связан закрытый ключ, иначе формирование ЭЦП окажется невозможным). Функции проверки ЭЦП у входящих сообщений, а также подписания и шифрования исходящих являются опциональными и могут выборочно "отключаться" и "включаться" через конфигурационные данные сервера.

Несколько усложним рассмотренный выше пример (см. рис. 3) в направлении мультисерверной обработки. Предположим, что кроме сервера PMS с именем alpha в нашей сети имеются еще два дополнительных сервера: beta и gamma. Причем сервисные функции Filter и Cut имеются в библиотеке Flib2 на сервере beta, а функция Sort — в библиотеке Flib3 на сервере gamma. На рис. 5 проиллюстрирована мультисерверная обработка, соответствующая следующему вызову метода Process:

```
MyConn.Connect("alpha");
```

```
...
```

```
PmsMessage Reply = Request.Process(MyConn,
"Flib.List, beta/(Flib2.Filter jpg,
gamma/Flib3.Sort, Flib2.Cut 10), Flib.GetFiles",
ReceiverCert);
```

Здесь переменные MyConn, Request, Reply и ReceiverCert имеют тот же смысл, что и на рис. 1.

На рис. 6 отображены экспериментальные оценки времени выполнения одиночных запросов при обычной конвейерной обработке и при мультисерверной конвейерной обработке для обоих рассмотренных примеров (см. рис. 3 и 5) в двух режимах: при применении только ЭЦП и при применении ЭЦП вместе с шифрованием данных. Эксперименты проводились в одинаковых условиях и с одним и тем же результатом выполнения за-

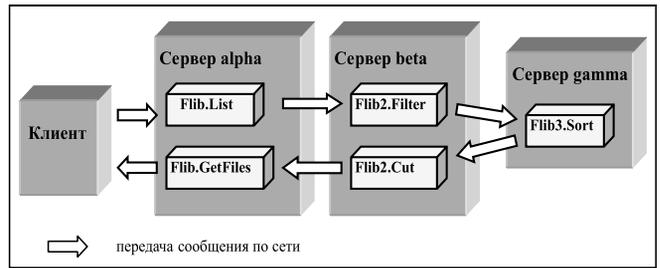


Рис. 5. Пример мультисерверной конвейерной обработки

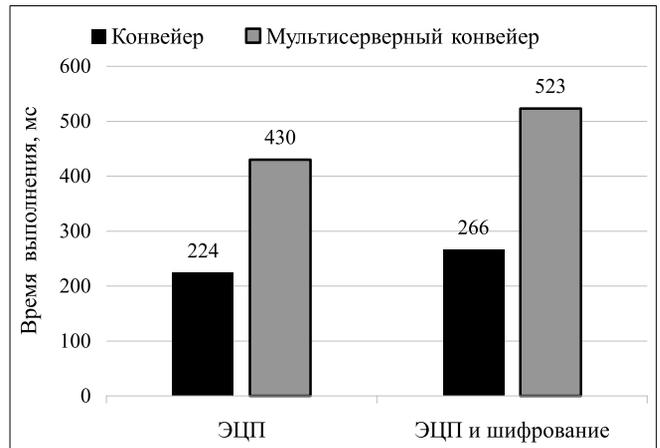


Рис. 6. Пример соотношения скоростей обычной конвейерной и мультисерверной конвейерной обработки

проса. Легко заметить, что в данных примерах переход к мультисерверной обработке привел к примерно двукратному увеличению времени выполнения ("накладные расходы" на распределенную обработку и межсерверное взаимодействие).

Как уже отмечалось, сервер PMS имеет многоканальную организацию, т. е. он создает отдельный обрабатывающий программный поток ("нить") для обслуживания каждого запроса. На рис. 7 приве-

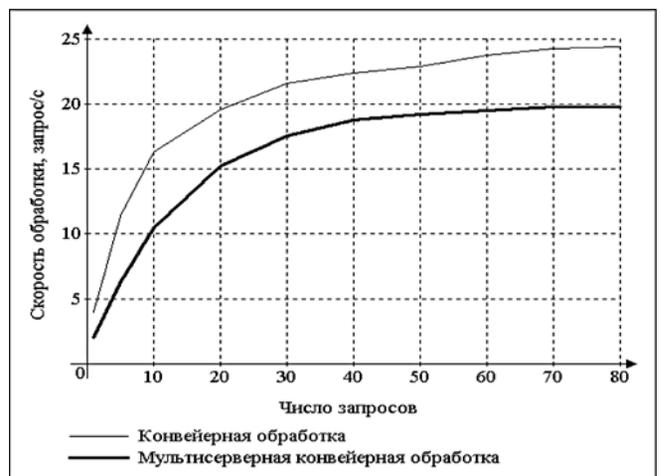


Рис. 7. Скорость обработки пакетов одновременных запросов

дены результаты экспериментальной оценки быстродействия PMS-конвейера в условиях высокой нагрузки — при параллельной обработке пакетов одновременно поступивших информационных запросов с применением криптозащиты (ЭЦП и шифрования). Характерные кривые, представленные на рис. 7, отражают зависимость скорости обработки запросов от их числа в пакете для обоих рассмотренных примеров (см. рис. 3 и 5). Скорость обработки вычислялась как частное от деления числа запросов в пакете на полное время его выполнения.

Как видно из рисунка, обе кривые ведут себя в целом одинаково. При увеличении числа запросов в пакете растет и скорость обработки, что объясняется положительным эффектом от многопоточной обработки запросов в сервере PMS. Например, при 20 запросах в пакете скорость обработки достигает 19,6 запросов в секунду для обычного конвейера и 15,2 — для мультисерверного. Однако при дальнейшем увеличении размеров пакета рост скорости обработки замедляется, а потом и совсем останавливается вследствие достижения предельной производительности.

В целом результаты данного эксперимента позволяют сформулировать следующие выводы:

- применение PMS-конвейера не разрушает положительного эффекта от многопоточной обработки запросов;
- в режиме конвейерной обработки PMS вполне позволяет поддерживать быстродействие до нескольких и даже нескольких десятков запросов в секунду с использованием средств криптозащиты, что обычно бывает достаточным для большинства информационных систем.

Как видно из правил организации мультисерверной обработки, если клиент установил соединение с сервером alpha, а в качестве параметра метода Process задал строку "beta/MyLib.MyFunc", то обработка запроса будет фактически выполнена на сервере beta, а серверу alpha достанется лишь роль посредника ("прокси-сервера"). Тем не менее, эта роль может заключаться не только в простом перенаправлении клиентского запроса в сервер beta, но и в выполнении функций информационной защиты. Предположим, например, что сервер beta размещен в частной локальной сети предприятия и недоступен клиентам из глобальной сети для прямого соединения, а сервер alpha занимает положение "пограничного" сервера, подключенного к обеим сетям: и к частной, и к глобальной. В этой ситуации сосредоточение функций информационной защиты в сервере alpha могло бы защитить всю частную сеть предприятия (включая сервер beta и, возможно, другие серверы) от проникновения в нее несанкционированных запросов из глобальной сети. Особенности применения PMS в сложных сетевых структурах рассмотрены в следующем разделе.

## Маршрутизация защищенных сообщений в мультисетевой среде

Интернет изначально строился как "сеть сетей". Частные локальные сети организаций, соединенные глобальной сетью, являются важнейшей составной частью "делового" Интернета. Статус частной сети предполагает высокую степень изолированности от остальной среды: прямое сетевое соединение между программами, одна из которых работает в частной сети, а другая — вне нее, невозможно без применения специальных средств (например, трансляторов сетевых адресов в IP-пакетах или серверов-посредников).

Если компоненты распределенной системы разнесены по разным частным локальным сетям предприятий, то возникает проблема организации безопасного взаимодействия между ними через глобальную сеть. Обычный способ ее решения заключается в применении технологии VPN (Virtual Private Network) [8] — технологии построения защищенных каналов взаимодействия через общедоступную глобальную сеть, позволяющей связать две или более удаленных локальных сети в одну территориально распределенную частную сеть с единым жестким администрированием, гарантирующим уникальность IP-адресов в пределах всей распределенной сети. Технология основана на передаче IP-датаграмм "поверх" другого протокола, оснащенного средствами информационной защиты (например, PPP или IPSec). Однако при наличии большого числа частных сетей, принадлежащих различным организациям, требование жесткого централизованного администрирования становится практически нереализуемым.

Как видно из предыдущего раздела, в функциональность сервера PMS заложено "поведение" прокси-сервера, обеспечивающего защиту данных. Для решения проблемы организации безопасного взаимодействия между удаленными частными сетями эта функциональность должна быть дополнена еще одним важным механизмом — средствами маршрутизации информационных запросов в мультисетевой среде. Эта маршрутизация является составной частью обработки элемента "вызов удаленного сервера" в конвейерном списке и осуществляется по следующим правилам.

- При обработке элемента "вызов удаленного сервера" в конвейерном списке сервер PMS сравнивает имя адресуемого сервера с собственным именем, содержащимся в его конфигурационных данных. Если эти имена совпадают, сервер PMS самостоятельно обрабатывает соответствующую этому вызову вложенную конвейерную строку. В противном случае весь элемент "вызов удаленного сервера" (вместе с именем сервера и вложенным конвейерным списком) передается на обработку другому серверу.

- Перед соединением с удаленным сервером сервер PMS выполняет собственно процедуру маршрутизации с использованием таблицы маршрутизации, которая является опциональной частью его конфигурационных данных. Используя имя удаленного сервера в качестве ключа, сервер PMS осуществляет поиск строки таблицы маршрутизации, соответствующей этому ключу, и извлекает из найденной строки Интернет-имя или IP-адрес удаленного сервера для выполнения сетевого соединения. Если же поиск строки заканчивается неудачно, то исходное имя удаленного сервера интерпретируется как его Интернет-имя.
- После соединения с удаленным сервером и получения от него сертификата для шифрования защищенного сообщения сервер PMS выполняет не только обычную проверку корректности полученного сертификата, но и выборочную проверку реквизитов его владельца (имя, организация, должность и т.п.) для защиты от фальсифицированного удаленного сервера. Поэтому кроме Интернет-имени или IP-адреса удаленного сервера строка таблицы маршрутизации может содержать и список контрольных значений реквизитов владельца для выполнения такой проверки. Если этот список отсутствует в найденной строке таблицы, то запрос сертификата у удаленного сервера и шифрование сообщения попросту не выполняются.
- Даже если используются всего два сервера PMS, при неудачной подготовке таблиц маршрутизации возможна ситуация "вечного заикливания" сообщения между ними. Для его предотвращения в заголовок сетевого сообщения добавляется специальное поле — однобайтовый обратный счетчик межсерверных передач. Это поле иницируется максимальным значением (конфигурируемый параметр) и декрементируется всякий раз, когда один сервер направляет защищенное сообщение другому без какой-либо обработки (аналог поля TTL в заголовке IP-пакета [7]). При обнулении счетчика обработка сообщения прекращается, а клиенту возвращается соответствующее диагностическое сообщение.

На рис. 8 проиллюстрирована маршрутизация информационного запроса в мультисетевой среде. Как видно из рисунка, рассматриваемая среда включает три частных локальных сети (LAN<sub>1</sub>, LAN<sub>2</sub> и LAN<sub>3</sub>), соединенные с глобальной сетью посредством "пограничных" серверов PMS (обозначены темными кубиками). Пред-

полагается, что каждый из "пограничных" серверов PMS оснащен двумя сетевыми интерфейсами, один из которых (внутренний) подключен к локальной сети, а другой (внешний) — к глобальной. Предположим также, что собственные имена этих серверов в их конфигурационных данных равны Pms1.ru, Pms2.ru и Pms3.ru соответственно и таковы же Интернет-имена их внешних интерфейсов в глобальной сети. В данном примере на эти серверы возлагаются исключительно служебные функции, связанные с маршрутизацией и защитой данных, а также с защитой частных сетей от проникновения в них несанкционированных информационных запросов. Рядом с каждым из этих серверов в прямоугольной рамке показан фрагмент его таблицы маршрутизации, имеющий отношение к примеру. Кроме того, в LAN<sub>2</sub> и LAN<sub>3</sub> имеются обрабатывающие серверы PMS с именами Filial2.1 и Filial3.1 и одинаковыми IP-адресами (например, 192.168.0.3), в которые загружены библиотеки сервисных функций Lib2 и Lib3 соответственно.

Рассмотрим пример совместного действия механизмов конвейерной обработки и маршрутизации запросов в мультисетевой среде при обработке следующего обращения к серверу Pms1.ru:

```
MyConn.Connect("внутренний_адрес_Pms1");
```

...

```
Ms.Process(null, MyConn, "Filial2.1/(Lib2.Fun1, Filial3.1/Lib3.Fun3, Lib2.Fun2)");
```

Здесь переменные MyConn, Request и Reply имеют тот же смысл, что и на рис. 1, "внутренний\_адрес\_Pms1" — сетевой адрес внутреннего сетевого интерфейса сервера Pms1.ru, а Fun1, Fun2 и Fun3 — имена сервисных функций.

Используя свою таблицу маршрутизации, сервер Pms1.ru определит, что любой запрос к серверу

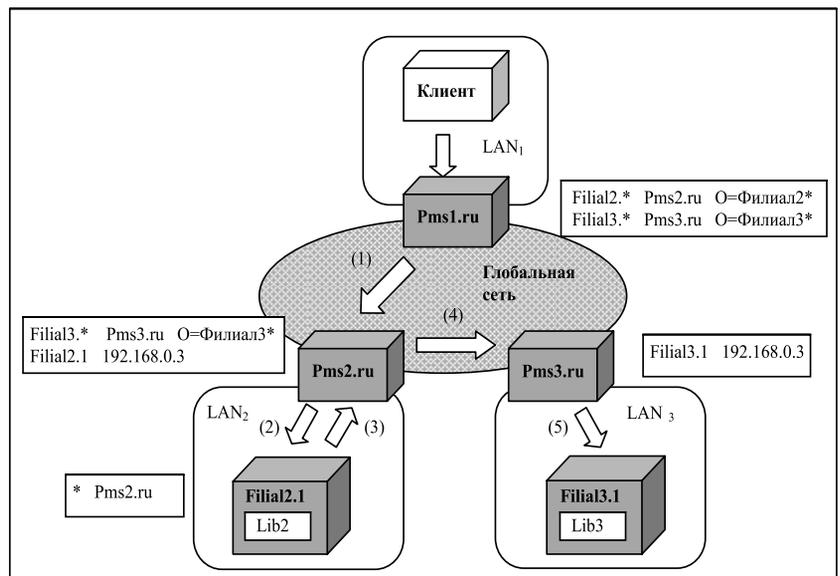


Рис. 8. Конвейерная мультисерверная обработка в мультисетевой среде

ру, имя которого начинается на "Filial2.", должен быть перенаправлен на сервер Pms2.ru. Предварительно сервер проверит ЭЦП, содержащуюся во входящем сообщении, опционально добавит к ней свою собственную подпись, установит соединение с сервером Pms2.ru, запросит сертификат у этого сервера для шифрования сообщения, проверит этот сертификат на корректность и проведет выборочную проверку реквизитов его владельца. В данном случае на эти реквизиты накладывается единственное ограничение: название организации владельца должно начинаться со строки "Филиал2", за которой может следовать все, что угодно (отдельные реквизиты обозначаются одной или двумя латинскими буквами точно так же, как у сертификатов в стандарте X509).

Обработка запроса в сервере Pms2.ru будет выполнена аналогично, с перенаправлением запроса на сервер PMS с IP-адресом 192.168.0.3, но с одним отличием: так как в строке таблицы маршрутизации отсутствуют контрольные значения реквизитов владельца сертификата, запрос сертификата у этого сервера и шифрование защищенного сообщения в данном случае выполняться не будут (в данном примере предполагается, что частная локальная сеть является безопасной зоной).

Собственно обработка конвейерного списка начнется на сервере Filial2.1 с сервисной функции Lib2.Fun1. При разборе конвейерного списка в этом сервере будет обнаружен элемент "вызов удаленного сервера" вида Filial3.1/Lib3.Fun3. Для обработки этого элемента сервер Filial2.1 выполнит обращение к серверу Filial3.1, причем серверы Pms2.ru и Pms3.ru будут использованы в качестве посредников (прямое соединение между серверами Filial2.1 и Filial3.1 невозможно, так как они находятся в разных частных сетях). Важную роль при этом сыграет единственная строка в таблице маршрутизации сервера Filial2.1, которая направляет все обращения к удаленным серверам на сервер Pms2.ru.

На рис. 8 передача информационного запроса между узлами сети обозначена белыми стрелками; цифры в скобках отображают очередность передач. Легко заметить, что совпадение IP-адресов у серверов Filial2.1 и Filial3.1 не создает никаких проблем для PMS.

### Заключение

Основная идея PMS заключается в отказе от гибкой модели организации взаимодействий на основе вызова методов удаленных объектов, характерной для электронных сервисов в архитектуре .NET, в пользу более жесткой модели обмена се-

тевыми сообщениями при фиксированной спецификации сервисных функций. Однако данная организация, разумеется, не предполагает никаких ограничений на содержание защищенных сообщений в распределенных системах. Каждое такое сообщение может заключать сколь угодно сложную структуру данных: от информационного запроса, содержащего десятки и сотни параметров (например, в формате XML), до отчета, включающего структурированные документы и графические приложения. В данной работе авторы попытались показать, что описанный подход позволяет предложить ряд готовых решений таких важных для разработчиков задач, как защита и маршрутизации информационных запросов в сложных мультисетевых средах и организация конвейерной обработки запросов в мультисерверной среде.

Описанная сетевая служба была реализована в форме двух дополняющих друг друга программных продуктов: сервера PMS (в форме постоянно активной Windows-службы) и библиотеки функций PMSBase.dll для клиентских приложений и библиотечных сервисных функций. Оба продукта реализованы на языке C# в среде Microsoft Visual Studio для среды MS Framework 4.0 с использованием сертифицированной криптосистемы "КриптоПро CSP" версии 3.6. Первые опыты ее использования и лабораторные эксперименты показали ее высокое быстродействие [6] и способность существенно сократить трудозатраты разработчиков на обеспечение защиты данных в мультисетевой среде.

### Список литературы

1. Шапошников И. В. Web-сервисы Microsoft.NET. СПб.: БХВ-Петербург, 2002. 336 с.
2. Мак-Дональд М., Шпуншта М. Microsoft ASP.NET 3.5 с примерами на C# 2008 и Silverlight 2 для профессионалов. М.: Вильямс, 2009. 1408 с.
3. Асратян Р. Э., Лебедев В. Н. Защита электронных сообщений в мультисетевой среде // Управление большими системами. 2013. Вып. 45. С. 95—111.
4. Згоба А. И., Маркелов Д. В., Смирнов П. И. Кибербезопасность: угрозы, вызовы, решения // Вопросы кибербезопасности. 2014. № 5. С. 30—38.
5. Щеглов А. В. Защита компьютерной информации от несанкционированного доступа. СПб.: Наука и техника, 2004. 384 с.
6. Асратян Р. Э. Интернет-служба защищенной обработки информационных запросов в распределенных системах // Программная инженерия. 2016. № 11. С. 490—497.
7. Снейдер Й. Эффективное программирование TCP/IP. Библиотека программиста. СПб.: Символ-Плюс, 2002. 320 с.
8. Хант К. TCP/IP. Сетевое администрирование. СПб.: Питер, 2007. 816 с.

**R. E. Asratian**, Leading Researcher, e-mail: rea@ipu.ru,  
**V. N. Lebedev**, Head of Laboratory, e-mail: lebvini@ipu.ru  
V. A. Trapeznikov Institute of control sciences of Russian academy of sciences

## Internet Service for Pipeline Processing of Protected Information Queries in Multi-Network Environment

*The principles of the organization of the new network service — Protected Message Service (PMS) — intended for protected queries processing in the distributed information systems oriented on operation in complex multi-server and multi-network structures are considered. Distinctive features of service are tight integration of functions of information data protection with functions of information exchange and information requests routing for multi-server processing. From the client point of view the service architecture is based on two main program classes: "Protected message" and "Network Connection". This classes offer necessary functionality not only for creating and protecting messages, but also for transferring them to remote server via established network connections for processing. Contrary to web-services, based on remote function call model, PMS-service is based on message processing model: all service functions receive object of "Protected message" class as a parameter and return another object of the same class as a result of processing. This approach opens a basic possibility of the organization of pipeline processing of one protected message by the sequence of service functions in a manner when the output of each function is forwarded to the input of the following one, and the result of last function returns to the client. The principles of the organization of pipeline processing of queries in a chain of servers are considered. The emphasis on routing of queries in the complex multi-network structures including a set of private local area networks is especially placed. Some experimental estimates of speed of pipeline multi-server processing are presented.*

**Keywords:** distributed systems, Web-technologies, Internet-technologies, network interactions, data security

### References

1. **Shaposhnikov I. V.** *Web-servisy Microsoft.NET* (Web-services Microsoft.NET), Saint-Petersburg, BHV-Peterburg, 2002, 336 p. (in Russian).
2. **Mak-Donal'd M., Shpushta M.** *Microsoft ASP.NET 3.5 s prime-rami na C# 2008 i Silverlight 2 dlja professionalov* (Pro ASP.NET 3.5 in C#2008 Includes Silverlight 2), Moscow, Vil'jams, 2009, 1408 p. (in Russian).
3. **Asratian R. E., Lebedev V. N.** Zashhita jelektronnyh soobshhenij v mul'ti-setevoj srede (Message protection in multi-network environment), *Upravlenie bol'shimi sistemami*, 2013, vol. 45, pp. 95—111 (in Russian).
4. **Zgoba A. I., Markelov D. V., Smirnov P. I.** Kiberbezopasnost': ugrozy, vyzovy, reshenija (Cyber security: threats, challenges,

decisions), *Voprosy kiberbezopasnosti*, 2014, no. 5, pp. 30—38 (in Russian).

5. **Shheglov A. V.** *Zashhita komp'yuternoj informacii ot nesankcionirovannogo dostupa* (Protection of computer information against illegal access), Saint-Petersburg, Nauka i tehnika, 2004, 384 p. (in Russian).
6. **Asratian R. E.** Internet-sluzhba zashhishhennoj obrabotki informacionnyh zaprosov v raspredeennyh sistemah (Internet service for protected information queries processing in distributed systems), *Programnaja inzhenerija*, 2016, vol. 7, no. 11, pp. 490—497 (in Russian).
7. **Snader J.** *Effektivnoe programirovanie TCP/IP* (Effective TCP/IP programming), Saint-Petersburg, Simvol-Pljus, 2002, 320 p. (in Russian).
8. **Hant K.** *TCP/IP. Setevoie administrirovanie* (TCP/IP. Network administration), Saint-Petersburg, Piter, 2007, 816 p. (in Russian).

## II Форум-выставка "IT в градостроительной сфере"

В 2017 году II Форум-выставка "IT в градостроительной сфере" пройдет с 21 по 24 ноября в зале Интерактивной карты города Москвы.

Организатором II Форума-выставки является Государственное бюджетное учреждение "Информационно-аналитический центр Комплекса градостроительной политики и строительства города Москвы "Мосстройинформ" — ГБУ "Мосстройинформ".

Целью проведения II Форума-выставки является популяризация применения информационных технологий на всех этапах жизненного цикла градостроительных объектов для автоматизации управления городским строительством.

В этом году форум-выставка будет представлен следующими направлениями:

- государственные услуги в градостроительном комплексе;
- проектирование, строительство, эксплуатация градостроительных объектов с применением информационных технологий;
- подготовка и переподготовка IT-специалистов в градостроительстве.

Заявки на участие в выставке и деловой программе форума принимаются до 03 ноября 2017 года.

Ответственное лицо за организацию II Форума-выставки:

Галиаскар Ильмира Индусовна, начальник управления градостроительных выставок

e-mail: GaliaskarII@str.mos.ru

тел.: 8 (499)250-35-82, доб. 291

Более подробная информация размещена на dom6.ru

## Сравнение параметров фильтра Габора для эффективного извлечения краев

*Приведена обобщенная формула фильтра Габора, составленная по обширному обзору источников. Проведено сравнение различных коэффициентов, влияющих на форму ядра, по метрике постоянной компоненты. Таким образом была определена оптимальная формула. Были выявлены значимые параметры, влияющие на качество извлечения краев. Предложена метрика оценки, по которой проведено сравнение различных отношений значимых параметров. В результате оптимальное соотношение значимых параметров было найдено.*

**Ключевые слова:** анализ изображений, фильтр Габора, длина волны, масштаб фильтра, извлечение краев, постоянная компонента

### Введение

С тех пор как в 1988 г. Джон Даугман предложил использовать фильтр Габора в задаче анализа изображений [1], был накоплен богатый опыт его применения. И вместе с тем явственно обозначилась проблема оптимального подбора параметров. Зачастую под разные задачи параметры подбираются специально в результате экспериментальной проверки.

Фильтр Габора доказал свою эффективность в задаче выделения краев и применяется при распознавании лиц [2–4], сканировании отпечатков пальцев и радужной оболочки глаза [5], анализе текстур [6, 7], в медицинской информатике [8], при поиске дефектов [9], анализе видеопотока [10, 11], в алгоритмах сегментации фотографий [12]. На базе банка фильтров Габора получают дескрипторы текстур [13]. Также он находит применение в задачах обеспечения роботов зрением [14, 15].

Чаще всего в компьютерном зрении применяется двумерный фильтр Габора, но в работе [16] описано применение трехмерного фильтра к задаче распознавания лиц.

Существенную роль в развитии практики применения фильтра Габора сыграли работы Даугмана [18], Петкова [21, 22], Курки [78], Бовик [83].

### 1. Фильтр Габора

Изначально преобразование Габора предлагалось для фильтрации одномерного сигнала [17]. Даугман заметил сходство работы фильтра Габора и "простых" клеток первичной зрительной коры головного мозга [18].

Ядро фильтра Габора представляет собой произведение элемента базиса Фурье и гауссиана. За счет

элемента базиса Фурье фильтр становится чувствителен к компонентам изображения определенной ориентации и пространственной частоты. Гауссиан необходим для пространственной локализации фильтра. В этом отношении свертка с фильтром Габора напоминает оконное преобразование Фурье, где функцией окна является гауссиан.

Формула Даугмана имеет специфический вид и не очень удобна для вычислений:

$$G = \exp(-\pi[(x - x_0)^2 a^2 + (y - y_0)^2 b^2]) \times \exp(-i2\pi(u(x - x_0) + v(y - y_0))), \quad (1)$$

где  $(x_0, y_0)$  — позиция центра окна свертки;  $(u, v)$  — вектор, определяющий предпочтительную пространственную частоту;  $a = \frac{1}{\sigma_x}$ ,  $b = \frac{1}{\sigma_y}$  — параметры

масштаба ядра фильтра.

Наложение фильтра на изображение осуществляется путем дискретной свертки с импульсной характеристикой или ядром фильтра с помощью сдвигающегося окна, поэтому для центра окна принимают значения  $x_0 = 0$ ,  $y_0 = 0$ . Чаще всего изображение для свертки представлено массивом яркости пикселей, но известны свертки изображения с каждым из цветовых каналов RGB [19]

$$R_{ij} = \sum_{k,l} G_{i-k, j-l} L_{k,l} \quad (2)$$

Еще одним вариантом наложения является формирование гексагональной поверхности ядра фильтра Габора (Hex-Gabor), подробно описанное в работе [41] и применяемое для задачи восстановления изображений.

По формуле Эйлера экспоненту из формулы (1) можно переписать в следующем виде:

$$\begin{aligned} & \exp(-i2\pi(ux + vy)) = \\ & = \cos(2\pi(ux + vy)) + i \sin(2\pi(ux + vy)). \end{aligned} \quad (3)$$

Такая запись фильтра образует квадратурную пару (рис. 1, см. третью сторону обложки), состоящую из фильтров для определения симметричных и антисимметричных компонентов. При этом в большинстве случаев достаточно применения фильтра для определения симметричных компонент:

$$G_{sym}(x, y) = \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \cos(2\pi(ux + vy)); \quad (4)$$

$$G_{antisym}(x, y) = \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \sin(2\pi(ux + vy)). \quad (5)$$

Как правило, фильтруют только по  $x$ -компоненте пространственной частоты, а остальное возлагается на поворот ядра фильтра. Таким образом, принимают  $v = 0$ . Также часто переходят от вектора  $(u, v)$  к фиксированной частоте  $F = \sqrt{u^2 + v^2}$  (циклов/пиксель).

## 2. Составление оптимальной формулы ядра

Проанализировав многие работы по применению фильтра Габора, можно составить обобщенную формулу:

$$G = C_1 \exp\left(-C_2 \left[ \frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right]\right) \exp(iC_3(ux + vy)). \quad (6)$$

У многих авторов вместо пространственной частоты фигурирует длина волны  $\lambda \in [2, \infty)$ , определяющая размеры элементов изображения, на которые будет реагировать фильтр. Длина волны имеет обратно пропорциональную зависимость с пространственной частотой  $\lambda = 1/F$ .

Масштабирующие коэффициенты  $C_1, C_2, C_3$  определяют форму кривой ядра фильтра и у разных авторов принимают разные значения, которые представлены в табл. 1.

Коэффициент  $C_1$  непосредственно определяет селективную способность фильтра. На рис. 2 (см. третью сторону обложки) изображено влияние вариантов коэффициента на график ядра (в порядке следования в табл. 1). С ростом  $\lambda$  и соответственно уменьшением  $F$  графики ядер с коэффициентами № 1–4 вырождаются практически в прямую, что негативным образом сказывается на селективной способности. Графики ядер с коэффициентами № 5, 6 вырождаются гораздо медленнее. Уменьшение  $\lambda$  приводит к вырождению ядер с коэффициентами № 3–5, график с коэффициентом № 6 при этом дает слишком большой экстремум.

Коэффициент  $C_2$  определяет форму гауссиана (рис. 3, см. третью сторону обложки). При  $C_1 = 1$

с ростом  $\lambda$ , графики ядер с номерами коэффициентов 1 и 5 практически совпадают, график с коэффициентом № 4 вырождается в прямую, график с коэффициентом № 2 становится ближе к № 1 и 5. С ростом  $F$ , и соответственным изменением  $\sigma_x, \sigma_y$ , уменьшается отрицательная зона у ядер с коэффициентами № 1, 4, и 5. Маленькая и большая амплитуда ядра фильтра одинаково плохо сказывается на его способности выделения границ, основанных на перепадах яркости. При слишком большой амплитуде фильтр окажется не чувствителен к мелким деталям, при слишком маленькой амплитуде перестанет быть инвариантен к освещенности. Оптимальным видится использование коэффициентов № 2 и 3. Учитывая соображения по поводу похожести работы фильтра Габора и простых клеток зоны V1 (первичной зрительной коры) зрительной системы [18], оптимальным будет коэффициент № 2.

Коэффициент  $C_3$  не оказывает особого влияния на график ядра фильтра. Тем не менее, для получения именно периодического колебания с отрицательными и положительными зонами на симметричном фильтре необходимо использовать коэффициент № 1 или 2. Знак для симметричного фильтра не имеет значения, для антисимметричного он определяет порядок следования отрицательных и положительных зон.

Ядро фильтра Габора должно иметь максимально близкую к нулю постоянную компоненту (*DC component*), так как это позволяет снизить уровень ложно выделяемых краев. В табл. 2 представлено

Таблица 1

Коэффициенты обобщенной формулы

Номер	$C_1$	$C_2$	$C_3$
1	$\frac{F^2}{\sigma_x \sigma_y}$ [22]	$F^2$ [3, 4, 23–25]	$-2\pi$ [1, 6, 11, 38, 43, 48, 51, 54, 55, 57–60, 62, 66–68, 70]
2	$\frac{F^2}{\pi \sigma_x \sigma_y}$ [3, 4, 23–25, 78]	$\frac{1}{2}$ [2, 6, 11, 13, 14, 20–22, 26–45, 47–65]	$2\pi$ [2–4, 13, 20, 23–29, 31–37, 39–42, 45, 47, 49, 50, 52, 53, 56, 61, 69, 71]
3	$\frac{1}{2\pi \sigma_x \sigma_y}$ [2, 13, 26–44]	1 [46]	1 [14, 22, 44, 46, 64, 65]
4	$\frac{1}{\sqrt{2\pi} \sigma_x \sigma_y}$ [45]	$\pi$ [1, 47, 62, 68–71]	$-1$ [30, 63]
5	1 [1, 6, 11, 14, 20, 46–70]	$\frac{F^2}{2}$ [22]	
6	$\frac{\sqrt{2}}{\sigma}$ [71]		

Таблица 2

Сравнение коэффициентов  $C_1$  и  $C_2$  по постоянной компоненте (первая цифра обозначает номер коэффициента  $C_1$ , вторая — номер коэффициента  $C_2$ )

Вариант комбинации коэффициентов	Значение DC component	Вариант комбинации коэффициентов	Значение DC component
2-2	2,726183204E-05	4-1	7,303063589E-03
3-2	5,452366409E-05	2-4	8,071898710E-03
1-2	8,564557128E-05	4-3	8,583319746E-03
4-2	1,536646879E-04	3-4	1,614379742E-02
<b>5-2</b>	<b>4,330750086E-04</b>	4-5	1,706274567E-02
6-2	5,447271355E-04	5-3	1,710524653E-02
2-1	9,161594046E-04	4-4	1,815117774E-02
2-5	1,513560835E-03	5-4	2,040816327E-02
3-1	1,832318809E-03	1-4	2,535861769E-02
2-3	2,153531600E-03	6-1	2,588868636E-02
1-1	2,878199655E-03	5-1	2,910778273E-02
3-5	3,027121671E-03	6-3	3,042707627E-02
3-3	4,307063200E-03	6-5	6,048585854E-02
1-5	4,754991602E-03	6-4	6,434424978E-02
1-3	6,765519053E-03	5-5	9,617627615E-02

сравнение различных комбинаций коэффициентов  $C_1$  и  $C_2$  по значению постоянной компоненты при  $F = 0,5$ :

$$DC = |mean(G)|. \quad (7)$$

Как можно заметить, наиболее близкие к нулю значения постоянной компоненты имеет ядро фильтра с коэффициентом  $C_2 = \frac{1}{2}$  (первые шесть

строк таблицы). Коэффициент  $C_1 = \frac{F^2}{\pi\sigma_x\sigma_y}$  (коэф-

фициент № 2) был предложен Курки [78], и его применение дает наименьшее значение постоянной компоненты, но, как говорилось выше, при росте  $\lambda$  ядро фильтра имеет очень маленькие значения, что негативно сказывается на селективной способности. То же самое касается коэффициентов № 1, 3, 4.

Оптимальная формула фильтра Габора должна иметь низкое значение постоянной компоненты (минимизация шума), и в то же время, высокую селективную способность (более четкое выделение краев на перепаде яркости, что дает более математически высокое значение свертки). Учитывая значение постоянной компоненты (табл. 2), а также влияние коэффициентов обобщенной формулы на селективную способность (см. рис. 2, 3 на третьей стороне обложки) этому требованию соответствует формула, приведенная Петковым [20, 21] (комбинация значений коэффициентов 5—2 по табл. 2 выделена жирным):

$$G = \exp\left(-\frac{1}{2}\left[\frac{x'^2 + \gamma^2 y'^2}{\sigma^2}\right]\right) \cos(2\pi Fx' + \varphi), \quad (8)$$

где  $\varphi$  определяет симметричность ядра (в радианах);  $\sigma$  — масштаб фильтра;  $\gamma$  — степень эллиптичности фильтра.

Для реагирования на компоненты определенной ориентации в фильтр Габора вводятся параметры поворота координатной сетки на угол  $\theta$  (рис. 4):

$$\begin{cases} x' = x\cos\theta + y\sin\theta; \\ y' = -x\sin\theta + y\cos\theta; \end{cases} \quad (9)$$

$$\theta_k = \frac{k\pi}{n_0}; \quad (10)$$

$k=0, \dots, n_0 - 1$  — число различных ориентаций; (11)

$$\Delta\theta = \frac{\pi}{n_0}. \quad (12)$$

Существует несколько распространенных вариантов числа ориентаций: 4 [6, 25, 28, 34, 40, 42, 56, 58, 62, 67, 69, 74, 76, 85], 6 [26, 31, 32, 36, 60, 75, 84], 8 [3, 11, 21, 23, 54]. Тем не менее данные нейрофизиологии [73, 88] позволяют говорить о формировании фильтров с 18 ориентациями. В работе [38] приведено описание банка фильтров со 180 ориентациями, что кажется весьма избыточным. Так как фильтр имеет ярко выраженную ориентационную избирательность, число ориентаций влияет на выявление деталей и, как следствие, уровень шума. Для компенсации этого негативного эффекта приходится увеличивать размер матрицы ядра фильтра, что увеличивает вычислительную емкость свертки. Для выявления незначительных деталей необходимо максимум 18 ориентаций.

Параметр  $\varphi$  [см. ф-лу (8)], отвечающий за симметричность ядра фильтра, был введен в качестве альтернативы квадратурной паре фильтров (4) и (5). Он

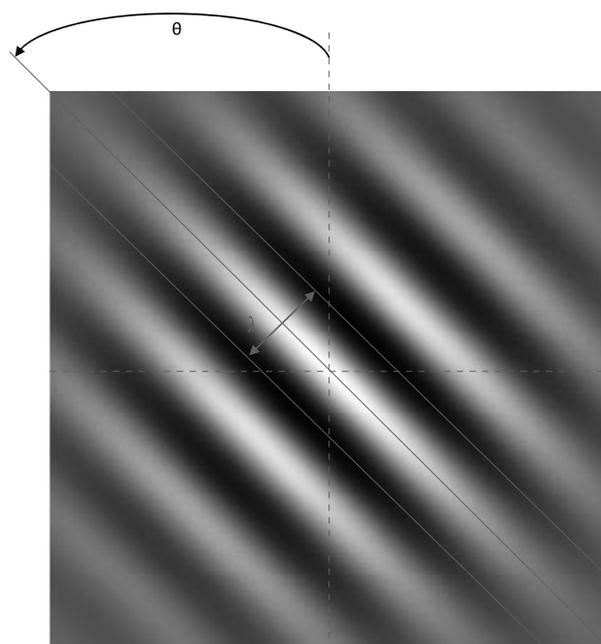


Рис. 4. Ядро симметричного фильтра

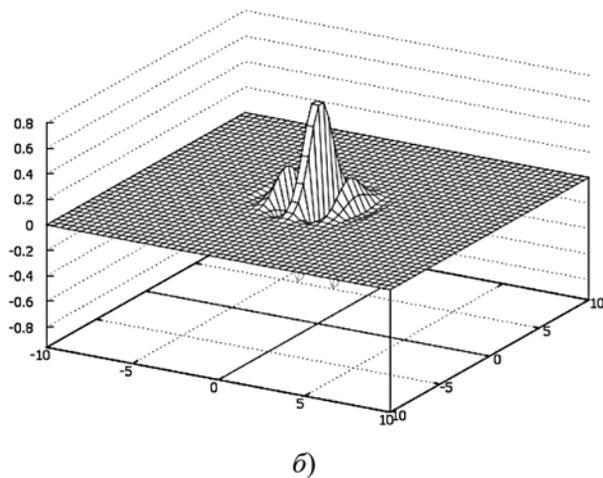
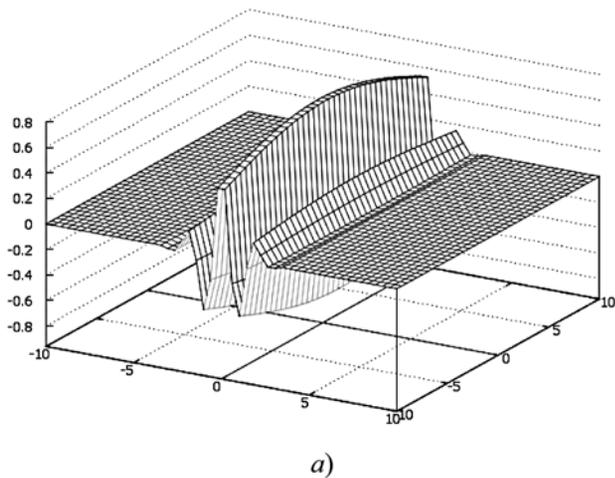


Рис. 5. Варианты ядра с разными  $\gamma$ :  
 а —  $\gamma = 0,1$ ; б —  $\gamma = 1$

может принимать значения  $-\frac{\pi}{2}$  или  $\frac{\pi}{2}$  для выявления антисимметричных компонент и  $-\pi, 0$  или  $\pi$  для симметричных компонент;  $\gamma \in (0, 1]$  определяет вытянутость ядра фильтра по оси ординат (рис. 5). Параметр введен исключительно для упрощения подбора оптимального ядра без разных значений  $\sigma_x$  и  $\sigma_y$  и

может быть выражен через эти параметры:  $\gamma = \frac{\sigma_x}{\sigma_y}$ .

В работе [73] приведена рекомендация по выбору значения параметра из интервала  $0,23 < \gamma < 0,92$ . Зачастую принимают  $\gamma = 0,5$ .

### 3. Длина волны и масштаб фильтра

Если с остальными параметрами ядра фильтра Габора все более или менее ясно, то длина волны и масштаб фильтра вызывают наибольшие дискуссии.

Таблица 3

Константы отношения

Номер	Константа $\sigma/\lambda$	Источники
1	0,16	[76]
2	0,4	[11]
3	$\frac{1}{2\sqrt{2\ln 2}} \approx 0,425$	[39]
4	0,5	[21, 48, 57, 73]
5	0,7	[57]
6	0,8	[87, используется в 56, 83]
7	0,975	[27]
8	$\sqrt{2} \approx 1,414$	[70]
9	$3\sqrt{\frac{\ln 2}{2}} \approx 1,766$	[14]
10	3,5	[45]
11	$2\pi \approx 6,283$	[69]
12	$\frac{1}{4\sqrt{2}\sin(\frac{\Delta\theta}{2})}$	[43]

Зависимость между пространственной частотой и масштабом фильтра становится очевидной при использовании фильтра Габора. К настоящему времени сложилось несколько подходов к определению этой зависимости: константы отношения (табл. 3); алгоритмы оптимизации; пропускная способность.

### Алгоритмы оптимизации

Подбор оптимальных параметров фильтра для формирования банка фильтров некоторые авторы рассматривают как задачу статистической оптимизации. Методы [5, 35, 40] сходным образом используют оптимальную модель для мультитекстурной сегментации, базирующуюся на вычислении энергии Габора.

В работе [5] приведено описание поиска оптимальных параметров с помощью метода роя частиц. Каждая частица в нем представлена вектором из четырех параметров, определяющих фильтр Габора (максимальная частота  $F_{\max}$ , масштабирующий фактор частот  $f$ , число частот  $M$  и число ориентаций  $n_0$ ). Целевую функцию при этом рассчитывают на основании энергии Габора и фазы. Оптимальный фильтр представлен тремя параметрами ( $F, \theta, \sigma$ ). В работе [35] он представлен антителом и применяется иммунный генетический алгоритм. В работе [40] для этой же цели служит алгоритм имитации отжига.

### Подбор через пропускную способность

Пропускная способность ( $B$ ), измеряемая в октавах, была предложена в работе [83] и использована в работах [4, 6, 16, 21, 33, 49, 54, 58, 61, 71, 74, 85]:

$$B = \log_2 \left( \frac{f_h}{f_l} \right). \quad (13)$$

Применяя к формуле (13) частоту Найквиста, определяют соотношение пропускной способности и масштаба фильтра через половину пиковой маг-

нитуды ядра. Приведем расчет для обобщенной формулы (6):

$$C_1 \exp\left(-C_2 \left[ \frac{(u-u_0)^2}{\sigma_u^2} + \frac{(v-v_0)^2}{\sigma_v^2} \right]\right) = \frac{1}{2} C_1; \quad (14)$$

$$-C_2 \left[ \frac{(u-u_0)^2}{\sigma_u^2} + \frac{(v-v_0)^2}{\sigma_v^2} \right] = \ln\left(\frac{1}{2}\right); \quad (15)$$

$$\frac{(u-u_0)^2}{\sigma_u^2} + \frac{(v-v_0)^2}{\sigma_v^2} = \frac{\ln 2}{C_2}. \quad (16)$$

Введем обозначение  $D^2 = \frac{\ln 2}{C_2}$ , соответственно,

$$D = \sqrt{\frac{\ln 2}{C_2}}.$$

Воспользуемся каноническим уравнением эллипса:

$$\frac{(u-u_0)^2}{\sigma_u^2 D^2} + \frac{(v-v_0)^2}{\sigma_v^2 D^2} = 1. \quad (17)$$

Частоты можно определить следующим образом:

$$\begin{cases} f_h = F_0 + A, \\ f_l = F_0 - A, \end{cases} \quad (18)$$

где  $A$  — большая полуось эллипса ядра (рис. 6).

Таким образом,

$$B = \log_2 \left( \frac{F_0 + \sigma_u D}{F_0 - \sigma_u D} \right). \quad (19)$$

Подставив в формулу (18)  $C_2 = \frac{1}{2}$ , получим формулу пропускной способности, используемую большинством авторов:

$$B = \log_2 \left( \frac{\frac{\sigma}{\lambda} + \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda} - \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}}} \right). \quad (20)$$

Соответственно,

$$\frac{\sigma}{\lambda} = \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}} \frac{2^B + 1}{2^B - 1}. \quad (21)$$

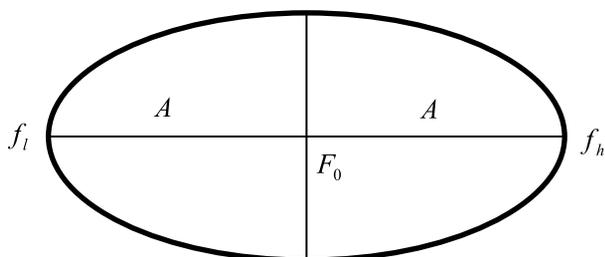


Рис. 6. Ядро фильтра в частотном пространстве

По данным нейронауки пропускная способность для нейронов первичной зрительной коры определяется в диапазоне  $B = [0,6...2,0]$  [87]. Зачастую принимают  $B = 1$ . В этом случае  $\sigma/\lambda \approx 0,56$ .

#### 4. Сравнение отношений

При подборе параметров длины волны и масштаба фильтра удобнее в качестве управляющего параметра использовать длину волны.

В качестве показателя сравнения эффективности подбора параметров можно применить отношение сигнал/шум SNR с несколькими оговорками. По определению, соотношение сигнал/шум является частным математического ожидания сигнала и математического ожидания шума. Сигнал в нашем случае определяется выявленным краем и может быть найден как среднее значение результата свертки. Шум определяется ложно выявленным краем или фоном на изображении. Фоновых пикселей на изображении больше, поэтому оценкой шума может служить среднее квадратическое отклонение результата свертки от медианы. В этом заключается главное отличие предлагаемой метрики от классического определения SNR. Этим также обусловлена необходимость минимизации метрики. Оценка по классической формуле не отражает правильности выделения краев, так как фоновые пиксели имеют ненулевой результат свертки с ядром фильтра и, соответственно, большое значение среднего квадратического отклонения от среднего.

$$SNR_E = \frac{\mu_L}{\sigma_L} \rightarrow \min, \quad (22)$$

где  $\mu_L = \frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q R_{i,j}$  — среднее значение результата свертки;

$\sigma_L = \sqrt{\frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q (R_{i,j} - Me)^2}$  — стандартное отклонение результата свертки;  $Me$  — медиана результата свертки.

В табл. 4 приведено сравнение рассчитанного для изображений (рис. 7, см. четвертую сторону обложки) значения  $SNR_E$  по формуле (22) для констант отношений (см. табл. 3) и отношения, определяемого пропускной способностью по формуле (21) при  $B = 1$ . Свертка выполнена с ядром, рассчитанным по формуле (8) ( $\gamma = 0,1, \lambda = 2, \varphi = 0$ ).

Как видно из табл. 4, наименьшие значения (выделены жирным) имеют соотношение  $\frac{\sigma}{\lambda} = 0,5$  (кон-

станта № 4 из табл. № 3) и соотношение  $\frac{\sigma}{\lambda} = 0,56$ ,

определяемое формулой (21). При этих соотношениях масштаба фильтра и длины волны получается наименьший шум при свертке изображений с ядром фильтра и более четкое выделения контура сегмента.

Сравнение значений для подбора  $\sigma/\lambda$ 

$\sigma/\lambda$	$SNR_E$					
	test	kaniza	popopout	Universum	Lena	Voynich
0,16	0,37519	0,15725	0,16211	1,44573	0,28532	0,20251
0,4	0,32453	0,1277	0,1401	1,64541	0,2665	0,19732
0,424661	0,31096	0,12424	0,13719	1,59282	0,2621	0,19549
0,5	0,26583	<b>0,12116</b>	<b>0,13502</b>	1,44253	0,24739	0,185
0,56	<b>0,23331</b>	0,12309	0,13751	<b>1,41101</b>	<b>0,23976</b>	<b>0,17719</b>
0,7	0,27976	0,14178	0,15475	1,64474	0,25656	0,19401
0,8	0,32802	0,15439	0,1658	1,84013	0,27801	0,21529
0,975	0,3623	0,16941	0,17862	1,9649	0,29374	0,23217
1,414214	0,37983	0,18048	0,18764	1,9541	0,30164	0,24315
1,766115	0,38257	0,18283	0,18943	1,93213	0,30298	0,24563
3,5	0,38491	0,18518	0,19114	1,89933	0,30424	0,24833
6,283185	0,38531	0,18562	0,19143	1,89165	0,30447	0,24889

### 5. Фильтр Габора в частотном пространстве

Классический фильтр Габора описан в пространстве декартовых координат сигнала. В случае необходимости перейти в пространство частот осуществляют смену базиса с помощью преобразования Фурье ядра фильтра:

$$G(u, v) = \exp\left(-\frac{1}{2}\left[\frac{(u - F_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right), \quad (23)$$

$$\sigma_u = \frac{1}{2\pi\sigma_x}; \quad \sigma_v = \frac{1}{2\pi\sigma_y}; \quad F_0 = f_h.$$

Переход в пространство частот облегчает подбор параметров фильтра. Фурье-образ ядра фильтра позволяет построить банк фильтров в виде так называемой розетки, по которой можно подобрать значения пространственных частот оптимальных для конкретной задачи.

В этом случае для проведения свертки необходимо сначала сделать Фурье-преобразование яркостного сигнала изображения, затем осуществить саму свертку с частотной характеристикой (23), и сделать обратное преобразование Фурье для получения результата. Для этого зачастую применяют быстрое преобразование Фурье.

Подбор параметров приведен в работе [84], он же используется в работах [28, 32, 36, 62]:

$$a = \left(\frac{f_h}{f_l}\right)^{s-1}; \quad (24)$$

$$\sigma_u = \frac{(a-1)f_h}{(a+1)\sqrt{\frac{\ln 2}{2}}}; \quad (25)$$

$$\sigma_v = \frac{\arctg\left(\frac{\pi}{2n_0}\right)\left[f_h - \left(\frac{\sigma_u}{f_h}\right)2\ln 2\right]}{\sqrt{2\ln 2 - \frac{(2\ln 2)^2 \sigma_u^2}{f_h^2}}}, \quad (26)$$

$s$  — число фильтров с разными масштабами;  $n_0$  — число фильтров с разными ориентациями.

Благодаря свойству обратимости преобразования Фурье, соотношение  $\frac{\sigma}{\lambda}$  влияет на качество вы-

деления краев в частотном пространстве так же, как и при свертке в декартовых координатах. При построении банка фильтров соотношения получают в том же диапазоне значений, указанных в табл. 2, и для них также справедлива оценка  $SNR_E$  (см. табл. 3).

### Заключение

В работе был дан обзор практики применения фильтра Габора в задачах компьютерного зрения. Были выявлены различные коэффициенты, определяющие формулу ядра фильтра. На основании анализа влияния коэффициентов обобщенной формулы на селективную способность фильтра, а также принимая во внимание значение постоянной компоненты фильтра можно заключить, что оптимальной является формула, предложенная Петковым (комбинация коэффициентов 5-2). Это подтверждает расчет постоянной компоненты и аналитическое сравнение влияния различных значений длины волны на форму графика ядра. Также было приведено описание способов подбора соотношения длины волны и масштаба фильтра. Применение предложенной метрики оценки выделения краев позволило говорить об оптимальности значения

$\frac{\sigma}{\lambda} = 0,5$  (константа № 4, см. табл. 3).

### Список литературы/References

1. **Daugman J. G.** Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1988, vol. 36, no. 7, pp. 1169—1179.
2. **Batool N., Chellappa R.** Fast detection of facial wrinkles based on Gabor features using image morphology and geometric constraints. *Pattern Recognition*, 2015, vol. 48, no. 3, pp. 642—658.

3. **Serrano A., Martin de Dengo I., Conde C., Cabello E.** et al. Analysis of variance of Gabor filter banks parameters for optimal face recognition. *Pattern Recognition Letters*. 2011. Vol. 32, N. 15. P. 1998–2008.
4. **Gonzalez-Ortega D., Diaz-Pernas F. J., Anton-Rodriguez M.** et al. MLP-based face recognition with Gabor filters and PCA. *Pattern Recognition and Image Analysis*. 2013. Vol. 23, N. 1. P. 10–25.
5. **He F., Liu Y., Zhu X.** et al. Multiple local feature representations and their fusion based on an SVR model for iris recognition using optimized Gabor filters. *EURASIP Journal on Advances in Signal Processing*, 2014. Vol. 95, no. 6. doi: 10.1186/1687-6180-2014-95.
6. **Jenssen R., Eltoft T.** Independent component analysis for texture segmentation. *Pattern Recognition*. 2003. Vol. 36, no. 10, pp. 2301–2315.
7. **Sagiv Ch., Sochen N. A., Zeevi Y. Y.** Integrated Active Contours for Texture Segmentation. *IEEE Transactions on Image Processing*, 2006, vol. 15, no. 6, pp. 1633–1646.
8. **Sommen van der F., Zinger S., Schoon E. J., de With P. H.N.** Supportive automatic annotation of early esophageal cancer using local gabor and color features. *Neurocomputing*, 2014, vol. 144, pp. 92–106.
9. **Hanmandlu M., Choudhury D., Dash S.** Detection of defects in fabrics using topothesy fractal dimension features. *Signal, Image and Video Processing*, 2015, vol. 9, no. 7, pp. 1521–1530.
10. **Santhaseelan V., Asari V. K.** Utilizing Local Phase Information to Remove Rain from Video. *International Journal of Computer Vision*, 2015, vol. 112, no. 1, pp. 71–89.
11. **Wagenaar D. A., Kristan W. B. Jr.** Automated Video Analysis of Animal Movements Using Gabor Orientation Filters. *Neuroinformatics*, 2010, vol. 8, no. 1, pp. 33–42.
12. **Yang H.-Y., Wang X.-Y., Wang Q.-Y., Zhang X.-J.** LS-SVM based image segmentation using color and texture information. *Journal of Visual Communication and Image Representation*, 2012, vol. 23, no. 7, pp. 1095–1112.
13. **Ahonen T., Pietikainen M.** Image description using joint distribution of filter bank responses. *Pattern Recognition Letters*, 2009, vol. 30, no. 4, pp. 368–376.
14. **Baunegaard L.** et al. A two-level real-time vision machine combining coarse- and fine-grained parallelism. *Journal of Real-Time Image Processing*, 2010, vol. 5, no. 4, pp. 291–304.
15. **Shimonomura K., Kushima T., Yagi T.** Binocular robot vision emulating disparity computation in the primary visual cortex. *Neural Networks*, 2008, vol. 21, no. 2–3, pp. 331–340.
16. **Wang Y., Chua Ch.-S.** Face recognition from 2D and 3D images using 3D Gabor filters. *Image and Vision Computing*, 2005, vol. 23, no. 11, pp. 1018–1028.
17. **Gabor D.** Theory of communication. *J. Inst. Elec. Eng.*, 1946, vol. 93, no. 26, pp. 429–457.
18. **Daugman J. G.** Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. 1985, *J. Opt. Soc. Am. A.*, vol. 2, no. 7, pp. 1160–1169.
19. **Nirmala S., Nagabhushan P.** Foreground text segmentation in complex color document images using Gabor filters. *Signal, Image and Video Processing*, 2012, vol. 6, no. 4, pp. 669–678.
20. **Petkov N.** Biologically motivated computationally intensive approaches to image pattern recognition. *Future Generation Computer Systems*, 1995, vol. 11, no. 4–5, pp. 451–465.
21. **Kruizinga P., Petkov N.** Nonlinear Operator for Oriented Texture. *IEEE Transactions on Image Processing*, 1999, vol. 8, no. 10, pp. 1395–1407.
22. **Kalliomaki I., Lampinen J.** On steerability of Gabor-type filters for feature detection. *Pattern Recognition Letters*, 2007, vol. 28, no. 8, pp. 904–911.
23. **Caleanu C., Huang D.-Sh., Gui V.** et al. Interest Operator versus Gabor filtering for facial imagery classification. *Pattern Recognition Letters*, 2007, vol. 28, no. 8, pp. 950–956.
24. **Bianconi F., Fernandez A.** Evaluation of the effects of Gabor filter parameters on texture classification. *Pattern Recognition*, 2007, vol. 40, no. 12, pp. 3325–3335.
25. **Li W., Mao K., Zhang H., Chai T.** Designing Compact Gabor Filter Banks for Efficient Texture Feature Extraction. In: *Proceedings of the 11th IEEE International Conference on Control Automation Robotics and Vision (ICARCV)*, 2010, pp. 1193–1197.
26. **Sirmacek B., Unsalan C.** Using local features to measure land development in urban regions. *Pattern Recognition Letters*, 2010, vol. 31, no. 10, pp. 1155–1159.
27. **Choras R. S.** Fuzzy Approach for Image Retrieval. *Pattern Recognition and Image Analysis*, 2007, vol. 17, no. 2, pp. 259–267.
28. **Abrishami Moghaddam H., Nikzad Dehaji M.** Enhanced Gabor wavelet correlogram feature for image indexing and retrieval. *Pattern Analysis and Applications*, 2013, vol. 16, no. 2, pp. 163–177.
29. **Autio I., Elomaa T.** Flexible view recognition for indoor navigation based on Gabor filters and support vector machines. *Pattern Recognition*, 2003, vol. 36, no. 12, pp. 2769–2779.
30. **Bigun J., du Buf J. M. H.** N-folded Symmetries by Complex Moments in Gabor Space and Their Application to Unsupervised Texture Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994, vol. 16, no. 1, pp. 80–87.
31. **Bissi L., Baruffa G., Placidi P.** et al. Automated Defect Detection in Uniform and Structured Fabrics using Gabor Filters and PCA. *Journal of Visual Communication and Image Representation*, 2013, vol. 24, no. 7, pp. 838–845.
32. **Choi S. E., Lee Y. J., Lee S. J.** et al. Age estimation using a hierarchical classifier based on global and local facial features. *Pattern Recognition*, 2011, vol. 44, no. 6, pp. 1262–1281.
33. **Clausi D. A., Jernigan M.** (ed.) Designing Gabor filters for optimal texture separability. *Pattern Recognition*, 2000, vol. 33, no. 11, pp. 1835–1849.
34. **Huang L.-L., Shimizu A., Kobatake H.** Robust face detection using Gabor filter features. *Pattern Recognition Letters*, 2005, vol. 26, no. 11, pp. 1641–1649.
35. **Li M., Staunton R. C.** Optimum Gabor filter design and local binary patterns for texture segmentation. *Pattern Recognition Letters*, 2008, vol. 29, no. 5, pp. 664–672.
36. **Nabti M., Bouridane A.** An effective and fast iris recognition system based on a combined multiscale feature extraction technique. *Pattern Recognition*, 2008, vol. 41, no. 3, pp. 868–879.
37. **Pan X., Ruan Q.-Q.** Palmprint recognition using Gabor-based local invariant features. *Neurocomputing*, 2009, vol. 72, no. 7–9, pp. 2040–2045.
38. **Rangayyan R. M., Ayres F. J.** Gabor filters and phase portraits for the detection of architectural distortion in mammograms. *Medical and Biological Engineering and Computing*, 2006, vol. 44, no. 10, pp. 883–894.
39. **Sastry Ch. S., Ravindranath M., Pujari A. K., Deekshatulu B. L.** A modified Gabor function for content based image retrieval. *Pattern Recognition Letters*, 2007, vol. 28, no. 2, pp. 293–300.
40. **Tsai D.-M., Wu S.-K., Chen M.-Ch.** Optimal Gabor filter design for texture segmentation using stochastic optimization. *Image and Vision Computing*, 2001, vol. 19, no. 5, pp. 299–316.
41. **Veni S., Narayanankutty K. A.** Vision-based hexagonal image processing using Hex-Gabor. *Signal, Image and Video Processing*, 2014, vol. 8, no. 2, pp. 317–326.
42. **Wang X., Ding X., Liu Ch.** Gabor filters-based feature extraction for character recognition. *Pattern Recognition*, 2005, vol. 38, no. 3, pp. 369–379.
43. **Weldon Th. P., Higgins W. E., Dunn D. E.** Efficient Gabor filter design for texture segmentation. *Pattern Recognition*, 1996, vol. 29, no. 12, pp. 2005–2015.
44. **Yu W., Sommer G., Daniilidis K.** et al. Using skew Gabor filter in source signal separation and local spectral orientation analysis. *Image and Vision Computing*, 2005, vol. 23, no. 4, pp. 377–392.
45. **Aujol J.-F., Gilboa G., Chan T., Osher S.** Structure-Texture Image Decomposition—Modeling, Algorithms, and Parameter Selection. *International Journal of Computer Vision*, 2006, vol. 67, no. 1, pp. 111–136.
46. **Bodnarova A., Bennamoun M., Latham S.** Optimal Gabor filters for textile flaw detection. *Pattern Recognition*, 2002, vol. 35, no. 12, pp. 2973–2991.
47. **Casasent D. P., Neiberg L. M.** Classifier and Shift-invariant Automatic Target Recognition Neural Networks. *Neural Networks*, 1995, vol. 8, no. 7–8, pp. 1117–1129.
48. **Chen Ch.-Ch., Tyler Ch. W.** Evidence for elongated receptive field structure for mechanisms subserving stereopsis. *Vision Research*, 2006, vol. 46, no. 17, pp. 2691–2702.
49. **Clark M., Bovik A. C., Gesler W. S.** Texture segmentation using Gabor modulation/demodulation. *Pattern Recognition Letters*, 1987, vol. 6, no. 4, pp. 261–267.
50. **Dunn D., Higgins W. E.** Optimal Gabor Filters for Texture Segmentation. *IEEE Transactions on Image Processing*, 1995, vol. 4, no. 7, pp. 947–964.
51. **Field D. J., Hayes A., Hess R. F.** Contour Integration by the Human Visual System: Evidence for a Local "Association Field". *Vision Research*, 1993, vol. 33, no. 2, pp. 173–193.
52. **Gonzaga S. L. de O.** A Method for Fingerprint Image Identification Based on Gabor Filter and Power Spectrum. *Pattern Recognition and Image Analysis*, 2010, vol. 20, no. 2, pp. 201–209.
53. **Gottschlich C.** Curved-Region-Based Ridge Frequency Estimation and Curved Gabor Filters for Fingerprint Image Enhancement. *IEEE Transactions on Image Processing*, 2012, vol. 21, no. 4, pp. 2220–2227.

54. Grigorescu S. E., Petkov N., Kruižinga P. Comparison of Texture Features Based on Gabor Filters. *IEEE Transactions on Image Processing*, 2002, vol. 11, no. 10, pp. 1160–1167.
55. Gu W., Xiang Ch., Venkatesh Y. V. et al. Facial expression recognition using radial encoding of local Gabor features and classifier synthesis. *Pattern Recognition*, 2012, vol. 45, no. 1, pp. 80–91.
56. Hamamoto Y., Uchimura Sh., Watanabe M. et al. A gabor filter-based method for recognizing handwritten numerals. *Pattern Recognition*, 1998, vol. 31, no. 4, pp. 395–400.
57. Hata M., Toyoura M., Mao X. Automatic generation of accentuated pencil drawing with saliency map and LIC. *The Visual Computer*, 2012, vol. 28, no. 6, pp. 657–668.
58. Jain A. K., Bhattacharjee S. Text Segmentation Using Gabor Filters for Automatic Document Processing. *Machine Vision and Applications*, 1992, vol. 5, no. 3, pp. 169–184.
59. Khan J. F., Adhami R. R., Bhuiyan Sh. M. A. A customized Gabor filter for unsupervised color image segmentation. *Image and Vision Computing*, 2009, vol. 27, no. 4, pp. 489–501.
60. Kumar A., Zhang D. Personal authentication using multiple palmprint representation. *Pattern Recognition*, 2005, vol. 38, no. 10, pp. 1695–1704.
61. Lahajnar F., Kovacic S. Rotation-invariant texture classification. *Pattern Recognition Letters*, 2003, vol. 24, no. 9–10, pp. 1151–1161.
62. Manthalkar R., Biswas P. K., Chatterji B. N. Rotation invariant texture classification using even symmetric Gabor filters. *Pattern Recognition Letters*, 2003, vol. 24, no. 12, pp. 2061–2068.
63. Mehrotra R., Namuduri K. R., Ranganathan N. Gabor filter-based edge detection. *Pattern Recognition*, 1992, vol. 25, no. 12, pp. 1479–1494.
64. Zhao Y. Q., Wang X. H., Wang X. F., Shih F. Y. Retinal vessels segmentation based on level set and region growing. *Pattern Recognition*, 2014, vol. 47, no. 7, pp. 2437–2446.
65. Raghu P. P., Yegnanarayana B. Segmentation of Gabor-Filtered Textures Using Deterministic Relaxation. *IEEE Transactions on Image Processing*, 1996, vol. 5, no. 12, pp. 1625–1636.
66. Rao K. R., Ben-Arie J. Lattice Architectures for Multiple-Scale Gaussian Convolution, Image Processing, Sinusoid-Based Transforms and Gabor Filtering. *Analog Integrated Circuits and Signal Processing*, 1993, vol. 4, no. 2, pp. 141–160.
67. Said H. E. S., Tan T. N., Baker K. D. Personal identification based on handwriting. *Pattern Recognition*, 2000, vol. 33, no. 1, pp. 149–160.
68. Sanger T. D. Stereo Disparity Computation Using Gabor Filters. *Biological Cybernetics*, 1988, vol. 59, no. 6, pp. 405–418.
69. Su Y.-M., Wang Jh.-F. A novel stroke extraction method for Chinese characters using Gabor filters. *Pattern Recognition*, 2003, vol. 36, no. 3, pp. 635–647.
70. Tsao T.-R., Chen V. A neural scheme for optical flow computation based on Gabor filters and generalized gradient method. *Neurocomputing*, 1994, vol. 6, no. 3, pp. 305–325.
71. Teuner A., Pichler O., Hosticka B. J. Unsupervised texture segmentation of images using tuned matched Gabor filters. *IEEE Transactions on Image Processing*, 1995, vol. 4, no. 6, pp. 863–870.
72. Chen J., Sato Y., Tamura Sh. Orientation Space Filtering for Multiple Orientation Line Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, vol. 22, no. 5, pp. 417–429.
73. Jones J. P., Palmer L. A. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophysiol.*, 1987, vol. 58, pp. 1233–1258.
74. Choras R. S., Andrysiak T., Choras M. Integrated color, texture and shape information for content-based image retrieval. *Pattern Analysis and Applications*, 2007, vol. 10, no. 4, pp. 333–343.
75. Diaz-Pernas F. J., Antón-Rodríguez M., Perozo-Rondón F. J., González-Ortega D. A multi-scale supervised orientational invariant neural architecture for natural texture classification. *Neurocomputing*, 2011, vol. 74, no. 18, pp. 3729–3740.
76. Randen T., Husoy J. H. Filtering for Texture Classification: A Comparative Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, vol. 21, no. 4, pp. 291–310.
77. Field D. J. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, 1987, vol. 4, no. 12, pp. 2379–2394.
78. Kyrki V. *Local and global feature extraction for invariant object recognition*. Ph. D. thesis, Lappeenranta University of Technology, 2002.
79. Ghose S., Oliver A., Marti R. et al. Statistical shape and texture model of quadrature phase information for prostate segmentation. *International Journal of Computer Assisted Radiology and Surgery*, 2012, vol. 7, no. 1, pp. 43–55.
80. Arrospeide J., Salgado L. Log-Gabor Filters for Image-based Vehicle Verification. *IEEE Transactions on Image Processing*, 2013, vol. 22, no. 6, pp. 2286–2295.
81. Kovesi P. *Invariant measures of image features from phase information*. Ph. D. thesis, University of Western Australia, 1996.
82. Serre T., Kouh M., Cadieu C. et al. A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex. Technical Report, MIT, Massachusetts, USA, 2005.
83. Bovik A. C., Clark M., Geisler W. S. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990, vol. 12, no. 1, pp. 55–73.
84. Manjunath B. S., Ma W. Y. Texture features for browsing, retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996, vol. 18, no. 8, pp. 837–842.
85. Jain A. K., Farrokhnia F. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 1991, vol. 24, no. 12, pp. 1167–1186.
86. Serre Th., Riesenhuber M. Realistic Modeling of Simple and Complex Cell Tuning in the HMAX Model, and Implications for Invariant Object Recognition in Cortex. Technical Report, MIT, Massachusetts, USA, 2004.
87. Pollen D. A., Ronner S. F. Visual Cortical Neurons as Localized Spatial Frequency Filters. *IEEE Trans. SMC*, 1983, vol. 13, no. 5, pp. 907–916.
88. Nicholls J. G. et al. *From neuron to brain*. 4th edition, 468 p. Sinauer Associates, 2001.

A. V. Kugaevskikh, PhD, Associate Professor, e-mail: a-kugaevskikh@yandex.ru  
Tyumen State Medical University, Tyumen, Russia, 625000

## Comparison Gabor Filter Parameters for Efficient Edge Detection

*In this paper I proposed generalized formula of Gabor filter. It was made according to the extensive review of sources. I carried out comparing of different coefficients influencing the form of a kernel on a DC component metrics. Thus the optimum formula was defined. The significant parameters (wavelength and scale of the filter) influencing quality of edges detection were revealed. The metrics of assessment on which comparing of different relations of significant parameters is carried out was offered. As a result the optimum ratio of significant parameters was found. It is proved that the equation offered by Petkov is optimum (equation 8). That was confirmed by the calculation of the DC component and the analytical comparison of the influence of different wavelengths on the form of the kernel graphs. Also, the description of the selection methods for the ratio of the wavelength and filter scale was performed. Quality assessment of edge detection metrics  $SNR_E$  was proposed. Its application made possible to speak of the optimality ratio the use of the half-response spatial frequency bandwidth (equation 23).*

**Keywords:** image analysis, gabor filter, wavelength, filter scale, edge detection, dc component

УДК 004.05

**С. М. Авдошин**, канд. техн. наук, проф., руководитель департамента программной инженерии факультета компьютерных наук, e-mail: savdoshin@hse.ru,  
**Е. Ю. Песочкая**, канд. экон. наук, доц., e-mail: epesotskaya@hse.ru,  
Национальный исследовательский университет "Высшая школа экономики" (НИУ ВШЭ)

## Интернет вещей: ритейл

*Рассматривается общий потенциал развития Интернета вещей в России, а также конкретные возможности использования Интернета вещей в ритейле. В рамках исследования проанализированы текущие тренды с изучением отдельных примеров решений в процессах закупок и обеспечения запасами. Исследованы основные процессы и возможности использования цифровых технологий для области торговли, приведены некоторые примеры целесообразности использования Интернета вещей для снижения издержек и получения дополнительной выгоды в маркетинге и обеспечении заказов. Сделаны выводы и предложены рекомендации по применению и использованию Интернета вещей в ритейле.*

**Ключевые слова:** цифровая трансформация, Интернет вещей, автоматизация, облачные технологии, мобильные технологии, мобильные приложения

### Введение

"Если вы думаете, что интернет изменил вашу жизнь, задумайтесь еще раз. Интернет вещей поменяет ваше представление об интернете и заново изменит все!" [1].

*Brendan O'Brien,*  
*основатель Aria Systems*

Скорее всего, среднестатистический покупатель и пользователь услуг ритейла будет знать совсем немного о явлении Интернета вещей (Internet of Things/IoT). Возможно, наиболее продвинутые пользователи слышали и знают о некоторых наиболее освещенных возможностях Интернета вещей, таких как шикарные дома или умные автомобили. Однако истинный потенциал Интернета вещей остается скрытым от большинства посетителей магазинов. Эксперты предсказывают, что Интернет вещей распространится очень быстро за следующие несколько лет и предложит целое направление новых служб и сервисов, улучшающих как жизнь потребителей, так и производительность компаний [2]. Когда Кевин Эштон впервые представил термин "Интернет вещей" во время презентации для Procter & Gamble в 1999 г., он даже не мог себе представить масштабы, которые приобретет этот термин в современном мире [3]. Первоначально технология, над которой работал Эштон, представляла собой замену штрихкодов на товарах, которые выполняют некоторые ручные операции по считыванию этих меток с помощью сканеров, на

электронные метки, которые могли бы собирать и передавать информацию не только о названии и местоположении продуктов, но и об их состоянии. Более 15 лет назад он заявил в своей презентации, что если бы предметы и процессы могли "говорить", то это изменило бы мир.

Последующее усовершенствование Интернета вещей заключалось в разработке устройств меньшего размера с большей вычислительной мощностью. В результате технического прогресса обычные в прошлом продукты оказались умными, взаимосвязанными с другими продуктами и программами, оборудованные датчиками, системами хранения данных и микропроцессорами [4].

Сейчас самое простое определение Интернета вещей — это сеть, к которой могут быть подключены самые разные предметы, приобретающие совершенно новые свойства и качества "умных вещей" за счет использования возможностей программирования и настроек, анализа данных, прогноза поведения, задания определенного сценария и умения взаимодействовать друг с другом [5].

Согласно прогнозам Gartner, в ближайшие годы число взаимосвязанных умных вещей увеличится в разы и может потенциально достигнуть 21 млрд к 2020 г. [6]. Этот тренд будет не только влиять на конечного пользователя, но также давать новые возможности различным индустриям и компаниям. Особенной прорыв уже наблюдается в области потребительского маркетинга, поскольку многие компании уже могут самостоятельно собирать и

использовать данные о своих клиентах, чтобы формировать более усовершенствованные ценностные предложения, учитывающие индивидуальные особенности клиентов, их предпочтения, геолокацию, средства и частоту передвижения, привычки и образ жизни.

Многие компании используют потенциал аналитических систем, чтобы иметь более подробное представление о клиентах. Например, некоторые страховые компании совершенствуют свои продуктовые портфели и структуру затрат благодаря аналитическим выкладкам по андеррайтингу и ценообразованию. Другие компании с помощью аналитики проводят экспериментальное изучение поведенческих моделей клиентов. Так, например, компания, владеющая ресторанами, может исследовать ценообразование и продвижение через франчайзинговую сеть. Эксперимент позволяет динамически регулировать цены на продукцию в зависимости от спроса, погодных условий, уровня запасов и близости времени закрытия заведений.

Обладая хорошим пониманием рынка, бизнес может трансформировать опыт продажи товаров. Компании аккумулируют информацию о покупках клиента, чтобы обеспечить более точные продажи и персонализированное обслуживание или даже предложить ему индивидуальный набор продуктов. Компания в области гостеприимства, используя геолокационные возможности, высылает на мобильные устройства находящихся неподалеку клиентов скидочные купоны. Таким образом, компания может в режиме реального времени отслеживать реакцию на свои предложения. Компания в области ипотечной недвижимости налаживает CRM-стратегию, чтобы клиенты на местах получали актуальные ссылки на объекты недвижимости.

Глобальные исследования Forrester говорят, что Интернет вещей является бизнес-трендом, при этом в отчете упоминается, что 23 % компаний уже используют Интернет вещей, а 29 % планируют начать делать это в течение следующих 12 месяцев [7]. Компании и фирмы используют Интернет вещей для преобразования своей бизнес-модели, переходя от разовых операций с продуктами к отношениям "продукт как сервис" и оптимизируя использование физических и финансовых ресурсов, а также создавая новые формы привлечения пользователей и клиентов.

Согласно норвежской исследовательской организации SINTEF, 90 % данных в мире были сформированы после 2010 г. [8]. Каждую секунду создаются более 205 000 новых гигабайтов, а это эквивалентно 150 млн книг. Это число данных, созданных в мире с помощью 10...20 млрд подключенных объектов с датчиками. При этом компания Cisco выяснила, что 99 % физических устройств пока не соединены [9], что открывает огромные возможности для экономики Интернета вещей.

Также аналитики считают, что по крайней мере одна компания будет обучать инвесторов с Wall Street, как проследить эволюцию от продукта к доходам от услуг. Например, Adobe ([www.adobe.com](http://www.adobe.com)) и Microsoft ([www.microsoft.com](http://www.microsoft.com)) уже перешли на продажу программного обеспечения как сервиса. Корпоративные клиенты и потребители будут автоматически спрашивать у продавцов о возможностях датчиков в мобильном приложении продукта при приобретении нового холодильника или кофемашины. Одна или несколько компаний будут сообщать о количестве подключенных продуктов, которыми они пользуются в работе с клиентами, а также о числе подключенных клиентов. Интернет вещей создаст огромные объемы данных, и аналитика этих данных станет специализированной дисциплиной в анализе данных.

### Интернет вещей в ритейле

Отрасль торговли на протяжении последних лет — одна из наиболее быстро развивающихся отраслей экономики России, опережающая по темпам роста многие отрасли, и ритейл до последнего времени был также очень успешен, показывая в среднем рост 25...35 % в год. Однако в 2015 г. проблемы в экономике вызвали снижение темпов роста доходов населения, что негативно повлияло на сферу розничной торговли. На первом месте оказались вопросы повышения эффективности бизнеса, в том числе построение оптимальной системы взаимодействия с поставщиками и повышение внутренней эффективности всех бизнес-процессов.

Также одним из вызовов ритейла последних лет является возрастающая фрагментированность покупателей в связи с ускорением темпа жизни. Это оказывает влияние на ассортимент, в частности, создает необходимость поддерживать большое число наименований. Поддержание широкого ассортимента, в свою очередь, требует создания достаточно сложной системы логистики от поставщика до торговой точки.

Цифровые технологии являются решением для многих этих проблем. Так, например, при поддержке информационных технологий от хорошего продавца уже не нужно ожидать абсолютного знания всего ассортимента, особенностей и деталей каждого продукта и его наличие — главным фокусом становится качественное обслуживание покупателя и удовлетворение его индивидуальных потребностей. Весь меняющийся ассортимент, его выкладку и наличие с легкостью запомнит компьютер и подскажет с помощью портативного помощника продавца.

Бизнес-процессы в торговле, как и везде, — планирование и закупка, взаимодействие с поставщиками, продажи и обслуживание в точках продаж, маркетинг и сервисное обслуживание (рис. 1).

Так, в ритейле можно выделить три этапа создания ценности: управление цепочкой поставок, ко-



Рис. 1. Основные процессы торговли

торые включают в себя закупку и управление запасами; операции в торговых точках, подразумевающие в том числе управление точками; взаимодействие с покупателем, что включает маркетинг, обеспечение заказов (т. е. непосредственно продажи) и сервисные услуги. На каждом этапе решения для Интернета вещей будут включать в себя совершенно разные приложения, сервисы, датчики; соответственно будут формироваться различные сценарии использования Интернета вещей. Эффекты тоже будут разными: в некоторых случаях мы уменьшим издержки и оптимизируем складские запасы, а в других сможем привлечь дополнительных покупателей или увеличить средний чек. Более того, многие аналитики уверены, что Интернет вещей является ключом к удовлетворенности и лояльности покупателей [10]. В каждой из трех звеньев возможны значительные варианты увеличения дохода и оптимизации издержек игроков за счет более качественного управления на основе анализа данных и прогнозирования спроса/продаж, использования "умных сумок" для автоматической оплаты товаров при покидании торговой точки, процессов "умной логистики". Все зависит от того, в какой процесс мы хотим внедрить Интернет вещей. Есть мнение, что к 2020 г. в мире будет больше "умных" устройств, чем людей.

### Управление цепочкой поставок и снабжение

В части снабжения и управления цепочкой поставок существуют системы (например, SAP Transportation Management), которые позволяют уменьшить затраты на содержание складов и повысить эффективность доставок. Автоматизация увеличивает прозрачность и уменьшает затраты, помогая менеджерам оптимизировать ресурсы в транспортной сети. Автоматическая система позволяет выбрать подходящую курьерскую службу с наименьшим тарифом для каждого заказа. В результате 98 % заказов доставляются вовремя, что помогает повысить доход и удовлетворенность потребителя. Новая система позволяет обрабатывать заказы на 50 % быстрее и повысить доступность товаров с 93 до 98 %. Эффект от экономии на тарифах курьерских служб составил 4 % [11].

RFID — одна из самых известных технологий, которую применяют в решениях для Интернета вещей. Так, RFID-метки помогают быстро и без участия живого консультанта-продавца предоставить клиенту всю интересующую его информацию о товаре.

RFID-метки помогают отслеживать товары по всей цепочке поставок и сообщать об их состоянии.

Технологии IoT не только дают исчерпывающую информацию о том, когда прибудет та или иная партия товара, но также помогают сократить расходы за счет более точного измерения внешних факторов, воздействующих на товар, и оценки их влияния, например, влияние повышенной температуры на протяжении всей цепочки поставки или тряски во время перевозки, влажности, повышенного давления на упаковку. Фактически Интернет вещей предлагает использовать новые стандарты контроля, эффективности, прозрачности и гибкости — ключевых признаков успешной цепочки поставок.

Нужно отметить, что с помощью высокотехнологичных решений ритейлеры действительно могут создавать для потребителей абсолютно новый уровень сервиса, сопоставляя данные о спросе и планировании пополнения запасов, основанных на потребительских данных в реальном времени. Это приводит к повышению точности прогнозирования, снижению затрат на логистику и оптимизации запасов. В зависимости от формата торговой точки и ее местоположения могут быть составлены ассортиментные матрицы, учитывающие специфику работы магазина. Высокая точность планирования и оперативный контроль ассортимента торговых точек создают основу для оптимального товарооборота розничной компании, позволяя снизить объемы невостребованных запасов, избежать дефицита и сократить возможные потери торговой выручки.

### Операции в торговых точках

Существует огромное число решений<sup>1</sup> для каждого бизнес-процесса в ритейле и множество сервисов для аналитики в торговых точках. Накопленные крупными торговыми сетями значительные объемы данных могут дать топ-менеджерам массу полезной аналитической информации: какие товары пользуются спросом, обеспечено ли их достаточное количество на складе, налажены ли поставки, какие магазины наиболее прибыльны и т. д. Для аналитики используются такие инструменты, как IBM (www.ibm.com), Microsoft, Oracle (www.oracle.com), SAP. Так, платформа SAP HANA (<http://www.sap.com/product/technology-platform/hana.html>) позволяет супермаркету с более 10 тыс. наименований

<sup>1</sup> <https://solutionexplorer.sap.com/solexp/ui>

товаров затратить на получение детализированной информации по ним не 15 дней, как при ручном анализе, а всего 5 мин. Система автоматически прогнозирует спрос и предлагает ряд решений для конкретной ситуации: планирование промокампаний, перемещение товаров на другие склады и магазины, возврат поставщикам и другие действия. В итоге время реакции на выявленные возможности увеличения продаж сокращаются с месяцев до нескольких дней. А инструмент SAP for Retail<sup>1</sup> позволяет в полной мере организовать контроль процессов на всех уровнях управления в торговой сети, сохранив при этом гибкость организации и способность реакции на изменения внешней среды. Решение задачи по построению системы корпоративного управления начинается со стратегического планирования, проектирования структуры бизнес-подразделений компании и показателей эффективности их работы и заканчивается контролем за результатами деятельности филиалов и отделов по календарным периодам. Сложность и важность решения такой задачи, скорость, с которой активная бизнес-среда требует реакции на изменения, определяет необходимость концентрации ресурсов и выбора инструментов, адекватных масштабам задачи. Возможности решения охватывают весь цикл создания добавленной стоимости и поддерживают все процессы, обеспечивающие эффективность бизнеса. Хотя сам по себе инструмент не предназначался для Интернета вещей, его компоненты можно легко использовать для создания новых приложений или сервисов в концепции Интернета вещей.

Достаточно успешно продемонстрировал себя сервис анализа потоков клиентов для более сфокусированного размещения товаров на витринах. Это стало возможно в том числе за счет использования средств анализа и алгоритмов Machine learning [12]. Суперкомпьютеры сегодня способны на основе накапливаемых данных создавать персонализированные торговые предложения для каждого отдельного человека. Персонализированная реклама и онлайн мерчандайзинг открывают огромные возможности для ритейла, особенно в перспективе прогнозирования спроса и покупок.

Для повышения эффективности операций в торговых точках магазины устанавливают множество датчиков и камер на всей торговой площади, что делает возможным отслеживание потоков покупателей в магазине в течение дня, выявление тех областей магазина, где покупателей наибольшее количество (зоны повышенного спроса).

С помощью Bluetooth- или beacon-точек можно отследить перемещение клиентов по торговым залам. Такие данные могут использоваться для оптимизации проводимых промоакций и других целей по стимулированию продаж за счет расположения

товаров в магазине. Зоны, где проходит меньше всего покупателей, обозначаются как неэффективные. Программа прогнозирует, как будут двигаться покупатели, если переместить товары, обладающие наибольшей популярностью, и предлагает наиболее эффективные варианты размещения ассортимента.

Более продвинутая система "умный пол" с помощью цифровой камеры фиксирует покупателя магазина по его обуви и отслеживает его дальнейшие передвижения, обращая большое внимание на места, где покупатели проводят больше всего времени. Все данные о поведении человека по сети будут моментально передаваться в аналитическую систему. Нужно отметить, что "умный пол", который фиксирует все передвижения потока посетителей в точке продаж, обходится дешевле, чем анализ стандартного видеопотока, и проанализировать данные "умного пола" можно в реальном времени, причем с минимальными усилиями.

В перспективе "умными" становятся не только пространство магазина, но и тележки, которые могут генерить и накапливать электроэнергию благодаря усилиям покупателей, определять состав покупок, находящихся внутри тележки для быстрой оплаты на терминале, прокладывать оптимальный путь постоянного покупателя по его отпечатку пальца.

"Умные рамки" могут позволить решить проблему очередей, более оптимально спланировать спрос на товары, сократить издержки на хранение и персонал, сократить время покупки для покупателя и увеличить поток (рис. 2).

Пока что для навигации по магазину наибольшую популярность на Западе набирает сервис, который помогает клиентам в выборе товаров с помощью приложения, привязанного к сети магазина. Приложение позволяет вводить либо продукты,



Рис. 2. Схема использования датчиков на примере "умных рамок" в магазине

<sup>1</sup> <http://go.sap.com/cis/solution/industry/retail.html>

нужные для покупки, либо блюда, которые планирует приготовить покупатель. Оно автоматически показывает точки внутри магазина, где находятся необходимые продукты, может самостоятельно выбирать среди подобных продуктов в зависимости от того, от какого продукта нужно избавиться, либо от чека покупателя. Дополнительно приложение "подсказывает" покупателю, какие продукты, не входящие в список, ему следует купить для получения лучшего вкуса, согласно рецептам, хранящимся в базе программы.

В 2013 г. был разработан прототип программного обеспечения Smart Vending<sup>1</sup> для торговых автоматов. Концепция такого автомата предполагает, что он оснащен сенсорным дисплеем, используя который, покупатель сможет войти в систему, чтобы получить персонализированное обслуживание. Новый софт позволяет собирать информацию о покупках и предпочтениях покупателей для ее последующего анализа. По его результатам при последующих покупках автомат сможет выдавать каждому покупателю наиболее подходящие варианты товаров и сервиса.

"Умный торговый автомат" предполагает различные возможности оплаты покупки, включая оплату со смартфона, получение дополнительных данных о пользователе из социальных сетей с его согласия, возможность отправки товара в качестве презента друзьям, отправки предложений продавцу по ассортименту и др. Стоит отметить, что по желанию покупателя совершить покупку в "умном автомате" можно будет и без идентификации в системе, как в обычном автомате.

### **Взаимодействие с покупателями**

Один из крупнейших ритейлеров Walmart решил соединить данные, получаемые при продаже продуктов, и... прогноз погоды. На основе этого компания выявила несколько паттернов поведения клиентов и теперь в режиме реального времени может предсказывать потребление тех или иных продуктов в зависимости от изменения погоды. Это помогает быстро формировать заказы на склад и вовремя пополнять полки магазина. Например, при росте температуры выше 26 °С и слабом ветре система формирует заказ на ингредиенты для салата. Nordstorm анализирует общие тренды на сайте Pinterest.com и предпочтения друзей покупателя. На основе полученных данных потребителю, который авторизовался в Сети, на входе предлагают актуальные для него предложения одежды. Гигант онлайн-торговли Amazon предложил своим постоянным клиентам с определенным набором товаров интересное устройство — Dash Button, которое подключается к сети Wi-Fi. Как только возникает по-

требность в закупке, клиент просто нажимает кнопку. Все остальное (формирование заказа и оплата) происходит автоматически, без участия клиента.

Существует много решений для пополнения домашнего холодильника. Используя метки и датчики для каждого типа продуктов, холодильник может сам делать заказ продуктов на дом. Потребителю остается только подтвердить время приезда курьера. Компания Nespresso разработала профессиональную кофемашину, подключенную к Интернету. Это отличный пример того, как Интернет вещей трансформирует розничную торговлю. Подключение кофемашин к Интернету может показаться просто рекламным трюком, но подобное решение дает реальные преимущества. Оно позволяет готовить кофе высочайшего качества, контролировать запасы кофейных зерен и необходимость в обслуживании. Так, отслеживая частоту использования, кофемашинка своевременно информирует о необходимости проведения технического обслуживания. Она контролирует температуру, вибрацию и давление, что гарантирует высочайшее качество напитка.

Среди существующих примеров удачных решений можно отметить дисконтные карты (управление программами лояльности). Так, постоянным покупателям в Сети выдается дискретная карта, привязанная к человеку, а все покупки клиента записываются в систему и анализируются самые популярные нестандартные покупки. На наиболее популярные для клиента продукты магазин может предоставлять ему дополнительную скидку, стимулируя покупателя делать покупки только в данном магазине.

Можно пойти еще дальше: если пользователь уже совершал покупку в магазине, существует возможность в режиме реального времени при посещении магазина предложить ему скидку или купон на покупку здесь и сейчас. Такие приятные сюрпризы хорошо сказываются на потребительском опыте и эмоциях от посещения магазина.

Что касается приложений, способствующих росту продаж за счет "умных" устройств и датчиков, для клиента, скачавшего приложение магазина, существует возможность получения промоакций в момент приближения к магазину (рис. 3).

Клиент узнает о специальных предложениях каждого магазина, а установленная в магазине beacon-точка идентифицирует клиентов, находящихся неподалеку, и сообщает им о скидках, специальных акциях, детальной информации по товару (отзывы, состав, подбор размера, похожие товары, наличие), поиску местонахождения товара в магазине по метке, а также мгновенной покупке или бронированию в другом магазине без очередей на кассе.

Магазин, в свою очередь, получает собственную выгоду от этого процесса, которая заключается в сборе аналитики о предпочтениях и покупках с привязкой к конкретному покупателю, что позво-

<sup>1</sup> <https://www.youtube.com/watch?v=roeU65hiCTI>



Рис. 3. Активный контакт с клиентом через приложение

ляет увеличить средний чек и обеспечить клиентский поток наравне со снижениями издержек на персонал за счет самостоятельного выбора и оплаты товара в магазине. Также магазин получает лучший контроль за ассортиментом (наполненность полок, перемещение товаров в цепочке поставок) и оптимизацию торгового пространства за счет информации о передвижениях клиентов и их истории действий в приложении.

Важнейшей с точки зрения обеспечения задач маркетинга является интеграция данных, связанных с поведением потребителя и приверженностью его к тем или иным брендам. На основе информации по всем функциональным областям из компонентов решения IoT, других систем и внешних баз данных маркетологи имеют возможность строить отчетность с помощью инструментов бизнес-аналитики, что позволяет контролировать потребительское поведение по всему циклу взаимодействия с клиентами.

Точное понимание предпочтений и финансовых возможностей потребителя, способность прогнозировать их изменение и формировать потребительские вкусы — цели, которые ставят перед собой крупнейшие продуктовые сети и ведущие ритейлеры.

### Заключение

Сегодня дальновидные ритейлеры пересматривают свои представления о том, как современные покупатели посещают магазины. Реагируя на изменение привычек потребителей, розничные торговые компании объединяют цифровые технологии, мобильный доступ и обслуживание покупателей в магазинах. Новейшие технологии имеют для клиентов ключевое значение. Чтобы опередить конкурентов, необходимо произвести впечатление на покупателей. Можно предположить, что те ритейлеры, которые будут внедрять цифровые технологии, ста-

нут более конкурентоспособными на российском рынке за счет повышения лояльности и удобства для клиента. Кроме того, они могут приблизиться к мировым лидерам, что создаст дополнительный потенциал для выхода на рынки других стран.

Уже сейчас ритейлеры развивают новые способы взаимодействия с потребительскими каналами и клиентами. Интернет вещей позволяет ритейлерам использовать новые сервисы для поддержки существующих процессов закупок, продаж, маркетинга, предсказания и анализа рынка и объемов потребительских товаров, соответственно приспосабливаться к постоянно меняющемуся рынку потребительских товаров. Среди наиболее популярных сегодня — это оптимизация управления цепочками поставок, мониторинг и учет товаров, оборудованных датчиками, online-управление складом и запасами, бесконтактные платежи и оплата товаров. Большинство усилий направлено на повышение продаж и оптимизацию издержек, в то время как следующим шагом развития Интернета вещей для ритейлера будет достижение удовлетворенности покупателей. Только так мелкие ритейлеры смогут бороться с конкуренцией со стороны крупных игроков, а крупные сети смогут выигрывать в войне за потребителя.

Уже сейчас некоторые компании приходят к пониманию, что Интернет вещей представляет собой наилучшую возможность выделиться на фоне конкурентов, но сейчас — в тяжелое для отрасли время — игроки сфокусированы на оптимизации бизнес-процессов и снижении затрат. Также в некоторых случаях сдерживающим фактором являются и сами потребители, обладающие консервативными привычками и стремящиеся к минимальным изменениям "привычного шоппинга". Не все покупатели готовы перейти на новый терминал самообслуживания, который самостоятельно просканирует все сделанные покупки и предложит оплатить их взмахом банковской карты или мобильного телефона. Многие по-прежнему больше доверяют традиционному живому общению, нежели экрану компьютера, и готовы простоять огромную очередь в супермаркете из-за недоверия к новым технологиям. Но не исключено, что уже через пять лет в наиболее продвинутых магазинах будут доступны только самые свежие товары, которые покупатели хотят видеть на прилавках, а в торговой зоне совсем не будет продавцов, кассиров и других сотрудников магазина.

### Список литературы

1. **O'Brien B.** The 'Internet of Things' // Aria Systems, 2017 [Official website]. URL: <https://www.ariasystems.com/blog/internet-things-quotes-consider/> (accessed 01.03.2017).
2. **Understanding** the Internet of Things (IoT) // London: GSM Association, 2014 [Official website]. URL: [http://www.gsma.com/connectedliving/wp-content/uploads/2014/08/cl\\_iot\\_wp\\_07\\_14.pdf](http://www.gsma.com/connectedliving/wp-content/uploads/2014/08/cl_iot_wp_07_14.pdf) (accessed 01.03.2017).

3. **Ashton K.** That 'Internet of Things' Thing. In the real world, things matter more than ideas // *RFID journal*, 2009 [Official website]. URL: <http://www.rfidjournal.com/articles/view?4986> (accessed 01.03.2017).

4. **Porter M. E., Heppelmann J. E.** How Smart, Connected Products are Transforming Competition // *Harvard Business Review*, 2014 [Official website]. URL: <https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition> (accessed 01.03.2017).

5. **Miorandi D., Sicari S., Pellegrini F., Chlamtac I.** Internet of things: Vision, applications and research challenges // *Ad Hoc Networks* 10 (2012) 1497–1516. URL: <https://irinsubria.uninsubria.it/retrieve/handle/11383/1762288/2389/IOT.pdf> (accessed 01.03.2017).

6. **Meulen R.** Enterprises to Bolster IoT Revenue // *Gartner Newsroom*, 2015 [Official website]. URL: <http://www.gartner.com/newsroom/id/3165317> (accessed 01.03.2017).

7. **Pelino M. Gillett F. E., Voce Christopher, Mai Michelle, Easton Laura.** TechRadar: Internet Of Things, Q1 2016 // *Forrester* 2016, URL: <https://www.10thmagnitude.com/wp-content/uploads/2016/07/10th-Magnitude-IoT-Orange-Paper-2016.pdf> (accessed 01.03.2017).

8. **Dragland A.** Bid Data — for Better Or Worth // *SINTEF*, 2013 [Official website]. URL: <https://www.sintef.no/en/latest-news/big-data--for-better-or-worse> (accessed 01.03.2017).

9. **Bradley J., Barbier J., Handler D.** Embracing the Internet of Everything To Capture Your Share of \$14.4 Trillion // *Cisco White Paper*, 2013 [Official website]. URL: [https://www.cisco.com/web/about/ac79/docs/innov/IoE\\_Economy.pdf](https://www.cisco.com/web/about/ac79/docs/innov/IoE_Economy.pdf) (accessed 01.03.2017).

10. **Kirk S.** The Big Book of MoT — Making the Most of the Monetization of Things // *Aria Systems*, White Paper, 2015 [Official website]. URL: [http://info.ariasystems.com/rs/705-IGF-584/images/E-book\\_The\\_Big%20Book%20of%20MoT\\_FINAL\\_11132015%201.pdf](http://info.ariasystems.com/rs/705-IGF-584/images/E-book_The_Big%20Book%20of%20MoT_FINAL_11132015%201.pdf) (accessed 01.03.2017).

11. **King B.** SAP Transportation Management // *SAP*, 2015 [Official website]. URL: <http://fm.sap.com/data/UPLOAD/files/SAP%20TM%2093%20Overview%20June%202015%20-%20Final.pdf> (accessed 01.03.2017).

12. **Holdowsky J., Mahto M., Raynor M. E., Cotteleer M.** Inside the Internet of Things (IoT) // *Deloitte University Press*, 2015 [Official website]. URL: [https://dupress.deloitte.com/content/dam/dup-us-en/articles/iot-primer-iot-technologies-applications/DUP\\_1102\\_Inside-TheInternetOfThings.pdf](https://dupress.deloitte.com/content/dam/dup-us-en/articles/iot-primer-iot-technologies-applications/DUP_1102_Inside-TheInternetOfThings.pdf) (assessed 01.03.2017).

**S. M. Avdoshin**, PhD, Professor, Head of Software Engineering School,  
Faculty of Computer Science, HSE, e-mail: [savdoshin@hse.ru](mailto:savdoshin@hse.ru),

**E. Yu. Pesotskaya**, Docent, Faculty of Computer Science, HSE, e-mail: [epesotskaya@hse.ru](mailto:epesotskaya@hse.ru)  
National Research University Higher School of Economics (HSE)

## Internet of Things: Retail

*The paper describes the possibility of using the Internet of Things in retail industry as well as potential of using internet of things in Russia. The research objective is to identify the possibilities of using Internet of Things in retail process such as supply, logistics, client service and marketing and to give the examples for getting economy and cost reduction in industry. The author provides an overview of possible tools and insights that transform retailers' businesses and recommendations of using internet of things technology in retail. Authors analyze IoT possibilities that can enhance and refine business processes, decrease operational expenditures, better understand and communicate with consumers, taking into considerations existing aspects and current worldwide trends of global transformation, digitalization and cloud technologies.*

**Keywords:** Digital Transformation, Internet of Things, Automation, Cloud Technologies, Mobile, Mobile application (apps)

### References

1. **O'Brien B.** The 'Internet of Things' [Electronic resource], *Aria Systems*, 2017 [Official website]. URL: <https://www.ariasystems.com/blog/internet-things-quotes-consider/> (accessed 01.03.2017).

2. **Understanding** the Internet of Things (IoT) // London: GSM Association, 2014 [Official website]. URL: [http://www.gsma.com/connectedliving/wp-content/uploads/2014/08/cl\\_iot\\_wp\\_07\\_14.pdf](http://www.gsma.com/connectedliving/wp-content/uploads/2014/08/cl_iot_wp_07_14.pdf) (accessed 01.03.2017).

3. **Ashton K.** That 'Internet of Things' Thing. In the real world, things matter more than ideas, *RFID journal*, 2009 [Official website]. URL: <http://www.rfidjournal.com/articles/view?4986> (accessed 01.03.2017).

4. **Porter M. E., Heppelmann J. E.** How Smart, Connected Products are Transforming Competition, *Harvard Business Review*, 2014 [Official website]. URL: <https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition> (accessed 01.03.2017).

5. **Miorandi D., Sicari S., Pellegrini F., Chlamtac I.** Internet of things: Vision, applications and research challenges, *Ad Hoc Networks* 10 (2012) 1497–1516. URL: <https://irinsubria.uninsubria.it/retrieve/handle/11383/1762288/2389/IOT.pdf> (accessed 01.03.2017).

6. **Meulen R.** Enterprises to Bolster IoT Revenue, *Gartner Newsroom*, 2015 [Official website]. URL: <http://www.gartner.com/newsroom/id/3165317> (accessed 01.03.2017).

7. **Pelino M. Gillett F E, Voce Christopher, Mai Michelle, Easton Laura.** TechRadar: Internet Of Things, Q1 2016, *Forrester* 2016. URL: <https://www.10thmagnitude.com/wp-content/uploads/2016/07/10th-Magnitude-IoT-Orange-Paper-2016.pdf> (accessed 01.03.2017).

8. **Dragland A.** Bid Data — for Better Or Worth, *SINTEF*, 2013 [Official website]. URL: <https://www.sintef.no/en/latest-news/big-data-for-better-or-worse> (accessed 01.03.2017).

9. **Bradley J., Barbier J., Handler D.** Embracing the Internet of Everything To Capture Your Share of \$14.4 Trillion, *Cisco White Paper*, 2013 [Official website]. URL: [https://www.cisco.com/web/about/ac79/docs/innov/IoE\\_Economy.pdf](https://www.cisco.com/web/about/ac79/docs/innov/IoE_Economy.pdf) (accessed 01.03.2017).

10. **Kirk S.** The Big Book of MoT — Making the Most of the Monetization of Things, *Aria Systems*, White Paper, 2015 [Official website]. URL: [http://info.ariasystems.com/rs/705-IGF-584/images/E-book\\_The\\_Big%20Book%20of%20MoT\\_FINAL\\_11132015%201.pdf](http://info.ariasystems.com/rs/705-IGF-584/images/E-book_The_Big%20Book%20of%20MoT_FINAL_11132015%201.pdf) (accessed 01.03.2017).

11. **King B.** SAP Transportation Management, *SAP*, 2015 [Official website]. URL: <http://fm.sap.com/data/UPLOAD/files/SAP%20TM%2093%20Overview%20June%202015%20-%20Final.pdf> (accessed 01.03.2017).

12. **Holdowsky J., Mahto M., Raynor M. E., Cotteleer M.** Inside the Internet of Things (IoT), *Deloitte University Press*, 2015 [Official website]. URL: [https://dupress.deloitte.com/content/dam/dup-us-en/articles/iot-primer-iot-technologies-applications/DUP\\_1102\\_Inside-TheInternetOfThings.pdf](https://dupress.deloitte.com/content/dam/dup-us-en/articles/iot-primer-iot-technologies-applications/DUP_1102_Inside-TheInternetOfThings.pdf) (assessed 01.03.2017).

Д. А. Печников, канд. техн. наук, доц., e-mail: 19pda72@bk.ru  
Военно-морская академия имени Адмирала Флота Советского Союза Н. Г. Кузнецова,  
г. Санкт-Петербург, Российская Федерация

## Применение методов интервальной оценки результатов контроля знаний в компьютерных системах тестирования

*Рассматривается применение моделей и процедур интервальной оценки результатов контроля знаний в компьютерных системах с применением методов, основанных на законах биномиального распределения и геометрического распределения, как частного случая отрицательного биномиального распределения. Предложен подход к переходу от точечных оценок эмпирической частоты успешного выполнения тестовых заданий к интервальным оценкам вероятности этой успешности, позволяющим определять погрешность и доверительный интервал полученных оценок. Разработаны модели критериально-ориентированного оценивания результатов контроля знаний, которые ориентированы на реализацию в компьютерных системах тестирования.*

**Ключевые слова:** доверительный интервал, интервальная оценка, компьютерная система тестирования, критериально-ориентированное тестирование, биномиальный закон распределения

### Введение

В тенденциях развития тренажно-обучающих систем (ТОС) отмечается следующий ряд основных противоречий:

- между ростом возможностей ТОС как средств моделирования управляемых объектов и неизменностью их дидактических характеристик;
- между необходимостью индивидуализации процесса подготовки и ограниченными психофизиологическими возможностями обучающихся по управлению обучением в группах с количественным составом более 5...7 обучаемых;
- между ростом числа и сложности задач управления обучением и неизменностью психофизиологических возможностей руководителей обучения.

В качестве средства разрешения этих противоречий рассматривается автоматизация в ТОС функций управления обучением.

Ключевой функцией, без автоматизации которой нельзя автоматизировать все остальные функции управления обучением, является функция педагогического контроля качества подготовки обучаемых. В педагогике эта функция определяется как функция текущего (корректирующего) контроля, которая имеет целью не столько констатацию уровня подготовки обучаемых, сколько осуществление корректуры этого уровня. Сегодня эта функция чаще всего реализуется с применением компьютерных систем тестирования (КСТ).

Текущий контроль, как показано в работах [1—5 и др.], может быть реализован только методами кри-

териально-ориентированного тестирования, идентифицирующими соотношения "усвоил — не усвоил" между конкретными обучаемыми и конкретными познавательными объектами, в отношении которых реализуется рассматриваемый процесс обучения. При этом сам тест должен полностью соответствовать содержанию обучения и разрабатываться тем преподавателем, который осуществлял обучение [2].

В настоящее время в КСТ наибольшее распространение получили тестовые задания (ТЗ) закрытого типа с единственным верным ответом. По имеющимся данным [2, 6—8] доля заданий в практике образовательного процесса колеблется от 86 до 98 %. Такая статистика определяет целесообразность первоочередного решения задач интервального оценивания в отношении ТЗ закрытого типа с единственным верным ответом.

В таких ТЗ успешность выполнения задания оценивается путем установления факта совпадения ответа обучаемого с заранее заданным правильным вариантом ответа. Любой другой вариант ответа является ошибочным. Для последующей статистической обработки результаты ("верно" — "неверно") ТЗ представляются в виде дихотомической переменной (0—1). Безошибочность (правильность) ответов обучаемого определяется в виде показателя частности  $B$ :

$$B = \frac{\sum_{i=1}^n j_i}{n}, \quad (1)$$

где  $i$  ( $i = \overline{1, n}$ ) — номер ТЗ;  $n$  — число ТЗ в тесте;  $j_i$  ( $j_i = 0; 1$ ) — результат выполнения ТЗ ( $j_i = 1$  — верный ответ,  $j_i = 0$  — неверный ответ).

Все существующие процедуры обработки результатов выполнения тестов в КСТ основаны на оценке (1).

Такие оценки являются точечными и не несут информации о погрешности, а потому не могут быть использованы в целях обеспечения оперативности процедуры тестирования (прекращения тестирования при получении оценки, имеющей заданную погрешность). Ниже предлагаются модели и процедуры интервального оценивания результатов выполнения систем ТЗ закрытого типа с единственным верным ответом, которые в отличие от точечных оценок несут в себе информацию как о своем наиболее вероятном значении, так и о его погрешности.

#### **Постановка задачи интервальной оценки результатов критериально-ориентированного тестирования**

Предлагаемые модели и процедуры базируются на том факте, что показатель "правильность" является дихотомической переменной, а процедура его определения в процессе решения тестирующей выборки из  $n$  заданий соответствует схеме Бернулли.

Для корректного применения схемы Бернулли должны быть выполнены следующие условия:

- каждое испытание имеет ровно два исхода, условно называемых успехом и неудачей;
- независимость испытаний: результат очередного эксперимента не должен зависеть от результатов предыдущих экспериментов;
- вероятность успеха должна быть постоянной (фиксированной) для всех испытаний.

Первое условие всегда выполняется за счет того, что в ТЗ закрытого типа результаты с единственным верным ответом представлены дихотомической переменной  $j_i$  ( $j_i = 0; 1$ ). Второе условие в критериально-ориентированном тестировании выполняется всегда за счет возможности предъявления всех ТЗ в случайном порядке. Третье условие выполняется только в случае, когда обучаемый никаких новых сведений об объектах тестирования из формулировки ТЗ не получает. Выполнение этого требования ограничено существованием у обучаемого возможности угадывания верного решения.

Возможность угадывания правильного решения имеет место во всех формах ТЗ закрытого типа, но в форме задания с единственным верным ответом вероятность такого угадывания наиболее высока. Для исключения этой возможности и достижения истинности результатов тестирования в тестологии используются два метода. Суть первого метода состоит в снижении вероятности угадывания путем увеличения в ТЗ числа дистракторов (альтернатив

решений). В тестологии [6—9 и др.] считается, что достоверность оценки выполнения систем закрытых ТЗ с единственным верным ответом достигается, если во всех ТЗ число дистракторов равно или больше пяти. Суть второго метода состоит во введении поправки на угадывание в результат (1) тестирования. В случае критериально-ориентированного тестирования используются фиксированные поправки, виды которых приведены в [6—10 и др.]. Оба метода реализуемы в любой из КСТ.

С учетом изложенного выше примем допущение о том, что вероятность правильного выполнения ТЗ определяется только знаниями обучаемого и не зависит от фактора угадывания. В соответствии с этим допущением безошибочность выполнения ТЗ обучаемым характеризуется некоторой постоянной вероятностью  $p = \text{const}$  правильного выполнения ТЗ, что и определяет полное соответствие процедуры критериально-ориентированного тестирования схеме Бернулли.

#### **Решение поставленной задачи интервальной оценки результатов критериально-ориентированного тестирования**

Пусть в результате решения обучаемым выборки из  $n$  ТЗ получен вектор (примеры векторов приведены в табл. 1)

$$\mathbf{J} = (j_1, \dots, j_i, \dots, j_n), \quad (2)$$

в котором  $j_i = 0; 1$  есть оценки успешности выполнения соответствующего задания ( $j_i = 1$  — ТЗ выполнено без ошибок;  $j_i = 0$  — ТЗ выполнено с ошибкой). Требуется оценить вероятность  $p$  безошибочного решения ТЗ.

Эмпирические данные, полученные по схеме Бернулли, подчиняются законам таких распределений, как биномиальное, геометрическое и отрицательное биномиальное. В целях интервальной оценки вероятности  $p$  безошибочного выполнения ТЗ целесообразно использовать те законы распределения, которые принципиально обеспечивают обработку любого числа ( $i, i = \overline{1, +\infty}$ ) ТЗ. Такому требованию соответствуют только законы биномиального и геометрического распределения.

Биномиальное распределение имеет место в тех случаях, когда последовательность испытаний Бернулли обрывается после проведения фиксированного числа  $n$  испытаний. При этом под биномиальной случайной величиной  $X$  понимается число успехов в серии из  $n$  испытаний Бернулли.

Вероятность  $p(\mathbf{J})$  появления вектора  $\mathbf{J}$  при биномиальном распределении оценивается как

$$p(\mathbf{J}) = C_n^k p^n - k q^k \text{ при } C_n^k = \frac{n!}{k!(n-k)!}, \quad (3)$$

где  $\mathbf{J}$  — рассматриваемый вектор результатов тестирования;  $p$  — вероятность безошибочного вы-

полнения ТЗ;  $q = 1 - p$  — вероятность выполнения задания с ошибкой;  $k$  — число заданий, выполненных с ошибкой;  $n$  — общее число заданий;  $n - k$  — число заданий, выполненных без ошибок.

Геометрическое распределение возникает при обрыве испытаний сразу же после первого успеха или неудачи. При этом вероятность  $p(\mathbf{J}; j_i = 1)$  успеха оценивается как

$$p(\mathbf{J}; j_i = 1) = p^h q, \quad (4)$$

а вероятность  $p(\mathbf{J}; j_i = 0)$  неудачи как

$$p(\mathbf{J}; j_i = 0) = q^h p, \quad (5)$$

где  $h(h = \overline{1, n})$  — номер  $i$  последнего ТЗ, выполненного без ошибки, или, наоборот, с ошибкой (число "неудач" до первого "успеха" или число "успехов" до первой "неудачи").

Для обеспечения возможности использования геометрического распределения в отношении векторов (2) любой конфигурации введем понятие вектора  $\mathbf{H}$  непрерывного результата (успеха или неудачи), под которым будем понимать любую часть вектора (2), каждый из элементов которой равен 1 или 0. Если все элементы вектора равны 1 ( $j_i = 1$ ) — вектор  $\mathbf{H}^+$  непрерывного успеха; если все элементы вектора равны 0 ( $j_i = 0$ ) — вектор  $\mathbf{H}^-$  непрерывной неудачи. Например, в табл. 1 для варианта 6 вектора  $\mathbf{J}$  это векторы  $\mathbf{H}_1^+ = (j_1, j_2)$ ,  $\mathbf{H}_2^- = (j_3, j_4, j_5, j_6, j_7, j_8, j_9)$ ,  $\mathbf{H}_3^+ = (j_{10}, j_{11}, j_{12})$ .

Вероятность появления вектора (1) можно оценивать с двух позиций:

1) как вероятность события  $L$ , заключающегося в том, что при выполнении  $n$  заданий обучаемый достиг  $(n - k)$  успехов (допущено  $k$  ошибок);

2) как вероятность события  $H$ , заключающегося в появлении в векторе (1)  $m$  векторов непрерывного результата.

Вероятность  $p(L)$  наступления события  $L$  оценивается по биномиальному закону. В соответствии с формулой (3) при определении доверительного интервала (ДИ) вероятности  $p$  успешного вы-

полнения ТЗ корректно использование следующих формул:

- оценка точного ДИ биномиального распределения [3]:

$$\left( \frac{(n-k)}{(n-k) + (k+1)F_{2(k+1), 2(n-k), 1-\varepsilon/2}}, \frac{(n-k+1)F_{2(n-k+1), 2k, 1-\varepsilon/2}}{k + (n-k+1)F_{2(n-k+1), 2k, 1-\varepsilon/2}} \right), \quad (6)$$

где  $n$  — число испытаний;  $k$  — число ошибок, а  $F_{f, g, \alpha}$  — квантиль порядка  $\alpha$  распределения  $F$  с  $f, g$  степенями свободы;  $\varepsilon$  — длина доверительного интервала.

- оценка приближенного ДИ биномиального распределения [3]:

$$\left( p^* - u_{1-\varepsilon/2} \sqrt{\frac{p^*(1-p^*)}{n}}, p^* + u_{1-\varepsilon/2} \sqrt{\frac{p^*(1-p^*)}{n}} \right) = \left( \frac{n-k}{n} - u_{1-\varepsilon/2} \sqrt{\frac{(n-k)k}{n^2}}, \frac{n-k}{n} + u_{1-\varepsilon/2} \sqrt{\frac{(n-k)k}{n^2}} \right), \quad (7)$$

где  $p^*$  — оценка эмпирической частоты правильного выполнения ТЗ;  $u_{1-\varepsilon/2}$  — квантили стандартного нормального распределения порядка  $1 - \varepsilon/2$ .

Геометрическое распределение является единственным дискретным распределением со свойством отсутствия памяти (последствия): знание об отсутствии успеха (неудачи) в предыдущих опытах никак не влияет на распределение оставшегося числа опытов до появления успеха (неудачи). Используем это свойство для оценки вероятности события  $H$ , которое заключается в появлении в векторе (2)  $m$  векторов непрерывного результата.

В соответствии с ним события  $h_s \in H$  появления в составе вектора (2) отдельных векторов  $\mathbf{H}_s$  непрерывного результата относятся к категории попарно несовместных событий. Поэтому вероятность  $p(H)$  любого из вариантов их реализаций (см. табл. 1) оценивается по формуле

$$p(H) = p(h_1 \cup \dots \cup h_s \cup \dots \cup h_m) = p(h_1) + \dots + p(h_s) + \dots + p(h_m), \quad (8)$$

где  $s(s = \overline{1, m}; m \leq n)$  — номер вектора непрерывного результата;  $n$  — число выполненных ТЗ;  $p(h_s)$  — вероятность появления отдельных векторов  $\mathbf{H}_s$  непрерывного результата, оцениваемая в соответствии с формулами (4), (5).

Для оценки в (8) неизвестного параметра  $p$  предлагается использовать метод правдоподобия, при котором в качестве оценки вероятности  $p$  принимается ее значение

$$\hat{p} = \arg \max_{p \in (0,1)} f(\mathbf{J}|p) = \arg \max_{p \in (0,1)} P(H), \quad (9)$$

Таблица 1

Некоторые варианты векторов результатов тестирования для  $n = 12$

Вариант	$j_1$	$j_2$	$j_3$	$j_4$	$j_5$	$j_6$	$j_7$	$j_8$	$j_9$	$j_{10}$	$j_{11}$	$j_{12}$
1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	0	1	1	1	1	1	1
3	1	1	1	1	1	1	0	0	0	0	0	0
4	1	0	1	0	1	0	1	0	1	0	1	0
5	0	1	0	1	0	1	0	1	0	1	0	1
6	1	1	0	0	0	0	0	0	0	1	1	1
7	1	1	0	0	1	1	1	1	1	0	1	1
8	1	1	1	1	0	1	1	1	0	1	1	1
9	1	1	1	1	0	0	0	1	1	1	1	0
10	1	1	0	1	1	0	0	1	1	0	1	0

Оценки результатов тестирования, приведенных в табл. 1

№ варианта	$n - k$	$k$	$l$	$m - l$	Точечная оценка по (1)	Интервальные оценки					
						биномиального распределения			геометрического распределения		
						Вероятнейшее значение по (3)	Границы ДИ по (6)		Вероятнейшее значение по (11)	Границы ДИ по (12)	
							Нижняя	Верхняя		Нижняя	Верхняя
1	12	0	1	0	1,000	1,000	0,735	1,000	0,923	0,754	1,000
2	6	6	1	1	0,500	0,500	0,211	0,789	0,500	0,182	0,818
3	6	6	1	1	0,500	0,500	0,211	0,789	0,500	0,182	0,818
4	6	6	6	6	0,500	0,500	0,211	0,789	0,500	0,182	0,818
5	6	6	6	6	0,500	0,500	0,211	0,789	0,500	0,182	0,818
6	5	7	2	1	0,417	0,417	0,152	0,723	0,400	0,089	0,711
7	9	3	3	2	0,750	0,750	0,428	0,945	0,647	0,343	0,951
8	10	2	4	2	0,833	0,833	0,516	0,979	0,667	0,367	0,966
9	8	4	2	2	0,667	0,667	0,349	0,901	0,625	0,317	0,933
10	7	5	4	4	0,583	0,583	0,277	0,848	0,550	0,234	0,866

максимизирующее вероятность  $p(H)$  появления при выполнении  $n$  ТЗ вектора  $\mathbf{J} = (j_1, \dots, j_i, \dots, j_n)$ , включающего  $m (m = \overline{1, n})$  векторов  $\mathbf{H}_s$  непрерывного результата.

Соответствующая функция  $W(p_s, p)$  правдоподобия для любой из реализаций события  $H \cup h_s$  может быть представлена в виде

$$W(p_s, p) = \prod_{s=1}^{s=m} p(\mathbf{H}_s) = \prod_{s=1}^{s=l} p(\mathbf{H}_s^+) \prod_{s=l+1}^{s=m} p(\mathbf{H}_s^-) = p^{(n-k) + (m-l)q^{k+l}}, \quad (10)$$

где  $k (k = \overline{0, n})$  — число ТЗ, выполненных с ошибкой;  $(n - k) ((n - k) = \overline{0, n})$  — число ТЗ, выполненных правильно;  $s (s = \overline{1, m})$  — номера векторов  $\mathbf{H}_s$  непрерывного результата;  $s (s = \overline{1, l})$ ,  $s (s = \overline{l+1, m})$  — соответственно номера векторов непрерывного успеха и непрерывной неудачи;  $l$ ,  $(m - l)$  — соответственно число векторов непрерывного успеха и непрерывной неудачи ( $m + l \leq n$ ).

Поиск оценки  $\hat{p}$  наибольшего правдоподобия целесообразно реализовать путем решения следующей задачи нелинейного программирования:

$$\left. \begin{aligned} & p^{(n-k) + (m-l)(1-p)^{k+l}} \rightarrow \max_p; \\ & p \in (0, 1). \end{aligned} \right\} \quad (11)$$

Для решения (11) можно использовать стандартную функцию "Solve (Поиск решения)" Microsoft Excel. ДИ оценки  $\hat{p}$  может быть определен как

$$\left( \hat{p} - u_{1-\varepsilon/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + u_{1-\varepsilon/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right). \quad (12)$$

Сравним оценки обучаемых, полученные обычным путем на основе предлагаемых моделей.

Пусть обучаемыми выполнено 13 заданий и получены результаты, приведенные в табл. 1. Оценки этих результатов приведены в табл. 2.

Данные, представленные в табл. 2, свидетельствуют, что вероятнейшие значения результатов тестирования, полученные по предлагаемым моделям (3), (11), практически совпадают с традиционными точечными оценками. Однако, в отличие от традиционных оценок, они еще и указывают соответствующий доверительный интервал. При этом модель (12), использующая закономерности геометрического распределения, по сравнению с моделью (11), основанной на законе биномиального распределения, всегда дает более осторожные оценки.

Таким образом, приведенные в табл. 2 оценки свидетельствуют, что применение предлагаемых моделей интервального оценивания результатов критериально-ориентированного тестирования обеспечивает переход от точечных оценок эмпирической частоты к оценкам вероятности правильного действия, которые имеют количественно оцениваемый ДИ. Последнее обеспечивает применение этих моделей для решения задачи автоматизации управления процессом критериально-ориентированного тестирования в целях обеспечения его оперативности путем остановки процедуры предъявления ТЗ после получения оценки заданной точности.

### Выводы

Процедуры контроля знаний в современных КСТ исключают возможности оценки погрешности результатов тестирования и прекращения процедуры тестирования при достижении заданной достоверности его результатов. Разработанный в статье переход от точечных оценок эмпирической частоты успешного выполнения тестовых заданий к интервальным оценкам вероятности этой успешности позволяет устранить эти недостатки в отно-

шении наиболее распространенной формы тестовых заданий — тестовых заданий закрытого типа с единственным верным ответом.

#### Список литературы

1. **Евсигнеев А. Е.** Концепция развития тестовой технологии контроля уровня обученности студентов в системе профессионального образования России. URL: [http://www.vevivi.ru/best/downloads.html&req=download&code=confirm\\_download&id=156184](http://www.vevivi.ru/best/downloads.html&req=download&code=confirm_download&id=156184) (дата обращения: 07.11.2016).
2. **Переверзев В. Ю.** Технология разработки тестовых заданий: справочное руководство. М.: Е-Медиа, 2005. 265 с.
3. **Ким В. С.** Тестирование учебных достижений. Усурийск: Изд-во УГПИ, 2007. 214 с.
4. **Печников А. Н., Печников Д. А.** Решение задач текущего педагогического контроля на основе анализа результатов критериально-ориентированного тестирования // Образовательные технологии и общество (международный электронный журнал). 2015. Т. 18, № 2. С. 489—513. URL: <http://ifets.ieee.org/russian/> (дата обращения: 07.11.2016).
5. **Иванов В. Н.** Метод автоматизированной оценки качества профессиональной деятельности в корабельных автоматизированных системах: Дис. ... канд. воен. наук. Петродворец: ВМирЭ им. А. С. Попова, 2009. 187 с.
6. **Аванесова Т. П., Печников А. Н., Шиков А. Н.** Электронное обучение: учебное пос. СПб.: ВАС, 2014. 73 с. URL: [http://pedlib.ru/books.php?part=Books&dir=7/0297&num\\_page=1](http://pedlib.ru/books.php?part=Books&dir=7/0297&num_page=1) (дата обращения: 07.11.2016).
7. **Аванесов В. С.** Композиция тестовых заданий. Учебная книга для преподавателей вузов, учителей школ, аспирантов и студентов педвузов. 4-е изд. М.: Центр тестирования, 2002. 237 с.

8. **Печников А. Н., Ветров Ю. А.** Проектирование и применение компьютерных технологий обучения. Ч. 1. Концепция систем автоматизированного обучения и моделирование процессов деятельности. Кн. 1. СПб: БГТУ, 2002. 195 с.

9. **Печников А. Н., Печников Д. А.** О соответствии результатов критериально-ориентированного тестирования целям текущего педагогического контроля // *Фундаментальные и прикладные исследования в современном мире*. 2015. № 10-5. С. 74—85. URL: [http://to-future.ru/wp-content/uploads/2015/06/ФиПИВСМ\\_№10\\_Том5.pdf](http://to-future.ru/wp-content/uploads/2015/06/ФиПИВСМ_№10_Том5.pdf) (дата обращения: 07.11.2016).

10. **Свиридов А. П.** Основы статистической теории обучения и контроля знаний. М.: Высшая школа, 1981. 126 с.

11. **Наумов В. Н.** Теоретические основы представления содержания профессиональной подготовки специалистов ВМФ в интеллектуальных обучающих системах: Дис. ... докт. воен. наук. Петродворец: ВМирЭ им. А. С. Попова, 1994. 296 с.

12. **Ван дер Варден Б. Л.** Математическая статистика / Перевод с немецкого. М.: Изд-во Иностранной литературы, 1960. 436 с. URL: <http://www.twirpx.com/file/275786/>

13. **Печников А. Н.** Комплекс методов и моделей оценки качества и эффективности деятельности операторов вооружения и военной техники в тренажерно-обучающих системах ВМФ: Дис. ... докт. техн. наук. СПб.: СПбГЭТУ, 2011. 343 с.

14. **Печников А. Н., Стручков А. М., Хекерт Е. В.** и др. Генератор оценки: Свидетельство о государственной регистрации программ для ЭВМ № 2009613362 от 26.06.2009.

15. **Платов А. Н.** Автоматизация управления контролем подготовки на тренажерах ВМФ: Дис. ... канд. техн. наук. СПб.: ЦНИИ МО РФ, 2000. 125 с.

16. **Пузырев В. Ю.** Диагностика деятельности операторов корабельных электромеханических систем в специализированных тренажерах ВМФ: Дис. ... канд. техн. наук. Новороссийск: НГМА, 2005. 129 с.

**D. A. Pechnikov**, Associate Professor, e-mail: 19pda72@bk.ru

Naval Academy named after Admiral of the Fleet of the Soviet Union NG Kuznetsov, St. Petersburg

## Using the Methods of the Interval Estimation of the Results of the Control of Knowledge in Computer Testing Systems

*This article discusses the use of models and procedures for interval estimation of the results of control of knowledge in computer systems using methods based on the laws of binomial distribution and the geometric distribution as special case of the negative binomial distribution. The proposed approach to the transition from the empirical point estimates of the relative frequency of a successful execution of the test tasks to the interval estimates of the probability of this success to determine the margin of error and confidence interval of the estimates. The developed model criteria-oriented assessment of results of knowledge control, which is focused on implementation in computer systems testing.*

**Keywords:** Interval estimation, confidence intervals, binomial distribution

#### References

1. **Evsigineev A. E.** *Koncepcija razvitiya testovoj tehnologii kontrolja urovnja obuchennosti studentov v sisteme professional'nogo obrazovanija Rossii*. URL: [http://www.vevivi.ru/best/downloads.html&req=download&code=confirm\\_download&id=156184](http://www.vevivi.ru/best/downloads.html&req=download&code=confirm_download&id=156184) (data obrashhenija: 07.11.2016) (in Russian).
2. **Pererzev V. Ju.** *Tehnologija razrabotki testovyh zadanij: spravochnoe rukovodstvo*, Moscow, E-Media, 2005. 265 p. (in Russian).
3. **Kim V. S.** *Testirovanie uchebnyh dostizhenij*. Ussurijsk: Izdatel'stvo UGPI, 2007. 214 p. (in Russian).
4. **Pechnikov A. N., Pechnikov D. A.** Reshenie zadach tekushhego pedagogicheskogo kontrolja na osnove analiza rezul'tatov kriterial'no-orientirovannogo testirovanija, *Obrazovatel'nye tehnologii i obshhestvo (mezhdunarodnyj elektronnyj zhurnal)*, 2015, vol. 18, no. 2, pp. 489—513. URL: <http://ifets.ieee.org/russian/> (data obrashhenija: 07.11.2016) (in Russian).

5. **Ivanov V. N.** *Metod avtomatizirovannoj ocenki kachestva professional'noj dejatel'nosti v korabel'nyh avtomatizirovannyh sistemah*: Dis. ... kand. voen. nauk. Petrodvorec: VMIRJe, 2009. 187 p. (in Russian).

6. **Avanesova T. P., Pechnikov A. N., Shikov A. N.** *Jelektronnoe obuchenie: uchebnoe posobie*. SPb.: VAS, 2014, 73 p. URL: [http://pedlib.ru/books.php?part=Books&dir=7/0297&num\\_page=1](http://pedlib.ru/books.php?part=Books&dir=7/0297&num_page=1) (data obrashhenija: 07.11.2016) (in Russian).

7. **Avanesov V. S.** *Kompozicija testovyh zadanij. Uchebnaja kniga dlja prepodavatelej vuzov, uchitelej shkol, aspirantov i studentov pedvuzov*. 4-e izd. Moscow, Centr testirovanija, 2002, 237 p. (in Russian).

8. **Pechnikov A. N., Vetrov Ju. A.** *Proektirovanie i primenenie komp'juternyh tehnologij obuchenija*. Ch. 1. Koncepcija sistem avtomatizirovannogo obuchenija i modelirovanie processov dejatel'nosti. Kn. 1. Saint-Petersburg, BGTU, 2002, 195 p. (in Russian).

9. **Pechnikov A. N., Pechnikov D. A.** O sootvetstvii rezul'tatov kriterial'no-orientirovannogo testirovanija celjam tekushhego pedagogicheskogo kontrolja, *Fundamental'nye i prikladnye issledovanija v*

sovremennom mire, 2015, no. 10–5, pp. 74–85. URL: [http://to-future.ru/wp-content/uploads/2015/06/FiPivSM\\_№10\\_Tom5.pdf](http://to-future.ru/wp-content/uploads/2015/06/FiPivSM_№10_Tom5.pdf) (data obrashhenija: 07.11.2016) (in Russian).

10. **Sviridov A. P.** *Osnovy statisticheskoj teorii obuchenija i kontrolja znanij*, Moscow, Vysshaja shkola, 1981, 126 p. (in Russian).

11. **Naumov V. N.** *Teoreticheskie osnovy predstavlenija sodержanija professional'noj podgotovki specialistov VMF v intellektual'nyh obuchajushhijh sistemah*: Dis. ... dokt. voen. nauk. Petrodvorec: VVMIR Je im. A. S. Popova, 1994. 296 p. (in Russian).

12. **Van der Varden B. L.** *Matematicheskaja statistika*. Perevod s nemeckogo, Moscow, Izd-vo Inostrannoj literatury, 1960. 436 p. URL: <http://www.twirpx.com/file/275786/> (in Russian).

13. **Pechnikov A. N.** *Kompleks metodov i modelej ocenki kachestva i jeffektivnosti dejatel'nosti operatorov vooruzhenija i voennoj tehniki v trenazherno-obuchajushhijh sistemah VMF*: Dis. ... dokt. tehn. nauk. Saint-Petersburg, SPbGJeTU, 2011. 343 p. (in Russian).

14. **Pechnikov A. N., Struchkov A. M., Hekert E. V.** i dr. *Generator ocenki*: Svidetel'stvo o gosudarstvennoj registracii programm dlja JeVM № 2009613362 ot 26.06.2009. (in Russian).

15. **Platov A. N.** *Avtomatizacija upravlenija kontrolem podgotovki na trenazherah VMF*: Dis. ... kand. tehn. nauk. Saint-Petersburg: I CNII MO RF, 2000. 125 p. (in Russian).

16. **Puzyrev V. Ju.** *Diagnostika dejatel'nosti operatorov korabel'nyh jelektromehaničeskijh sistem v specializirovannyh trenazherah VMF*: Dis. ... kand. tehn. nauk. Novorossijsk: NGMA, 2005. 129 p. (in Russian).

УДК 004.832.34

**Л. М. Кадыева**<sup>1, 2</sup>, студент, МАИ (НИУ), математик, e-mail: leylakadyeva@gmail.com,

**С. Е. Левин**<sup>1</sup>, технический директор, РКСС-ПС, e-mail: levin@pkcc-ps.ru,

**Я. Н. Окрент**<sup>1</sup>, гл. науч. сотр., РКСС-ПС, e-mail: yakov.okrent@pkcc-ps.ru,

**Д. И. Лоскутов**<sup>2</sup>, студент, e-mail: dmitry.loskutov@pkcc-ps.ru,

**А. М. Романенков**<sup>2</sup>, канд. техн. наук, доц., e-mail: romanaleks@gmail.com

<sup>1</sup> Российская корпорация средств связи — Программные системы (РКСС-ПС), Москва

<sup>2</sup> Московский авиационный институт (Национальный исследовательский университет), МАИ, Москва

## Использование нейронечетких вычислений для диспетчеризации на железнодорожной станции со сложной топологией

*Рассмотрена методика нейронечетких вычислений для выявления параметров идентификации нечетких систем и вычисления приоритетов операций в рамках системы поддержки принятия решений диспетчерской службой железнодорожной станции со сложной топологией путей. Методика реализована в виде программного модуля. Приведены результаты моделирования и рекомендации по использованию разработанной методики.*

**Ключевые слова:** идентификация нелинейных зависимостей, нейронечеткая сеть, алгоритм Сугено, проектирование нечетких систем, очередь с приоритетом, задача нахождения оптимальной последовательности выполнения операций

### Введение

Планирование и контроль выполнения планов железнодорожных перевозок невозможны без анализа использования задействованных технических средств, учета и оценки их работы.

Наибольшую сложность вызывает планирование диспетчерских задач на железнодорожных станциях, представляющих собой многомерные системы большой размерности:

- сложная топология путей (значительное число входов и выходов, мест погрузки и разгрузки, платформ, тупиков, депо, светофоров, стрелок, перекрестков, развязок и т.д.);
- большое количество подвижного состава (локомотивы, пустые и груженные вагоны).

Одной из наиболее затратных составляющих в процессе организации эксплуатации станции является задействование находящегося на ней тягового состава (локомотивы, тепловозы) для сортировки

вагонов и перевозки полезных грузов. Поэтому повышение эффективности эксплуатации станционного хозяйства путем минимизации времени задействования тягового состава является актуальной задачей, имеющей научное и практическое значение.

Для решения данной задачи в качестве модели железнодорожной станции была использована математическая модель перевозки грузов внутри станции представленная в работе [1]. Модель состоит из нескольких модулей, каждый из которых выполняет определенную функцию:

- минимизация бюджета времени использования локомотивного парка;
- минимизация бюджета времени использования вагонного парка;
- определение оптимального плана погрузки/выгрузки;
- нахождение оптимального пути движения составов и т. д.

Модель железнодорожной станции рассматривалась состоящей из следующих компонентов:

- источники пустых вагонов (тупики, упоры, депо и пр.);
- платформы;
- въезды на станцию;
- выезды со станции;
- топология станции (ж/д пути, стрелки, перекрестки, светофоры и пр.);
- технология проведения железнодорожных операций.

Для решения задачи минимизации времени задерживания локомотивов, обеспечивающих продвижение вагонов от одного компонента станции к другому, что соответствует повышению эффективности использования бюджета локомотивного парка, в данной работе рассматривается методика построения нейронечеткого модуля. Этот модуль является дальнейшим развитием математической модели, изложенной в работе [1], и представляет собой реализацию методики многоэтапной оптимизации функционирования локомотивного и вагонного парков, формирующей оптимальную последовательность выполнения технологических операций на железнодорожной станции.

Для определения оптимальной последовательности выполнения технологических операций разрабатываемая модель (и ее реализация — программный модуль) формирует очередь диспетчерских операций с учетом рассчитанных приоритетов их выполнения.

Приоритетом выполнения диспетчерской задачи будем называть величину, определенную на интервале (0, 1) и характеризующую степень срочности исполнения данной задачи. Очередь с приоритетом — это последовательность элементов, в которой включение либо исключение элемента выполняется в соответствии с их приоритетами. Очередь с приоритетным исключением — это очередь, в которой включение элементов осуществляется в конец, а при исключении проводится поиск элемента с максимальным приоритетом. Очередь с приоритетным включением — такая очередь, в которой последовательность элементов все время поддерживается упорядоченной по убыванию приоритета. То есть при включении элемента выполняется поиск места, куда он будет включен, а затем происходит сама процедура включения. При исключении же будет удален элемент, стоящий в начале. Очередь с приоритетным включением непосредственно используется в представленной математической модели.

При использовании константных значений в качестве приоритетов возникает следующая проблема: при постоянном поступлении более значимых задач время задач с меньшим приоритетом будет постоянно расти, что может привести к невыполнению последних. Поэтому авторами предложен

метод нахождения динамических приоритетов, снижающий простои в работе и реализованный в нейронечетком модуле.

Нейронечеткий модуль рассчитывает конечный приоритет операции, основываясь на двух величинах — времени ожидания операции в очереди и собственном приоритете операции (рассчитывается с помощью предложенной математической модели). В процессе добавления новой технологической операции в очередь ей присваивается начальный приоритет, заданный диспетчером железнодорожной станции исходя из срочности выполнения этой конкретной операции. После постановки данной операции в очередь начинается отсчет времени ожидания, которое также влияет на последующий перерасчет ее приоритета. Такой постоянный динамический процесс вычисления значения приоритета позволяет оптимизировать порядок и время выполнения операций диспетчером железнодорожной станции.

Для сбора и накопления статистических данных, необходимых для демонстрации работы диспетчера железнодорожной станции, был смоделирован процесс поступления технологических операций в диспетчерский пункт. Каждая такая операция представляет собой технологический план, описывающий точную последовательность действий локомотива и обслуживающего персонала с указанием промежуточных остановок тепловоза и средних затрат времени. В качестве тестового задания было сгенерировано получение 15 операций в течение 1000 мин, что с учетом изменения приоритетов задач, уже стоящих в очереди, составляет две выборки длиной в 60 элементов (рис. 1), где первая выборка  $X_p = (x_{p,1}, \dots, x_{p,60})$  характеризуется временем ожидания в очереди, а вторая выборка  $X_t = (x_{t,1}, \dots, x_{t,60})$  — собственным приоритетом операций. Для исследования зависимости сгенерированных векторов про-

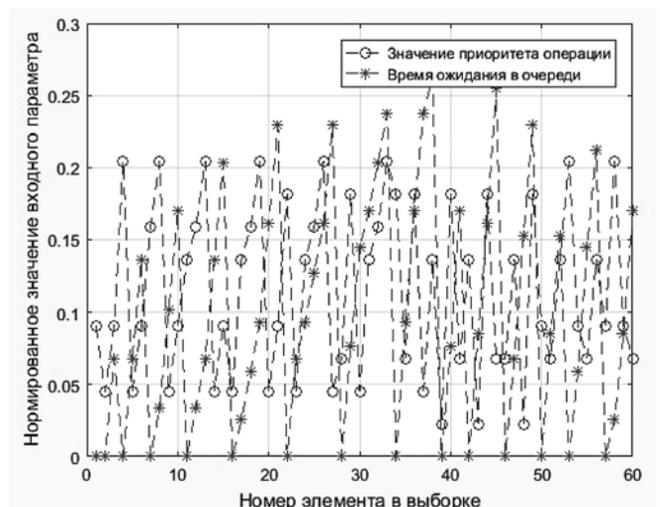


Рис. 1. Статистические данные

веден корреляционный анализ, по данным которого установлена линейная независимость векторов, что позволяет построить качественную оптимизационную модель. Коэффициент корреляции составил  $-8 \cdot 10^{-2}$ .

### Нейронечеткий модуль для расчета приоритетов

Процесс нахождения зависимости конечного приоритета (далее *функция приоритета*) для постановки операции в очередь диспетчера на выполнение усложнен тем, что помимо параметров необходимо найти еще и общий вид функции.

Наиболее известные алгоритмы, применяемые для решения подобных задач, такие как, например, метод Марквардта или методы нелинейного регрессионного анализа, позволяют приблизить расчетные данные к экспериментальным, но в то же время приводят к ряду проблем. Во-первых, они дают большие погрешности при попытке прогнозирования поведения *функции приоритета*, во-вторых, подобные методы по большей части работают с функциями заданного вида, в то время как в данной работе функция вообще неизвестна. Кроме того, понятия "приоритет диспетчерской задачи" и "время ожидания в очереди" являются нечеткими с точки зрения логики, поэтому было решено использовать метод двухэтапной идентификации нелинейных зависимостей с помощью нечетких баз знаний, основанный на работах Ротштейна [2, 3], основные положения работы которого можно сформулировать следующим образом.

1. Принцип лингвистичности входных и выходных переменных. В соответствии с этим принципом входы объекта и его выход рассматриваются как лингвистические переменные, которые оцениваются качественными терминами.

2. Принцип формирования структуры зависимости "вход-выход" в виде нечеткой базы знаний. Нечеткая база знаний представляет собой совокупность правил ЕСЛИ <входы>, ТО <выход>, которые отражают опыт эксперта и его понимание причинно-следственных связей в рассматриваемой задаче принятия решения (управления, диагностики, прогнозирования и др.).

3. Принцип двухэтапной настройки нечетких баз знаний. В соответствии с этим принципом построение модели нелинейного объекта осуществляется в два этапа, которые называются структурной и параметрической идентификацией. Настраиваемыми параметрами являются веса нечетких правил ЕСЛИ-ТО и формы функций принадлежности.

Первый этап построения математической модели — структурная идентификация — представляет собой процесс формирования нечеткой базы знаний Сугено [9], которая используется для систем подобного рода. Настройка параметров такой базы осуществляется по экспериментальным данным:

для модели зависимости  $y = f(X)$ , где  $X$  — вектор входных независимых переменных, а  $y$  — соответствующее значение выхода. Существует обучающая выборка из  $M$  пар экспериментальных данных, связывающих входы  $X_r = (x_{r,1}, x_{r,2}, \dots, x_{r,n})$  с выходом  $y$  исследуемой зависимости  $(X_r, y_r)$ ,  $r = \overline{1, M}$  (см. таблицу).

Входной вектор				Значение выхода
$x_{1,1}$	$x_{1,2}$	...	$x_{1,n}$	$y_1$
...	...	...	...	...
$x_{r,1}$	$x_{r,2}$	...	$x_{r,n}$	$y_r$
...	...	...	...	...
$x_{M,1}$	$x_{M,2}$	...	$x_{M,n}$	$y_M$

Предположим, что известна следующая нечеткая база знаний Сугено: "Если  $x_1 = \tilde{a}_{1j}$  и  $x_2 = \tilde{a}_{2j}$ , ...,  $x_n = \tilde{a}_{nj}$ , то  $y = b_j$ ,  $j = \overline{1, m}$ , где  $b_j$  — некоторые действительные числа".

Для начального наполнения базы знаний было выбрано использование каждого термина входной переменной только в одном правиле. Чтобы получить дифференцируемые отношения "входы — выход", использовали гауссовы функции принадлежности:

$$\mu_j(x_i) = e^{-\frac{1}{2} \left( \frac{x_i - c_{ij}}{\sigma_{ij}} \right)^2},$$

где  $\mu_j(x_i)$  — функция принадлежности нечеткого множества  $\tilde{a}_{ij}$ ;  $c_{ij}$  и  $\sigma_{ij}$  — параметры функции принадлежности.

Входному вектору  $X_r = (x_{r,1}, x_{r,2}, \dots, x_{r,n})$  соответствует такой результат нечеткого вывода:

$$y_r^f = \beta_{rj} \sum_{j=1}^m (b_0 + b_{j1}x_{r,1} + \dots + b_{jn}x_{r,n}),$$

где  $\beta_{rj}$  — относительная степень выполнения  $j$ -го правила для входного вектора  $X_r$ . При использовании в качестве  $t$ -нормы умножения относительная степень выполнения правила рассчитывается так:

$$\beta_{rj} = \frac{\exp\left(-\frac{1}{2} \sum_{i=1}^n \left( \frac{x_{ri} - c_{ij}}{\sigma_{ij}} \right)^2\right)}{\sum_{k=1}^m \left( \exp\left(-\frac{1}{2} \sum_{i=1}^n \left( \frac{x_{ri} - c_{ik}}{\sigma_{ik}} \right)^2\right) \right)}.$$

Второй этап — параметрическая идентификация исследуемой зависимости путем нахождения таких параметров, которые минимизируют отклонение результатов нечеткого моделирования от экспериментальных данных.

Настраиваемыми параметрами при идентификации будут коэффициенты в заключениях правил  $B = (b_1, \dots, b_m)$  и параметры функций принадлежности нечетких термов: координаты максимумов  $C = (c_{11}, \dots, c_{n1}, c_{22}, \dots, c_{n2}, \dots, c_{1m}, \dots, c_{nm})$  и коэффициенты концентраций  $S = (\sigma_{11}, \dots, \sigma_{n1}, \sigma_{22}, \dots, \sigma_{n2}, \dots, \sigma_{1m}, \dots, \sigma_{nm})$ . Задача настройки состоит в нахождении вектора  $(B, C, S)$ , такого, чтобы целевой функционал достигал минимума:

$$J(B, C, S) = \sqrt{\frac{1}{M} \sum_{r=1}^M (y_r - F(B, C, S, X_r))^2} \rightarrow \min,$$

где  $F(B, C, S, X_r)$  — результат вывода по нечеткой базе знаний Сугено с параметрами  $(B, C, S)$  для входного вектора  $X_r$ .

Пусть для  $r$ -й строчки выборки данных ошибка вычисляется по формуле  $E^{<r>} = (y_r - y_r^f)^2 = e^2$ . Тогда, согласно градиентному методу Коши [9], новые значения управляемых переменных рассчитываются по формуле

$$b_{ij}^{<r+1>} = b_{ij}^{<r>} - \alpha \frac{\partial E^{<r>}}{\partial b_{ij}}, \quad i = \overline{1, n}, \quad j = \overline{1, m},$$

где  $\alpha$  — длина шага, задающая скорость обучения. Частная производная имеет простое аналитическое представление:

$$\frac{\partial E^{<r>}}{\partial b_{ij}} = 2e\beta_{ij}x_{r,i}.$$

Находя частные производные ошибки  $E^{<r>}$  по управляемым переменным  $b_j, c_{ij}, \sigma_{ij}$ , получаем следующие правила обучения:

$$b_j^{<r+1>} = b_j^{<r>} - 2\alpha e\beta_{rj};$$

$$c_{ij}^{<r+1>} = c_{ij}^{<r>} - \alpha e\beta_{rj}(b_j^{<r>} - y_r) \frac{x_{ri} - c_i^{<r>}}{(\sigma_{ij}^{<r>})^2};$$

$$\sigma_{ij}^{<r+1>} = \sigma_{ij}^{<r>} - \alpha e\beta_{rj}(b_j^{<r>} - y_r) \frac{x_{ri} - c_i^{<r>}}{(\sigma_{ij}^{<r>})^2},$$

где  $i = \overline{1, n}, j = \overline{1, m}, r = \overline{1, M}$ . Для каждой пары "входы — выход" вычисляется ошибка — разница между полученным и ожидаемым значениями выхода, и модифицируются параметры нечеткой базы знаний [5].

ANFIS (*adaptive neuro-fuzzy inference system*) — это адаптивная сеть на основе системы нейронечеткого вывода Такаги—Сугено [9]. ANFIS является одним из первых вариантов гибридных ней-

ронечетких сетей — нейронной сети прямого распространения сигнала особого типа. Архитектура нейронечеткой сети изоморфна нечеткой базе знаний. В нейронечетких сетях используются дифференцируемые реализации треугольных норм (умножение и вероятностное ИЛИ), а также гладкие функции принадлежности. Это позволяет применять для настройки нейронечетких сетей быстрые алгоритмы обучения, основанные на методе обратного распространения ошибки.

Отличительная черта нейронечеткого вывода Такаги—Сугено состоит в следующем: правила алгоритма нечетки только в части **IF**, тогда как часть **THEN** представляет собой функциональную зависимость, т. е. правила в базе знаний являются своего рода переключателями с одного линейного закона на другой. В таком случае границы подобластей размытые, а это означает, что могут выполняться сразу несколько законов, но с разными степенями. Степень принадлежности входного вектора  $X^* =$

$(x_1^*, \dots, x_n^*)$  к значениям  $d_j = b_{j0} + \sum_{i=1}^n b_{ij}x_i$  рассчитывают следующим образом:

$$\mu_{d_j}(X^*) = \prod_{i=1}^n \mu_j(x_i^*), \quad j = 1, \dots, m.$$

Тогда результирующее значение выхода  $y$  находят как суперпозицию линейных законов, выполняемых в данной точке  $X^*$   $n$ -мерного пространства:

$$y = \frac{\sum_{j=1}^m \mu_{d_j}(X^*)d_j}{\sum_{j=1}^m \mu_{d_j}(X^*)}.$$

Такой подход к решению задачи представляет собой комбинацию методов наискорейшего спуска и обратного распространения ошибки, который оптимизирует как линейные, так и нелинейные параметры нечеткой базы знаний. Данный метод позволяет одновременно настраивать и заключения правил, и функции принадлежности термов входных переменных [9]. Так же по данному методу можно получить достаточно простые правила настройки параметров посылок правил.

Для реализации процесса моделирования описанной системы был использован набор инструментов Fuzzy Logic Toolbox пакета прикладных программ MATLAB. Данная технология предоставляет широкие возможности для создания, модификации и управления системами нечеткого вывода, настройки и использования ANFIS-систем и т. д.

## Демонстрация результатов разработки нейронечеткого модуля

Построенная нечеткая система моделирует зависимость конечного приоритета от приоритета выполнения задачи и времени нахождения задачи в очереди, что явилось результатом проведения двухэтапной процедуры проектирования нечеткой системы Сугено. Исходная система приведена на рис. 2, на котором видно, что начальные данные моделирования приняты за константную функцию, равную нулю, и далеки от экспериментальных данных.

Для синтезирования нечетких правил из экспериментальной выборки был использован метод субтрактивной кластеризации [9]. Это быстрый однопроходный алгоритм без итерационных процедур оптимизации. Для проведения кластерного анализа необходимо указать радиус в диапазоне [0, 1], который определяет, как далеко от центра кластера могут находиться его элементы. Данный параметр задан скаляром 0,5 (данные получены экспериментально). В результате была получена нечеткая модель Сугено с восемью правилами. Значение нормы отклонений составило 0,0055 (рис. 3).

Далее были настроены параметры нечеткой модели с помощью ANFIS-обучения в целях повышения точности. Значение разности квадратов на данном этапе составило  $7,9 \cdot 10^{-4}$ , что в 1,43 раза лучше, чем на предыдущем этапе. Таким образом удалось добиться роста качества моделирования. Сравнение входных экспериментальных данных с результатами вычислений ANFIS-системы можно увидеть на рис. 3, на котором точками обозначены экспериментальные данные, а звездочками — данные, полученные нейронечетким модулем. Видно, что эти два значения для каждого элемента выборки совпадают.

В результате была получена нейронная сеть с двумя входными переменными и одной выходной переменной (рис. 4), которая имеет следующую структуру.

- *Первый скрытый слой.* Каждый узел слоя представляет собой терм с соответствующей функцией принадлежности  $\mu$ . Входы сети соединены только со своими термами. Число узлов первого слоя равно сумме мощностей терм-множеств входных переменных. В результате работы программы было получено по 8 узлов первого слоя на каждую входную переменную, итого — 16 узлов. На выход узла подается степень принадлежности значения входной переменной соответствующему терму.
- *Второй скрытый слой.* Каждый узел данного слоя соответствует одному нечеткому правилу. Узел второго слоя соединен только с теми узлами первого, которые формируют antecedentes (посылки) соответствующего правила. Выходом узла является степень выполнения правила, ко-

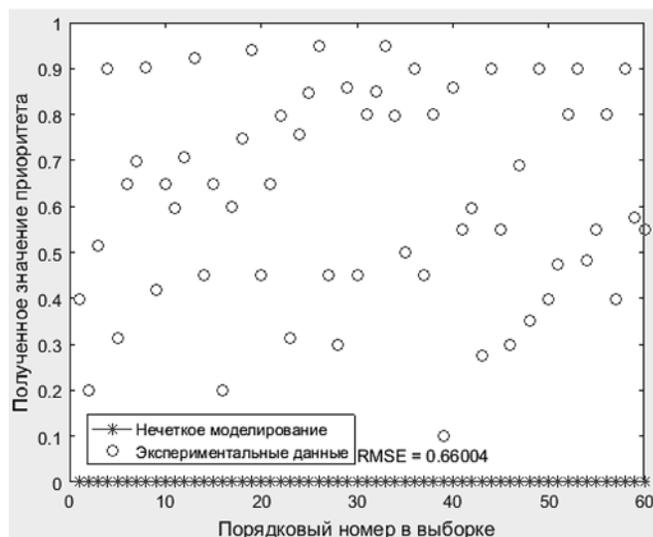


Рис. 2. Данные исходной системы

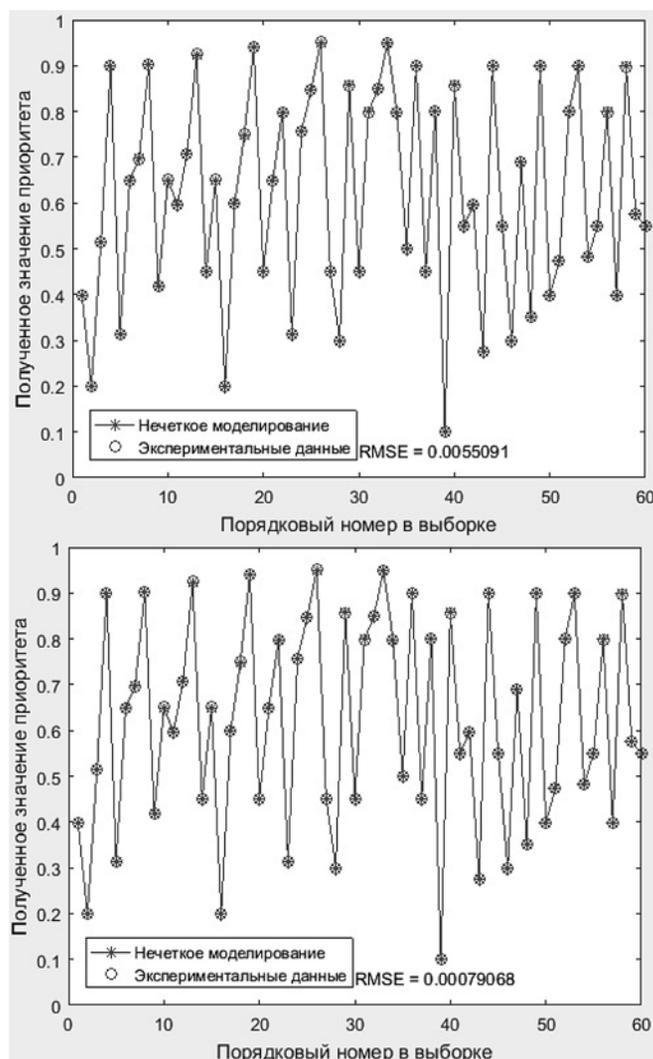


Рис. 3. Сравнение полученных результатов с экспериментальными данными (сверху — после проведения структурной идентификации, снизу — после проведения параметрической идентификации)

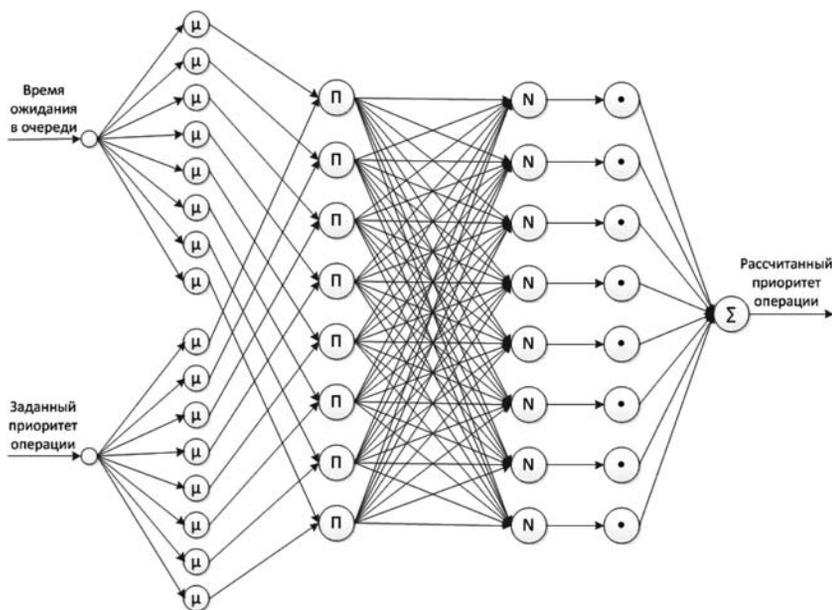


Рис. 4. Полученная нейронная сеть

торая рассчитывается как произведение входных сигналов.

- *Третий скрытый слой.* На данном слое происходит нормализация степеней выполнения правил — каждый узел рассчитывает относительную степень выполнения нечеткого правила.
- *Четвертый скрытый слой.* На данном слое рассчитываются заключения правил. Каждый узел вычисляет вклад одного нечеткого правила в выход сети.
- *Пятый скрытый слой.* Агрегирование результата, полученного по различным правилам. Единственный узел этого слоя суммирует вклады всех правил.

После построения и обучения нейронной сети ANFIS-система строит поверхность принятия решений, которая представлена на рис. 5. Коэффици-

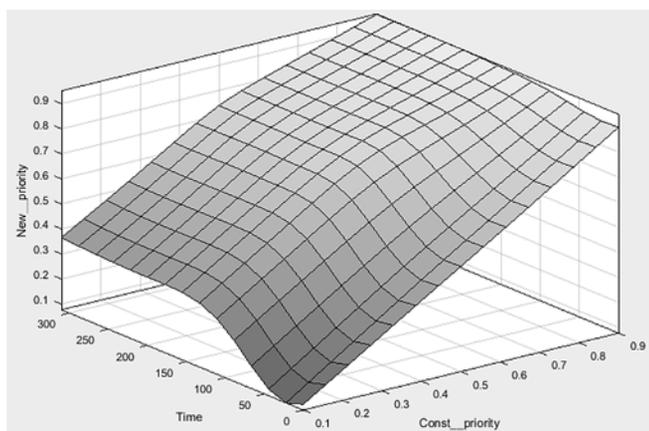


Рис. 5. Поверхность принятия решений

циент детерминации (критерий  $R^2$ ), показывающий связь между экспериментальными и теоретическими данными, равен 0,9998, что говорит об адекватности составленной модели.

После проведения ряда экспериментов исследования показали, что разработанный оптимизационный модуль позволил сэкономить около 19 % времени работы локомотивного парка, сократив его с 13,57 до 10,97 ч при поступлении пяти задач в течении суток.

## Заключение

Разработана методика построения нейронечеткого модуля вычисления приоритетов диспетчерских задач в рамках задачи оптимизации работы железнодорожной станции.

Собранные статистические данные были исследованы на коррелируемость. Результаты показали,

что данные независимы и могут быть успешно использованы для обучения ANFIS-системы.

Построенная в два этапа нечеткая модель является адекватной, согласно критерию  $R^2$ .

Разработанный нейронечеткий модуль интегрирован с оптимизационным алгоритмом, в результате чего удалось минимизировать бюджет времени использования локомотивов на железнодорожной станции и среднесуточный пробег вагонов.

## Список литературы

1. Кадыева Л. М., Левин С. Е., Окренг Я. Н. Комбинированный алгоритм линейной оптимизации с поиском максимального потока на графе // Молодой ученый. 2016. Т. 19 (123). С. 8—13.
2. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети. Винница: УНІВЕРСУМ-Вінниця, 1999. 320 с.
3. Ротштейн А. П., Кательников Д. И. Идентификация нелинейных зависимостей нечеткими базами знаний // Кибернетика и системный анализ. 1998. № 5. С. 53—61.
4. Орлов А. И. Прикладная статистика. М.: Экзамен, 2004. 110 с.
5. Штовба С. Д. Проектирование нечетких систем средствами MATLAB. М.: Горячая линия — Телеком, 2007. 288 с.
6. Цыпкин Я. З. Основы информационной теории идентификации. М.: Наука, 1984. 320 с.
7. Jang J.-S. R. ANFIS: Adaptive-Network-Based Fuzzy Inference System // IEEE Trans. Systems & Cybernetics. 1993. Vol. 23. P. 665—685.
8. Nauck D., Klawonn F., Kruse R. Foundations of Neuro-Fuzzy Systems. John Wiley & Sons., 1997. 320 p.
9. Takagi T., Sugeno M. Fuzzy Identification of Systems and Its Applications to Modeling and Control // IEEE Trans. on Systems, Man and Cybernetics. 1995. Vol. 15 (1). P. 116-132.

**L. M. Kadyeva**, Student, Mathematician, e-mail: leylakadyeva@gmail.com  
**S. E. Levin**, Chief Technical Officer, PKCC-PS, e-mail: levin@pkcc-ps.ru  
**Yu. N. Okrent**, Chief Scientific Officer, e-mail: yakov.okrent@pkcc-ps.ru  
Russian Corporation of Communication — Software Systems, Moscow, 105005  
**D. I. Loskutov**, Student (MAI), e-mail: dmitry.loskutov@pkcc-ps.ru  
**A. M. Romanenkov**, PhD, Associate Professor, e-mail: romanaleks@gmail.com  
Moscow Aviation Institute (National Research University), Moscow, 125080

## The Development of Neuro-Fuzzy Module Calculating the Priority Dispatching Tasks to Railway Station

*The theoretical position, methodological procedures and recommendations aimed at identifying fuzzy identification system parameters. There is analysis of the collected statistical data intended for training of a neural network. The considered model is tested for adequacy ( $R^2$  criterion). The structure of a neural network is shown. Spend the development of neuro-fuzzy module for calculating the dispatch task priorities in the decision support system of dispatching service railway station with a complex topology of ways. An example of the program module and analysis of the results are shown.*

**Keywords:** identification of nonlinear dependencies, neuro-fuzzy network, Sugeno algorithm of fuzzy systems design, priority queue, the problem of finding the optimal sequence of operations

### References

1. **Kadyeva L. M., Levin S. E., Okrent Ja. N.** Kombinirovannyj algoritm linejnoj optimizacii s poiskom maksimal'nogo potoka na grafe, *Molodoj uchenyj*, 2016, vol. 19 (123), pp. 8–13 (in Russian).
2. **Rotshtejn A. P.** *Intellektual'nye tehnologii identifikacii: nechetkaja logika, geneticheskie algoritmy, nejronnye seti*, Vinnica: UNIVERSUM-Vinnicia, 1999, 320 p. (in Russian).
3. **Rotshtejn A. P., Katel'nikov D. I.** Identifikacija nelinejnyh zavisimostej nechetkimi bazami znaniy, *Kibernetika i sistemnyj analiz*, 1998, no. 5, pp. 53–61 (in Russian).
4. **Orlov A. I.** *Prikladnaja statistika*. Moscow, Jekzamen, 2004, 110 p. (in Russian).
5. **Shtovba S. D.** *Proektirovanie nechetkih sistem sredstvami MATLAB*. Moscow, Gorjachaja linija — Telekom, 2007, 288 p. (in Russian).
6. **Cypkin Ja. Z.** *Osnovy informacionnoj teorii identifikacii*, Moscow: Nauka, 1984, 320 p. (in Russian).
7. **Jang J.-S. R.** ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Trans. Systems & Cybernetics*, 1993, vol. 23, pp. 665–685.
8. **Nauck D., Klawonn F., Kruse R.** *Foundations of Neuro-Fuzzy Systems*, John Wiley & Sons., 1997, 320 p.
9. **Takagi T., Sugeno M.** Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Trans. on Systems, Man and Cybernetics*, 1995, vol. 15 (1), pp. 116–132.

---

---

### Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Н. В. Яшина*.

Сдано в набор 09.06.2017. Подписано в печать 21.07.2017. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT817. 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.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.